Oracle Insurance

Insbridge Rating and Underwriting

RateManager User Guide

Modules

Volume 1 Program Explorer

Release 4.7.1

November 2013

Oracle Insurance Insbridge Rating and Underwriting RateManager User Guide

Release 4.7.1

Part # E50921-01

Library #50929-01

November 2013

Primary Author: Mary Elizabeth Wiger

CONTENTS

PREFACE

GETTING STARTED

CHAPTER 1

CHAPTER 2

CHAPTER 3

CHAPTER 4

CHAPTER 5

CHAPTER 6

CHAPTER 7

CHAPTER 12

CHAPTER 13

CHAPTER 14

CHAPTER 15

CHAPTER 18

CHAPTER 19

CHAPTER 20

CHAPTER 21

CHAPTER 22

CHAPTER 23

CHAPTER 24

CHAPTER 25

CHAPTER 26

SUPPORT

APPENDIX A

APPENDIX B

GLOSSARY

INDEX

LIST OF FIGURES

Preface

Welcome to the *Oracle Insurance Insbridge Rating and Underwriting RateManager User Guide.* This guide describes the features and functionality of Oracle Insurance Insbridge Rating and Underwriting RateManager (RateManager). RateManager is a component within the Oracle Insurance Insbridge Rating and Underwriting (IBRU) 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 Rating and Underwriting Framework Administrator User Guide.
- You can view this and other IBRU 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 Rating and Underwriting RateManager (RateManager) using the top and side menus as well as right click menus, which will be pointed out as they occur. Your browser's tool bars will not be displayed.

If you need 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: *IBRU requires Microsoft Internet Explorer. Other browsers are not compatible at this time.*

NOTE: *If you have a popup blocker installed, you will 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 need to step away. This prevents any unintentional deletions or entries and protects the integrity of your work.

INACTIVITY/TIMEOUT

RateManager, a web-based application, will log 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.

Manual 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

RateManager allows users to manage rate programs through the definition of variables, algorithms and underwriting rules. The RateManager system uses the same overall structure for any rate program. Within each line of business (Auto, Home, Fire, etc.), you can have an unlimited number of programs under an unlimited number of folders. Within each program, the system uses inputs and translates those into variables both table and calculated. These inputs and variables are then used in algorithms and underwriting rules, producing specific results.

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. Also, you may also want to review any applicable underwriting manuals.

Each line of business will have a set of coverages that makes up the base rate. The rate manual should define what is to be included in those base rates.

DATA PREPARATION

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

1. Start with the rating manual. Highlight all data elements that will 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. Each line of business will have a set of coverages that make up the base rate. Make sure you have defined what is to be included in those base rates.

For Example

A Homeowners Base Rate may include:

- Home/dwelling: This may be a flat dollar amount (In RateManager this would most likely be an Input)
- 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 (Input or Table Variable)
- Medical Payment: This may be a flat dollar amount (Input or Table Variable)

The home dwelling dollar may be calculated on:

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

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

3. 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 will require alteration for all variables linked to it.
    - Decreases rating time because the program has to access the database less.

4. Outline all algorithms that will be needed to produce rates for the program.

5. 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.

6. Create your Excel data tables. Each table must be implemented in a separate **tab-delimited** workbook. 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

Restrictions

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

Name Fields:

Most name fields, variables, algorithms, inputs, 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:

The Notes and Change Control fields restricts:

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**
Apostrophe	'
Tilde	~
Bar + H	|H

<table>
<tr><td>Bar + h</td><td>|h</td></tr>
<tr><td>Bar + L</td><td>|L</td></tr>
<tr><td>Bar + l</td><td>|l</td></tr>
</table>

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:

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

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:

- **Lines of Business:** Names must be unique. Duplicates are not allowed.
- **Sublines:** It is recommended that names be unique per line of business.
- **Subline Folders:** It is recommended that names be unique.
- **Version Folders:** It is recommended that names be unique.

  For subline 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.
- **Inputs:** Names are unique names per subline.
- **Result Variables:** Names are unique per subline.
- **Table Variables:** Global table variable names are unique per subline. Local table variable names are unique per program.
- **Calculated Variables:** Global calculated variable names are unique per subline. Local calculated variables are unique per program.
- **Algorithms:** Global algorithm names are unique per subline. Local algorithm names are unique per program.
- **Driver Assignment:** Global driver assignments names are unique per subline. Local driver assignment names are unique per program.
- 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 an input called ‘Manager’ you will be allowed to create an algorithm called ‘Manager’ as well. Each element will have an identifying icon to indicate what kind of element it is.
- **MultiLines, Releases, Test Cases, Impact Analysis Reports, etc.:** Names are all unique per module. You can use the same name across modules. For example, you can create a MultiLines program called ‘Test’ and a release called ‘Test’ and so on.

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.

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 multi-byte 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.

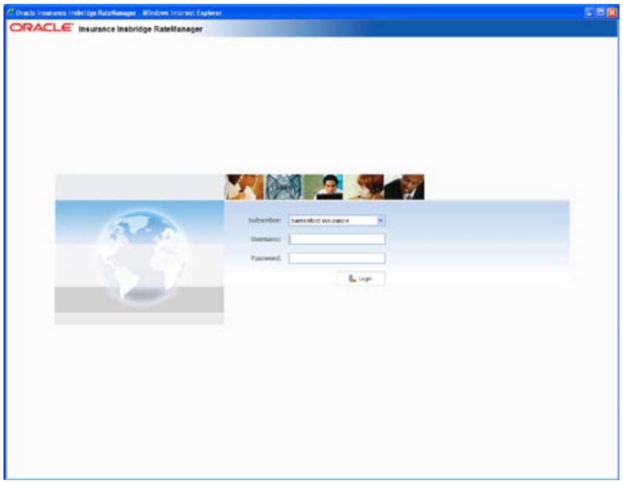
LOGGING INTO THE SYSTEM

Prior to accessing RateManager, you should have received:

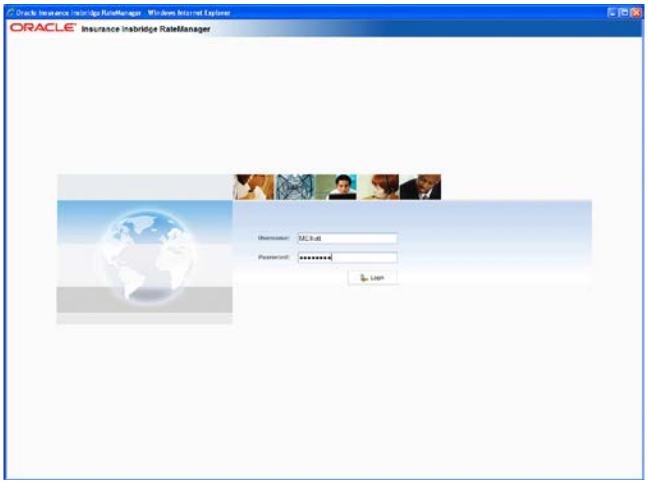
- **RateManager URL:** RateManager can be accessed by going to **http://<yourserver>/rm**, where <yourserver> is the name of the server where RateManager is installed.
- **Username:** usernames are required.
- **Password:** passwords are required.
- **Subscriber:** subscriber information may not be required.

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.


Oracle Insurance Insbridge RateManager - Windows Internet Explorer
ORACLE Insurance Insbridge RateManager
Subscriber:
Username:
Password:
Login

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


Oracle Insurance Insbridge RateManager - Windows Internet Explorer
ORACLE Insurance Insbridge RateManager
Username:
Password:
Login

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

Figure 1 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 will be 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 fields will be 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

Logging out will remove you from RateManager.

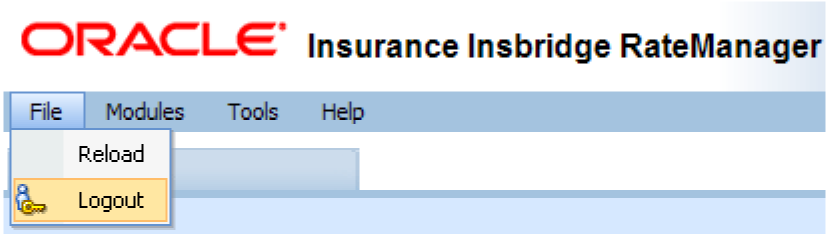


Figure 2 Logging Out

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.

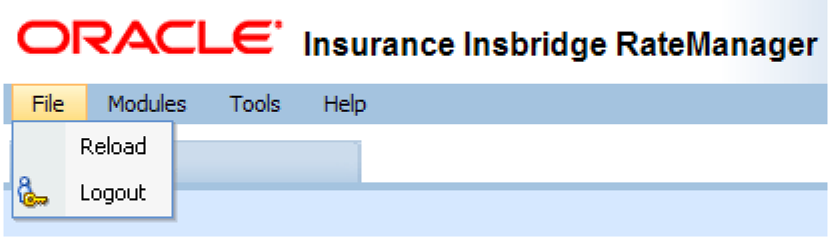
Internet Explorer

RateManager requires an ActiveX controller. Most of the time you will be prompted to install this controller when you begin to use RateManager. A message bar will be displayed at the top of the screen. You can run the ActiveX controller from there. If you experience issues with some RateManager screens in Internet Explorer 7 or Internet Explorer 8, the ActiveX Control may need to be re-registered. Please contact your system administrator for the RateManager for the **RateManager_IE7_IE8.exe** included in the installation download.

RateManager Tabs and Menus

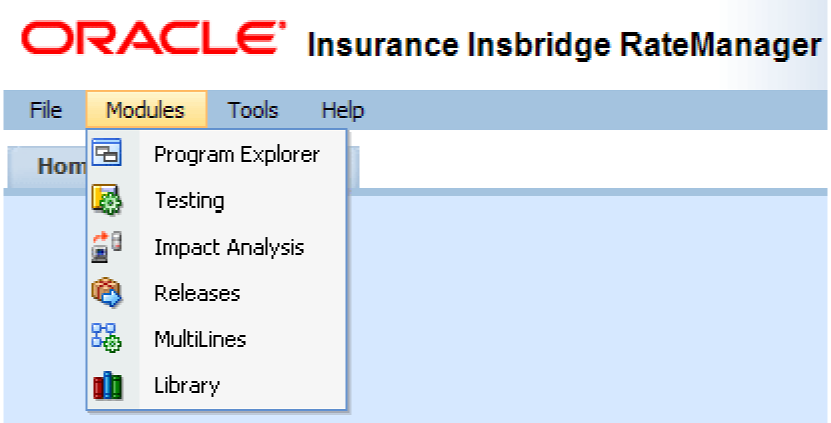
RateManager allows for multiple tabs to be utlized. The tabbed authoring interface allows for multiple activities to be open across separate tabs without losing any functionality. Each tab functions independently. You can view the Program Explorer, then edit a program version, and then create a new algorithm 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 will open either a new tab or a popup window. The new menus contain the same features that were in previous releases of RateManager.



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:

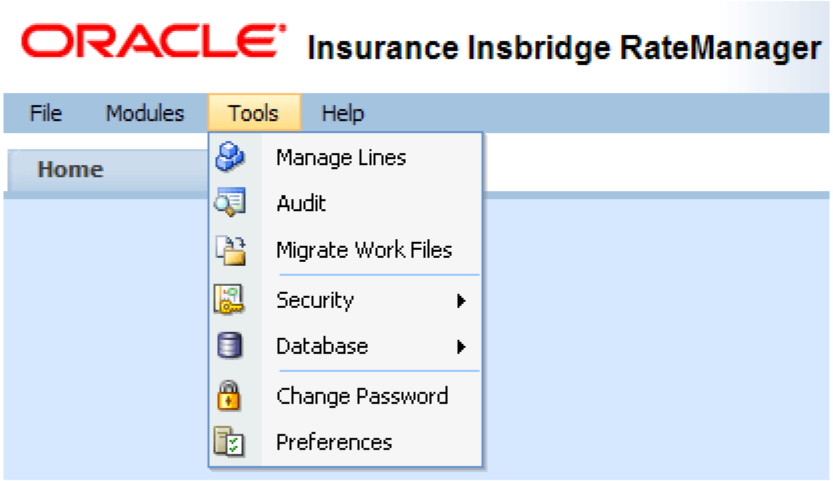
- **Program Explorer:** the program explorer holds sublines, folder, programs and program versions. Many of the procedures and functions performed under the line of business are found here.
- **Testing:** opens the testing module. This area allows you to test your rating program during development without loading packages.
- **Impact Analysis:** opens 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 will not be displayed. Please see the RateManager Modules User Guide Volume 2 for more information.

- **MultiLines:** a collection of programs from all lines of business combined with output mappings and assigned to execute in sequence that returns a single or multiple results.

- **Library:** the library is where custom XML files used in mappings as well as imported and template programs are stored and managed.

This guide will focus on the features and functionality in the Module menu.



The Tools menu:

- **Manage Lines:** opens a screen where administrators can manage lines of business. This is an advanced system feature and may not be available to all users.

- **Audit:** contains the audit logs of system users. In previous releases, this area was called Logs. This is an advanced system feature and may not be available to all users.

- **Migrate Work Files:** this option migrates the work files from previous releases into the current release. This is an advanced system feature and may not be available to all users.

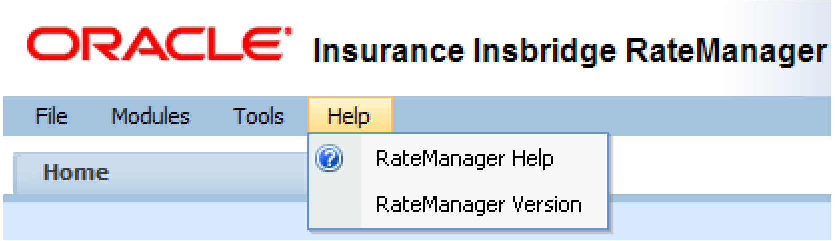
- **Security:** is where users, user groups and sessions are managed. This is an advanced system feature and may not be available to all users.

- **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.

- **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 will be displayed.

- **Change Password:** allows users to change their password.

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

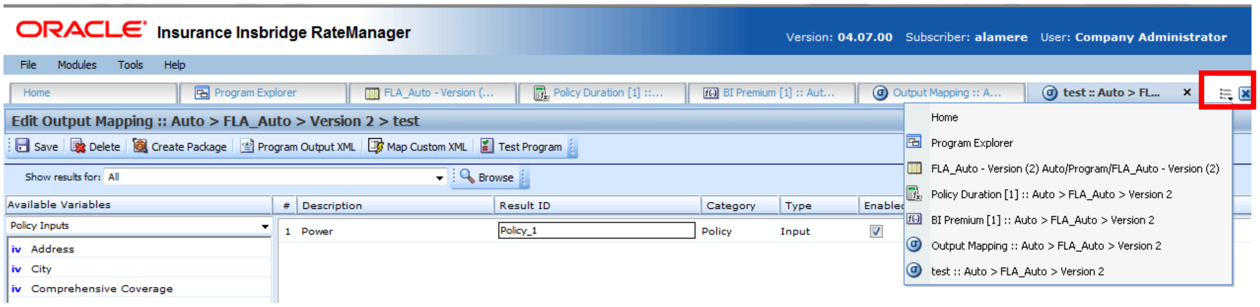


The Help menu:

- **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 hand of the screen to see a list of all the open tabs. You can navigate to any tab in the list by selecting it.


ORACLE Insurance Insbridge RateManager
Version: 04.07.00 Subscriber: alamere User: Company Administrator
File
Modules
Tools
Help
Home
Program Explorer
FLA_Auto - Version (...
Policy Duration [1] ::...
BI Premium [1] :: Aut...
Output Mapping :: A...
test :: Auto > FL...
Save
Delete
Create Package
Program Output XML
Map Custom XML
Test Program
Browse
Home
Program Explorer
FLA_Auto - Version (2) Auto/Program/FLA_Auto - Version (2)
Policy Duration [1] :: Auto > FLA_Auto > Version 2
BI Premium [1] :: Auto > FLA_Auto > Version 2
Output Mapping :: Auto > FLA_Auto > Version 2
test :: Auto > FLA_Auto > Version 2

Edit Output Mapping :: Auto > FLA_Auto > Version 2 > test

Show results for: All

Available Variables: Policy Inputs

- iv Address
- iv City
- iv Comprehensive Coverage

#	Description	Result ID	Category	Type	Enabled
1	Power	Policy_1	Policy	Input	☑

NOTE: *If you have a popup window open, you will have to close the popup before you can click on 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 will not be able to 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.

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 IBRU system is installed. Every line activated assumes the default locale. The date mask, the decimal symbol and digit grouping symbol are automatically filled with the defaults from the server. The line of business (LOB) can select a different locale from the server. The values set at the LOB level will apply as the default locale to every program created in the line regardless of subline however, the program can select a different locale if necessary. The updated locale will apply only to the program where the locale was changed.

The locale hierarchy is: the IBRU server determines the LOB locale. The LOB 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 Lines screen:** For more on setting a locale in the Manage Lines screen, please see the RateManager Tools guide.
- **In Programs**

To verify a locale for the LOB, enter the Edit the Line of Business screen in the Manage Lines option. The locale, date and number format are listed for the selected LOB. The Manage Lines option requires administrative 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 will be 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 LOB level. If you do not select a locale, the locale that the LOB has will be used.

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

Locale when Packaging

When you package a program (SRP), the Locale that is currently selected will be the Locale that the SRP uses. An SRP will use the Locale that has been selected at the program level. If you update the Locale and re-package, the Locale will be 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 will replace the previous version.

Locale in Global Variables

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

Locale in Table Variables

Table variables will 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 will retain the date and number formatting of the system where they were created. Date and number formats for global elements will assume the date and number format of the LOB 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 LOB 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 LOB level will affect the default locale on new programs only. Existing programs will not be affected by any changes. Existing programs will already have a locale set.

Changing a Locale at the program level will update 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 will be used when the program is packaged.

Locked programs can have the Locale updated. Locales will be 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 will be 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 will not be allowed. The list of special characters can be found in Restrictions.

Date formats are defined in the LOB. Any program in the line will use 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 IBRU 4.5 or later 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 an input 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 LOB. Any program in the line will use the defined format. A different format can be entered in the program.

- Most separators, i.e. comma (,), period (.), or space ( ) will be 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.

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 multi-level setup allows you to place an unlimited number of folders and programs underneath a subline of business.

VERSIONING

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

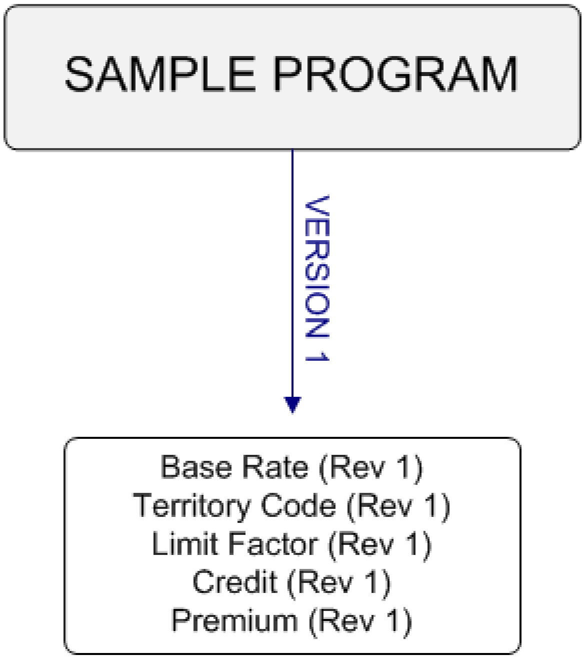
- **Program Versions**
- **Element Revisions**

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

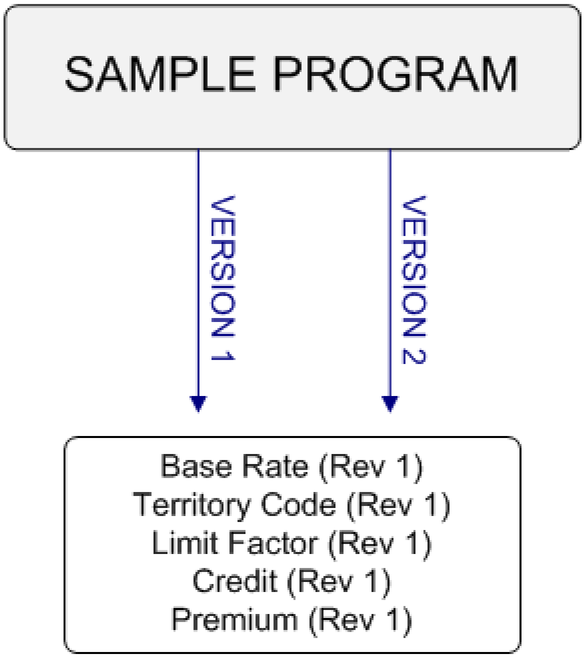
Program Versioning

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.

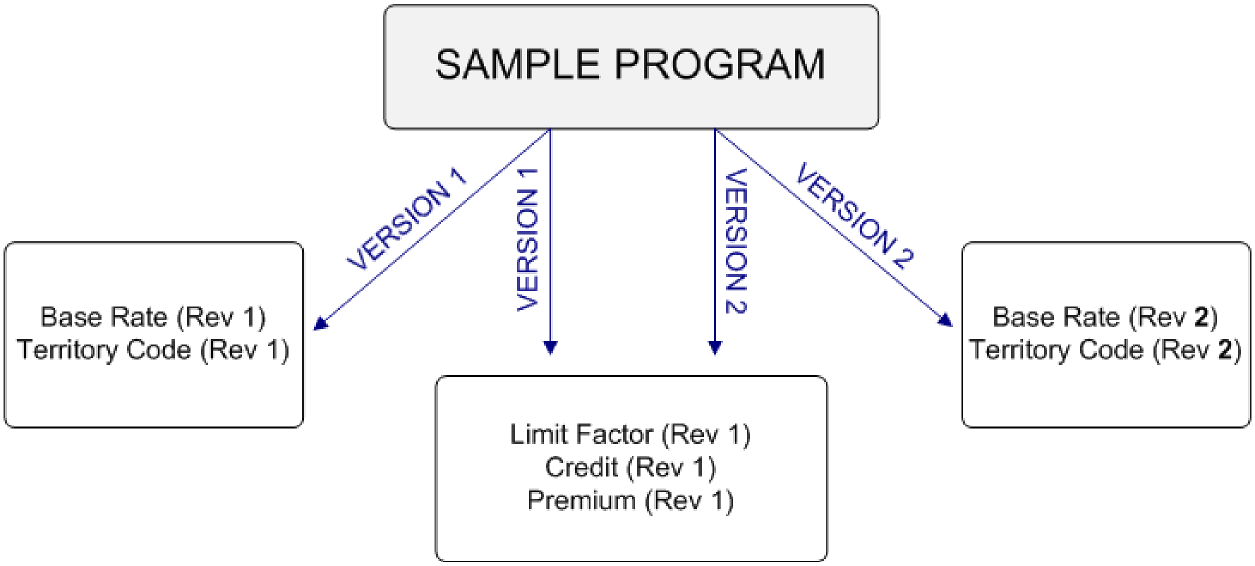
Sample Program represents an initial program before any new versions have been created.



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

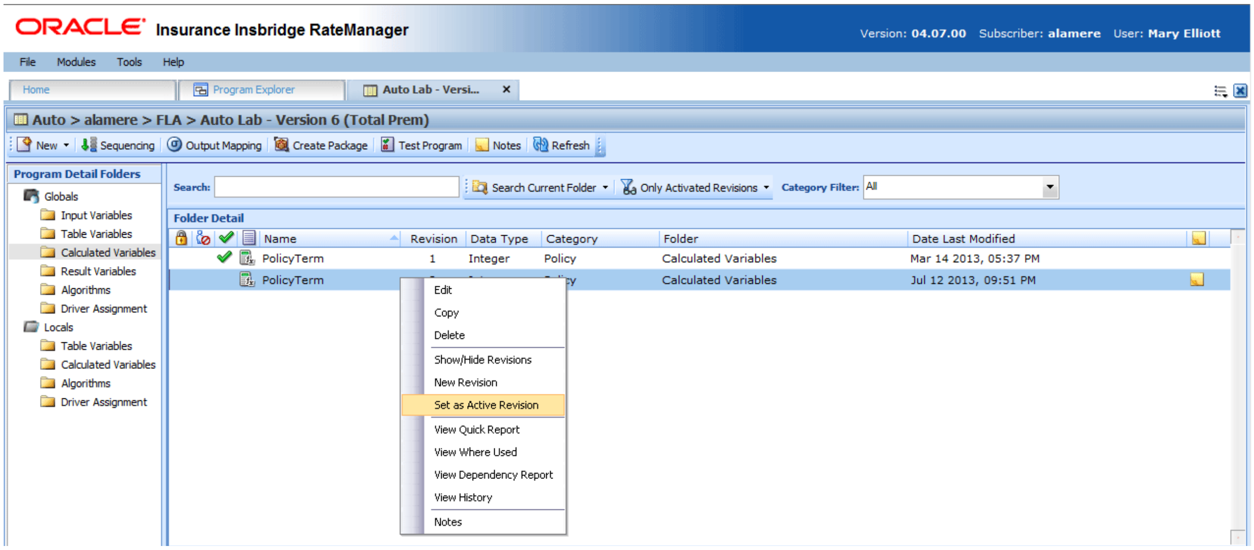


In order to load a rate change, you will need to create a second revision of all the variables and algorithms that reflect the rate change. In this example, if the Base Rate variable and the Territory Code variable change, you would make a second revision for both of variables. Version 1 would still 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 3 is an example of revisions for a specific table variable, shown by using the **Show/Hide Revisions** option on the right click menu of the variable.



✓	Name	Revision	Data Type	Category	Folder	Date Last Modified
✓	PolicyTerm	1	Integer	Policy	Calculated Variables	Mar 14 2013, 05:37 PM
	PolicyTerm	2	Integer	Policy	Calculated Variables	Jul 12 2013, 09:51 PM

Figure 3 Program Element Revisions

If you wanted ClassFactor to use Revision 2, you would select Revision 2 and 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.

NOTE: *All table variables, calculated variables, algorithms and drivers assignments can be versioned: including any global instances of these elements.*

HOME TAB

Once you have logged in, you will be on your Home Tab. The Home tab has a control panel that contains links to all the major components of RateManager. 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 will always be the first tab and cannot be closed.

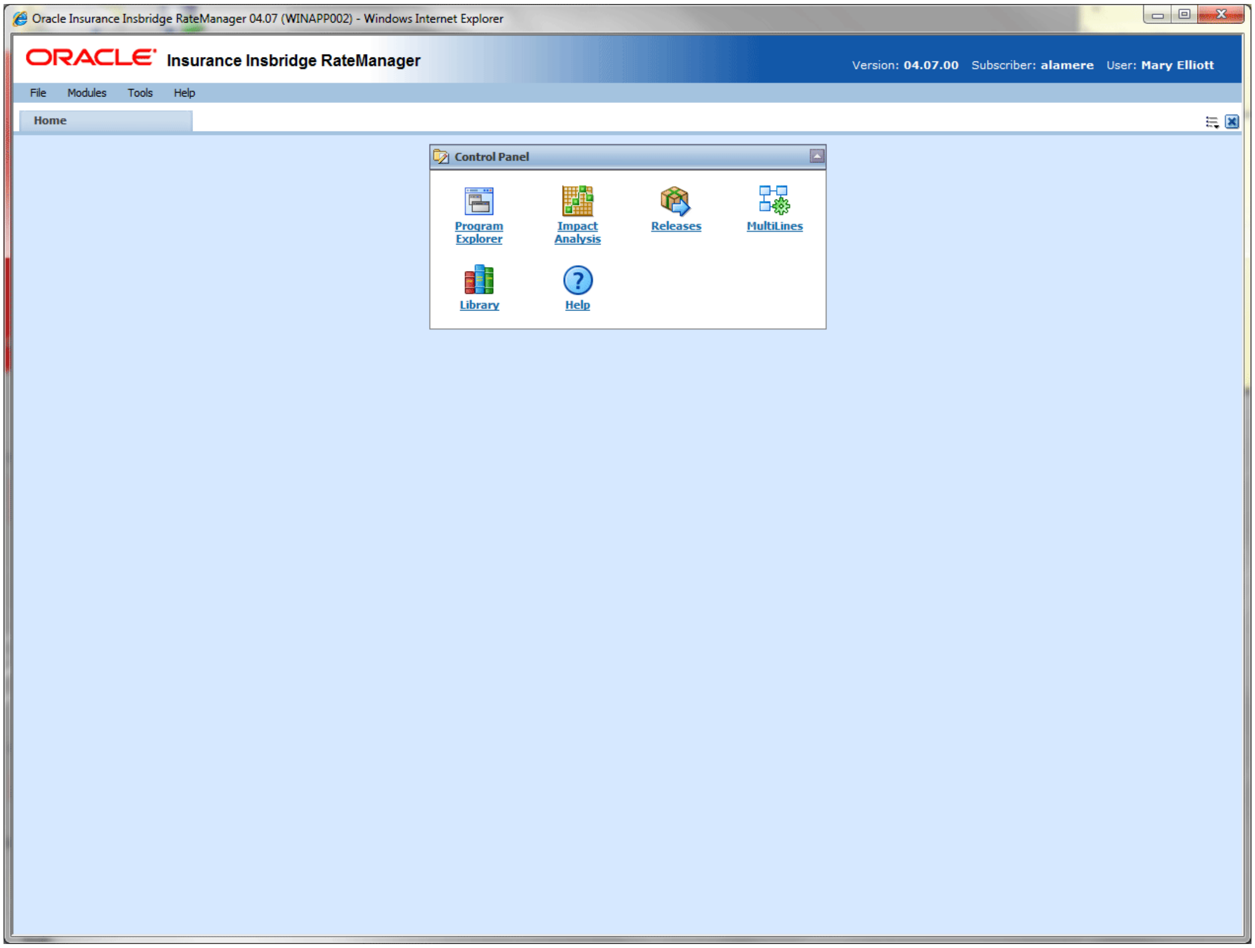

Oracle Insurance Insbridge RateManager 04.07 (WINAPP002) - Windows Internet Explorer
ORACLE Insurance Insbridge RateManager
Version: 04.07.00 Subscriber: alamere User: Mary Elliott
File
Modules
Tools
Help
Home
Control Panel
Program Explorer
Impact Analysis
Releases
MultiLines
Library
Help

Figure 4 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 hand corner of the panel. Depending upon your rights, you can have up to seven links:

- **Programs:** opens the Program Explorer.
- **Impact Analysis:** opens the Impact Analysis screen.
- **Testing:** opens the Testing area.

- **MultiLines:** opens the MultiLines area.
- **Library:** opens the Library area.
- **Releases:** opens the Releases area.
- **Help:** opens on-line help.

If you do not have rights to an area, a link will not be listed in your Control Panel. For example, in Figure 4, this user does not have rights to Releases. Her Control Panel does not contain links for that option.

Using Snaps

Inside the snap is a link to the program or element where you recently worked. Double click on the name of the program or element. A new tab will be 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 will be unchecked from your preferences and will no longer be 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 will leave the snap on the Home Tab but reduce the details. The snap top banner will be 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. You will know 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 will be 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 will drop the snap in place.

Sorting Snap Information

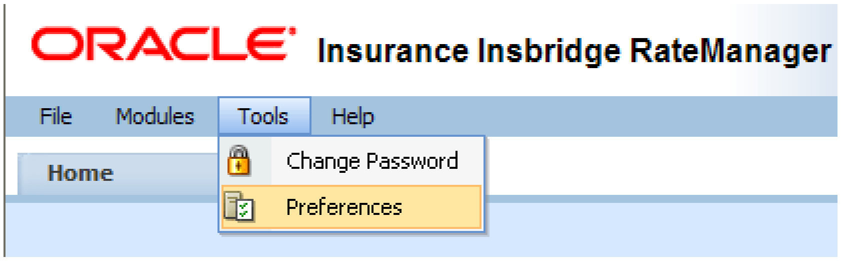
Snap information is listed by Date Last Opened, newest to oldest. To rearrange the information in the snap, click the column header. Information will be sorted either alphabetically or numerically, depending on the column header.

Adding Snaps to the Home Tab

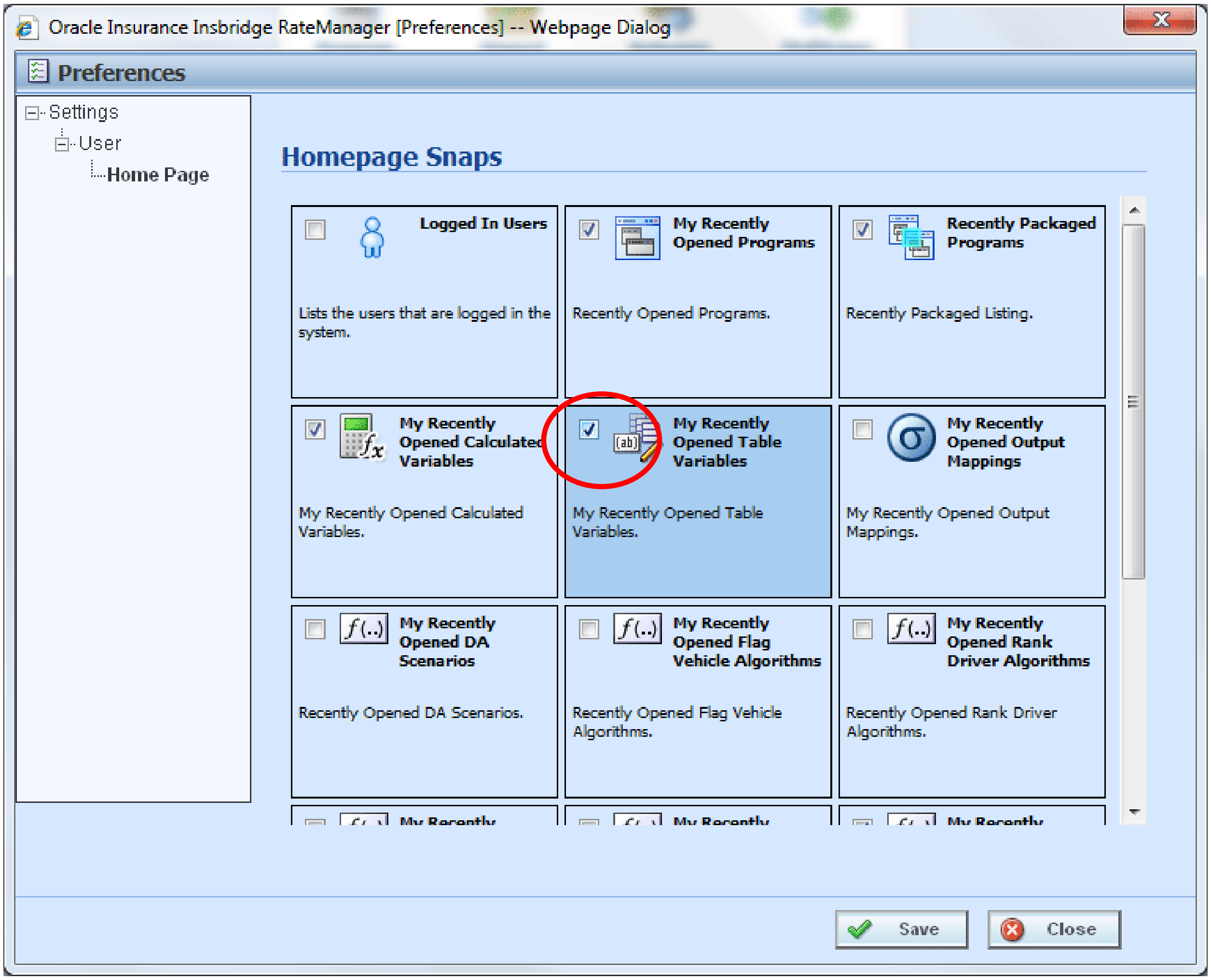
Preferences is where you select the snaps you want to place on your Home tab. The Control Panel snap does not have an option to add or remove.

To Select Your Snaps

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



2. This will open the **Preferences** window.


Oracle Insurance Insbridge RateManager [Preferences] -- Webpage Dialog
Preferences
Settings
User
Home Page
Save
Close

Homepage Snaps

☐ Logged In Users – Lists the users that are logged in the system.

☑ My Recently Opened Programs – Recently Opened Programs.

☑ Recently Packaged Programs – Recently Packaged Listing.

☑ My Recently Opened Calculated Variables – My Recently Opened Calculated Variables.

☑ My Recently Opened Table Variables – My Recently Opened Table Variables.

☐ My Recently Opened Output Mappings – My Recently Opened Output Mappings.

☐ My Recently Opened DA Scenarios – Recently Opened DA Scenarios.

☐ My Recently Opened Flag Vehicle Algorithms – Recently Opened Flag Vehicle Algorithms.

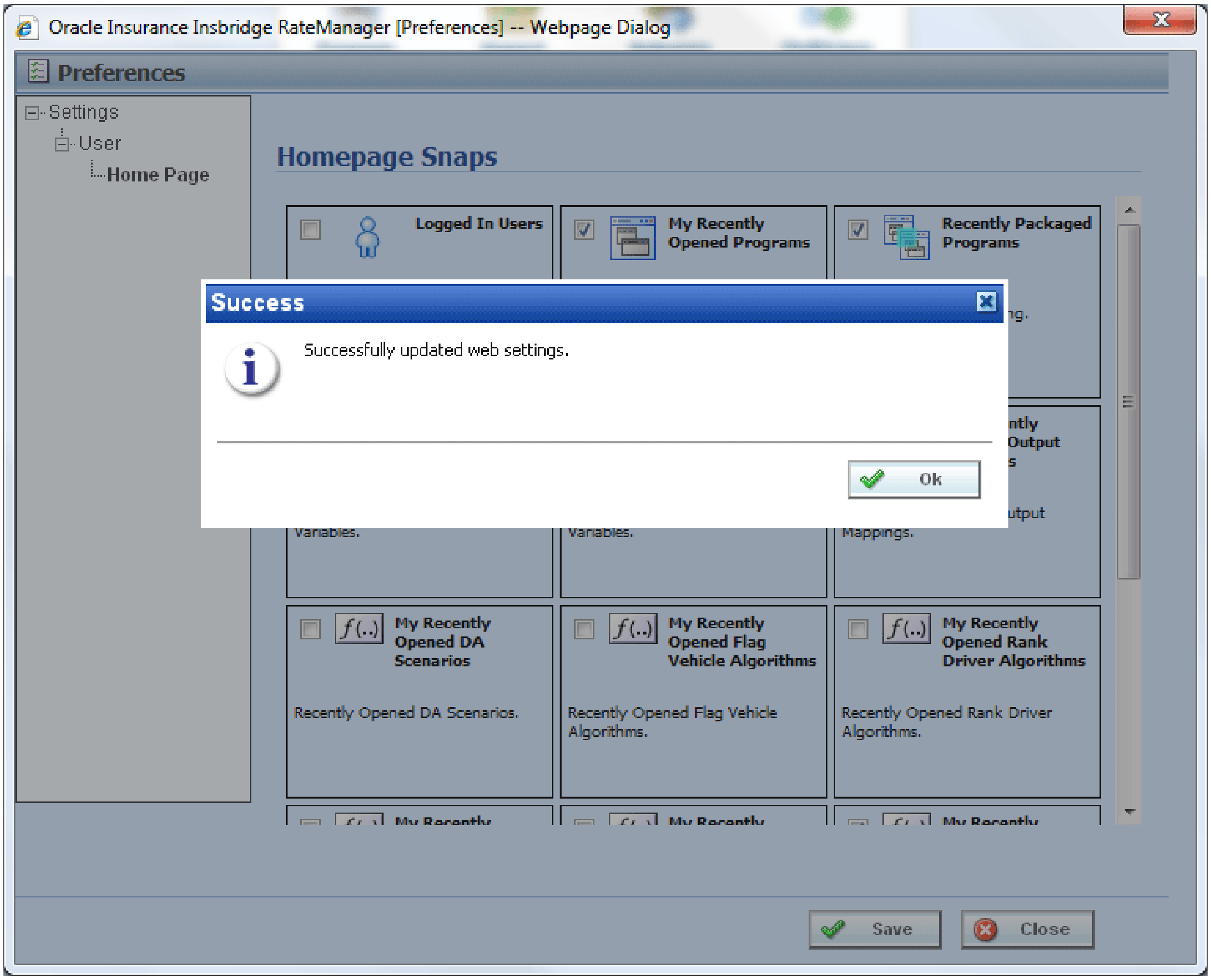
☐ My Recently Opened Rank Driver Algorithms – Recently Opened Rank Driver Algorithms.

Figure 5 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*" will display 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 will allow users to view video segments about the features and functionality contained in RateManager.


Oracle Insurance Insbridge RateManager [Preferences] -- Webpage Dialog
Preferences
Settings
User
Home Page
Homepage Snaps
Logged In Users
My Recently Opened Programs
Recently Packaged Programs
Success
Ok
My Recently Opened DA Scenarios
My Recently Opened Flag Vehicle Algorithms
My Recently Opened Rank Driver Algorithms
Save
Close

Successfully updated web settings.

Figure 6 Updated Preferences

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

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

6. Click **Close** to close the preferences window. Your HOME tab will display the snaps you have selected.

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

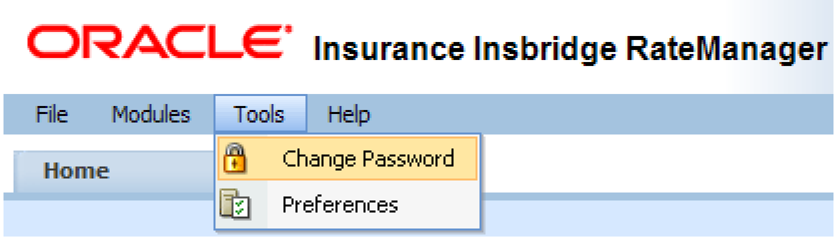
Changing Your Password

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

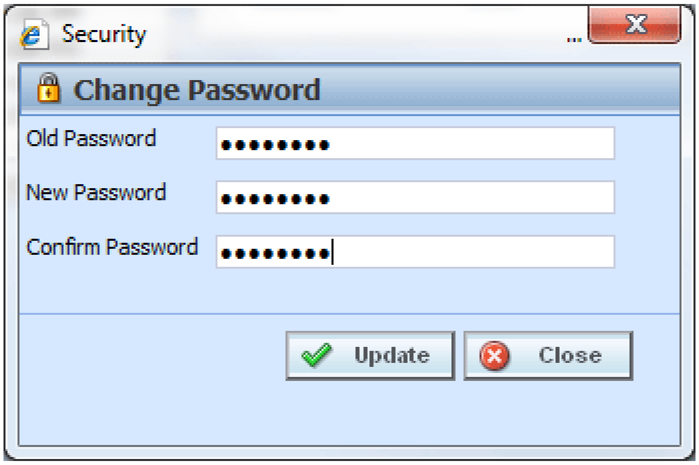
It is recommended that you change your password periodically to prevent other users from logging in and making changes using your user ID.

To Change Your Password

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



2. This will open the **Change Password** window.



Old Password ••••••••

New Password ••••••••

Confirm Password ••••••••

Figure 7 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.

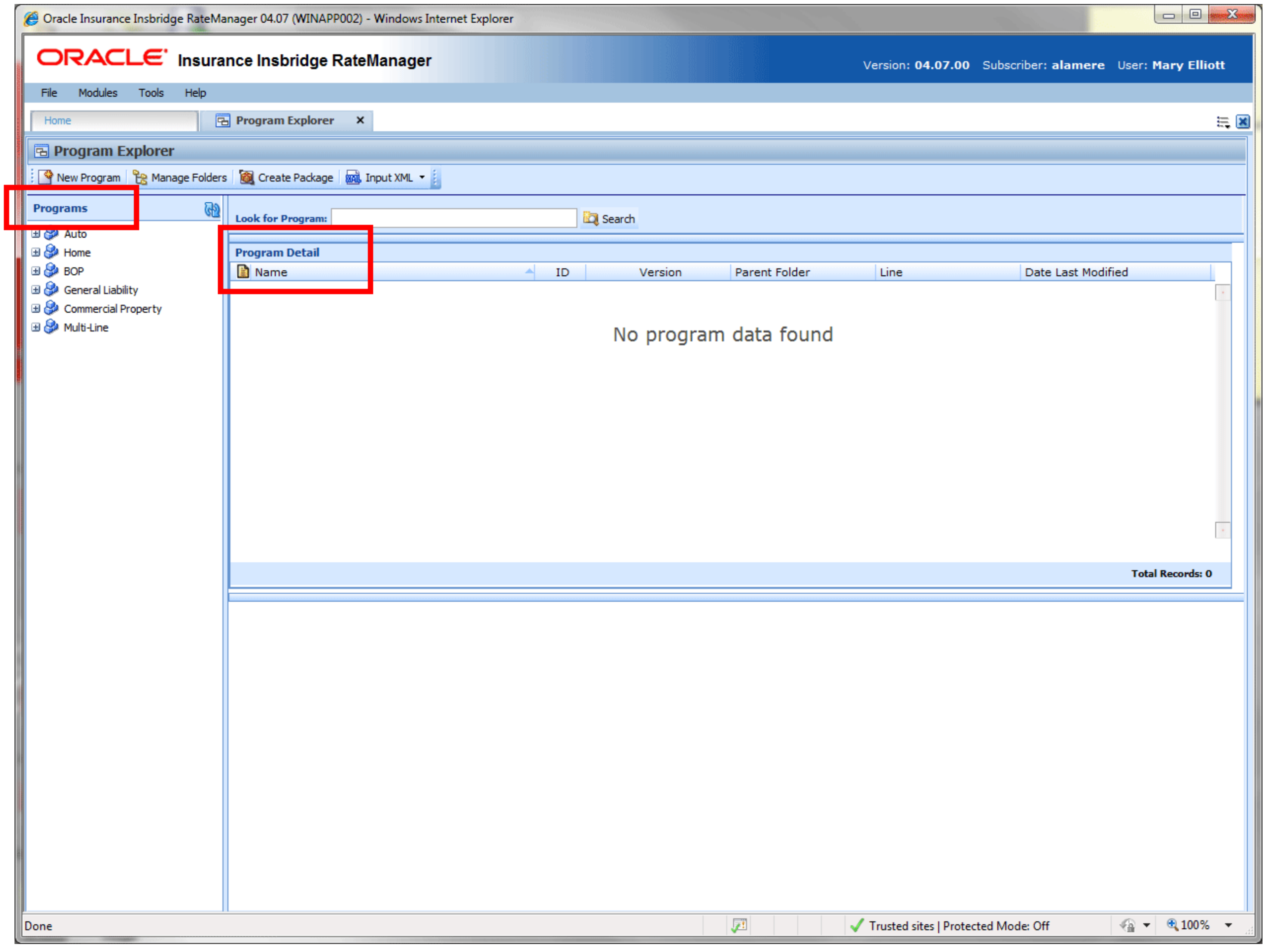
   Your password must be at least six, but fewer than twelve, alphanumeric characters.

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

Chapter 3

PROGRAM EXPLORER

Program Explorer serves as a central location for creating and managing programs and accessing program elements. Active lines of business that you have permission to view will be displayed on the left hand side of the tab. Underneath the lines will be the sublines, and then the folders, programs and globals that belong to that subline. All program elements; inputs, variables, algorithms, sequencing, output mapping and categories can be found within the Program Details tab of any selected program.



Name	ID	Version	Parent Folder	Line	Date Last Modified

No program data found

Figure 8 Program Explorer

Program Explorer is divided into two areas:

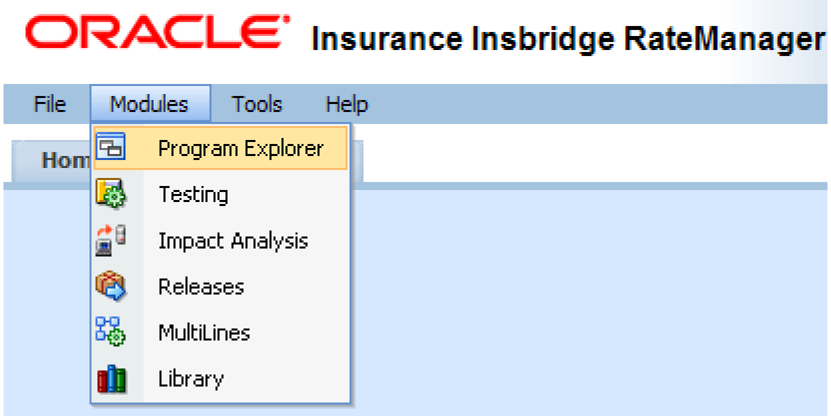
- **Programs:** Programs is a navigation column on the left hand side of the tab. Programs displays the lines of business you have rights to. As you expand the line, you will be able to see the sublines, folders, and programs currently in the LOB.
- **Program Detail:** Program detail contains the details 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.

- Select the **line of business** to view the **sublines**.
- Select the **sublines** 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.

You must have rights to view the lines and work in Program Explorer.

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 Program	**New Program:** Begins the process of creating a new program. You must select the line→subline→folder where the new program should be created.
Manage Folders	**Manage Folders:** Opens the Manage Program Folders function for the selected line→subline.
Create Package	**Create Package:** Opens the Create Package function for the selected program version.
Input XML ▾	**Input XML:** This option brings up the XML for: • **Line:** If you select the subline, or the subline→folder or

	subline→folder→program, the line inputs XML will be displayed. • **Program Version:** If you select a program version, the input XML for this program version will be displayed. This option will be available only if you select a program version.

Search:

The ***Search*** field allows you to search for a particular program. Select the subline where you want to search. Type or paste the program name you are looking for and then click **Search**. Searches are at the subline level. You cannot search one folder only. If you select a folder, the search will be done at the subline level.

As you type, matches may populate the field underneath. 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 will be displayed.

PROGRAMS

Programs is the navigation column on the left hand side of the Program Explorer. Icons will be displayed as the fields are expanded. Lines, Folders and Programs will be displayed after they are created.

PROGRAMS FIELDS

Fields are labeled with the current name assigned to them.

🖧	**Lines:** The active lines that you have access to will be displayed. Click the line to view the sublines.
□	**Sublines:** Every line has a default subline. If any user created sublines have been added, they will be listed. Click the subline to view the folders.
📁	**Subline Folders:** All subline folders are user created and must be created before you can create or apply programs. Click the folder to view the programs inside.
🗂	**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. Click the folder to view contents.
🗔	**Programs:** RateManager programs. Click the program to view all the versions.
🌐	**Globals:** Double click the Globals icon to open up the Global Details tab for the subline.
🔄	**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. Lines and Program Version Folders will not have a right click menu.

🖧	**Lines:** Has one option • **New Subline:** Allows you to create a new subline. This menu option is available to users with Subline Management rights.
□	**Sublines:** Has five options • **Create Versioning Package:** Allows you to create a global versioning package. • **Map Custom XML:** Brings up the Map Custom XML for the global inputs in this subline. • **Edit Subline:** Opens the Subline Editing window. The default subline cannot be edited. • **Clear Subline:** Removes all contents, programs and data, from the subline. • **Delete Subline:** Deletes the entire subline.
📁	**Subline Folders:** Has four options • **New Program:** Creates a new program in this folder. • **New Folder:** Brings up the New Folder screen where you can create a new folder. • **Edit Folder:** Brings up 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 five options • **Program Properties:** Brings up the Edit Program screen where you can make edits to the versioning criteria and other program information. • **Move Program:** Allows you to move the program from one folder into another within the same subline. • **Delete Program:** Deletes the selected program. • **Map Callouts:** Allows you to map callouts for the selected program. • **Program Difference Report:** Brings up the Program Difference Wizard.
🌐	**Global:** One option, Open. Opens the Global Details tab for the subline.

PROGRAM DETAIL

The Program Detail area is where the details of the element you have selected from Programs will be displayed. The item you select in the Program Folders area will determine what information is displayed in the Program Detail area.

PROGRAM DETAIL FIELDS

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

SELECTION	DISPLAYS
Line	Information for the sublines that fall under the selected line. **Icon:** There are two possible icons - Indicates the **Default** subline - Indicates **Template generated** sublines **Name:** The name of the subline. **ID:** Not used for sublines. **Version:** Not used for sublines. **Parent Folder:** Not used for sublines. **Line:** The line of business where the subline is located. **Date Last Modified:** The date the subline was last modified.
Subline	Information for the folders that were created under the selected subline. **Icon:** There will only be a folder 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. **Line:** The line of business 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:** There will only be a program 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. **Line:** The line of business where the program is located. **Date Last Modified:** The date the program was last modified.

SELECTION	DISPLAYS
Program	The program versions for the selected program. **Icon:** There are two possible icons - Indicates the **User** created program - Indicates **Template** created program **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. **Line:** The line of business 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 will indicate if 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 will be listed.
Program Version Folder	The program versions for the selected program. **Icon:** There are two possible icons - Indicates the **User** created program - Indicates **Template** created program **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. **Line:** The line of business 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 will indicate if this program version has a note.
Global	No change.

Right Click Menu Options:

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

SELECTION	DISPLAYS
Line	Information for the sublines that fall under the selected line. Sublines have five options: • **Create Versioning Package:** Allows you to create a global versioning package. • **Map Custom XML:** Brings up the Map Custom XML for the global inputs in this subline. • **Edit Subline:** Opens the Subline Editing window. The default subline cannot be edited. • **Clear Subline:** Removes all contents, programs and data, from the subline. • **Delete Subline:** Deletes the entire subline.
Subline	Information for the folders that were created under the selected subline. Folders have two options: • **Edit Folder:** Brings up the Edit Folders screen where you can edit the folder. • **Delete Folder:** Deletes the selected folder.
Folder	Information for the programs that were created under the selected folder. Programs have six options: • **Program Properties:** Brings up the Edit Program screen where you can make edits to the versioning criteria and other program information. • **Move Program:** Brings up the Move Program screen where you can move a program within the subline. • **Delete Program:** Deletes the selected program. • **Map Callouts:** Brings up the Callout Mapping screen where you can enter the mappings for program callouts or library callouts. • **Program Difference Report:** Brings up the Program Version Report. • **See Program Versions:** Refreshes the tab with the program versions.
Program and Program Version Folders	The program versions for the selected program. Program versions have 11 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:** Brings up 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. You must have rights to delete program versions. • **Test Program:** Opens Test Program, which allows you to test and debug a program within RateManager. • **Map Custom XML:** Brings up the Input Variable Mapping screen for this

	program version. This screen is where you can enter, view, import or upload input mappings. ● **Notes:** Enter, edit and view notes to the program version. ● **Program Version Report:** Brings up the Program Version Report. ● **Unassign From Program Version Folder:** removes a program version from the program version folder where it was assigned. ● **Assign To Program Version Folder:** Brings up the Assign Version to Program Version Folder screen. Here you can assign a program version to a folder. Template generated programs will also have: ● **Reconcile:** This option will reconciled two program versions.
🌐 **Global**	No Change.

LINES AND SUBLINES OF BUSINESS

Lines are managed in the Manage Lines option of the Tools menu. The Manage Lines option is available to system administrators and users with manage line permissions. If Lines are not set up prior to working in Program Explorer, please contact your system administrator.

SUBLINES

Lines are separated into *sublines*. Sublines are classifications that fall in between lines and program folders. Sublines allow you to distinguish program sources and also keeps the globals unique within that subline. Globals are subline specific and are not shared across sublines. There are two types of sublines, user created and template generated.

- Indicates **User Created** sublines

- Indicates **Template generated** sublines

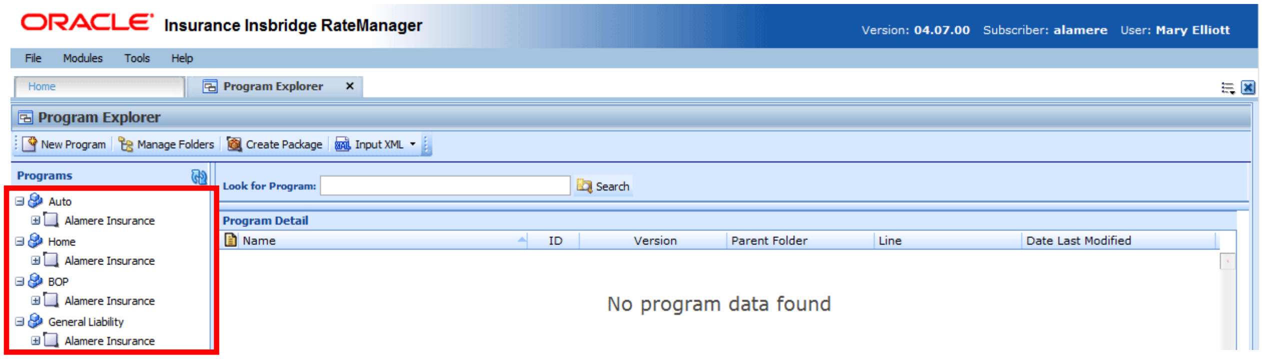
User Created Sublines

A user created subline is a subline that was created by a user with subline management permissions. User created sublines hold user created programs, imported programs and public XML programs. Most user created sublines can be edited, cleared or deleted. The exception is the default subline which can only be cleared.

Each line automatically has one user created subline placed underneath it. This subline is named with your subscriber description and is the *default* subline. The default subline cannot be edited or deleted and is always listed first.

The subscriber description was entered by the system administrator when your subscriber account was set up. If your subscriber description is Alamere, then under every line will be a subline named Alamere.

NOTE: *If you want to change the default subline name, please contact your system administrator and ask to have your subscriber description changed.*



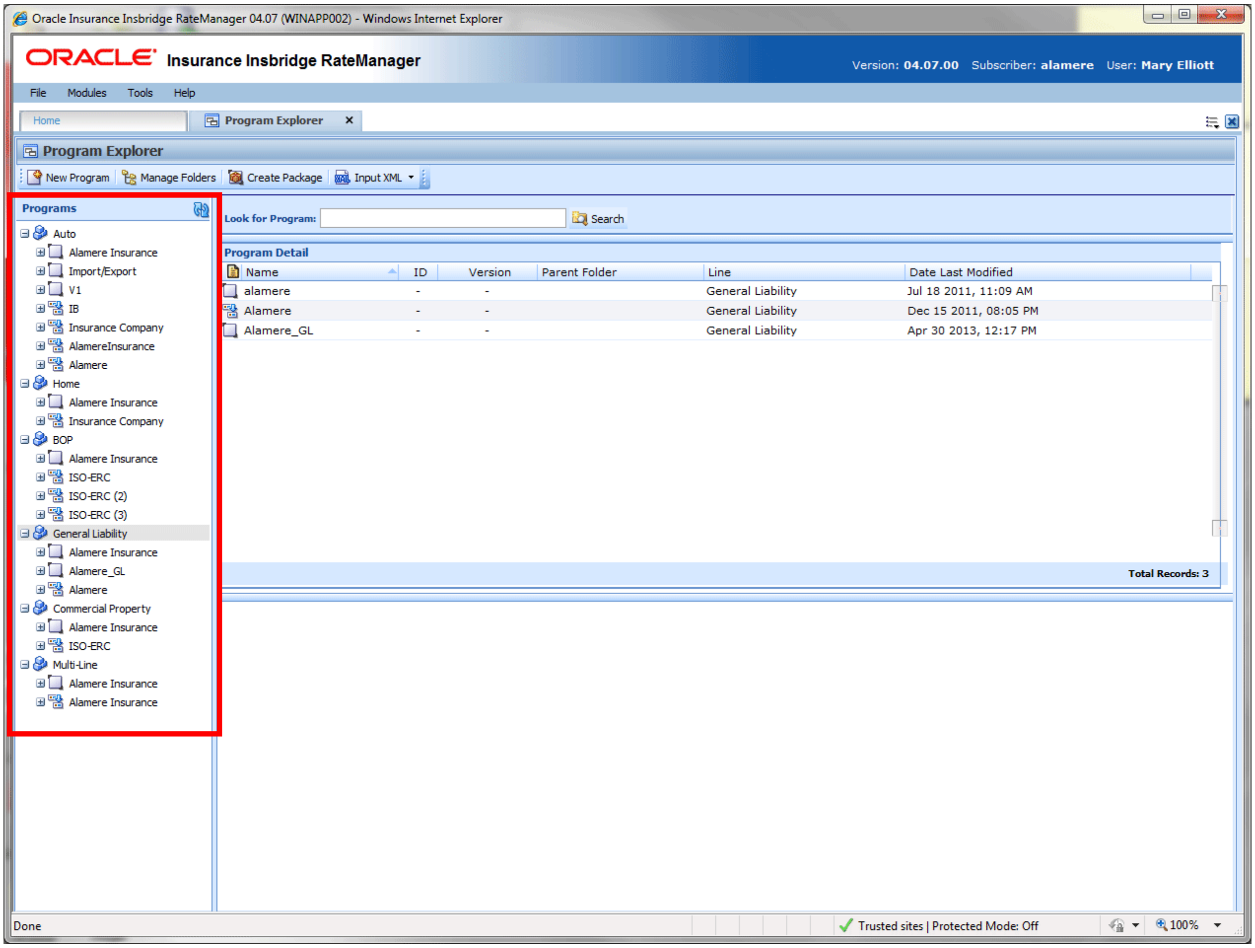
No program data found

Figure 9 Default Sublines under Each Line of Business

Template Sublines

Until you have applied a *template*, any new programs that you create are placed under the default subline or a user created subline. Templates are programs that have been uploaded into the RateManager Library. For more on templates, imports and libraries, please see Library.

When a template is applied, a new subline is created for each template *source*. The source is the creator of the template and also the name of the new subline. For example, if a template created by the Content Services group is brought into the General Liability line, a new subline called Content Services is placed under the General Liability line.


Oracle Insurance Insbridge RateManager 04.07 (WINAPP002) - Windows Internet Explorer
ORACLE Insurance Insbridge RateManager
Version: 04.07.00 Subscriber: alamere User: Mary Elliott
File
Modules
Tools
Help
Home
Program Explorer
Program Explorer
New Program
Manage Folders
Create Package
Input XML
Look for Program:
Search
Done
Trusted sites | Protected Mode: Off
100%

Programs

- Auto
  - Alamere Insurance
  - Import/Export
  - V1
  - IB
  - Insurance Company
  - AlamereInsurance
  - Alamere
- Home
  - Alamere Insurance
  - Insurance Company
- BOP
  - Alamere Insurance
  - ISO-ERC
  - ISO-ERC (2)
  - ISO-ERC (3)
- General Liability
  - Alamere Insurance
  - Alamere_GL
  - Alamere
- Commercial Property
  - Alamere Insurance
  - ISO-ERC
- Multi-Line
  - Alamere Insurance
  - Alamere Insurance

Program Detail

Name	ID	Version	Parent Folder	Line	Date Last Modified
alamere	-	-		General Liability	Jul 18 2011, 11:09 AM
Alamere	-	-		General Liability	Dec 15 2011, 08:05 PM
Alamere_GL	-	-		General Liability	Apr 30 2013, 12:17 PM

Total Records: 3

Figure 10 Program Explorer Populated with Lines, User Created Sublines and Template Sublines

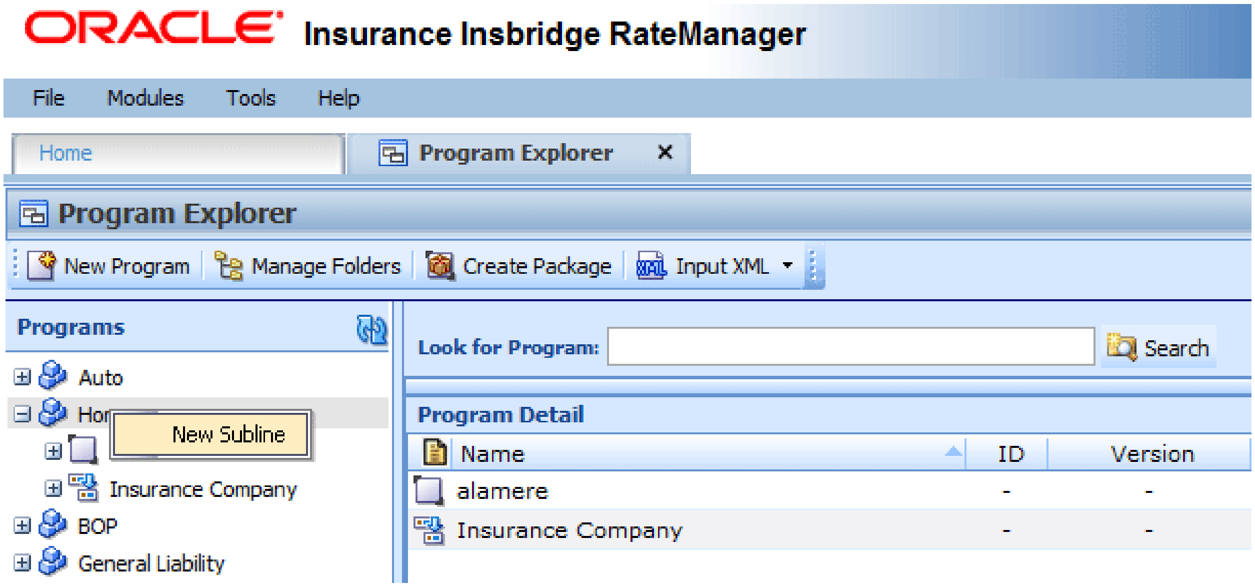
Each line can have multiple template generated sublines. Multiple template generated sublines can be created for each source. You can create a new template generated subline for every template applied if you need to. You also may be required to create a new template generated subline if there are category or input differences between existing templates and the new template you want to apply. Template sublines names cannot be edited. Template sublines can be cleared and deleted.

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

New User Created Sublines

New user created sublines can be created at any time. A new template generated subline is created only when a template is applied from the Library. Any new subline is displayed under the corresponding line of business. For example, if you applied an auto template, a new subline will be created under the auto line. The new subline will be named after the source. If you select the Auto line and create a new subline, the new user created subline will be listed there.

1. Select the line of business where you want to create a new user created subline.



Name	ID	Version
alamere	-	-
Insurance Company	-	-

Figure 11 Right Click for New Subline

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

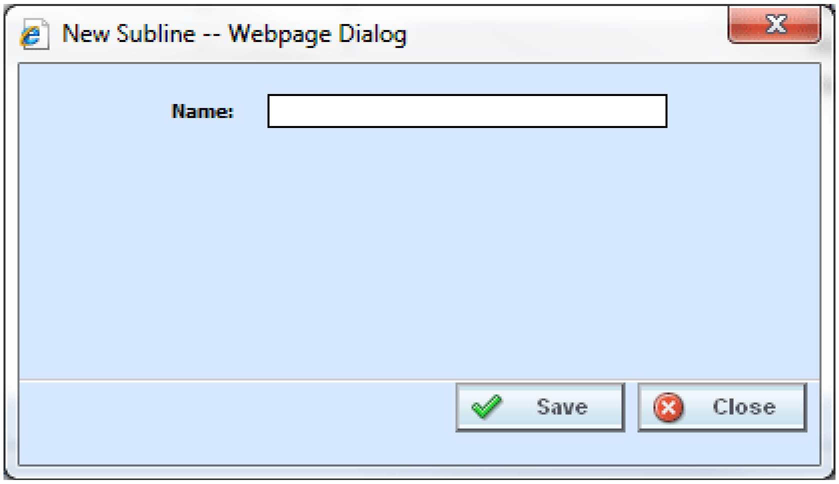
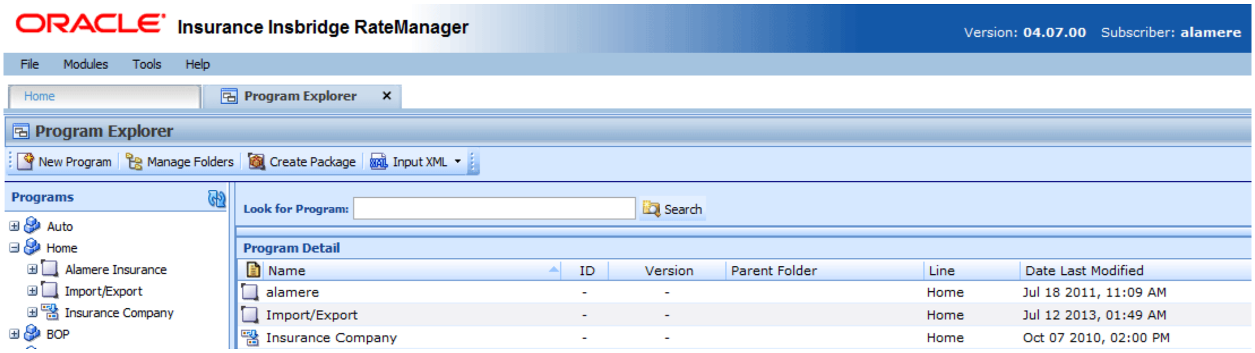


Figure 12 New User Created Subline

3. Enter in a Name.
4. Click Save to save your entry. After the Program Explorer tab refreshes, the new subline will be listed.



Name	ID	Version	Parent Folder	Line	Date Last Modified
alamere	-	-		Home	Jul 18 2011, 11:09 AM
Import/Export	-	-		Home	Jul 12 2013, 01:49 AM
Insurance Company	-	-		Home	Oct 07 2010, 02:00 PM

Figure 13 Program Explorer with New User Created Subline

Editing Sublines

Only user created sublines can be edited. Editing allows for the name to be changed.

1. Select the line of business where you want to edit a user created subline.

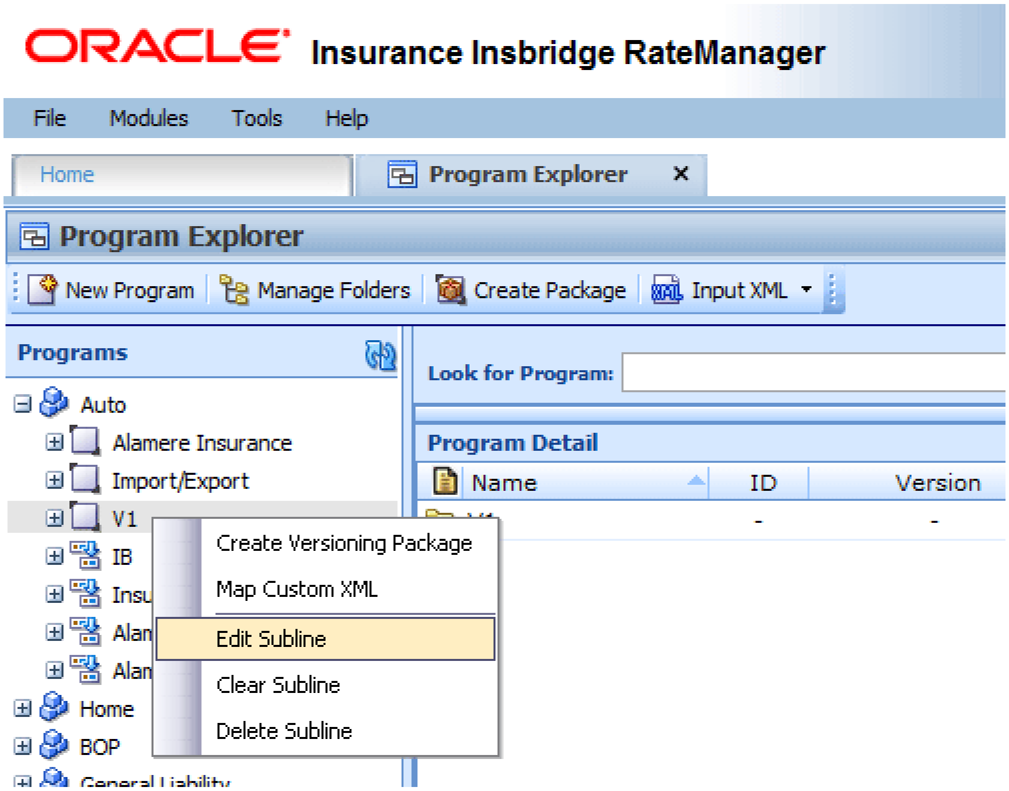
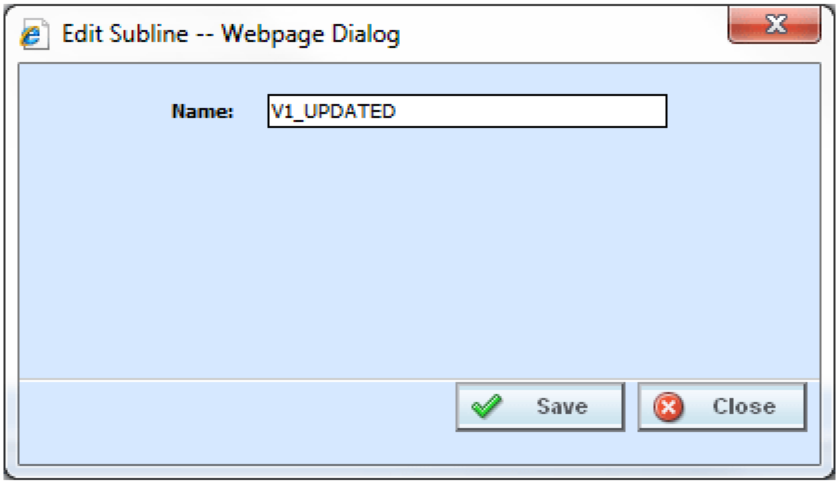


Figure 14 Editing a Subline

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



Name: V1_UPDATED

Figure 15 Editing a Subline on the Edit Subline Screen

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

Clearing Sublines

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

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

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

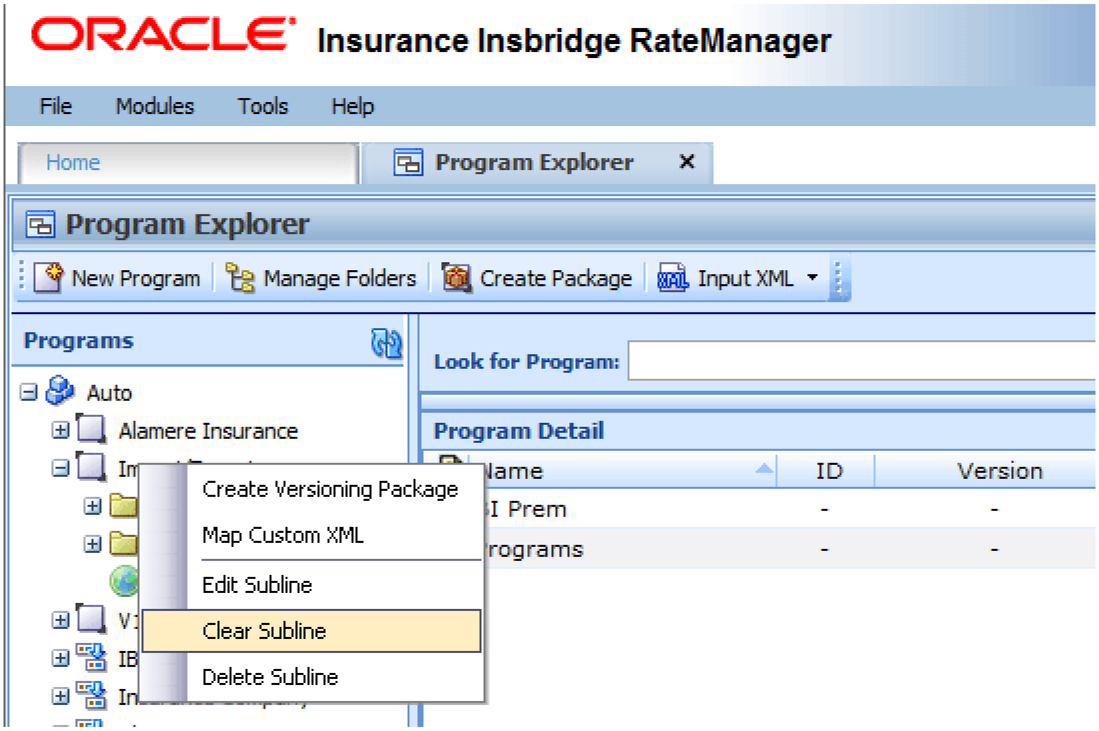
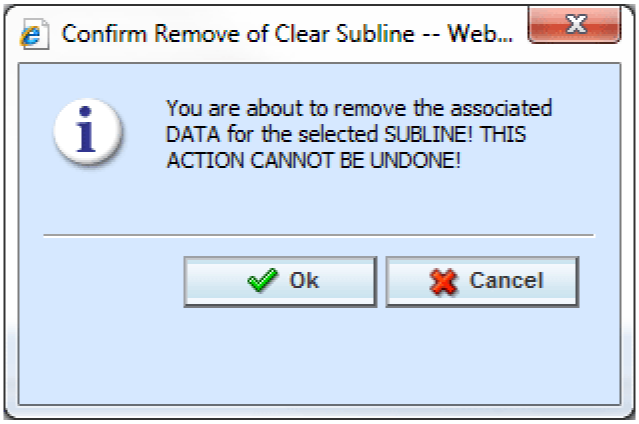


Figure 16 Clearing a Subline

1. Select the line of business where you want to clear a subline.
2. Right click and select Clear Subline.

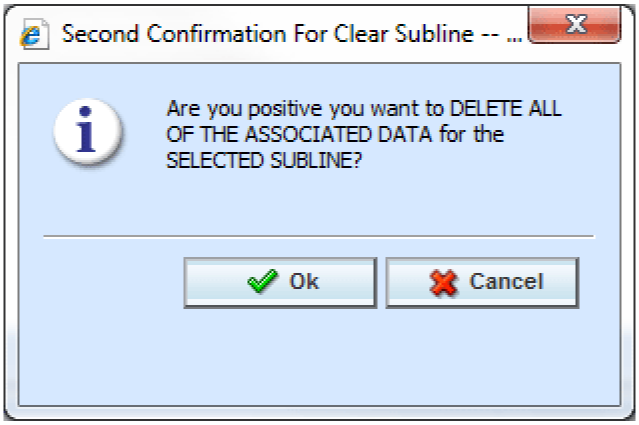


Confirm Remove of Clear Subline -- Web...

You are about to remove the associated DATA for the selected SUBLINE! THIS ACTION CANNOT BE UNDONE!

Figure 17 First Confirmation to Clear the Subline

3. Click OK to clear the subline.



Second Confirmation For Clear Subline -- ...

Are you positive you want to DELETE ALL OF THE ASSOCIATED DATA for the SELECTED SUBLINE?

Figure 18 Second Confirmation to Clear the Subline

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

Deleting Sublines

Deleting a subline removes the subline from the system. Everything in the subline, including folders, programs, and data will be deleted.

Deleting a subline should be done only when nothing in the subline is needed. Make sure this is the action you want to take. Once a subline is deleted, there is no undo. You will be asked to confirm your selection twice before the subline is deleted. The default subline in any line cannot be deleted. Any other user created subline or template generated subline can be deleted at any time.

You must have subline management permissions to delete a subline. If the delete subline option is grayed out, you may not have permissions or the subline is the default subline for the line.

1. Select the line of business where you want to delete a subline.

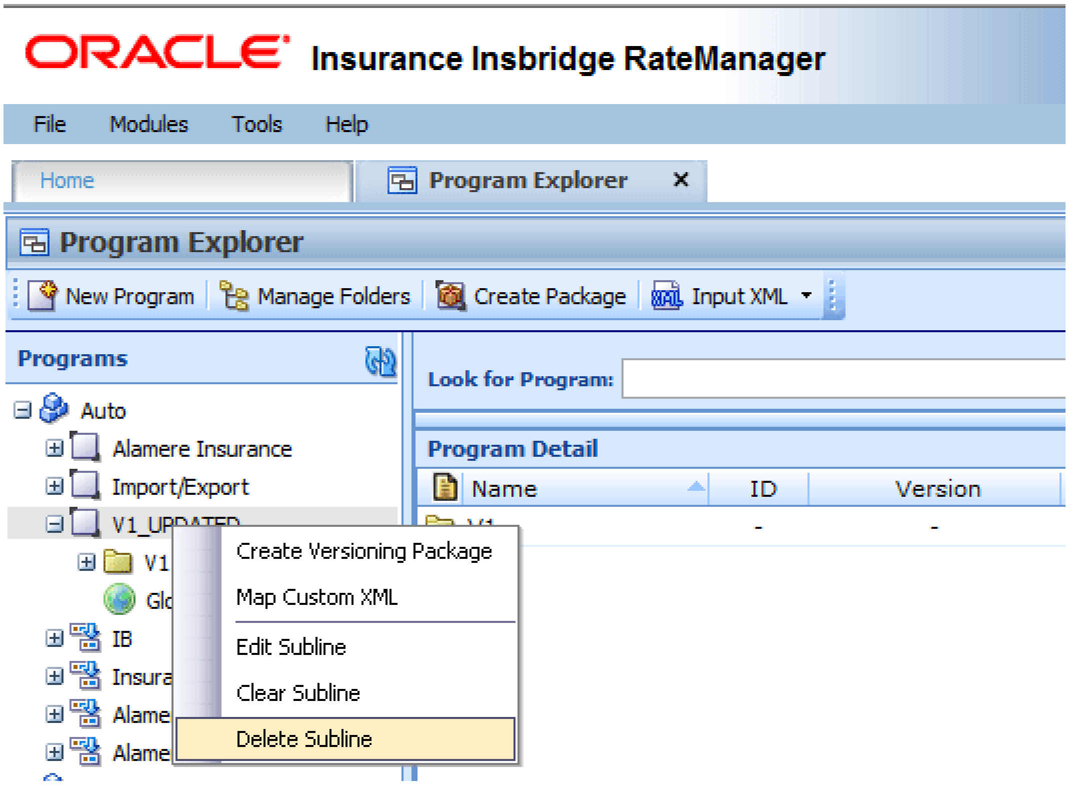
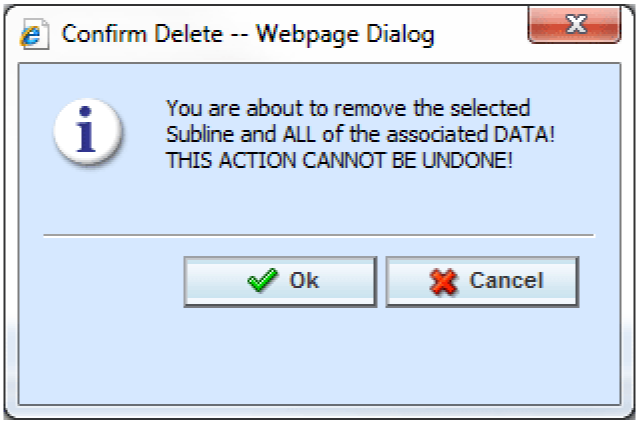


Figure 19 Deleting a Subline

2. Right click and select Delete Subline.

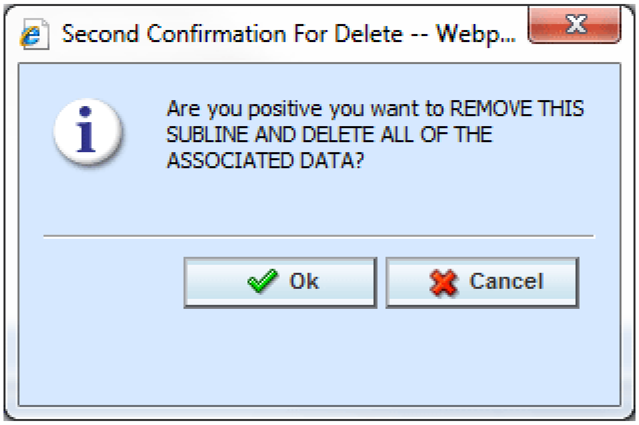


Confirm Delete -- Webpage Dialog

You are about to remove the selected Subline and ALL of the associated DATA! THIS ACTION CANNOT BE UNDONE!

Figure 20 First Confirmation to Delete the Subline

3. Click OK to delete the subline.



Second Confirmation For Delete -- Webp...

Are you positive you want to REMOVE THIS SUBLINE AND DELETE ALL OF THE ASSOCIATED DATA?

Figure 21 Second Confirmation to Delete the Subline

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

Chapter 5

FOLDERS

The RateManager file management system uses multiple folders and subfolders. This is a multi-level set up that allows you to place an unlimited number of folders and subfolders underneath a subline. 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:

- **Subline Folders:** are the standard folders that reside under the subline. These folders are visible at all times. Programs reside under subline folders. At least one subline 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 subline folder.

Subline Folders:

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

Subline 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 will be 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 subline folder where they were created. Each program version folder can contain multiple subfolders and/or multiple program versions.

The program version folder structure will be replicated in each program created in the subline folder. This means that if you create a program version folder called “archive” under the subline 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

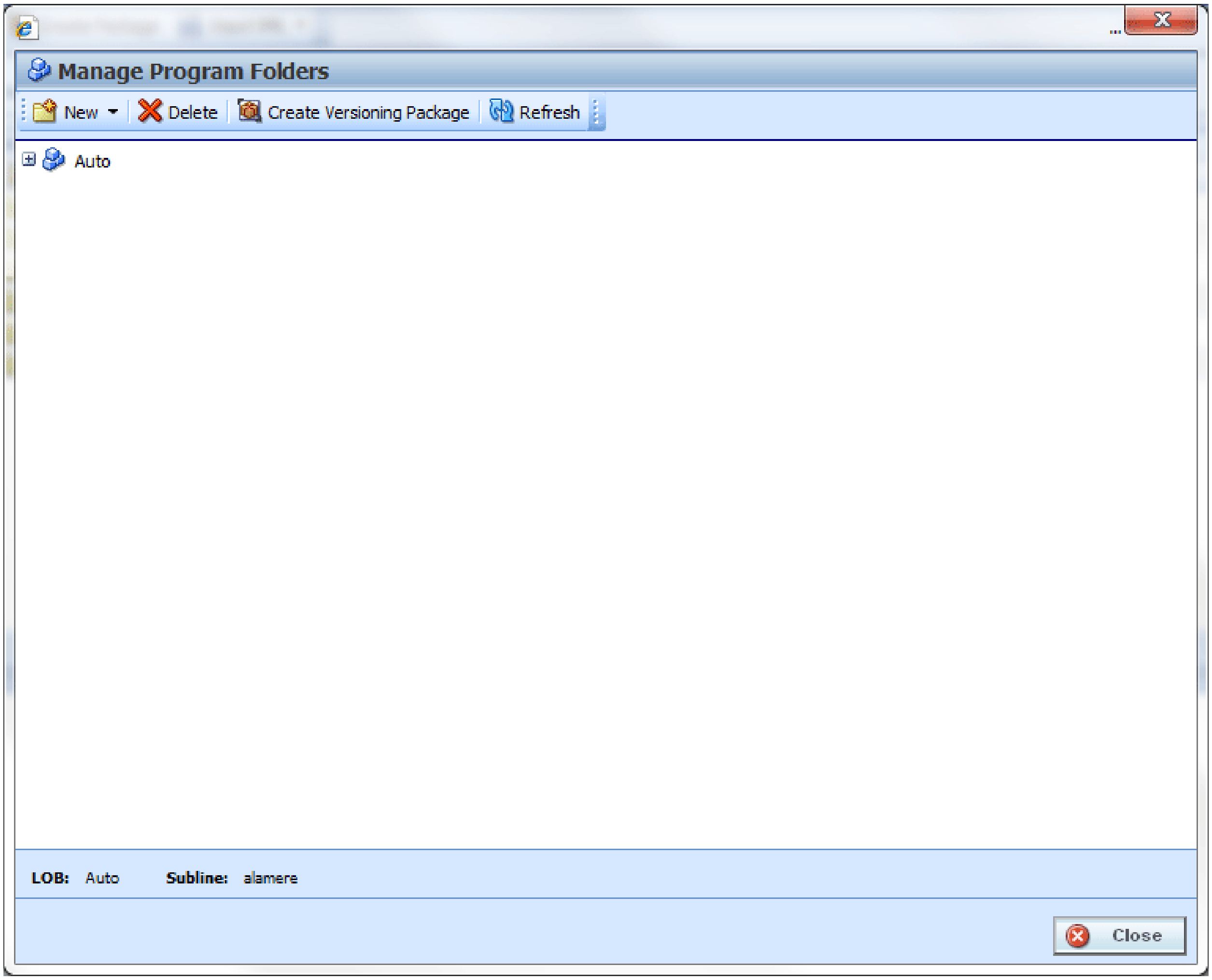
Subline folders and program version folders are managed on the Manage Program Folders screen. The Manage Program Folders screen allows you to:

- Create a New Folder
- Rename a Folder
- Delete a Folder
- Create a Versioning Package: see Creating a Global Versioning Package.

Navigating to Manage Program Folders:

Subline 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 line and the subline before the Manage Program Folders screen is shown.

1. In Program Explorer, select the line and then select the subline you need.
2. Click **Mange Folders** to open the Manage Program Folders screen.

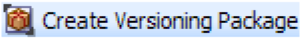


LOB: Auto Subline: alamere

Figure 22 Blank 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 will not be displayed. For example, if you do not have the rights to delete a folder, the delete option will not be displayed.

	New: Creates a new folder.
	Delete: Deletes the currently selected folder.
	**Create Versioning Package:** Creates a global versioning package that is used by SoftRater for automatic version selection. See Creating a Global Versioning Package.
	Refresh: Refreshes the screen with the current information.

MANAGE PROGRAM FOLDERS FIELDS

The Manage Program Folders screen contains all current subline folders and program version folders. Folders expand out into a tree view. The line of business is displayed at the top. Expand the line to see the subline. Expand the subline to see the folders underneath and so on. At the bottom of the screen is an information bar that contains the Line and Subline 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 will depend upon the type folder selected. The options for a subline 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.

Subline Folders has 4 options:

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

Program Version Folders has 3 options:

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

Program Version Folders cannot have subline folders underneath.

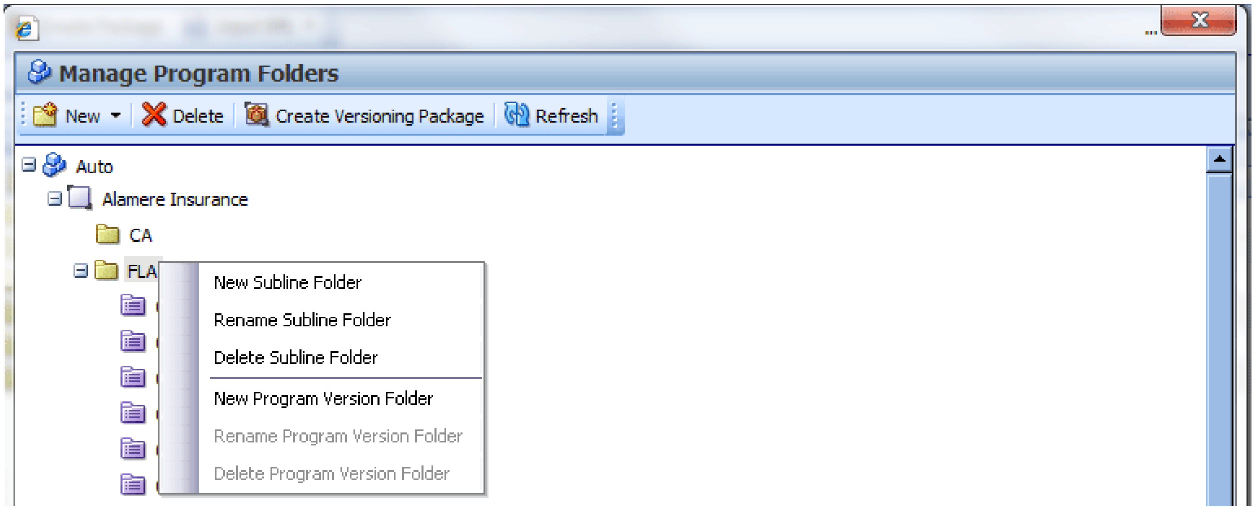


Figure 23 Right Click Menu for Subline Program Folders

Manage Program Folders has six right click menu options:

- **New Subline Folder:** Creates a new subline folder under the folder you selected.
- **Rename Subline Folder:** Allows you to rename the folder.
- **Delete Subline 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:** Allows you to rename the folder.
- **Delete Program Version Folder:** Deleted the selected folder, This option may not be active if you do not have the rights.

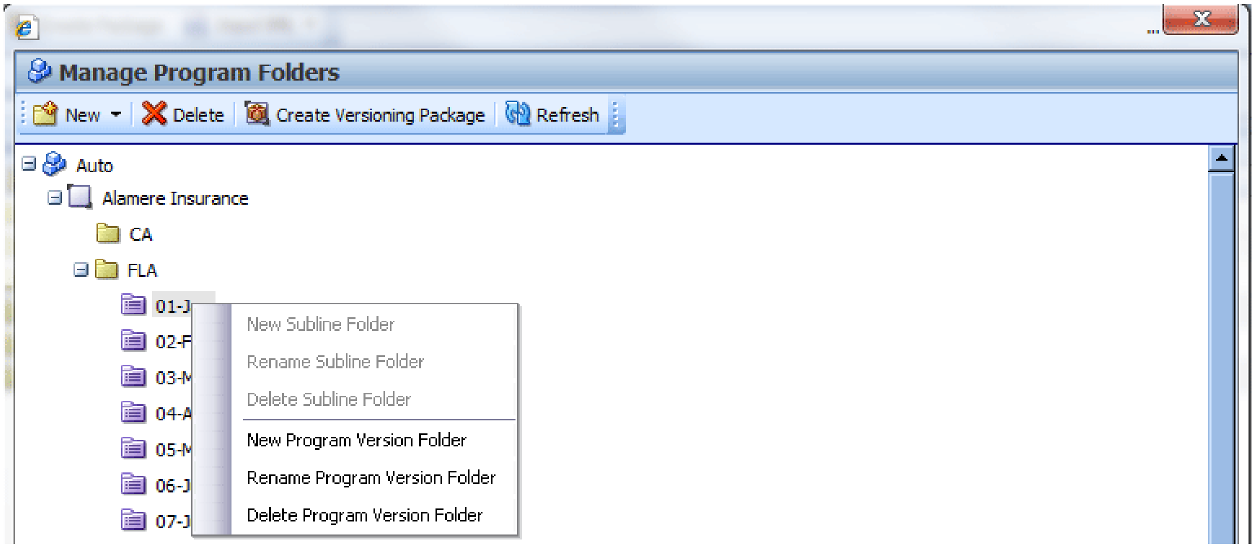
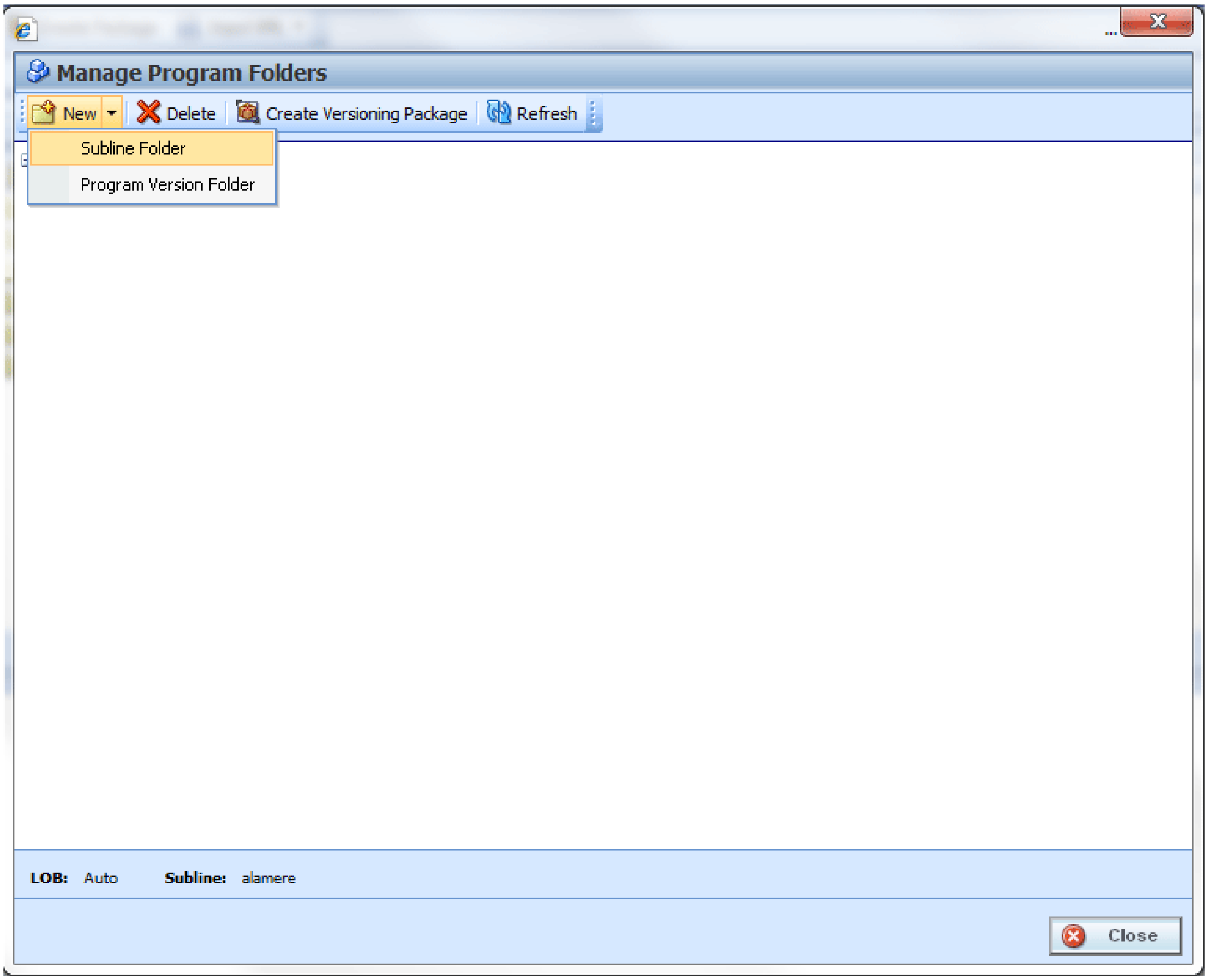


Figure 24 Right Click Menu for Manage Program Version Folders

CREATING A SUBLINE FOLDER

Folders have to be created before programs. If no folders have been created, the Parent folder will be the line of business. The subline will be directly underneath. The first folder created must fall under the subline folder. For any other folder, you can select the folder pathway.

1. Select the subline you want. If available, select the folder where you want to create a new subline folder.



LOB: Auto Subline: alamere

Figure 25 Creating New Folder

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

3. A blank box is listed under the folder you selected. 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.

A new subline folder can also be created by right clicking the subline, and selecting New Subline Folder.

Renaming a Subline Folder

Folders can be renamed at any time in the Manage Program Folder screen or on the Program Explorer→Edit Folder feature.

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

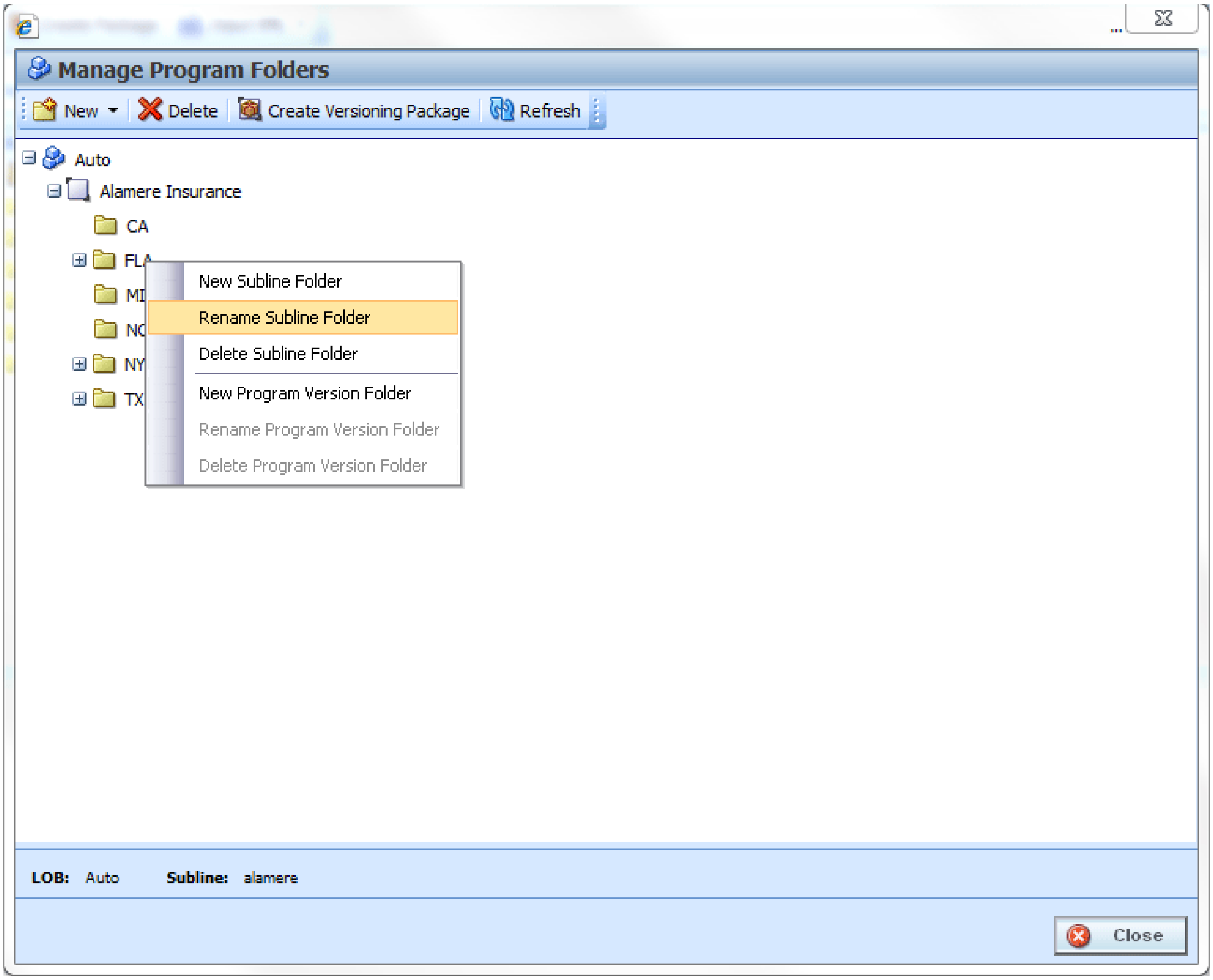


Figure 26 Renaming a Subline Folder

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

3. The subline 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.

NOTE: *Oracle Insurance recommends that the Folder Name not contain any spaces and that the Folder Names be unique, regardless of the pathway. For example, in Figure 26 Renaming a Subline Folder, no other folder under the Auto line of business, regardless of subline, should be called DallasCounty, the name of the newly entered folder. Names cannot be duplicated within the subline folder level.*

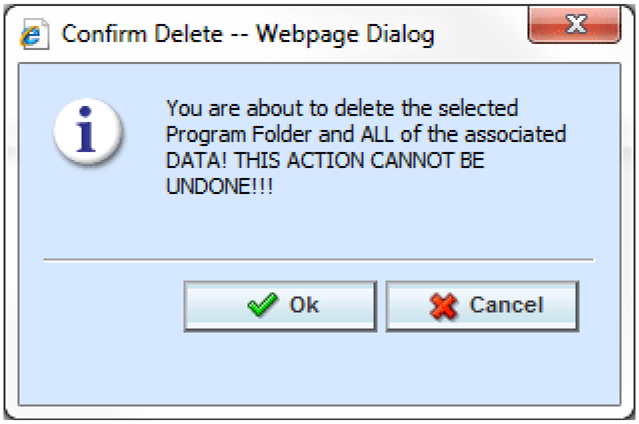
DELETING A SUBLINE FOLDER

If a subline 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 hand 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 will have to move the subfolder to a visible folder. All visible parent folders are located on the left hand 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 Subline Folder** from the menu.

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

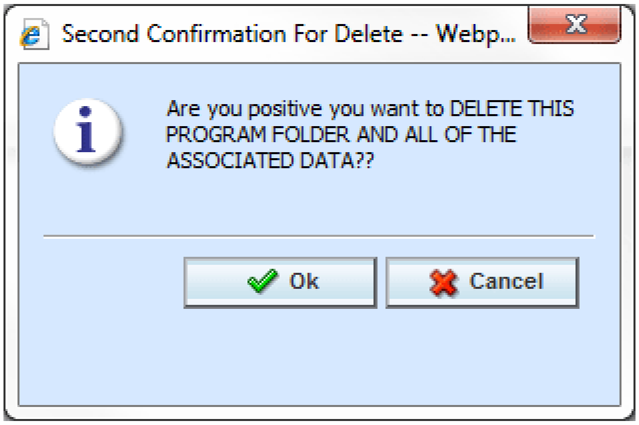


You are about to delete the selected Program Folder and ALL of the associated DATA! THIS ACTION CANNOT BE UNDONE!!!

Figure 27 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.



Are you positive you want to DELETE THIS PROGRAM FOLDER AND ALL OF THE ASSOCIATED DATA??

Figure 28 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.

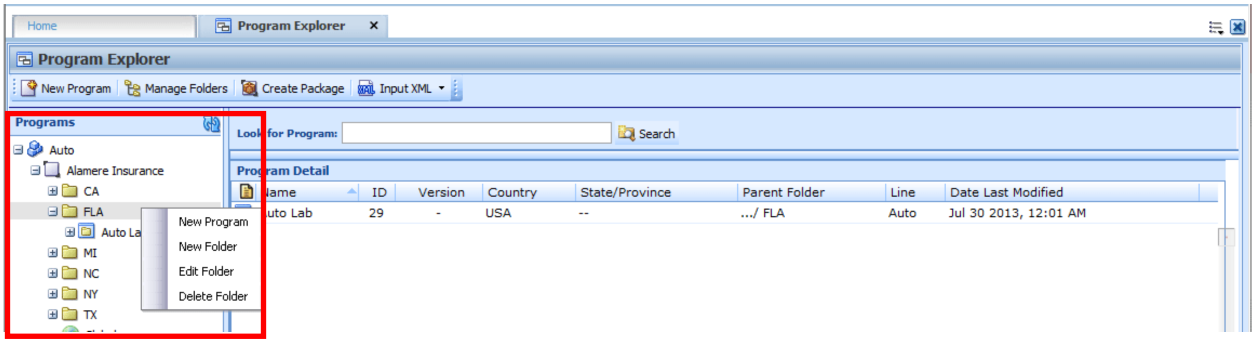
EDITING A SUBLINE FOLDER

Editing a subline folder allows you to change the folder name and move the folder to a different location within the same subline. You cannot move a folder outside of the subline where it was created. When you change the pathway of any folder, the folders underneath are changed as well.

On the Program Explorer tab, the Programs area and the Program Detail area both have right click menus containing subline folder options. The Edit Folder option is only found in the right click menus on the Program Explorer tab.

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

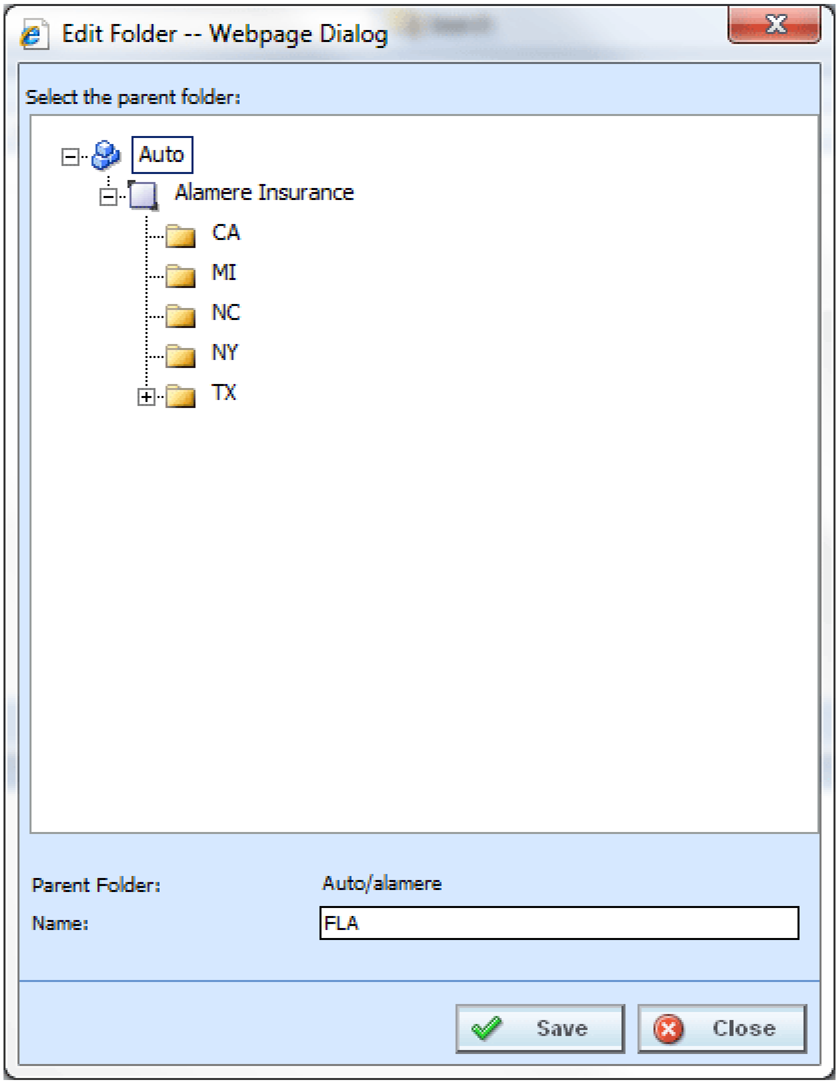
1. In the Program Explorer tab, select the folder you want to edit and right click. Select **Edit Folder**.



ame	ID	Version	Country	State/Province	Parent Folder	Line	Date Last Modified
uto Lab	29	-	USA	--	.../ FLA	Auto	Jul 30 2013, 12:01 AM

Figure 29 Folder Options Found on the Right Click Menu of Programs

2. The **Edit Folder** screen is shown.



Parent Folder: Auto/alamere

Name: FLA

Figure 30 Edit Folder

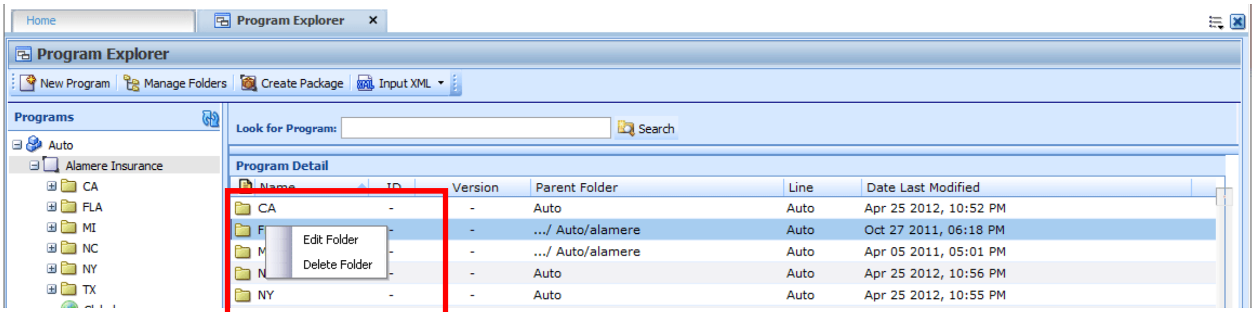
3. Make your changes and click **Save**. The Edit Folder screen closes and the Program Explorer tab refreshes with the changes you made. Click **Close** to go back to the Program Explorer tab without saving changes.

Moving a Subline Folder

Moving a subline folder allows you to change the folder location within the same subline. 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.

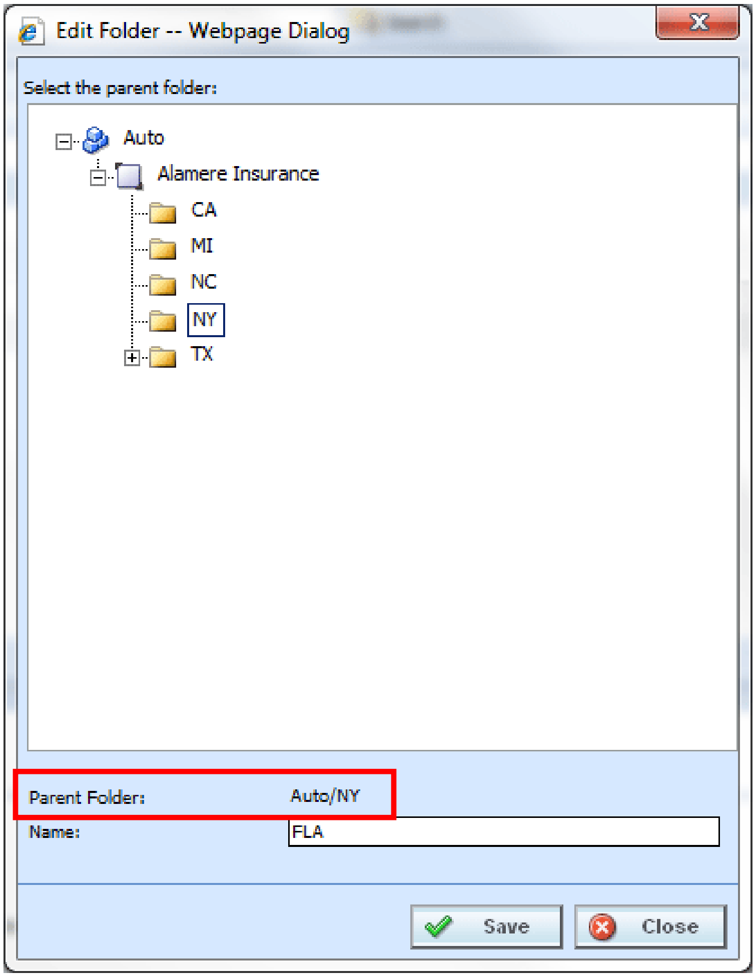
1. In the Program Explorer tab, select the folder you want to move and right click. Select **Edit Folder**.



Name	ID	Version	Parent Folder	Line	Date Last Modified
CA	-	-	Auto	Auto	Apr 25 2012, 10:52 PM
F	-	-	.../ Auto/alamere	Auto	Oct 27 2011, 06:18 PM
M	-	-	.../ Auto/alamere	Auto	Apr 05 2011, 05:01 PM
N	-	-	Auto	Auto	Apr 25 2012, 10:56 PM
NY	-	-	Auto	Auto	Apr 25 2012, 10:55 PM

Figure 31 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.



Parent Folder: Auto/NY

Name: FLA

Figure 32 Moving a Subline Folder

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

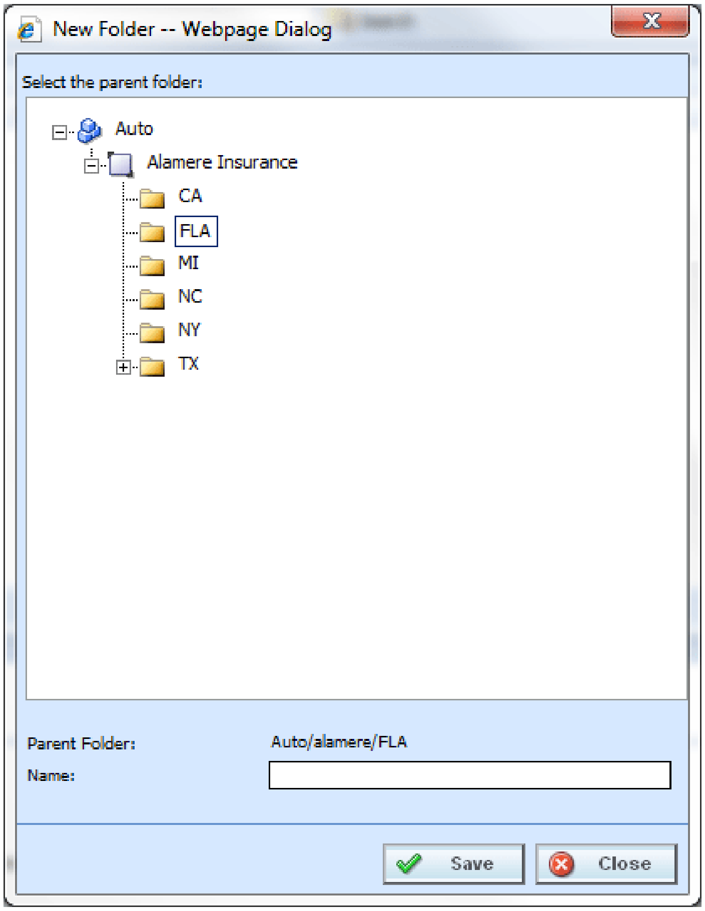
Creating a Subline Folder on Program Explorer

A new subline folder also can be created using the right click menu. On the Program Explorer tab, the Programs area has a right click menu containing subline folder options. You cannot create the first subline folder using this option. This option can be used only after a subline folder has been created in the Manage Folders area.

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

1. In the Program Explorer tab→Programs area right click any folder in the line of business→Subline you want. Select **New Folder** and the New Folder screen is shown.

2. Select the folder pathway. The pathway is displayed in the Parent Folder field. On the tree view, the final destination where the new folder will be placed will have a box around it.


New Folder -- Webpage Dialog

Select the parent folder:

Auto
Alamere Insurance
CA
FLA
MI
NC
NY
TX

Parent Folder: Auto/alamere/FLA

Name:

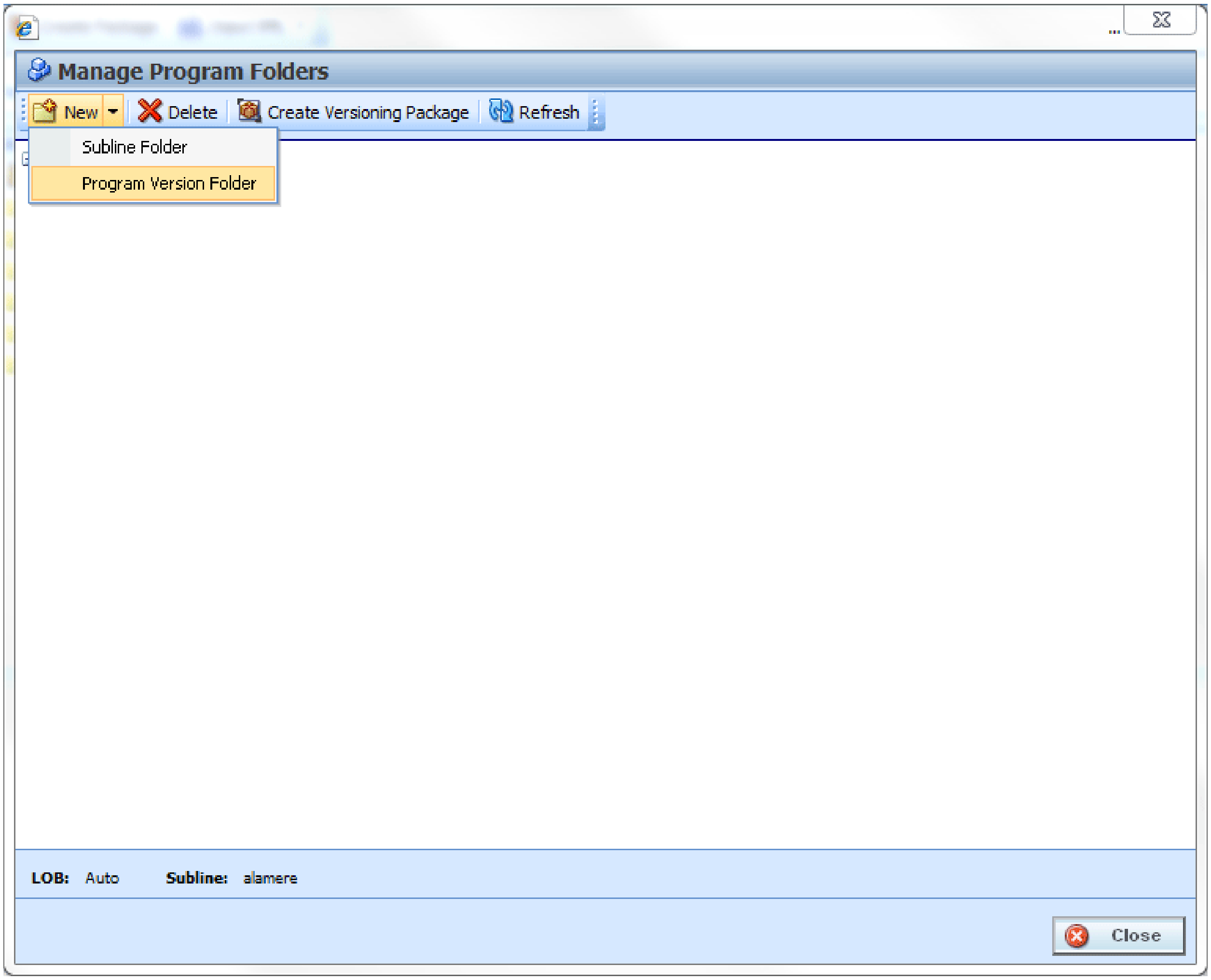
Figure 33 Creating New Folder

3. Enter in a **Name**. Click **Save** to save your folder. The screen refreshes and the new folder is displayed in the Program Explorer tab.

Creating a Program Version Folder

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

1. Select the line of business and the subline you want. Select the subline 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.



LOB: Auto Subline: alamere

Figure 34 Creating New Program Version Folder

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

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

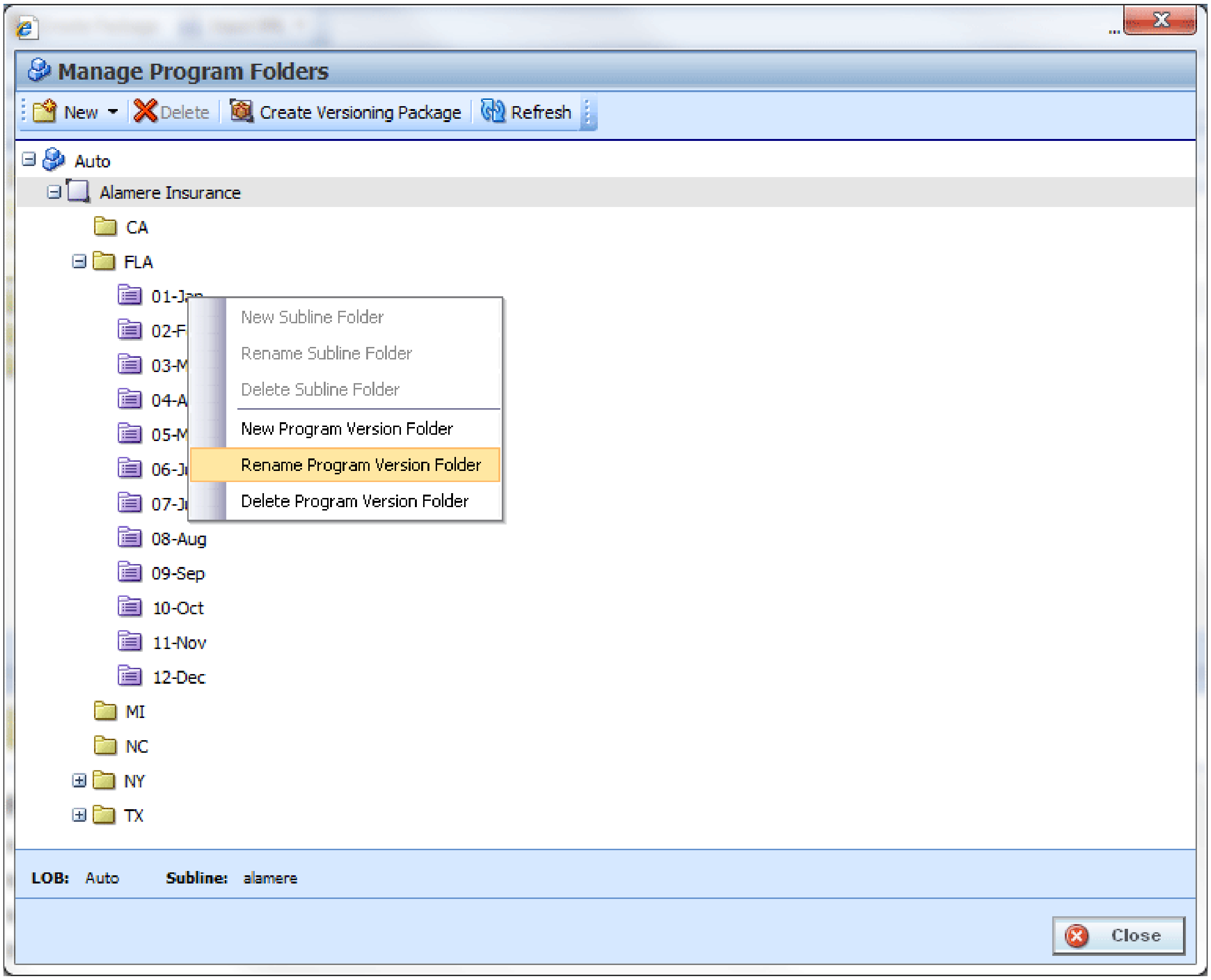
4. Click out of the box to exit the field and save your entry. The screen refreshes and the new folder is displayed in the Manage Program Folders screen.

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.



LOB: Auto Subline: alamere

Figure 35 Renaming a Subline 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.

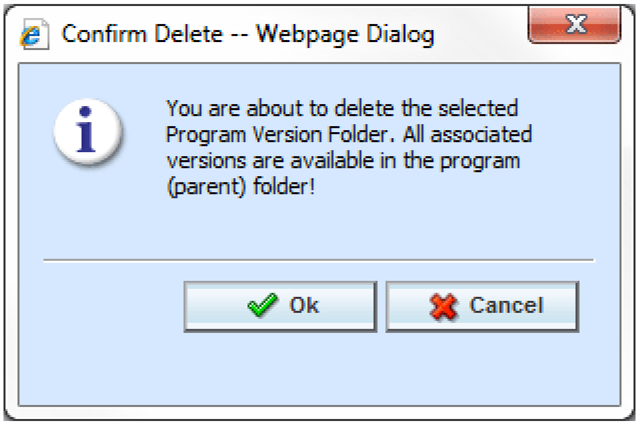
NOTE: *Oracle Insurance recommends that the Folder Name not contain any spaces and that the Folder Names be unique, regardless of the pathway. Names cannot be duplicated within the subline folder level.*

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 does, programs do not get deleted but any program version subfolders are removed.

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

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



You are about to delete the selected Program Version Folder. All associated versions are available in the program (parent) folder!

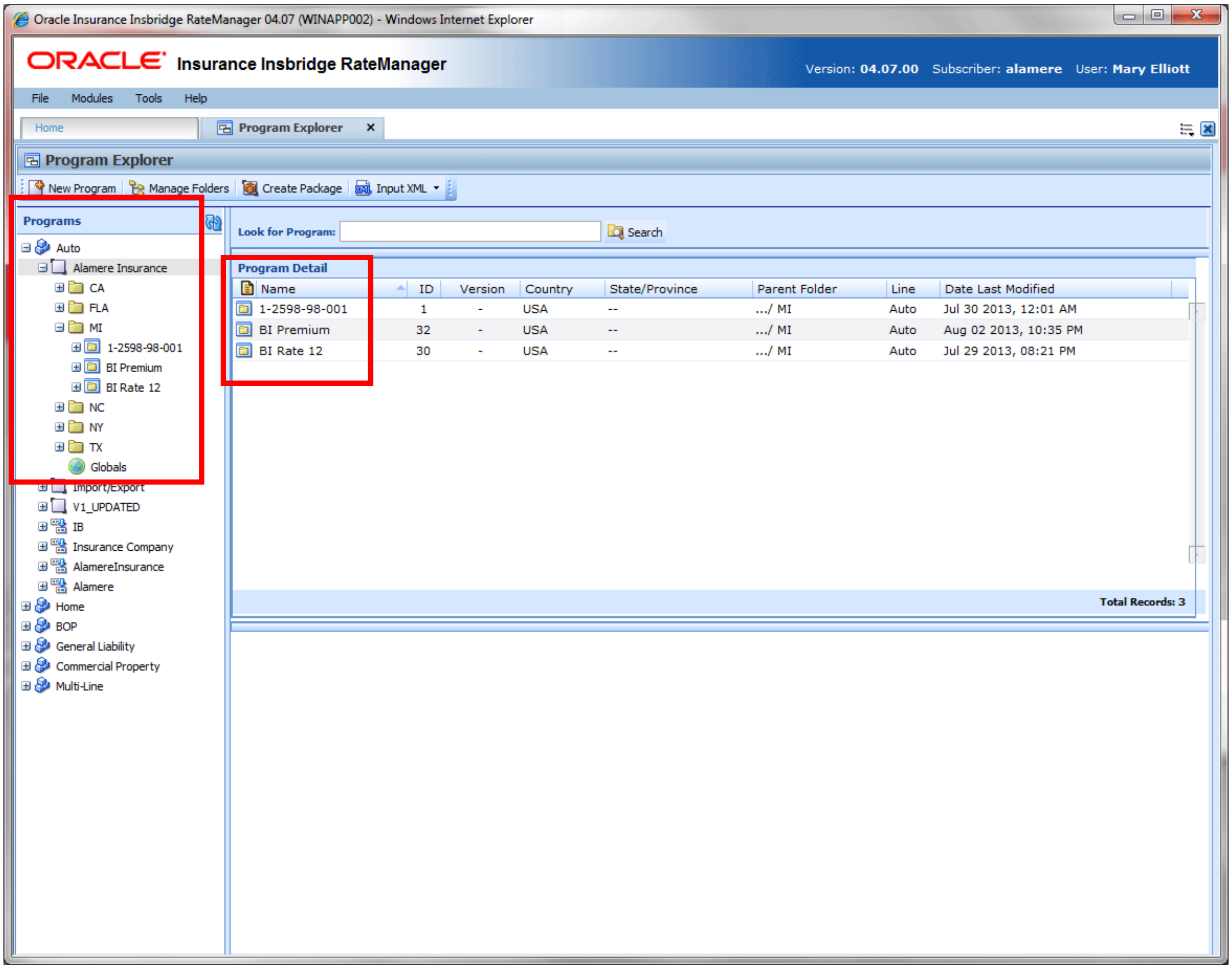
Figure 36 Deleting a Program Version Folder

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

4. The folder and any program version subfolders will be deleted.

PROGRAMS

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



Name	ID	Version	Country	State/Province	Parent Folder	Line	Date Last Modified
1-2598-98-001	1	-	USA	--	.../ MI	Auto	Jul 30 2013, 12:01 AM
BI Premium	32	-	USA	--	.../ MI	Auto	Aug 02 2013, 10:35 PM
BI Rate 12	30	-	USA	--	.../ MI	Auto	Jul 29 2013, 08:21 PM

Figure 37 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.

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 associated with a template are locked as well.

Sublines can contain both standard and template type programs, except for the default subline. The default subline will only contain standard programs. You will not be allowed to apply any template programs into your default subline.

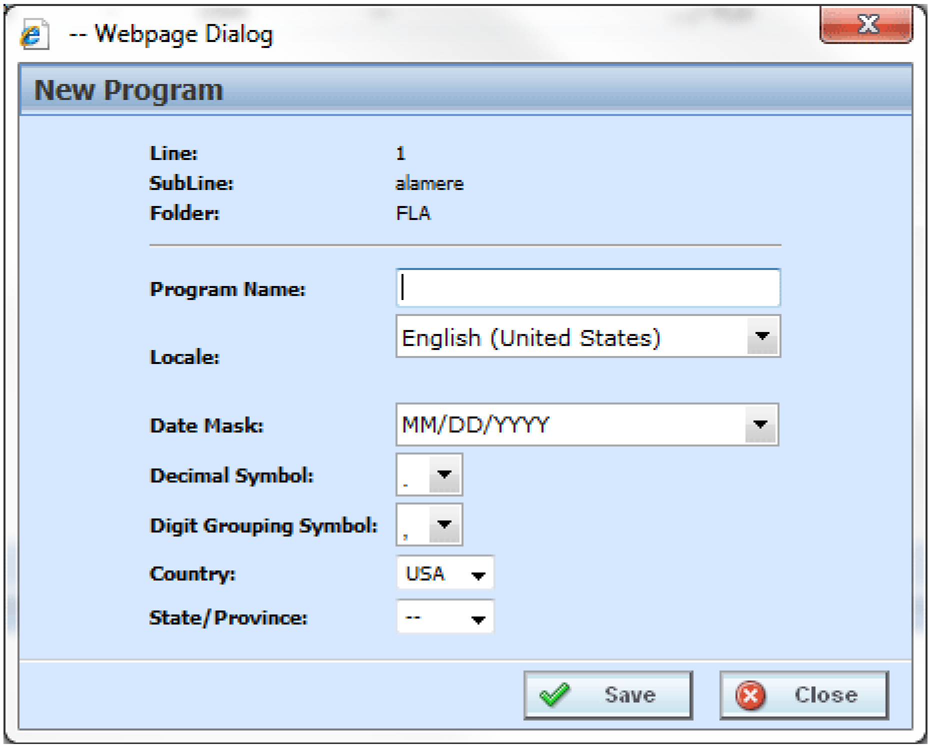
RateManager supports an unlimited number of programs and all programs are subline specific. For example, creating **Program 1** for **Carrier A** under the default subline does not create **Program 1** for **Carrier A** under another subline.

Creating a Program

New programs can be added at any time to any folder. Template created sublines can have standard programs added to them. There is no limit to the number of programs you can enter.

1. In Programs, select the subline and folder where you want to create a new program.

2. Click **New Program** from the top bar menu or right click 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.



Line: 1
SubLine: alamere
Folder: FLA

Program Name:
Locale: English (United States)
Date Mask: MM/DD/YYYY
Decimal Symbol: .
Digit Grouping Symbol: ,
Country: USA
State/Province: --

Figure 38 Creating a New Program

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

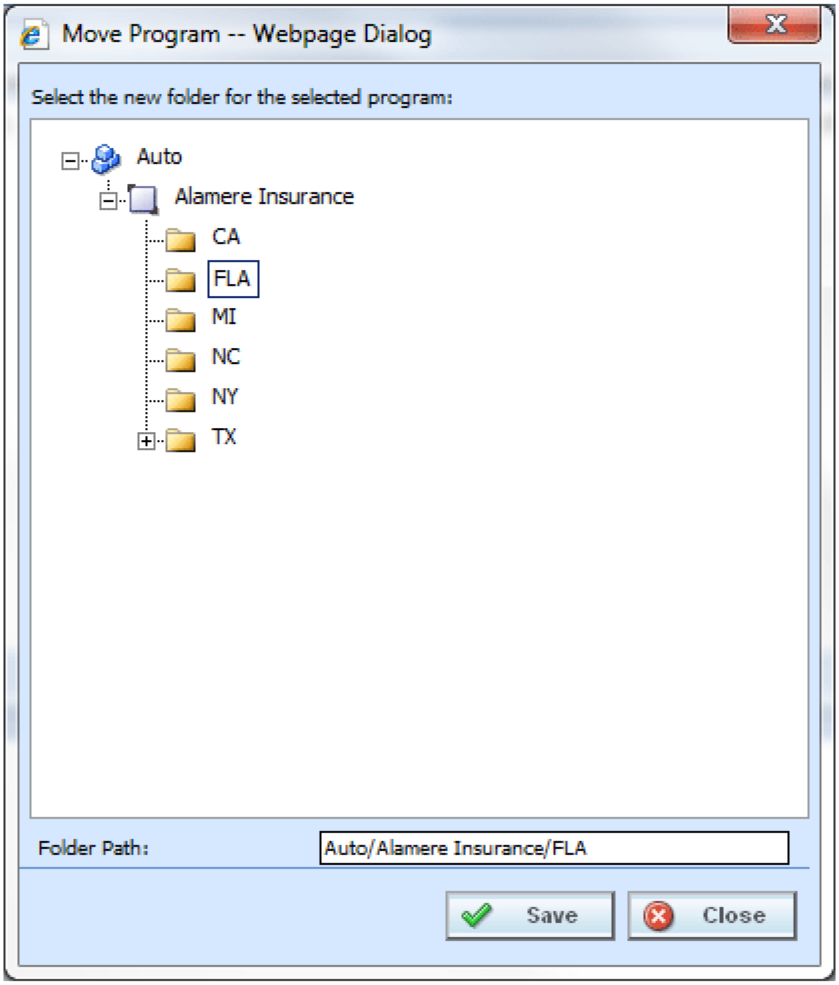
The Locale, Date Mask, the Decimal Symbol and Digit Grouping Symbol are automatically filled with the defaults from the line of business (LOB). The values set at the LOB apply as the defaults to every program created in the line regardless of subline. The program can select a different locale if necessary. 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 subline 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.



Move Program -- Webpage Dialog

Select the new folder for the selected program:

Folder Path: Auto/Alamere Insurance/FLA

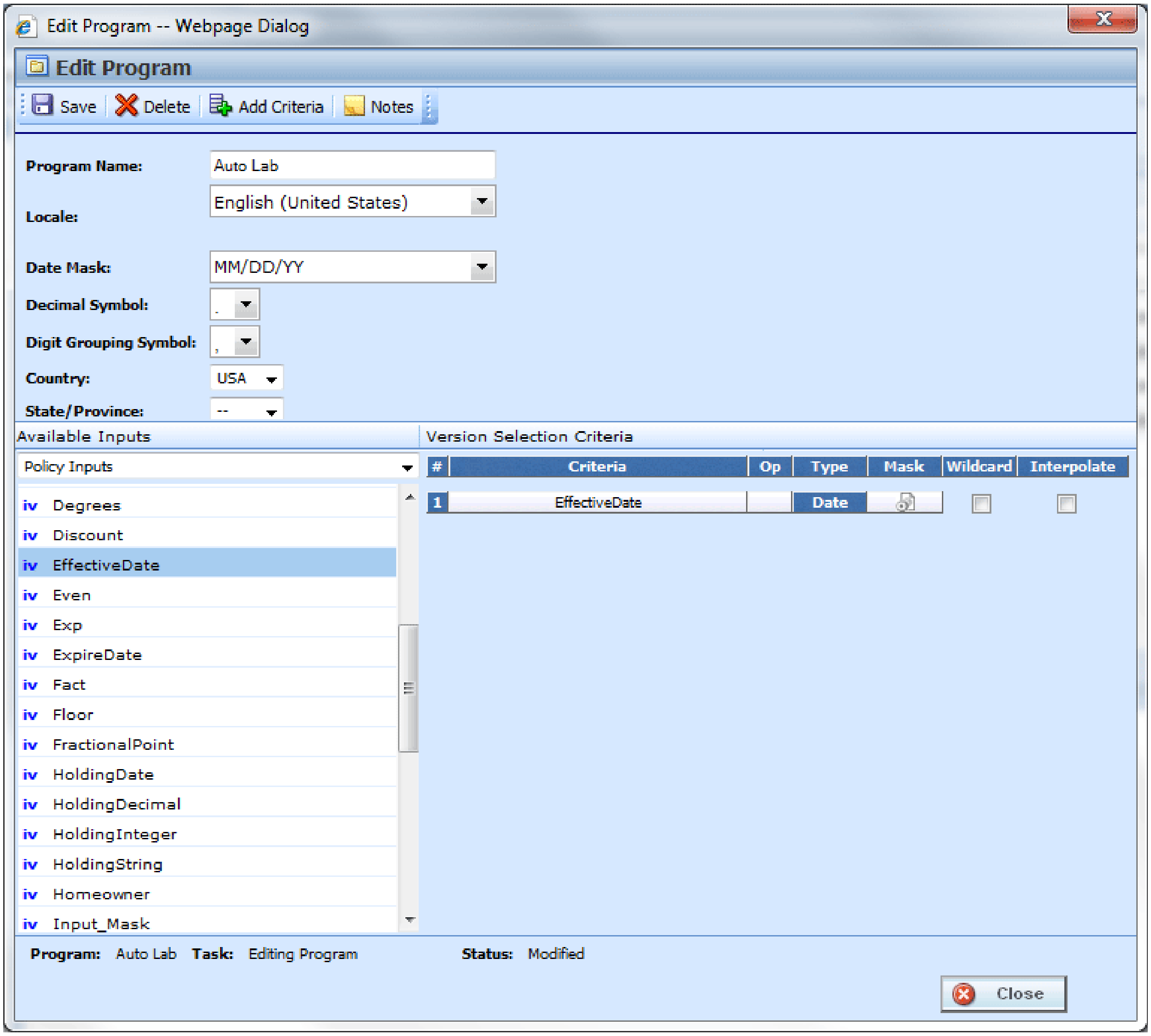
Figure 39 Selecting Destination for Moving a Program

3. The Folder Path at the bottom of the popup 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 will be moved to the selected folder.

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.



Program Name:	Auto Lab
Locale:	English (United States)
Date Mask:	MM/DD/YY
Decimal Symbol:	.
Digit Grouping Symbol:	,
Country:	USA
State/Province:	--

Available Inputs

Policy Inputs

- iv Degrees
- iv Discount
- iv EffectiveDate
- iv Even
- iv Exp
- iv ExpireDate
- iv Fact
- iv Floor
- iv FractionalPoint
- iv HoldingDate
- iv HoldingDecimal
- iv HoldingInteger
- iv HoldingString
- iv Homeowner
- iv Input_Mask

Version Selection Criteria

#	Criteria	Op	Type	Mask	Wildcard	Interpolate
1	EffectiveDate		Date		□	□

Program: Auto Lab **Task:** Editing Program **Status:** Modified

Figure 40 Editing a Program

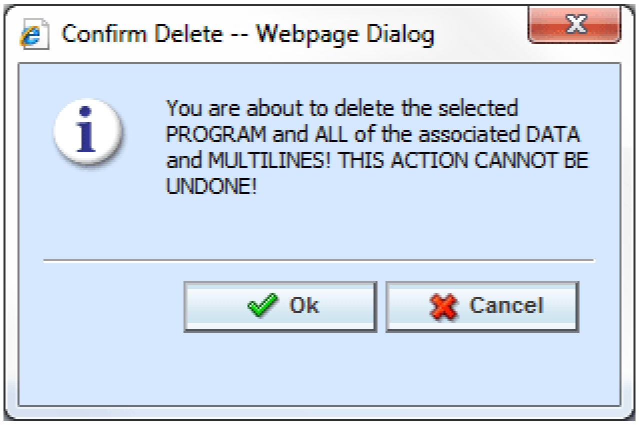
Deleting a Program

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.

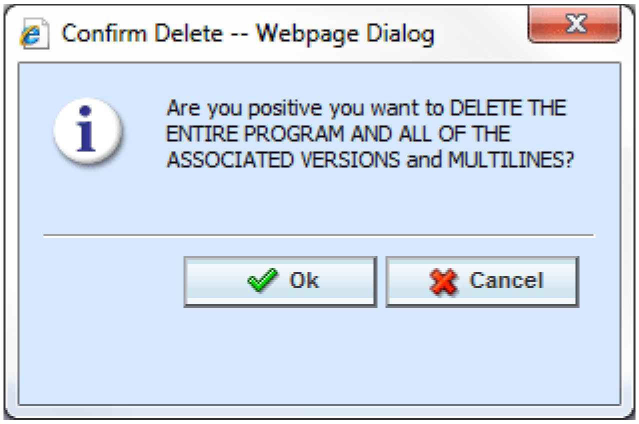


You are about to delete the selected PROGRAM and ALL of the associated DATA and MULTILINES! THIS ACTION CANNOT BE UNDONE!

Figure 41 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.



Are you positive you want to DELETE THE ENTIRE PROGRAM AND ALL OF THE ASSOCIATED VERSIONS and MULTILINES?

Figure 42 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.

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 input. For example, if you have a program with three versions, you can select which version to use by the effective date (a policy level input). 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 input criteria you want to use. This is the input that determines which version of the program should be used.

Step 2: At the program version level, you enter the defining parameters of the input.

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 inputs can be used as Version Selection Criteria*.

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 input criteria.
Add Criteria	**Add Criteria:** Adds the selected input as criteria.
Notes	**Notes:** Allows you to enter a note. Please see Notes for more information.

VERSIONING CRITERIA FIELDS

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 are three fields in the main body 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 Inputs

The inputs available for selection are listed. You also can create a new policy level input by clicking <New Input Variable>. Please see Adding a New Input from Another Area for more information.

Versioning Criteria: Version Selection Criteria

#: Sequential order of criteria selected from **Available Inputs**.

Criteria: Displays the name of each input 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 input 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.

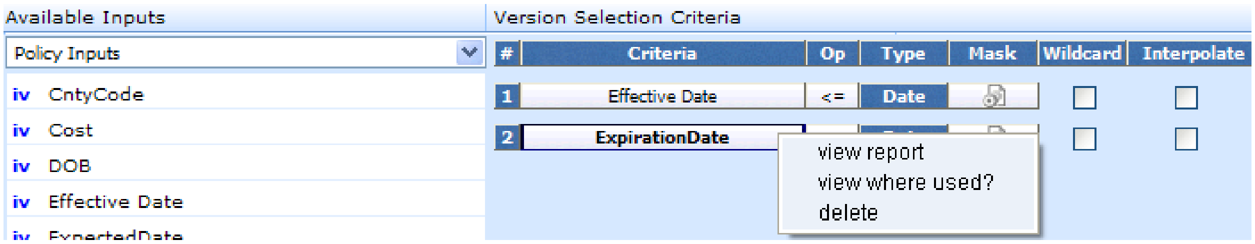
Interpolate: A check in this field instructs RateManager to use interpolation to determine the correct value for the table variable. See Interpolation for more information.

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.



#	Criteria	Op	Type	Mask	Wildcard	Interpolate
1	Effective Date	<=	Date		□	□
2	ExpirationDate				□	□

Figure 43 Right Click Menu for the Setting Versioning Criteria

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

- **View Report:** Displays Input information; XML Input 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 input you want to use from the listing of **Available Inputs**. You can either select the input and click **Add Criteria** or double-click the input. The input is displayed in the Version Selection Criteria area.

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 an input as criteria, click on 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 popup menu.

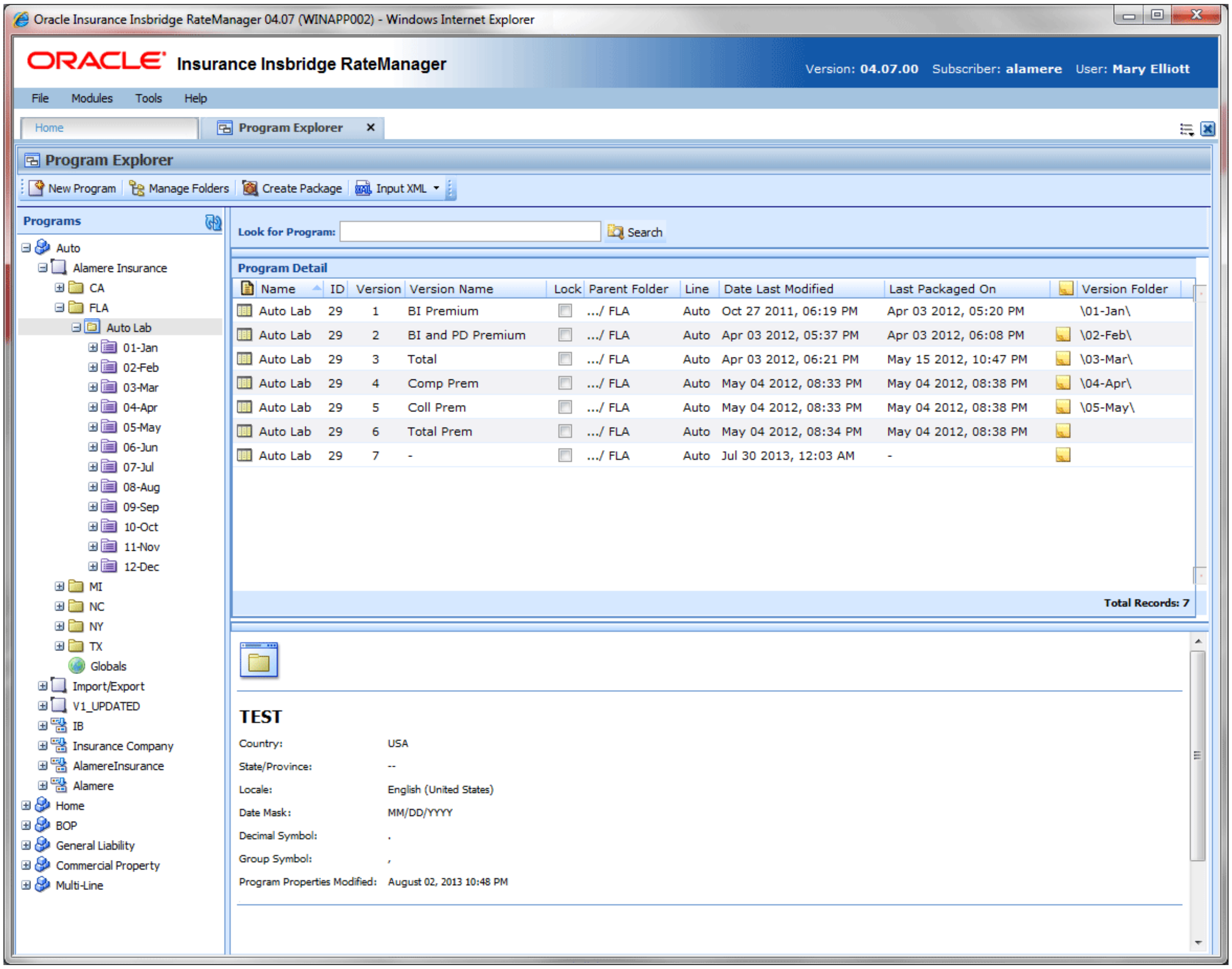
2. Click **Save** to save your work. Click **Close** if you are finished editing criteria.

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



Program Detail

Name	ID	Version	Version Name	Lock	Parent Folder	Line	Date Last Modified	Last Packaged On		Version Folder
Auto Lab	29	1	BI Premium	□	.../ FLA	Auto	Oct 27 2011, 06:19 PM	Apr 03 2012, 05:20 PM		\01-Jan\
Auto Lab	29	2	BI and PD Premium	□	.../ FLA	Auto	Apr 03 2012, 05:37 PM	Apr 03 2012, 06:08 PM	▨	\02-Feb\
Auto Lab	29	3	Total	□	.../ FLA	Auto	Apr 03 2012, 06:21 PM	May 15 2012, 10:47 PM	▨	\03-Mar\
Auto Lab	29	4	Comp Prem	□	.../ FLA	Auto	May 04 2012, 08:33 PM	May 04 2012, 08:38 PM	▨	\04-Apr\
Auto Lab	29	5	Coll Prem	□	.../ FLA	Auto	May 04 2012, 08:33 PM	May 04 2012, 08:38 PM	▨	\05-May\
Auto Lab	29	6	Total Prem	□	.../ FLA	Auto	May 04 2012, 08:34 PM	May 04 2012, 08:38 PM	▨	
Auto Lab	29	7	-	□	.../ FLA	Auto	Jul 30 2013, 12:03 AM	-	▨	

Total Records: 7

TEST

Country: USA

State/Province: --

Locale: English (United States)

Date Mask: MM/DD/YYYY

Decimal Symbol: .

Group Symbol: ,

Program Properties Modified: August 02, 2013 10:48 PM

Figure 44 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 a standard program or under a template generated program. Program versions can also be brought in under a template generated program. 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: Designates a program version that was created by you. User created program versions can be created by you, imported, or revisioned from a template program version. Most often, user created program versions are displayed in the default subline, but may be present if you create a new program version in a template generate program.

Template Created: Designates a program version that came from a template. Template created program versions also can be displayed under copied template generated programs. Template created program versions are displayed in template generated sublines.

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 that is 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. Any globals associated with a template are locked as well. 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.

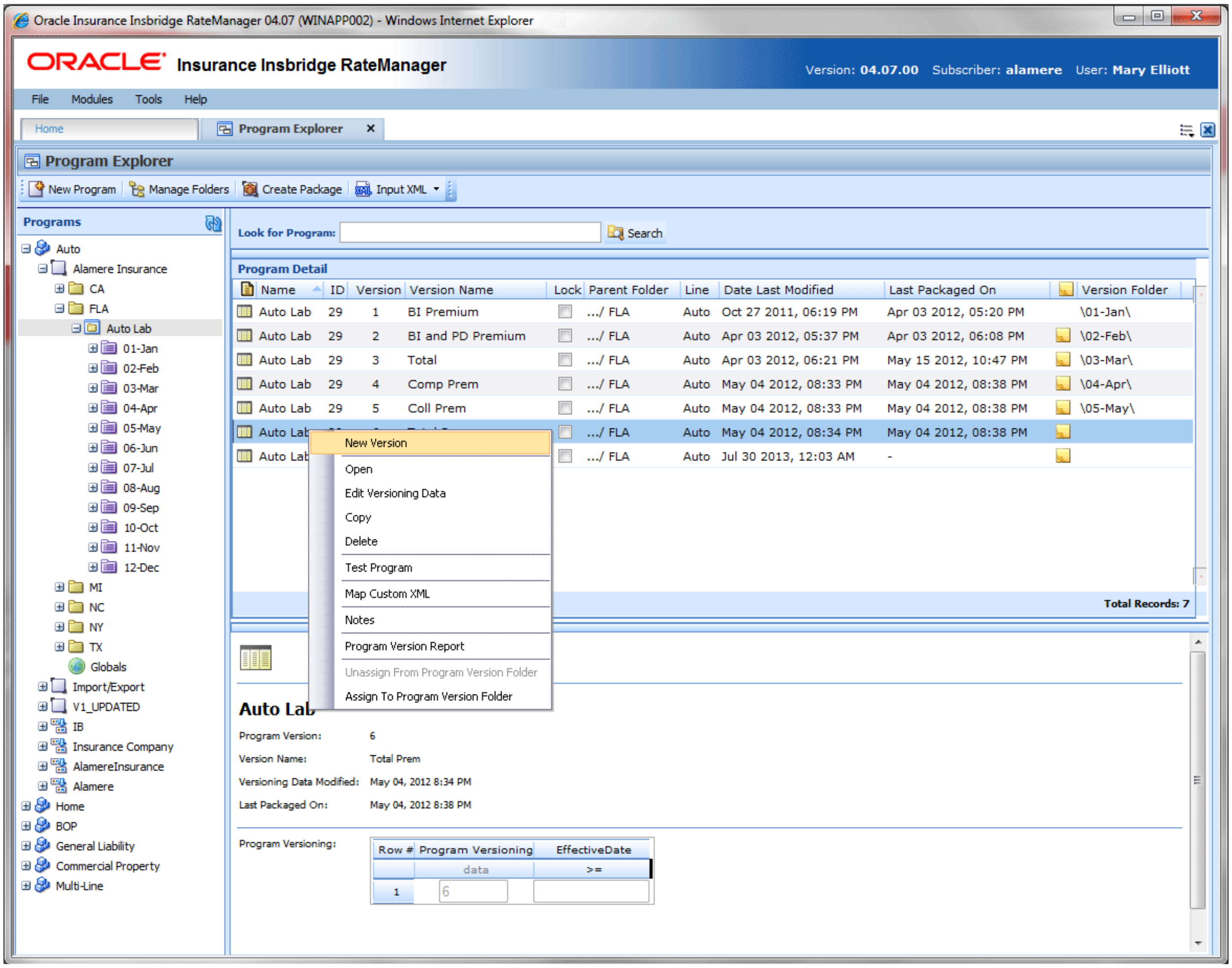
Template generated sublines can contain both user created and template created program versions. Default sublines can only contain user created program versions. In template generated program versions, the name is not open for editing. All other items can be edited as needed.

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 subline 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.



Program Detail

Name	ID	Version	Version Name	Lock	Parent Folder	Line	Date Last Modified	Last Packaged On		Version Folder
Auto Lab	29	1	BI Premium	□	.../ FLA	Auto	Oct 27 2011, 06:19 PM	Apr 03 2012, 05:20 PM		\01-Jan\
Auto Lab	29	2	BI and PD Premium	□	.../ FLA	Auto	Apr 03 2012, 05:37 PM	Apr 03 2012, 06:08 PM		\02-Feb\
Auto Lab	29	3	Total	□	.../ FLA	Auto	Apr 03 2012, 06:21 PM	May 15 2012, 10:47 PM		\03-Mar\
Auto Lab	29	4	Comp Prem	□	.../ FLA	Auto	May 04 2012, 08:33 PM	May 04 2012, 08:38 PM		\04-Apr\
Auto Lab	29	5	Coll Prem	□	.../ FLA	Auto	May 04 2012, 08:33 PM	May 04 2012, 08:38 PM		\05-May\
Auto Lab				□	.../ FLA	Auto	May 04 2012, 08:34 PM	May 04 2012, 08:38 PM		
Auto Lab				□	.../ FLA	Auto	Jul 30 2013, 12:03 AM	-		

Total Records: 7

Auto Lab

Program Version: 6

Version Name: Total Prem

Versioning Data Modified: May 04, 2012 8:34 PM

Last Packaged On: May 04, 2012 8:38 PM

Program Versioning:

Row #	Program Versioning	EffectiveDate
	data	>=
1	6	

Figure 45 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

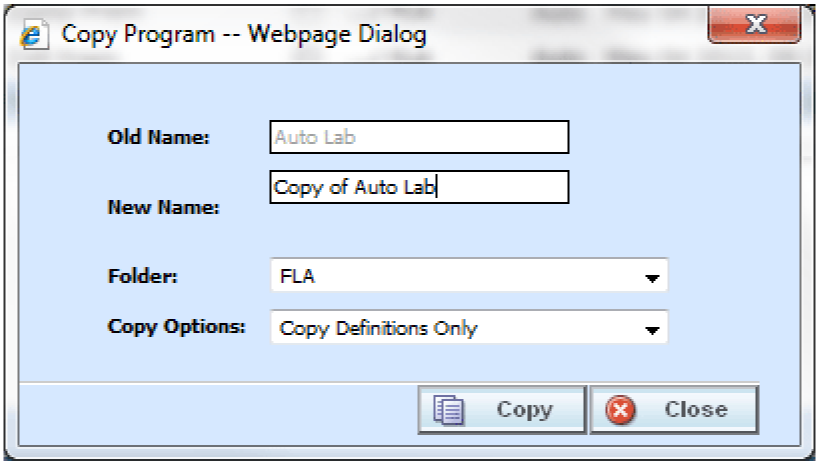
You can copy a program version at any time within the same subline. Program versions can be copied to any folder under the subline where the program is located.

1. In Programs, select the subline 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 popup.

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.



Copy Program -- Webpage Dialog

Old Name: Auto Lab

New Name: Copy of Auto Lab

Folder: FLA

Copy Options: Copy Definitions Only

Figure 46 Copy Program Version

5. Selecting **Copy** copies the program version to the folder you selected. Selecting Close closes the popup 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.

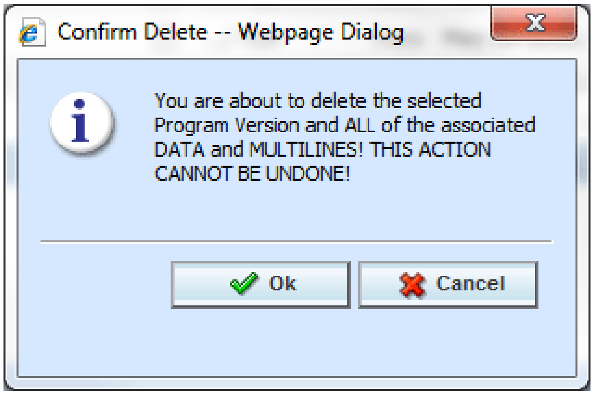
Deleting a Program Version

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 subline 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**.

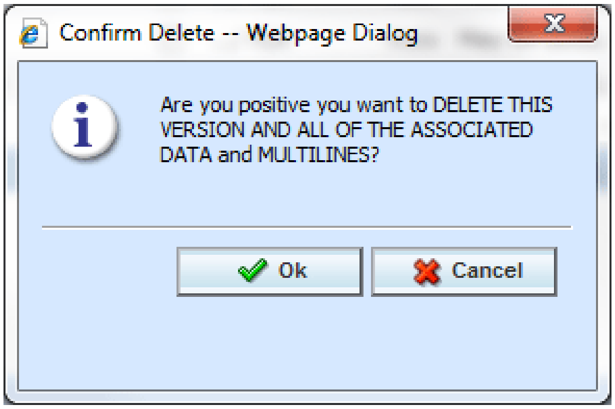


You are about to delete the selected Program Version and ALL of the associated DATA and MULTILINES! THIS ACTION CANNOT BE UNDONE!

Figure 47 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.



Are you positive you want to DELETE THIS VERSION AND ALL OF THE ASSOCIATED DATA and MULTILINES?

Figure 48 Confirming Delete

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

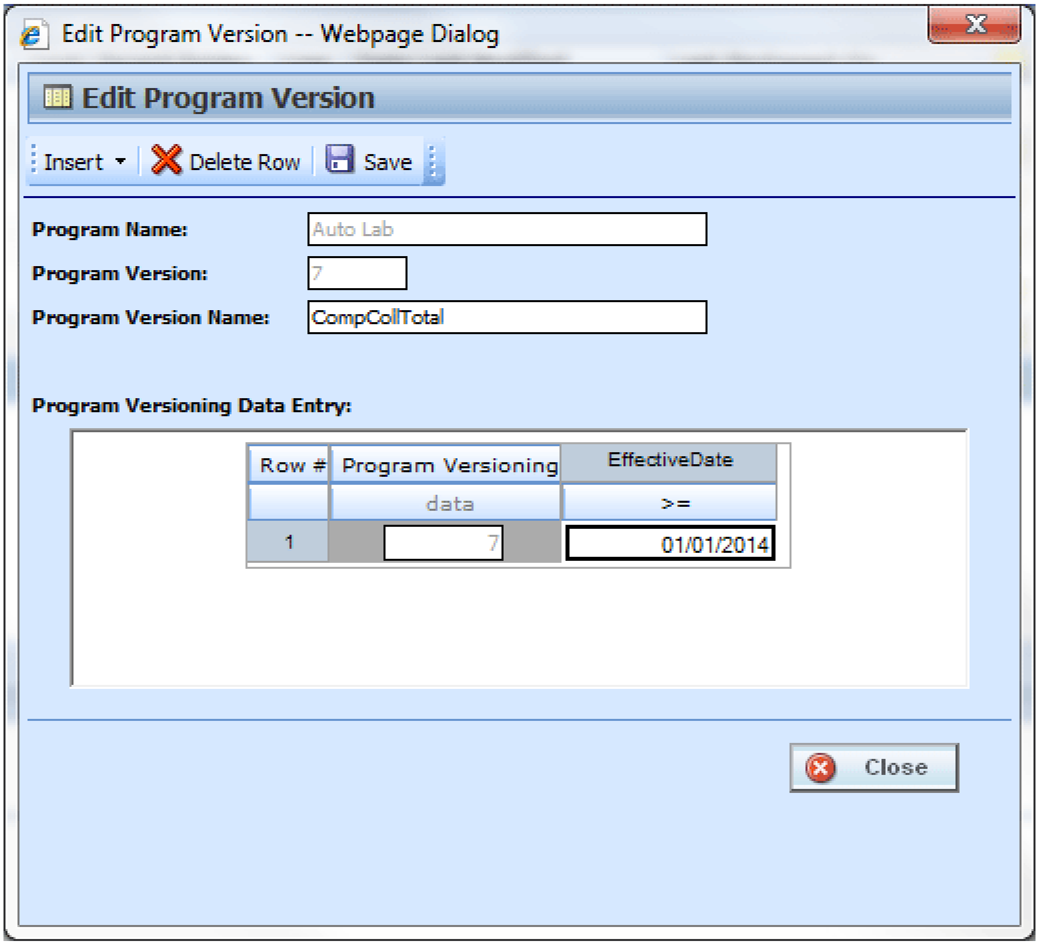
NOTE: *Make sure this is the action you want to take. Program Version deletion cannot be undone.*

Editing a Program Version

Program version information can be edited at any time. Use the Edit Versioning Data option when you want to change the description 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.



Program Name: Auto Lab

Program Version: 7

Program Version Name: CompCollTotal

Program Versioning Data Entry:

Row #	Program Versioning data	EffectiveDate >=
1	7	01/01/2014

Figure 49 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.

PROGRAM VERSIONING DATA

Versioning data is used when you want to define which version of a program you want to use based on a policy level input 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 input); 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 inputs. 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.

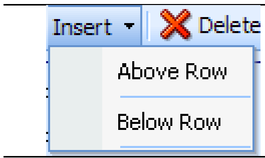
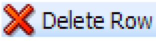
Setting up versioning data is a two-step process:

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

Step 2: At the program version level, enter the defining parameters of the input.

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 FIELDS

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 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 MultiLines and Releases.

Versioning Data Table:

The data table contains the policy level inputs 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:

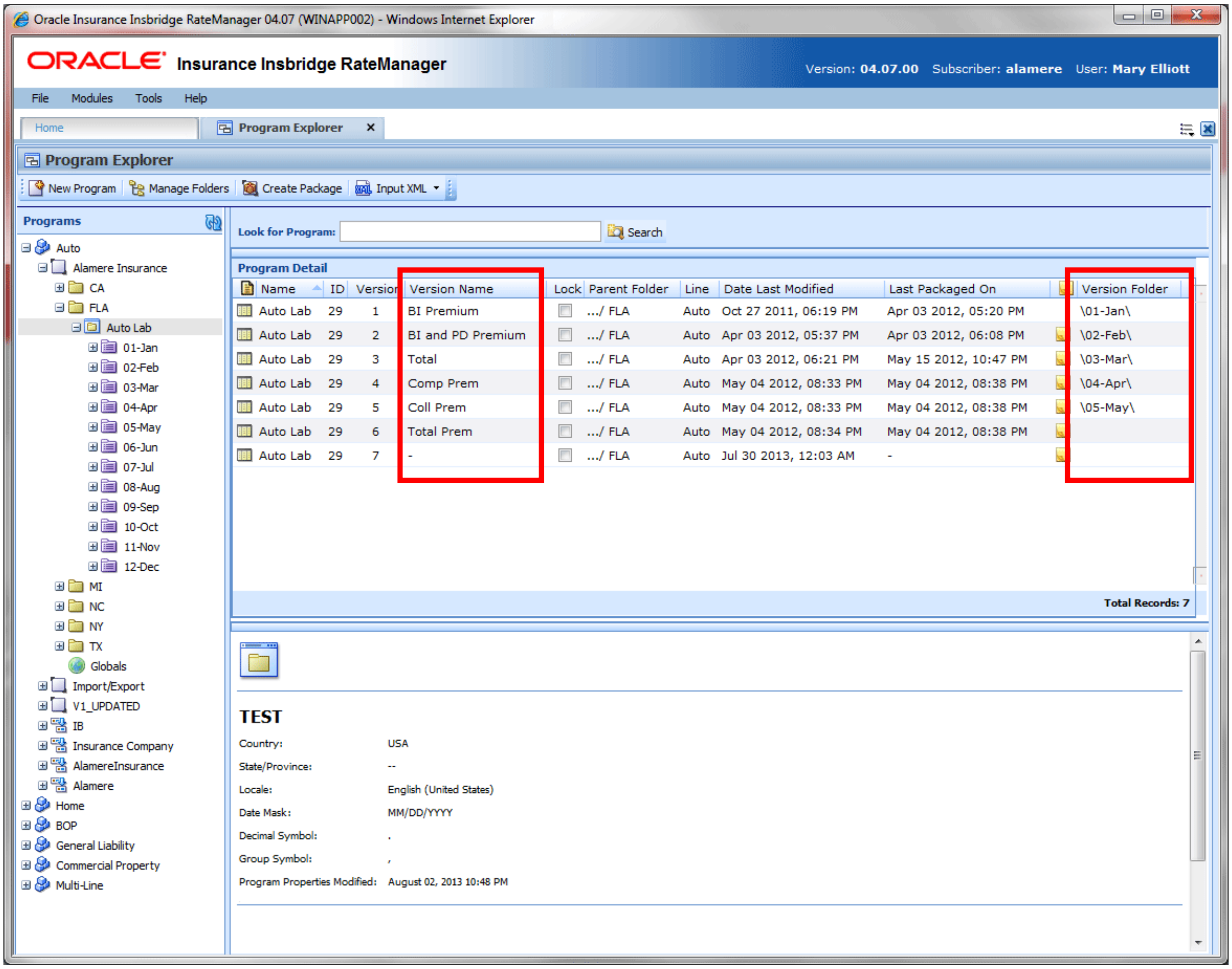
1. To remove a row in the data table, click on the row and select **Delete Row** from the menu bar. The row will be removed.

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.



Name	ID	Version	Version Name	Lock	Parent Folder	Line	Date Last Modified	Last Packaged On	Version Folder
Auto Lab	29	1	BI Premium	☐	.../ FLA	Auto	Oct 27 2011, 06:19 PM	Apr 03 2012, 05:20 PM	\01-Jan\
Auto Lab	29	2	BI and PD Premium	☐	.../ FLA	Auto	Apr 03 2012, 05:37 PM	Apr 03 2012, 06:08 PM	\02-Feb\
Auto Lab	29	3	Total	☐	.../ FLA	Auto	Apr 03 2012, 06:21 PM	May 15 2012, 10:47 PM	\03-Mar\
Auto Lab	29	4	Comp Prem	☐	.../ FLA	Auto	May 04 2012, 08:33 PM	May 04 2012, 08:38 PM	\04-Apr\
Auto Lab	29	5	Coll Prem	☐	.../ FLA	Auto	May 04 2012, 08:33 PM	May 04 2012, 08:38 PM	\05-May\
Auto Lab	29	6	Total Prem	☐	.../ FLA	Auto	May 04 2012, 08:34 PM	May 04 2012, 08:38 PM	
Auto Lab	29	7	-	☐	.../ FLA	Auto	Jul 30 2013, 12:03 AM	-	

Total Records: 7

TEST

Country: USA
State/Province: --
Locale: English (United States)
Date Mask: MM/DD/YYYY
Decimal Symbol: .
Group Symbol: ,
Program Properties Modified: August 02, 2013 10:48 PM

Figure 50 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

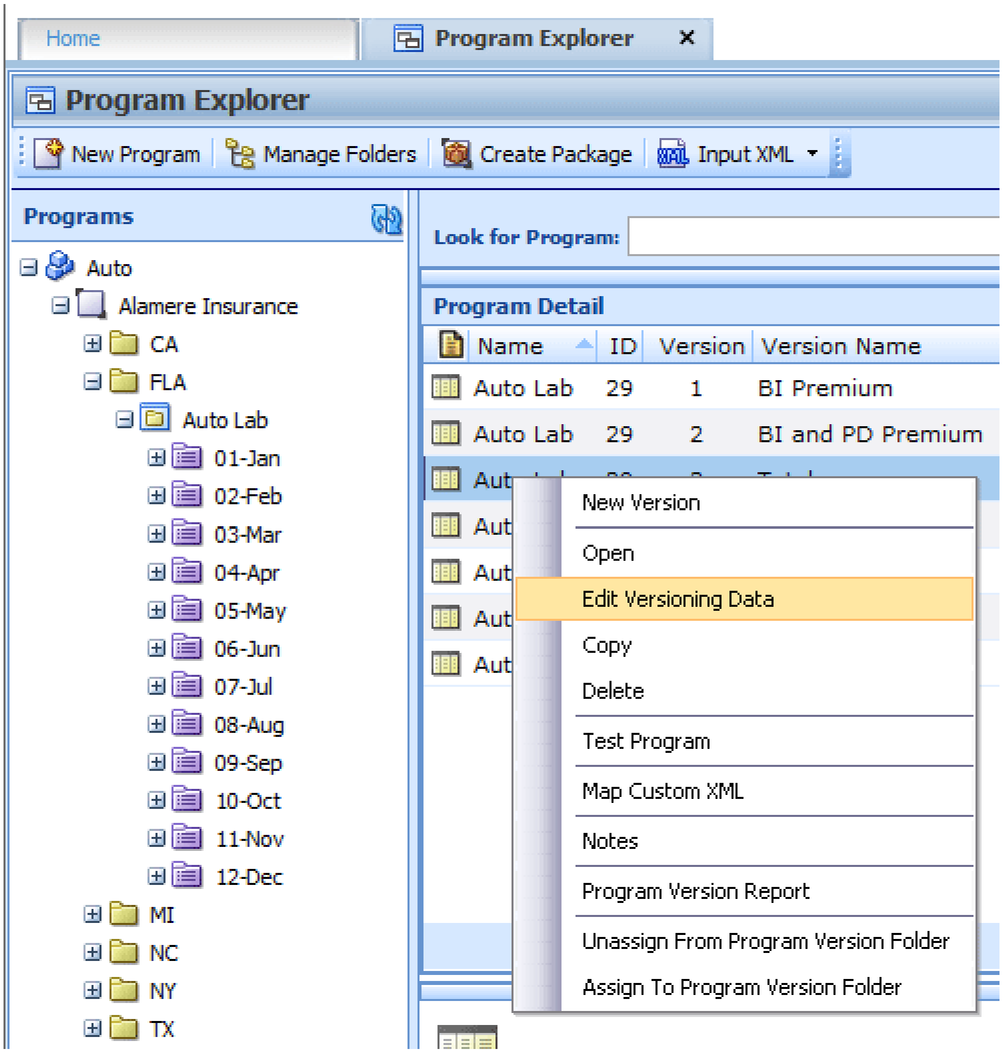
Program Version Names can assist with locating a specific version quickly by looking for a name instead of a program version number. Program version names can be 40 alphanumeric characters.

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.

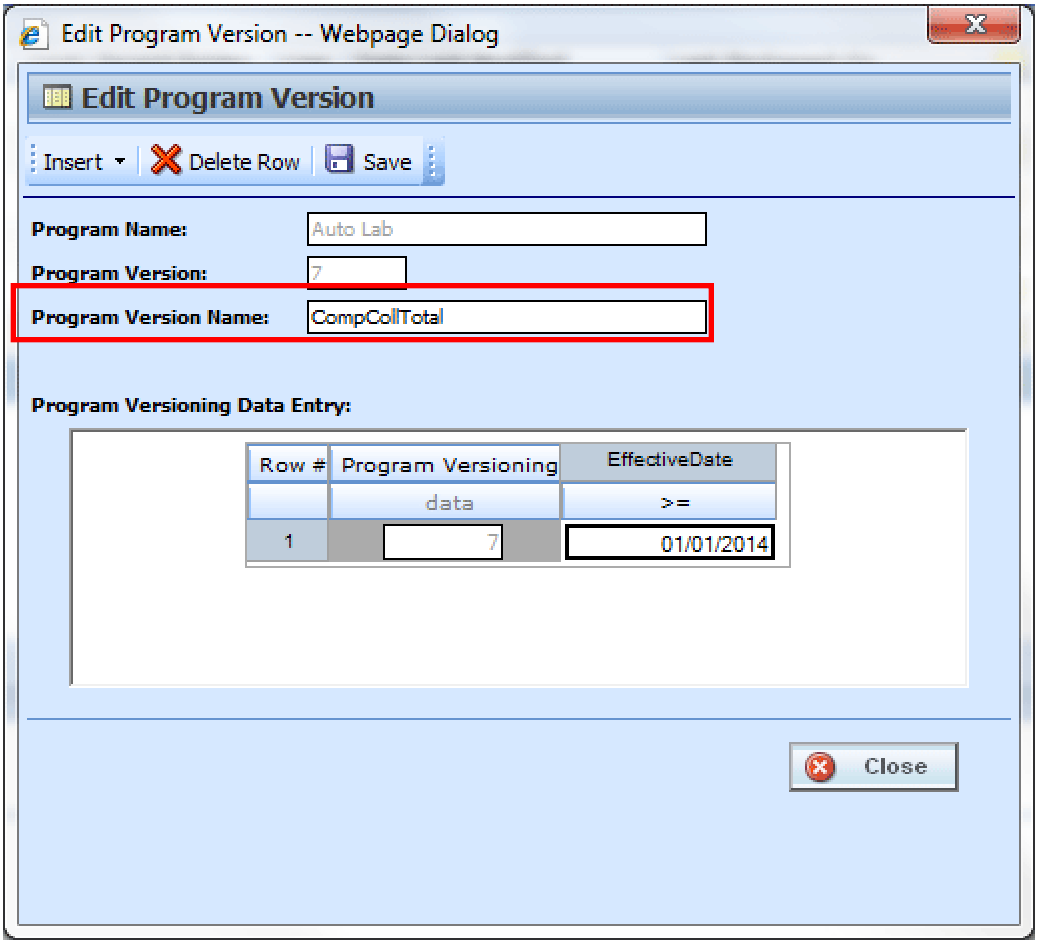


Name	ID	Version	Version Name
Auto Lab	29	1	BI Premium
Auto Lab	29	2	BI and PD Premium
Aut			
Aut			
Aut			
Aut			
Aut			

Figure 51 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.



Program Name:	Auto Lab
Program Version:	7
Program Version Name:	CompCollTotal

Program Versioning Data Entry:

Row #	Program Versioning data	EffectiveDate >=
1	7	01/01/2014

Figure 52 Entering a Program Version Name

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

4. Click **Save**.

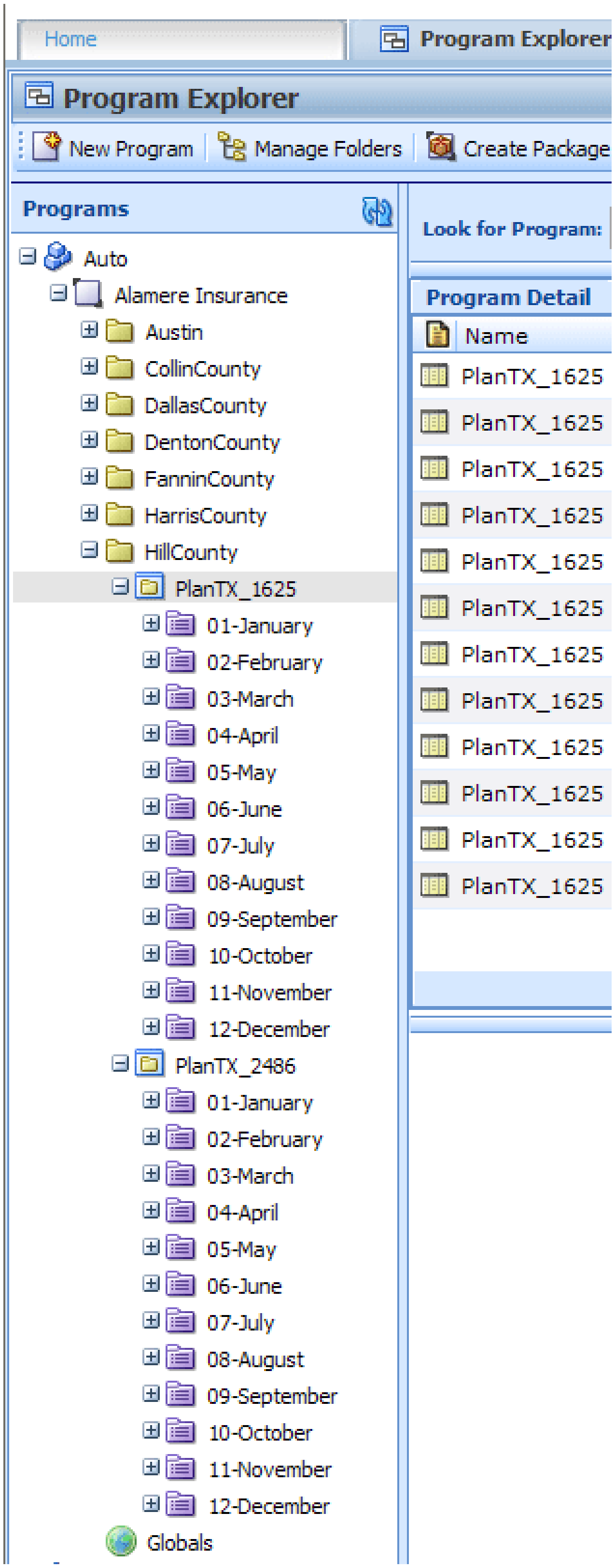
The Edit Program Version screen closes. The program name is displayed in the Version Name column in the Program Detail area.

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 is performed on the Program Explorer tab. For more on creating program version folders, please see Creating a New Program Version Folder.



Program Version Folder Structure:

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

The program version folder structure is duplicated in each program under the subline 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.

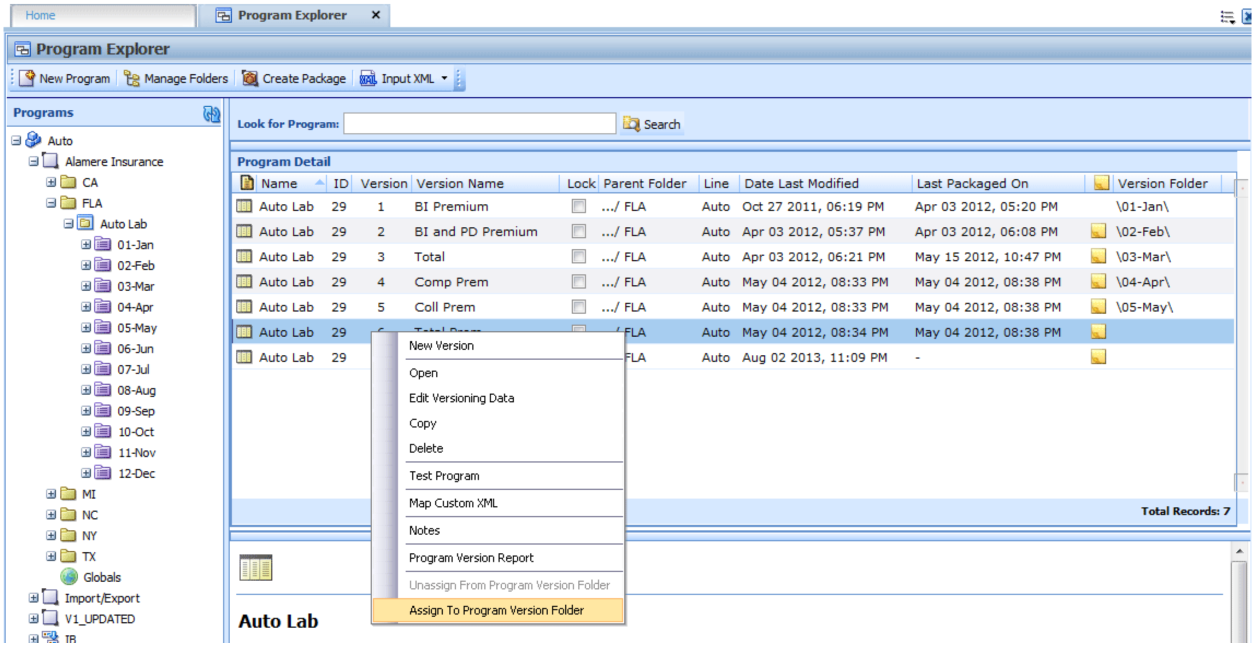
Program versions can be assigned and unassigned to program version folders at any time, and do not affect the integration or the execution of SoftRater.

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.



Name	ID	Version	Version Name	Lock	Parent Folder	Line	Date Last Modified	Last Packaged On	Version Folder
Auto Lab	29	1	BI Premium	☐	.../ FLA	Auto	Oct 27 2011, 06:19 PM	Apr 03 2012, 05:20 PM	\01-Jan\
Auto Lab	29	2	BI and PD Premium	☐	.../ FLA	Auto	Apr 03 2012, 05:37 PM	Apr 03 2012, 06:08 PM	\02-Feb\
Auto Lab	29	3	Total	☐	.../ FLA	Auto	Apr 03 2012, 06:21 PM	May 15 2012, 10:47 PM	\03-Mar\
Auto Lab	29	4	Comp Prem	☐	.../ FLA	Auto	May 04 2012, 08:33 PM	May 04 2012, 08:38 PM	\04-Apr\
Auto Lab	29	5	Coll Prem	☐	.../ FLA	Auto	May 04 2012, 08:33 PM	May 04 2012, 08:38 PM	\05-May\
Auto Lab	29	6	Total Prem	☐	.../ FLA	Auto	May 04 2012, 08:34 PM	May 04 2012, 08:38 PM	
Auto Lab	29				FLA	Auto	Aug 02 2013, 11:09 PM	-	

Figure 53 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.

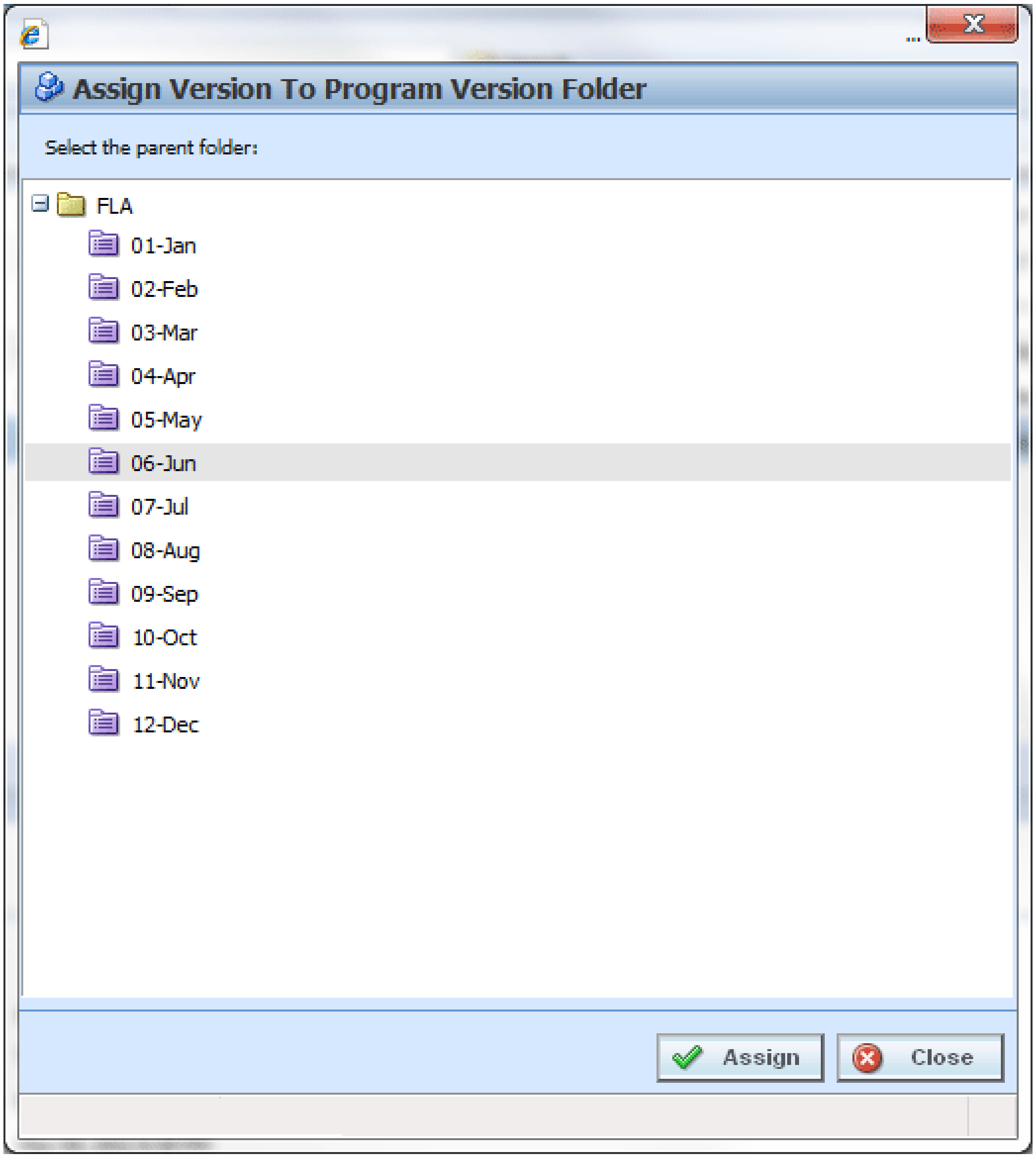


Figure 54 Assigning a Program to a Program Version Folder

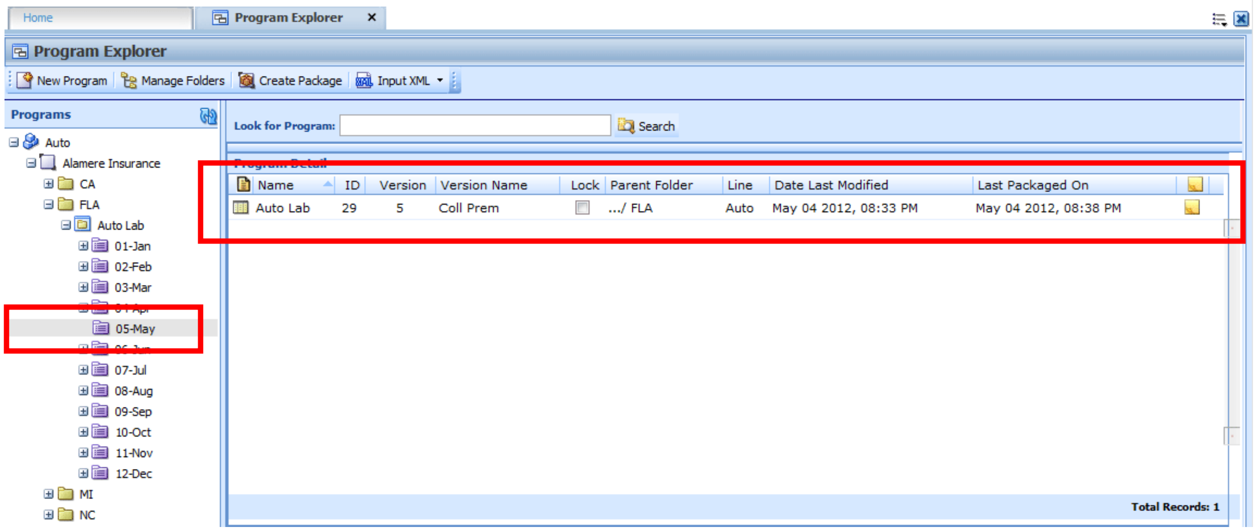
3. **Select the desired program version folder** from the listing.

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.



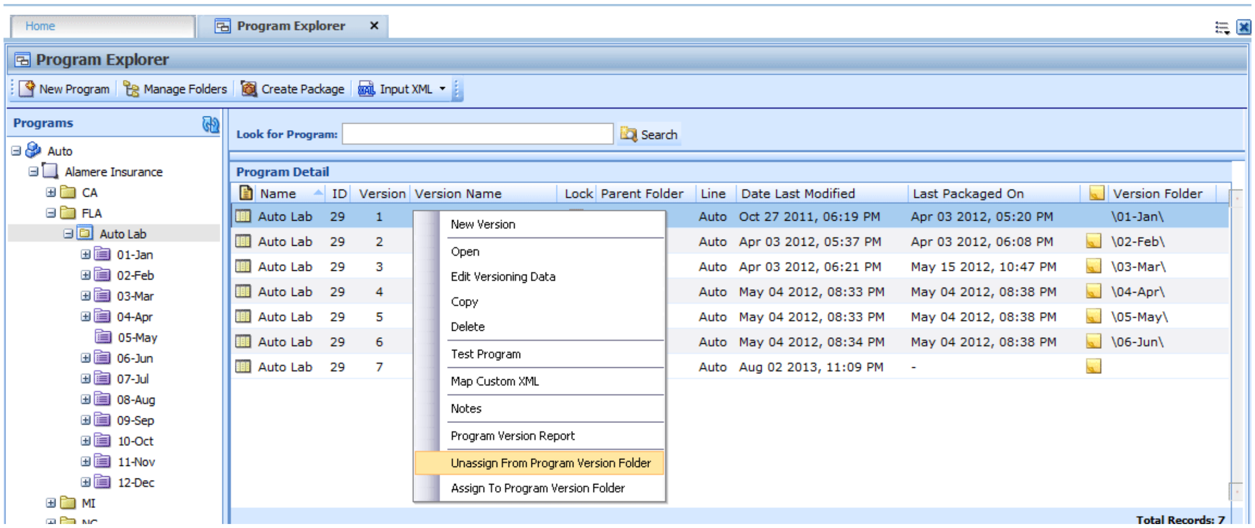
Name	ID	Version	Version Name	Lock	Parent Folder	Line	Date Last Modified	Last Packaged On
Auto Lab	29	5	Coll Prem	□	.../ FLA	Auto	May 04 2012, 08:33 PM	May 04 2012, 08:38 PM

Total Records: 1

Figure 55 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.



Name	ID	Version	Version Name	Lock	Parent Folder	Line	Date Last Modified	Last Packaged On		Version Folder
Auto Lab	29	1				Auto	Oct 27 2011, 06:19 PM	Apr 03 2012, 05:20 PM		\01-Jan\
Auto Lab	29	2				Auto	Apr 03 2012, 05:37 PM	Apr 03 2012, 06:08 PM		\02-Feb\
Auto Lab	29	3				Auto	Apr 03 2012, 06:21 PM	May 15 2012, 10:47 PM		\03-Mar\
Auto Lab	29	4				Auto	May 04 2012, 08:33 PM	May 04 2012, 08:38 PM		\04-Apr\
Auto Lab	29	5				Auto	May 04 2012, 08:33 PM	May 04 2012, 08:38 PM		\05-May\
Auto Lab	29	6				Auto	May 04 2012, 08:34 PM	May 04 2012, 08:38 PM		\06-Jun\
Auto Lab	29	7				Auto	Aug 02 2013, 11:09 PM	-		

Total Records: 7

Figure 56 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. Shown below is an example input file.

```
<rate rateId=“99” policyId=“10418124”>
  <heading>
 <program parent_id=“100” program_id=“28”>
  <m i=“119” n=“EffectiveDate” v=“02012012”/>
 </program>
 <program parent_id=“100” program_id=“29”>
  <m i=“119” n=“EffectiveDate” v=“05012012”/>
 </program>
 <program parent_id=“200” program_id=“29” program_ver=“3”>
  <m i=“119” n=“EffectiveDate” v=“03012012”/>
 </program>
  </heading>
  <c i=“1” desc=“Vehicle”>
 <m i=“386” n=“VehicleId” v=“1”/>
 <m i=“182” n=“Model Year” v=“2010”/>
  </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.

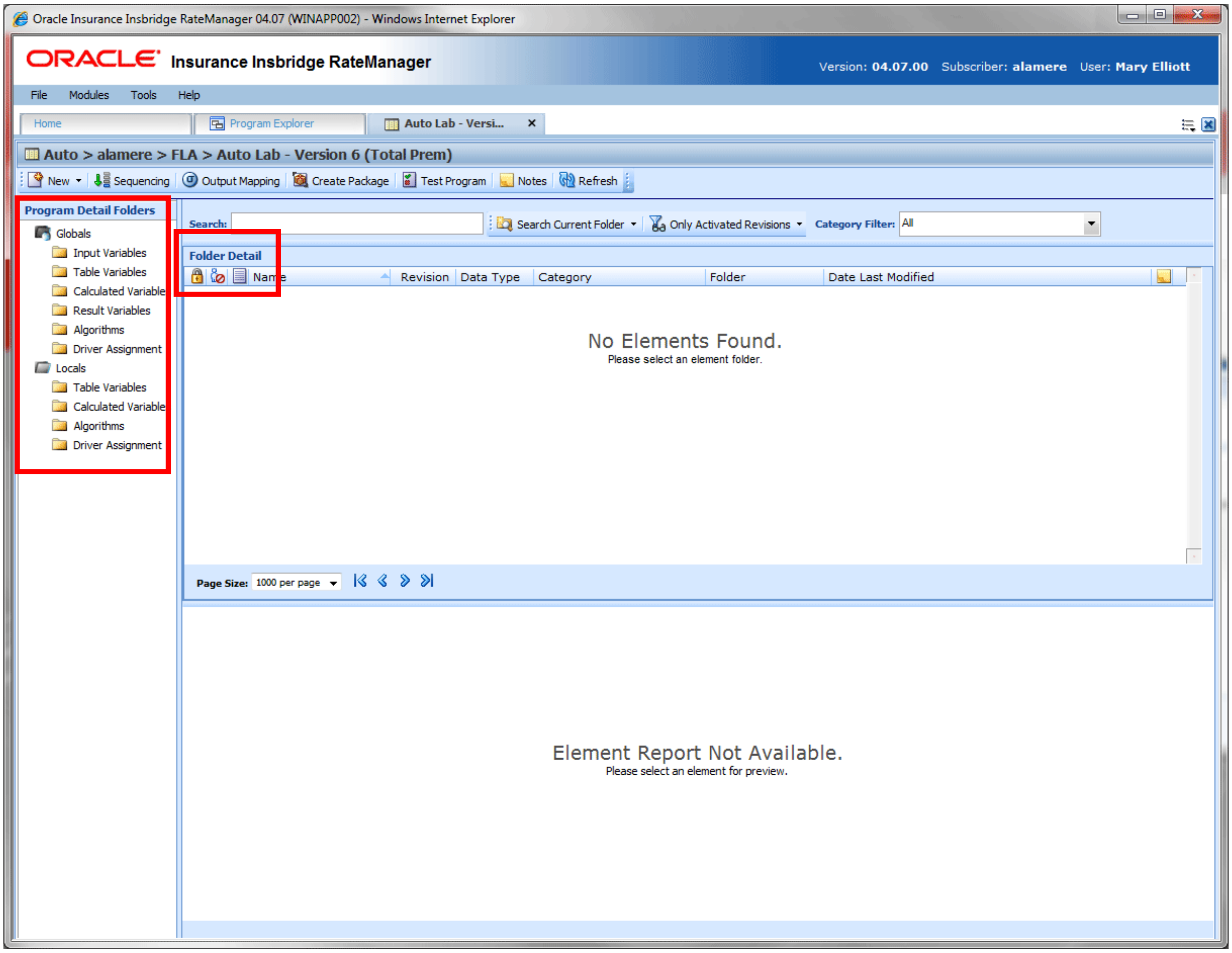
- The 3rd program call is parent_id = 200, program_id = 29, and program_ver = 3. In this case, automatic versioning is overridden with a specific version call.

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

1. Select which Policy inputs will be used to select a program version: see Editing a Program.
2. Specify the input 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 hand side of the tab. A listing of the elements contained in the folders is displayed in the right hand side of the tab.



No Elements Found.
Please select an element folder.

Element Report Not Available.
Please select an element for preview.

Figure 57 Program Details

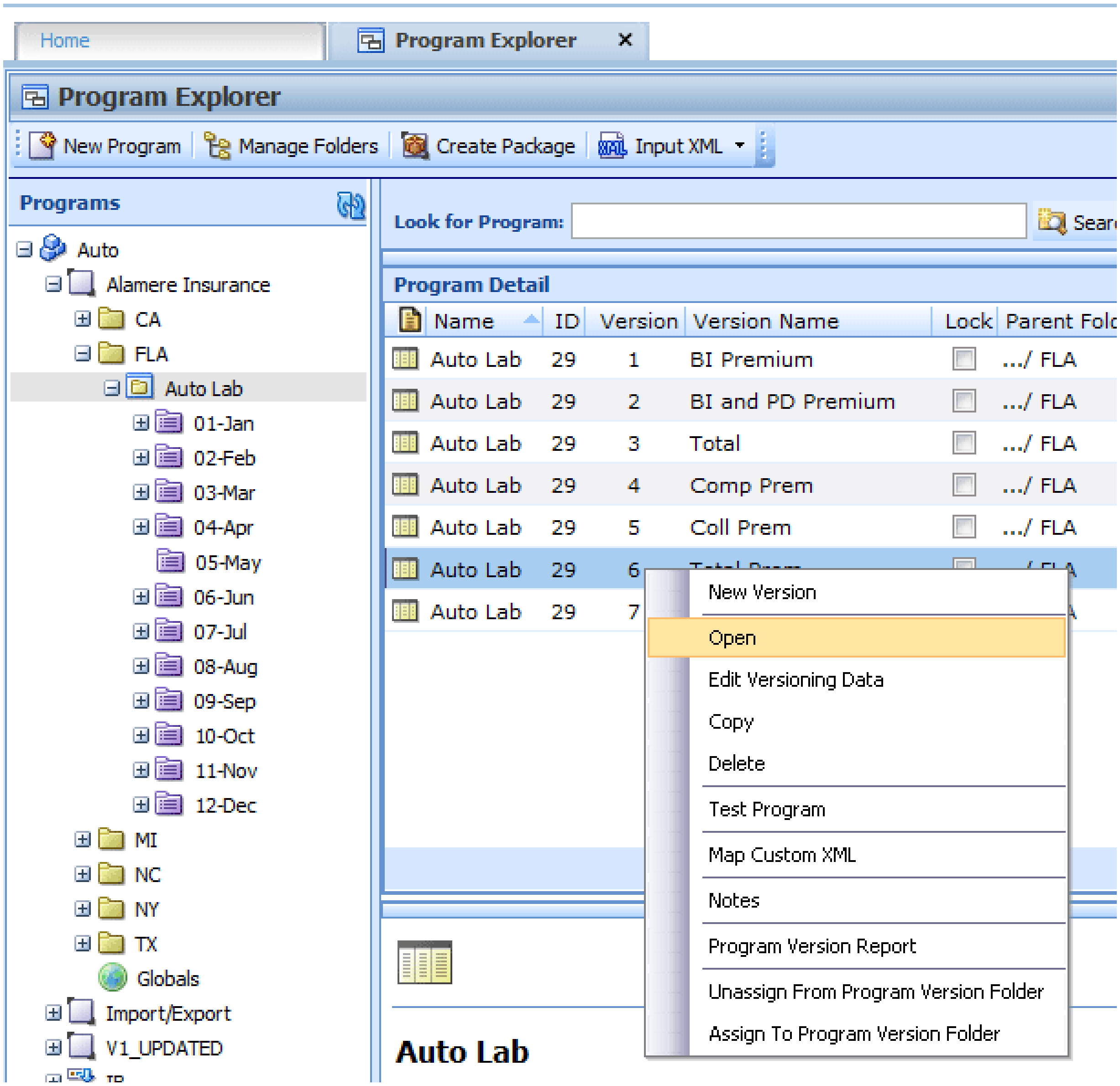
Program Details is divided into two areas:

- **Program Detail Folders:** Program Details Folders is a navigation column on the left hand side of the tab. Global element folders belonging to the subline are displayed on top. Local element folders belonging to the program are underneath. These are fixed folders. They do not expand or have right click menus. You can double click the folder to view details.

- **Folder Detail:** Folder detail lists the individual elements contained in the folders 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.



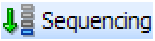
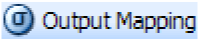
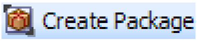
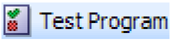
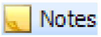
Name	ID	Version	Version Name	Lock	Parent Fold
Auto Lab	29	1	BI Premium	☐	.../ FLA
Auto Lab	29	2	BI and PD Premium	☐	.../ FLA
Auto Lab	29	3	Total	☐	.../ FLA
Auto Lab	29	4	Comp Prem	☐	.../ FLA
Auto Lab	29	5	Coll Prem	☐	.../ FLA
Auto Lab	29	6	Total Prem	☐	.../ FLA
Auto Lab	29	7			A

Figure 58 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.

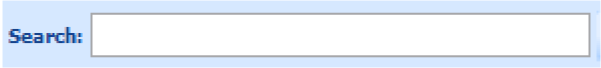
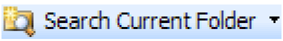
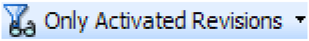
PROGRAM DETAILS FUNCTIONALITY

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. Inputs, Result Variables, Calculated Variables, and Tables Variables, allows you to select **New** from the element and open the corresponding tab or popup 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:

The **SEARCH** fields allow you to search for a particular element by name. Select the folder 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:”.
	**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 Inputs 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 hand side of the tab. Global element folders belonging to the subline are displayed on top. Local element folders belonging to the program are underneath. These are fixed folders. They do not expand or have right click menus. You can double click the folder to view details.

Program Detail Folder Fields

📁	**Global:** The folders underneath are the global elements of the subline.
📁	**Locals:** The folders 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 Inputs in the Program Detail Folders area, a list of inputs 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 Fields

The fields in the Element Listing area are the same for both global and local elements. Inputs and result variables can be managed at the global level only.

The Element Details area does not have any fields, only the details of the selected element.

NOTE: *Driver Assignment is available in the Auto LOB only. No other line has this option.*

Element Listing Fields:

SELECTION	DISPLAYS
Inputs	**Icon:** There are three icons: Locked. A lock in this field indicates that the input is locked. In-Use. Indicates that the input is being worked on by another user. Input. This icon indicates an Input. **Name:** The name of the input. **Data Type:** The type of input, i.e. date, decimal, integer or string. **Category:** The category the input is in. **Folder:** The folder the input is located in. **Date Last Modified:** The date and time the input was last updated. **Note:** A note icon indicates a note for this input. **XML ID:** The XML ID that is used to call the input.
Table Variables	**Icon:** There are four icons: 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. Table Variable. This icon indicates a Table Variable. **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.

SELECTION	DISPLAYS
Calculated Variables	**Icon:** There are three icons: 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. 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 three icons: 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 four 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. Advanced Option. Indicates that this algorithm contains an advanced option. 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 LOB ONLY	**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. f(..) 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.
REVISIONS	**Will display a green checkmark ✓ for the active revision.**

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 will be grayed out.

SELECTION	RIGHT CLICK MENU
Inputs	• **Edit:** Brings up the Edit Input Variable screen where you can make edits to the category location, data type and name. • **Copy:** Creates a copy of the input. • **Delete:** Deletes the inputs. You must have rights to delete inputs. • **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 Input. Please see View History for more information. • **Notes:** Enter, edit and view notes to the input.

SELECTION	RIGHT CLICK MENU
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 subline. 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 (Tab-Delimited):** Exports a table that places all the data from a table variables data table 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. • **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 subline. 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. Global algorithms, regardless of type, may be copied within the subline. 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 LOB 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. Global driver assignments, regardless of type, may be copied within the subline. 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 LOB 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.

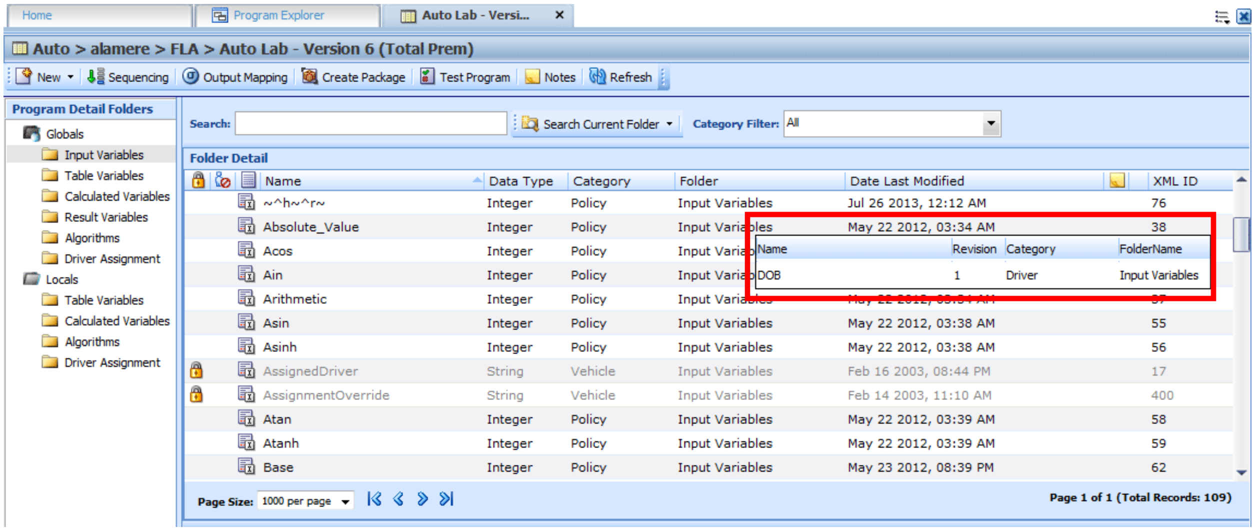
Moving Your Cursor over an Element:

When you hold your cursor over an element listed on the Program Details tab, an information box is displayed. The box identifies the element and any restrictions.

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.



Auto > alamere > FLA > Auto Lab - Version 6 (Total Prem)

Search:

Category Filter: All

Folder Detail

Name	Data Type	Category	Folder	Date Last Modified	XML ID
~^h~^r~	Integer	Policy	Input Variables	Jul 26 2013, 12:12 AM	76
Absolute_Value	Integer	Policy	Input Variables	May 22 2012, 03:34 AM	38
Acos	Integer	Policy	Input Varia		
Ain	Integer	Policy	Input Varia		
Arithmetic	Integer	Policy	Input Variables	May 22 2012, 03:34 AM	57
Asin	Integer	Policy	Input Variables	May 22 2012, 03:38 AM	55
Asinh	Integer	Policy	Input Variables	May 22 2012, 03:38 AM	56
AssignedDriver	String	Vehicle	Input Variables	Feb 16 2003, 08:44 PM	17
AssignmentOverride	String	Vehicle	Input Variables	Feb 14 2003, 11:10 AM	400
Atan	Integer	Policy	Input Variables	May 22 2012, 03:39 AM	58
Atanh	Integer	Policy	Input Variables	May 22 2012, 03:39 AM	59
Base	Integer	Policy	Input Variables	May 23 2012, 08:39 PM	62

Name	Revision	Category	FolderName
DOB	1	Driver	Input Variables

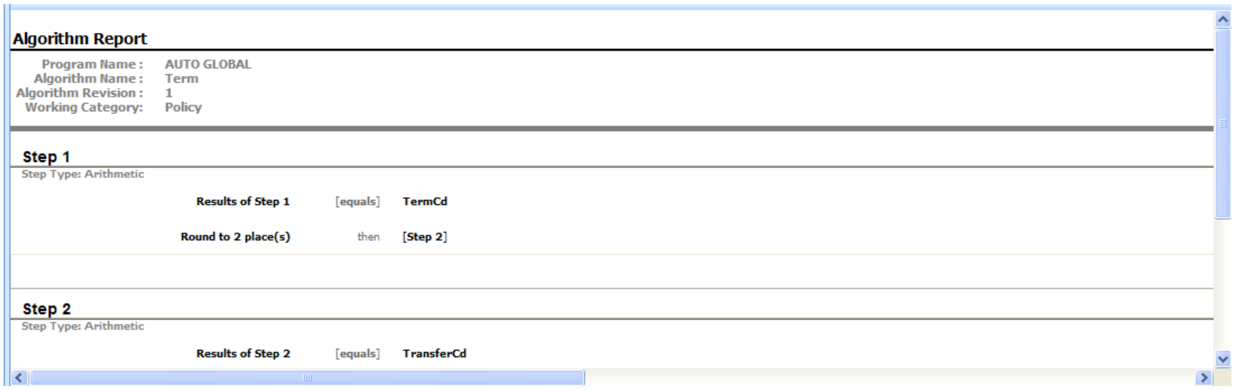
Page Size: 1000 per page

Page 1 of 1 (Total Records: 109)

Figure 59 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.



Algorithm Report

Program Name : AUTO GLOBAL
Algorithm Name : Term
Algorithm Revision : 1
Working Category: Policy

Step 1
Step Type: Arithmetic

Results of Step 1 [equals] TermCd

Round to 2 place(s) then [Step 2]

Step 2
Step Type: Arithmetic

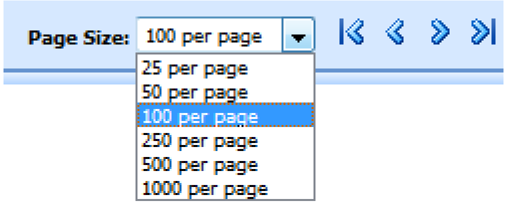
Results of Step 2 [equals] TransferCd

Figure 60 Element Details

Page Sizing:

At the bottom of the element listing are the page size and page navigation options. *Page Size* allows you to 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.



Page Size: 100 per page

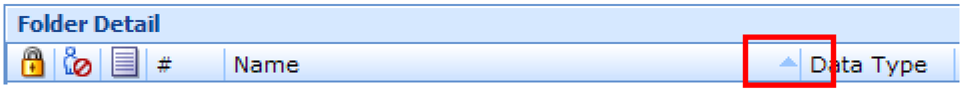
The *Page Navigation* options allow you to move back and forth between pages.

⏮	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 fields 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 9

GLOBALS

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 subline, it is referred to as a global element. If used properly, global elements can greatly reduce the time taken to load and maintain the same tables or logic in multiple programs.

Variables, algorithms, and driver assignments can be created at the global level and can be used in any program in the same subline. In addition, inputs and result variables also are created at the global level.

Globals are subline 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 coding is standard and used in every program in the subline, a global table variable can be created and used as an established option for all of the programs in the subline. This is also optimal for maintenance.

When a global is changed, it automatically updates in every program.

Global Elements:

- Global Inputs
- Global Variables
  - Result
  - Table
  - Calculated
- Global Algorithms
- Global Driver Assignments

Global inputs, variables, algorithms, and driver assignments can be created by the user, 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 globals can be added to a template generated subline and you can create new revisions. These new globals are open for editing, copying, or deletion, unless the program version is locked.

Global Rules

- At the global level, packages cannot be created and program testing cannot be done.

- Only global variables, inputs or results may be used in a global algorithm or driver assignment. You cannot use any local elements.

- Global elements are available for use by any of the programs under the subline where the element is defined.

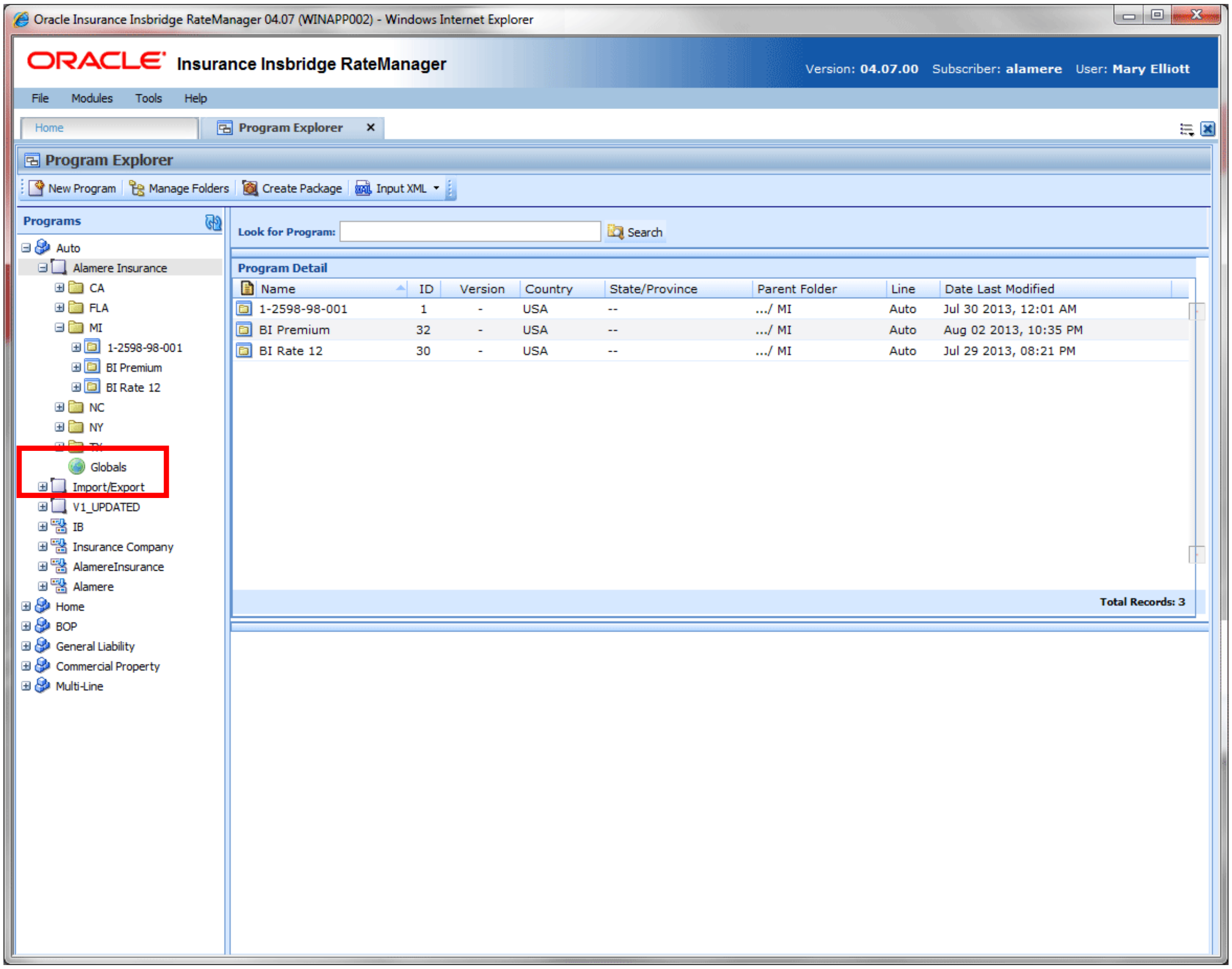
- Global elements have the active revision set at the program level. The default active revision is revision 1. If there is another revision that you want to use, you will have to go into the program and set the revision there. You cannot change the active revision in the global tab.

Globals Tab

The Globals tab is available from the Program Explorer. This tab lists only the global elements associated with the selected subline. 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 Globals tab.

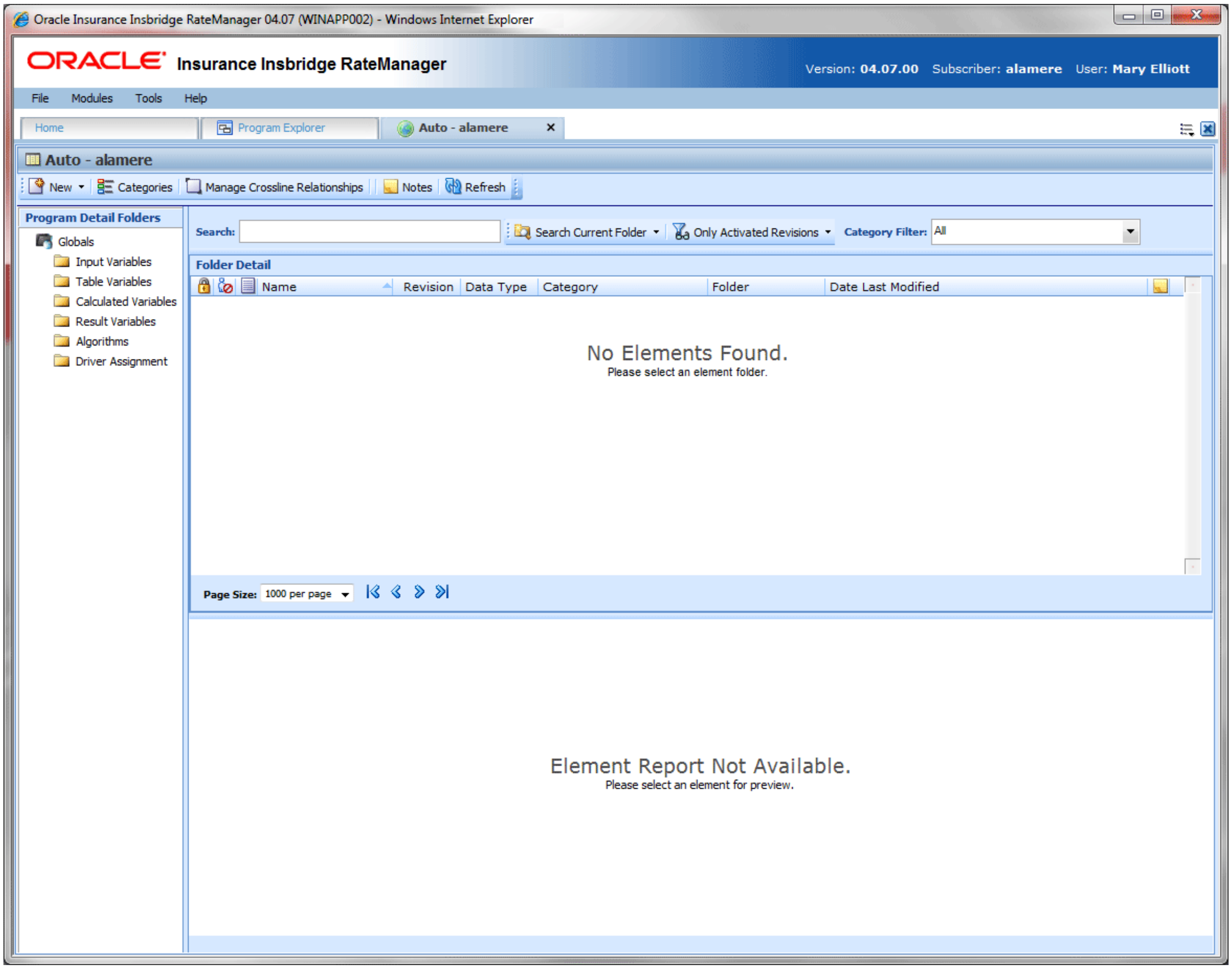
To access the Globals tab:

1. Select the line of business you need and then select the subline you need.
2. A Global icon 🌐 is listed underneath the folders.
3. Double click the global icon and to open the Globals Tab.



Name	ID	Version	Country	State/Province	Parent Folder	Line	Date Last Modified
1-2598-98-001	1	-	USA	--	.../ MI	Auto	Jul 30 2013, 12:01 AM
BI Premium	32	-	USA	--	.../ MI	Auto	Aug 02 2013, 10:35 PM
BI Rate 12	30	-	USA	--	.../ MI	Auto	Jul 29 2013, 08:21 PM

Figure 61 Selecting Globals for the Subline



No Elements Found.
Please select an element folder.

Element Report Not Available.
Please select an element for preview.

Figure 62 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. Inputs, Result Variables, Calculated Variables, and Tables Variables, allow you to click New from the element and open the corresponding tab or popup. You must select the type of Algorithm or Drivers Assignment you want to create from the menu.
Categories	**Categories:** Opens the Category screen where you can view and manage categories.
Manage Crossline Relationships	**Manage Crossline Relationships:** Opens the Crossline Relationships screen where you can setup relationships between lines of business.
Notes	**Notes:** Opens the Notes screen for the selected element.
Refresh	**Refresh:** Refreshes the screen with the current information.

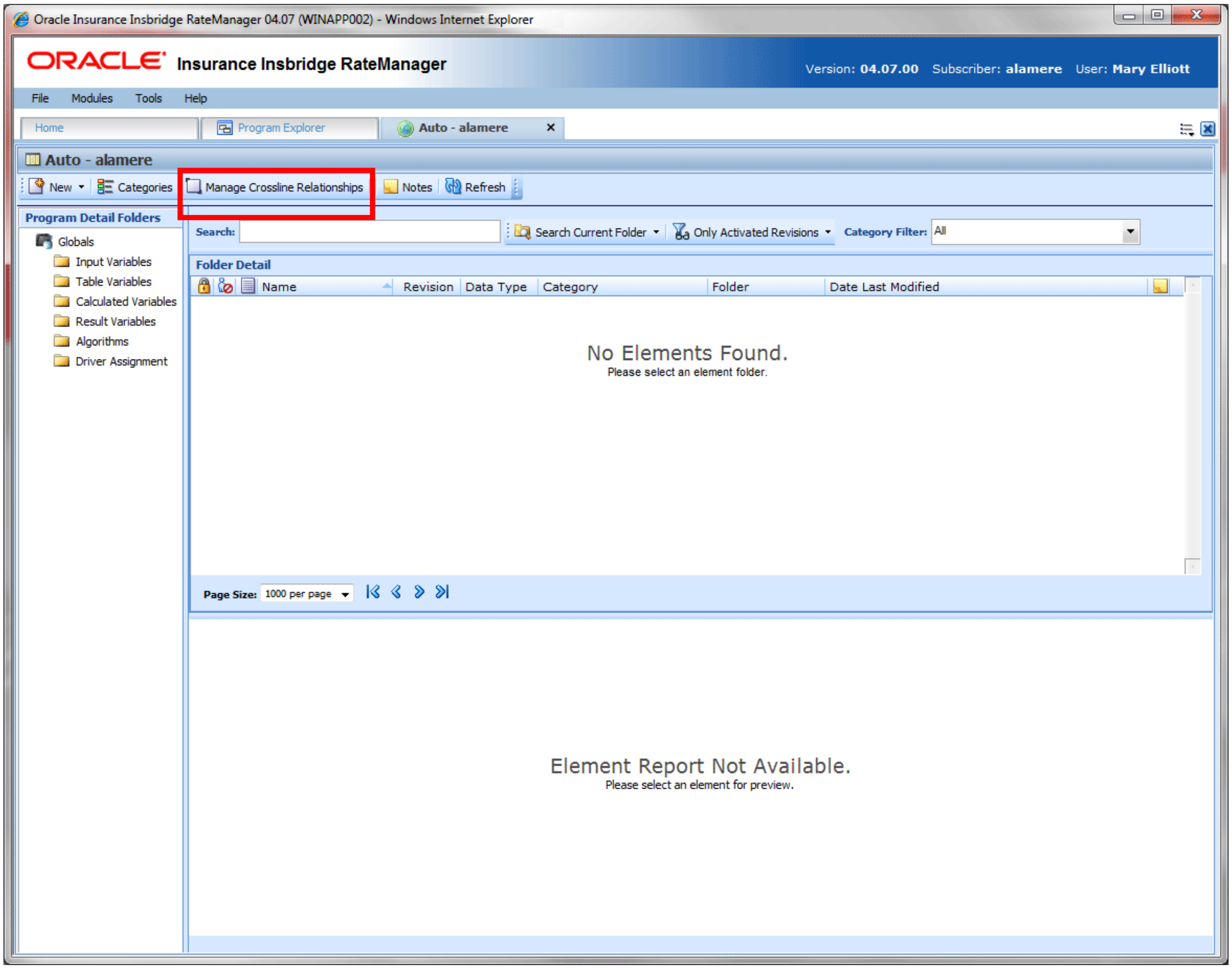
Globals Tab Fields:

The fields on the Global tab perform like the fields on the Program Details tab with the exception of setting revisions. Revisions cannot be set at the global level.

MANAGE CROSSLINE RELATIONSHIPS

Manage Crossline Relationships allows for a subline to callout to programs in another subline in a separate line of business. Each subline will be able to set up a single relationship with one other subline per line of business.

For example, the Alamere subline in the auto line of business can establish a relationship with the ISO-ERC subline in the BOP line of business. The Alamere subline can set up new relationships with other sublines in other lines of business but cannot set up a second relationship in BOP. If another subline was needed in BOP, then the subline selection would have to be updated.



No Elements Found.
Please select an element folder.

Element Report Not Available.
Please select an element for preview.

Figure 63 Selecting to Manage Crossline Relationships

You can setup a relationship with your own subline in another line of business. The Alamere subline in the Home line of business can set up a relationship with the Alamere subline in Auto. This allows a program in the Alamere subline of Home to make a callout to a program in the Alamere subline of Auto.

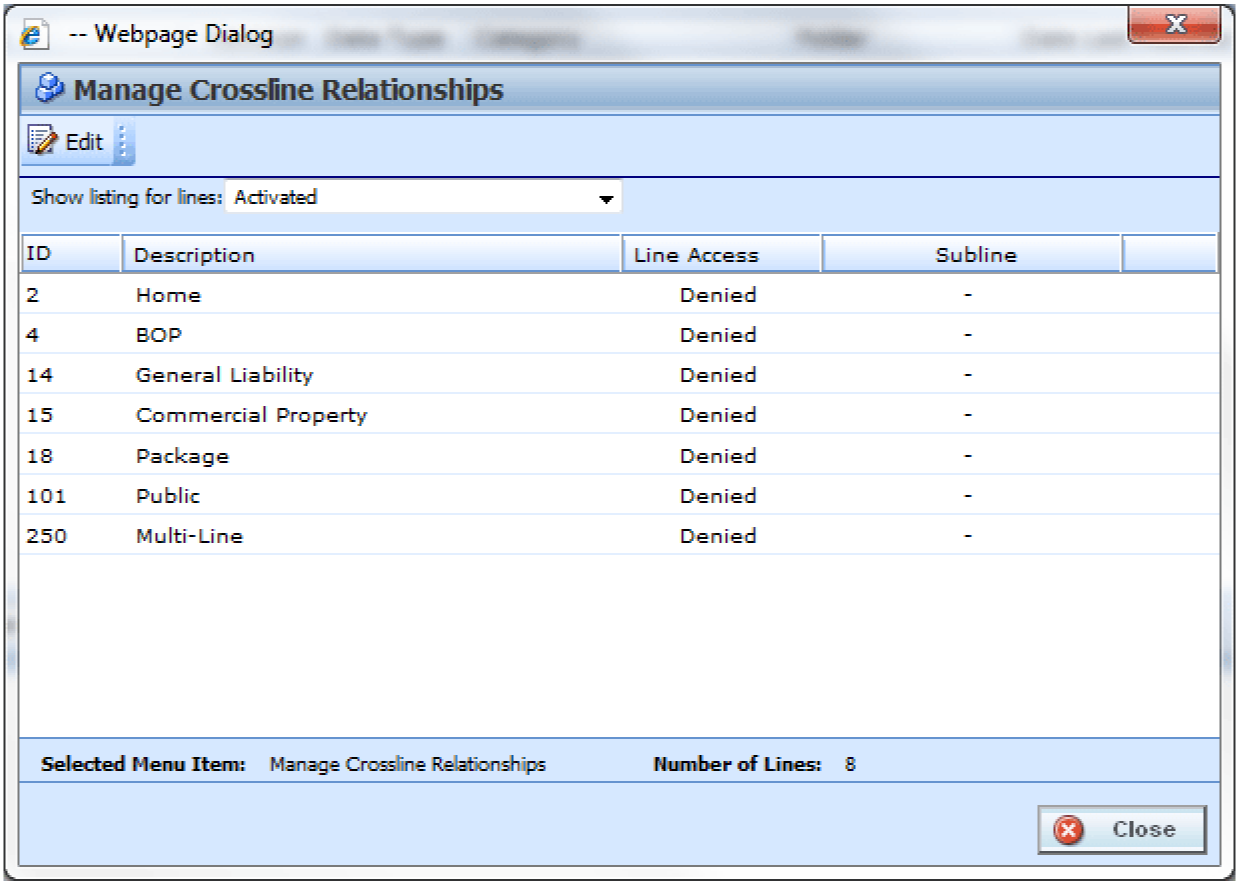
You cannot create a relationship within the same line of business. The Alamere subline in the auto line of business cannot create a relationship with the ISO-ERC subline in the auto line of business. If you are accessing the Manage Crossline Relationship screen from the Auto line of business, Auto will not be an option on the screen.

Relationships are granted per subline per line of business. There are no default relationships. You cannot create a relationship with a subline that does not contain any programs.

Relationships are needed if you want to make a callout to a program in another line of business. Relationships must be granted prior to creating any callouts. Until a relationship is granted, the option to callout to another line of business will not be available.

Navigating to Manage Crossline Relationships

1. Select the subline where you want to establish a crossline relationship.
2. Click the Globals icon to open the globals tab.
3. Click the Manage Crossline Relationship button in the top bar menu.



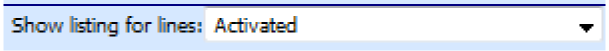
Show listing for lines: Activated

ID	Description	Line Access	Subline
2	Home	Denied	-
4	BOP	Denied	-
14	General Liability	Denied	-
15	Commercial Property	Denied	-
18	Package	Denied	-
101	Public	Denied	-
250	Multi-Line	Denied	-

Selected Menu Item: Manage Crossline Relationships Number of Lines: 8

Figure 64 Editing Crossline Relationships

Functionality:

	Edit: The Edit button opens the screen where you can grant access to the selected subline.
 Show listing for lines: Activated	**Show listing for Lines:** Filters the lines of business by: • **All:** All lines are displayed. • **Activated:** Only active lines are displayed. • **Deactivated:** Only deactivated lines are displayed.

Fields:

ID: The XML ID for the line of business

Description: The Name description for the line of business

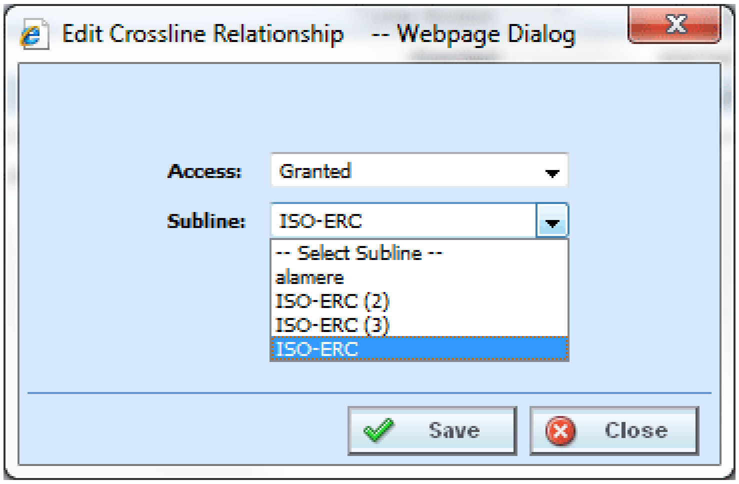
Line Access: Whether access has been granted or denied

Subline: The subline name that this subline can call out to.

Granting Access

To grant access, make sure you are in the line of business and subline from where you want to callout. This is the location of the calling program.

1. On the Manage Crossline Relationships screen, select the listing you want to view; All lines, only Activated lines or only Deactivated lines. The screen will refresh with your selection.
2. Select the line you where you want to create a relationship. This is the target line of business.
3. Click the Edit button.



Access: Granted

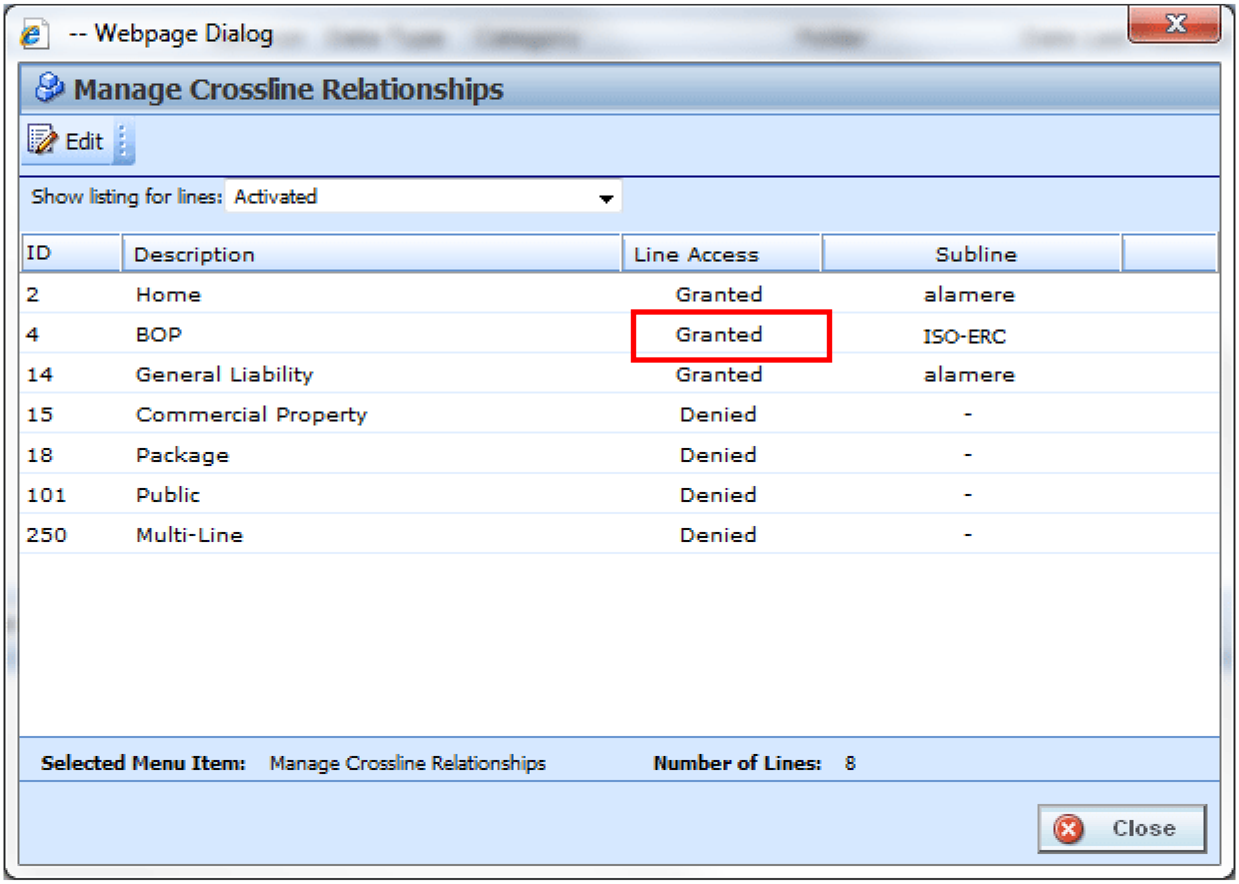
Subline: ISO-ERC

Figure 65 Granting Access to an LOB

4. Select Granted from the Access option.

5. Select the subline 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 Crossline Relationship screen will show that line access has been granted to the selected subline.



Show listing for lines: Activated

ID	Description	Line Access	Subline
2	Home	Granted	alamere
4	BOP	Granted	ISO-ERC
14	General Liability	Granted	alamere
15	Commercial Property	Denied	-
18	Package	Denied	-
101	Public	Denied	-
250	Multi-Line	Denied	-

Selected Menu Item: Manage Crossline Relationships Number of Lines: 8

Figure 66 Completed Crossline Relationship

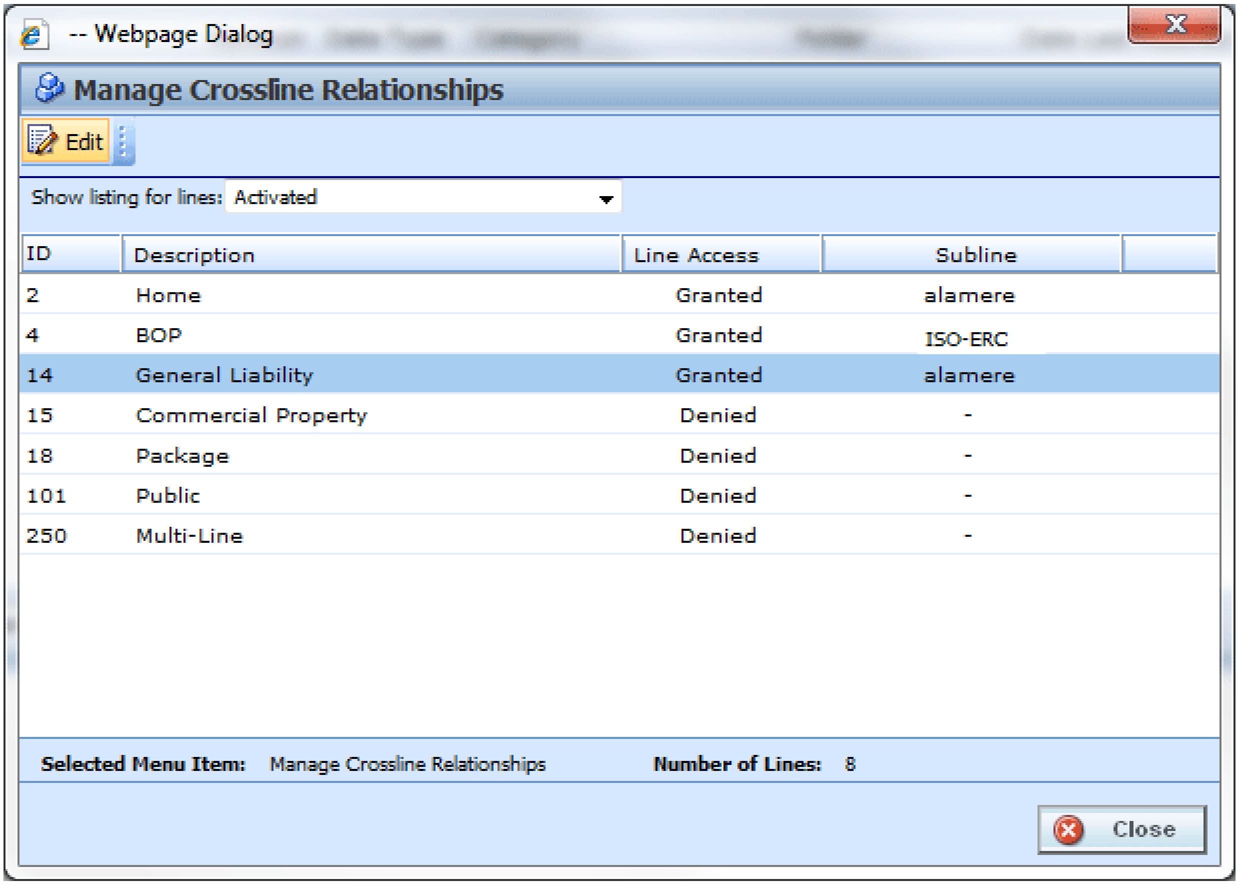
After access has been granted, there are additional checkpoints that must be made before a callout can be implemented. For more information on working with Callouts, please see Chapter 18 Callouts.

Denying Access to Crossline Relationships

If access is no longer needed, you can deny access by changing the line access to Denied. To deny access, make sure you are in the line of business where you created the relationship.

1. On the Manage Crossline Relationships screen, select the listing you want to view; All lines, only Activated lines or only Deactivated lines. The screen will refresh with your selection.

2. Select the line you where you want to deny access. This is the location of the program you no longer want to callout to.



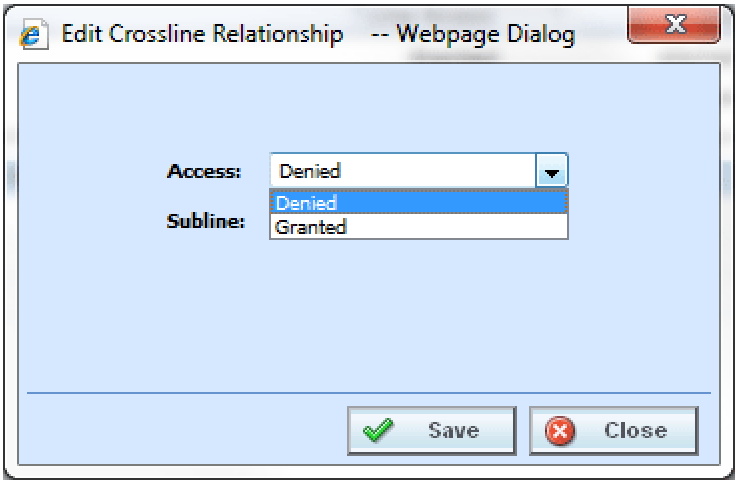
Show listing for lines: Activated

ID	Description	Line Access	Subline
2	Home	Granted	alamere
4	BOP	Granted	ISO-ERC
14	General Liability	Granted	alamere
15	Commercial Property	Denied	-
18	Package	Denied	-
101	Public	Denied	-
250	Multi-Line	Denied	-

Selected Menu Item: Manage Crossline Relationships **Number of Lines:** 8

Figure 67 Editing Access for a Crossline Relationship

3. Click the Edit button.

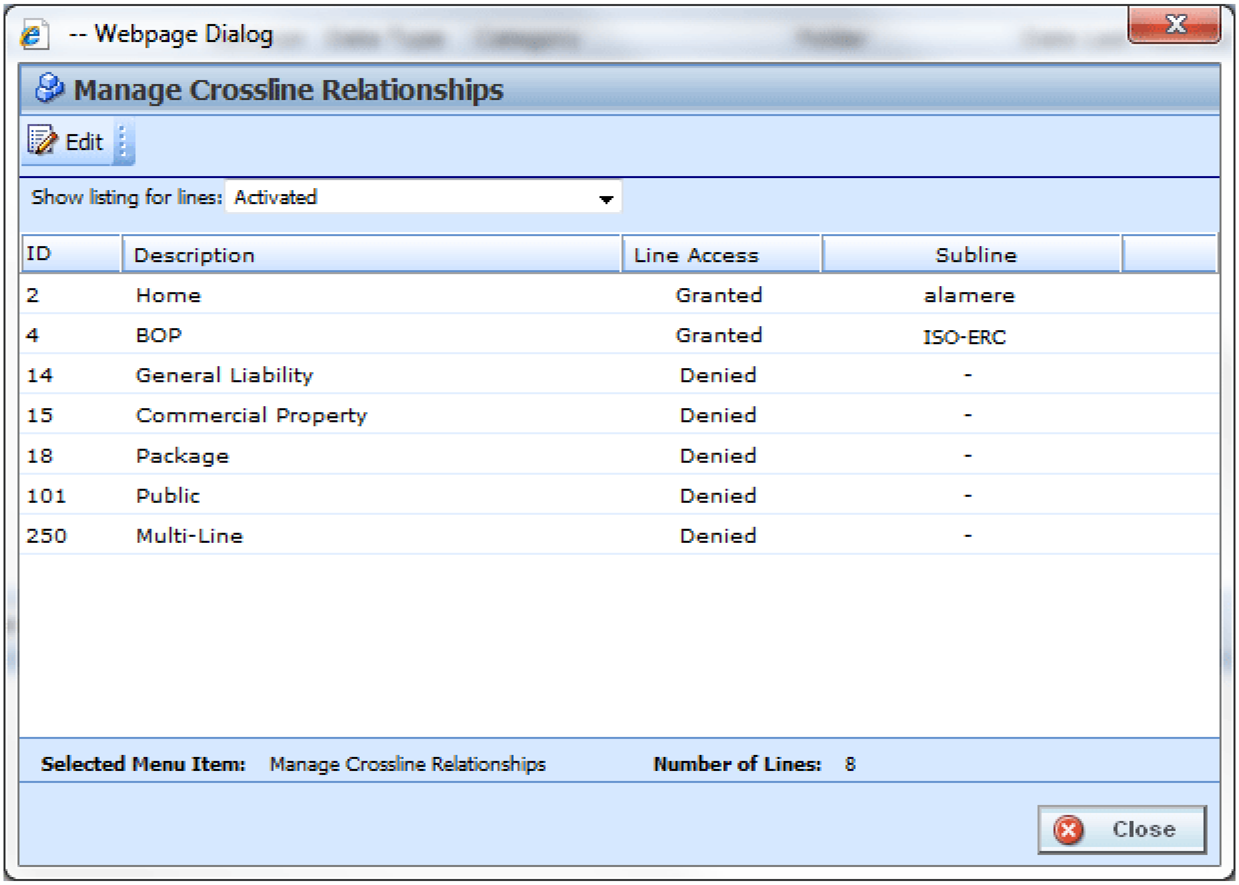


Access: Denied

Subline:

4. Select Denied from the Access option.

5. Click Save to save your entry. The screen will close and the Manage Crossline Relationship screen will show that line access has been denied to the selected subline.



Show listing for lines: Activated

ID	Description	Line Access	Subline
2	Home	Granted	alamere
4	BOP	Granted	ISO-ERC
14	General Liability	Denied	-
15	Commercial Property	Denied	-
18	Package	Denied	-
101	Public	Denied	-
250	Multi-Line	Denied	-

Selected Menu Item: Manage Crossline Relationships

Number of Lines: 8

Figure 68 Completed Crossline Relationship

Chapter 10

CATEGORIES

The category concept is central to program loading in RateManager. Categories define inputs, 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. In other words, 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 inputs together, so that the set will run for every instance of the group. For example, in calculating Driver Accident Points with an input of ViolationCode, you need 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 an input 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=“09/13/1980”/>
   <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>
```

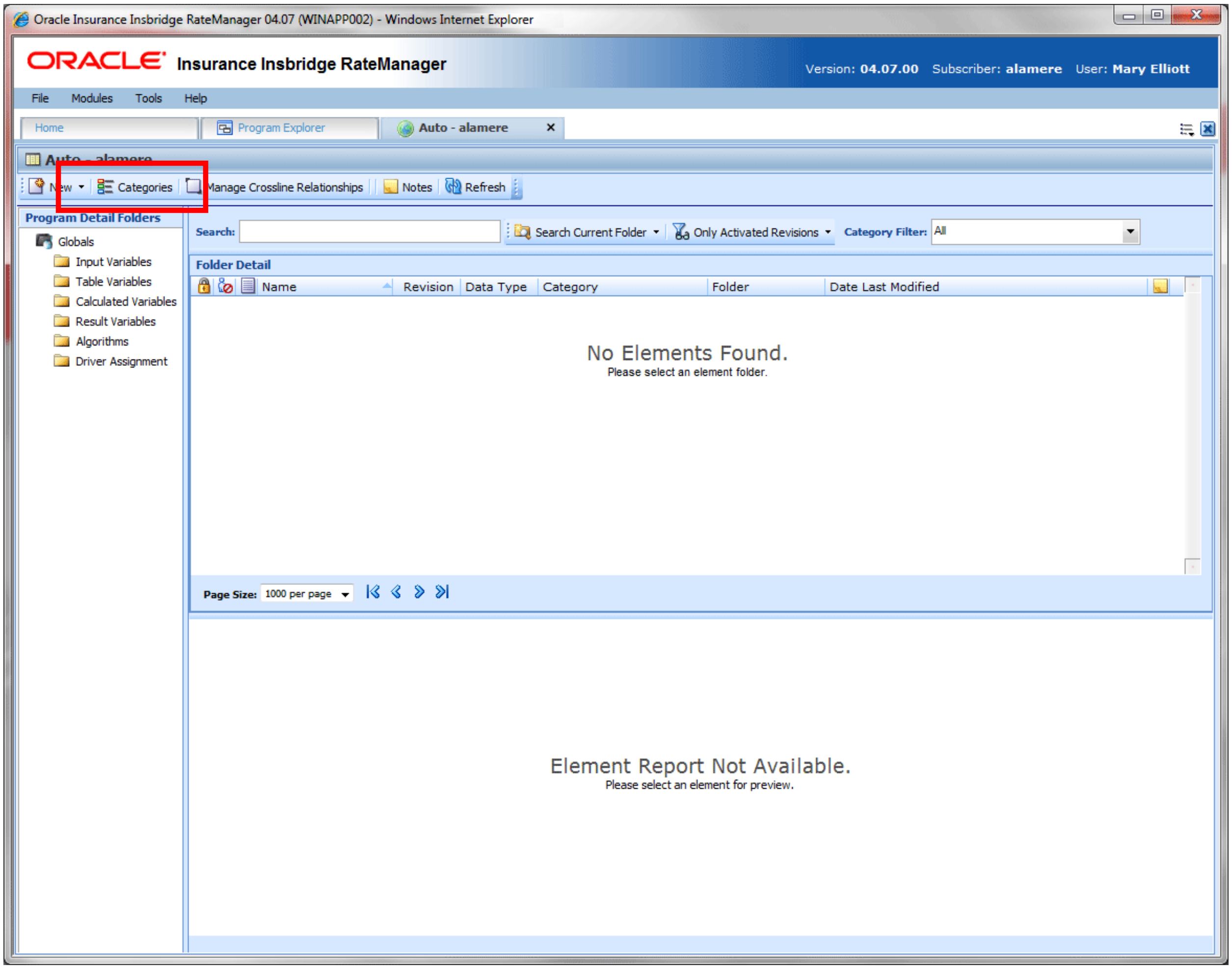
Navigating to Categories:

Categories are subline specific and are not an option on the Program Explorer tab or the Program Details tab. You must select the subline where you want to view or manage categories and then select the globals option for that subline. This opens the Globals tab where you can access the Categories screen.

1. Modules→Program Explorer. This opens the Program Explorer.

2. Select the Line of Business→Subline→**Globals**. This opens the Globals tab.

3. **Categories** are available from a button located on the menu bar.



No Elements Found.
Please select an element folder.

Element Report Not Available.
Please select an element for preview.

Figure 69 Categories Option on the Globals Tab

Working with Categories

The Categories screen lists all the categories for the subline. Every line has a system created Policy category (XML ID 0) that cannot be edited or deleted.

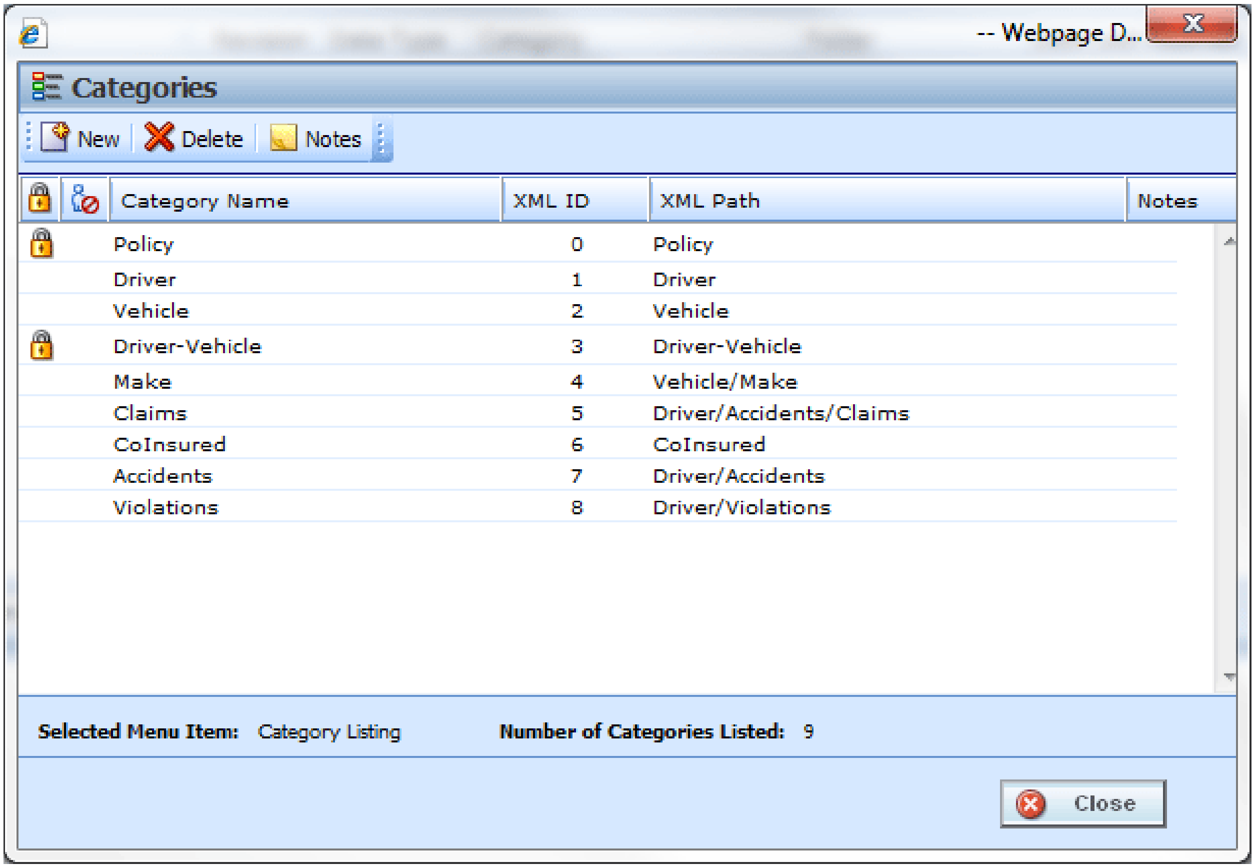
In the Auto LOB, the categories Driver, Vehicle, and Driver-Vehicle cannot be deleted or moved. The XML IDs for these categories cannot be changed and are always the same regardless of RateManager version, subline, or program version. The XML ID for the Driver category is 1, the Vehicle category is 2 and for the Driver-Vehicle the category is 3. The Driver and Vehicle categories will allow for the name to be edited. The Driver-Vehicle category cannot have any other category underneath it.

You can create as many categories as you need. Categories utilizes a parent/child relationship that allows for categories to be placed underneath each other or on the same level. This creates a multi-level set up where you can place an unlimited number of categories underneath the top level or any other category.

Top is the 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.

Categories that are locked cannot be edited or deleted. Locked categories have a lock icon.

NOTE: *In the auto line of business, your initial working category options are Policy, Driver, Vehicle and Driver-Vehicle. To create a variable that used both driver and vehicle inputs, you would need to define this variable in a Driver-Vehicle working category.*



		Category Name	XML ID	XML Path	Notes
🔒		Policy	0	Policy	
		Driver	1	Driver	
		Vehicle	2	Vehicle	
🔒		Driver-Vehicle	3	Driver-Vehicle	
		Make	4	Vehicle/Make	
		Claims	5	Driver/Accidents/Claims	
		CoInsured	6	CoInsured	
		Accidents	7	Driver/Accidents	
		Violations	8	Driver/Violations	

Selected Menu Item: Category Listing **Number of Categories Listed:** 9

Figure 70 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 New 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	**New:** Creates a new category.
Delete	**Delete:** Deletes the currently selected category.
Notes	**Notes:** Opens the Notes screen for the category.

CATEGORIES FIELDS

Categories can be sorted by the Category Name and XML ID fields. To change the order, click the field 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:

Shows a listing of categories for the subline. To open a category, double-click it.

Icon: There are two icon fields:

Locked. A lock in this field indicates that the category is locked.

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 if a number is not entered and 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 will depend upon the category. Policy will have three options, driver and Vehicle will have four options, and Driver-Vehicle will have two options. All User created categories will have five options.

To view the right click menu, select the category and right click.

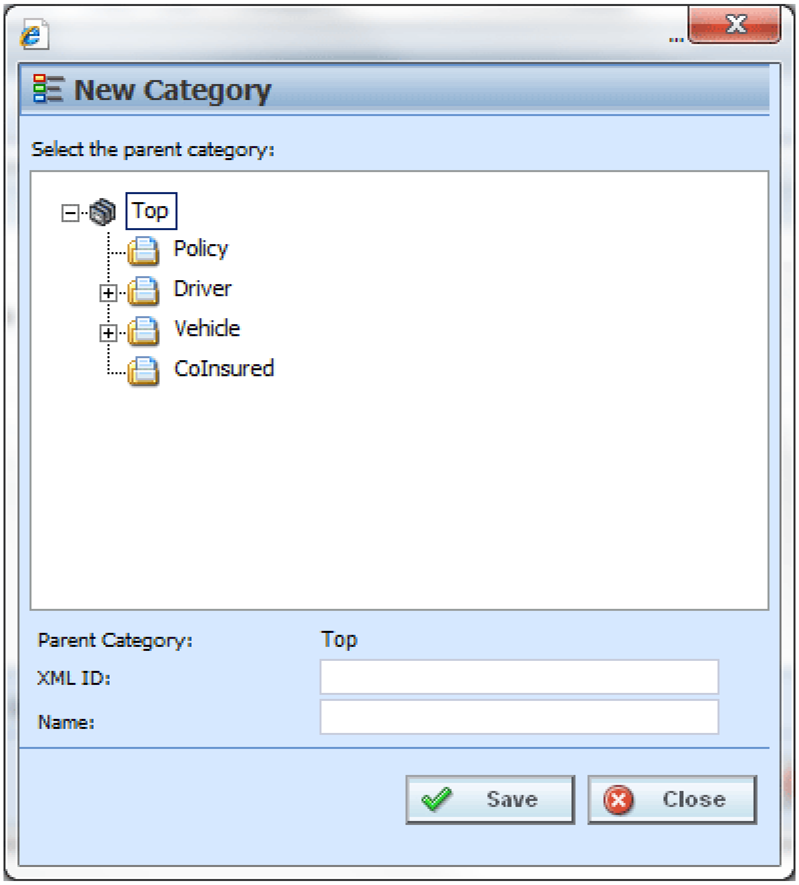
- **New:** Opens the New Category screen where you can create a new category. New categories cannot be created under Driver-Vehicle.

- **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

1. Navigate to the **Category** screen for the subline where you want to create a new category.
2. Click **New**. You also can highlight most existing categories and right click. Select **New** from the menu.
3. A **New Category** popup is shown.



New Category

Select the parent category:

- Top
  - Policy
  - Driver
  - Vehicle
  - CoInsured

Parent Category: Top

XML ID:

Name:

Figure 71 Creating New Category

4. 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 will have a box around it.

5. **OPTIONAL:** If you need 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.

6. 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.

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.*

Editing a Category

Editing a category allows you to change the category's name or change the parent category.

Changing a parent category allows you to change the category location within the same subline. 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 will always be 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.

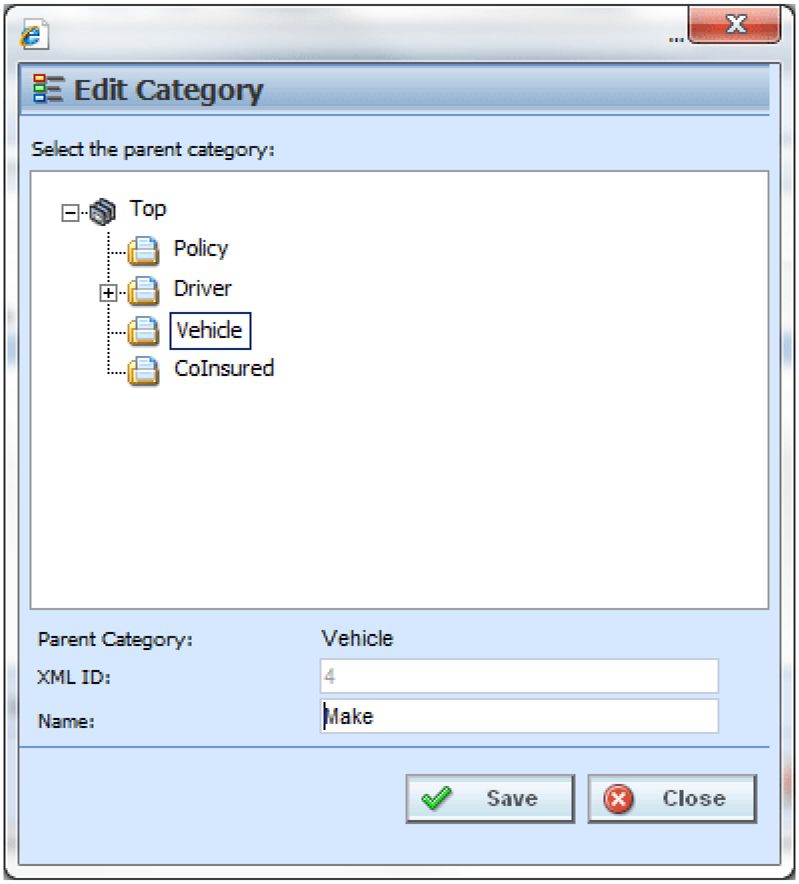
WARNING: ***If you have already started creating your program, changing the parent category is not recommended.***

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 subline 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** popup is displayed.



Select the parent category:

Parent Category: Vehicle

XML ID: 4

Name: Make

Figure 72 Editing a Category

4. To change file location, choose a new parent category. The folder you have selected to move will be 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 will be available to be assigned to a new category.

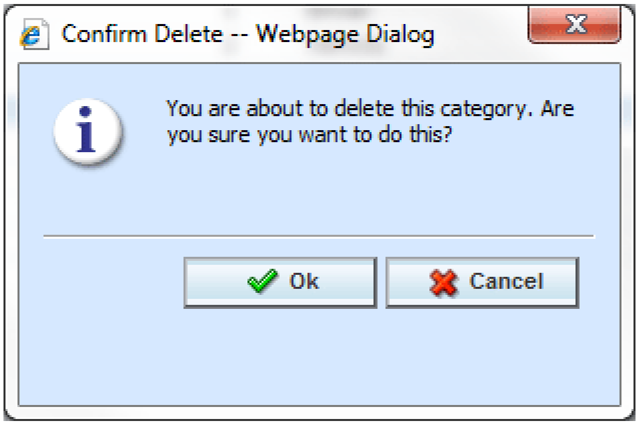
In the Auto LOB, the categories Driver, Vehicle and Driver-Vehicle cannot be deleted.

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 subline 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 popup menu.



You are about to delete this category. Are you sure you want to do this?

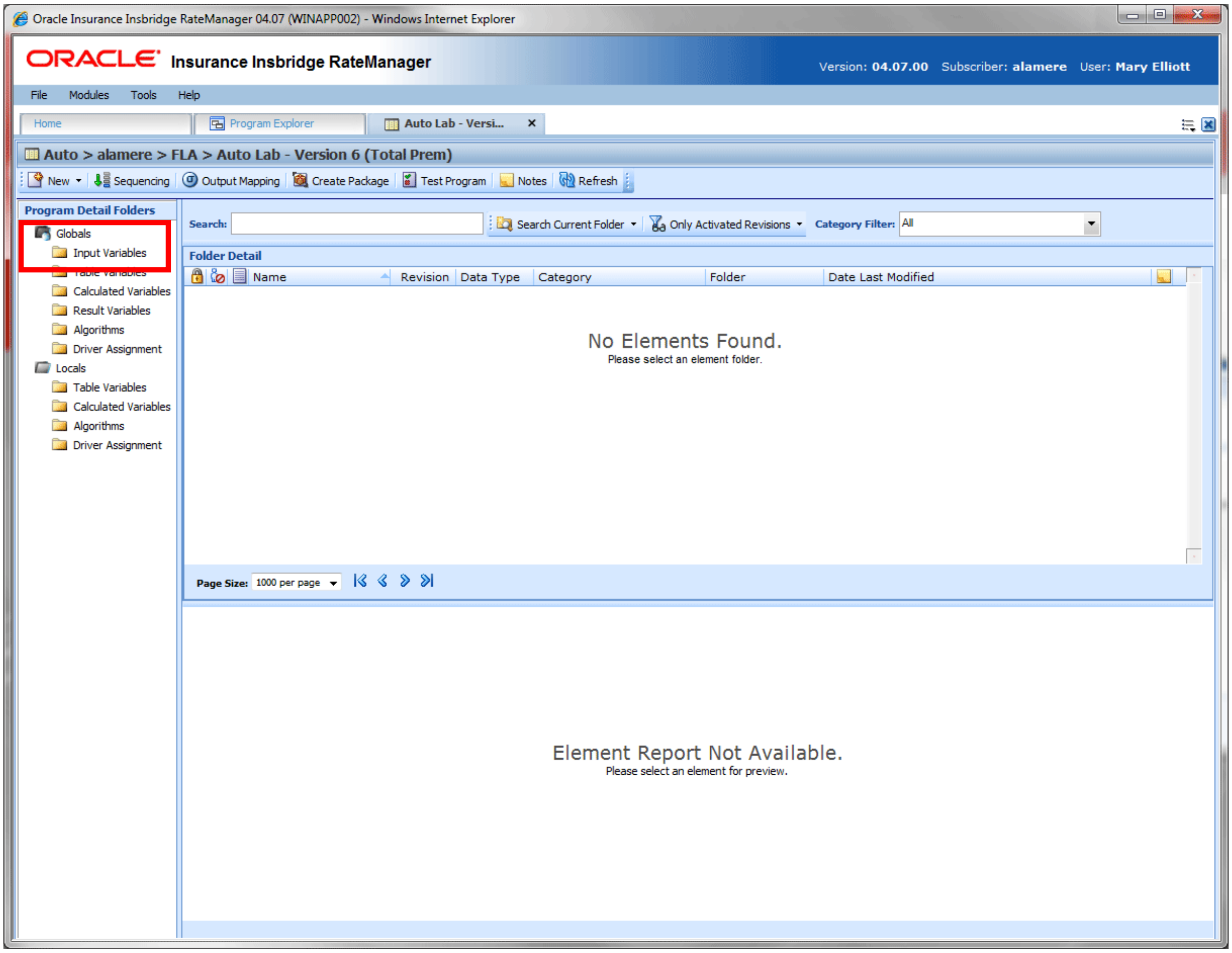
Figure 73 Deleting a Category

3. The category will be deleted and the category listing will be refreshed. If the category is still being used by an element, you will receive an error message.

INPUTS

Inputs are a way for you to pass information into the SoftRater engine. Inputs are subline specific and managed at the Global level. There are two areas to view and manage inputs, either from the Globals tab or from the Program Details tab. When selected, inputs will be listed in the Folder Details area.

There are user created inputs, inputs brought in from a template, and system inputs. User created inputs are open and can be edited, copied, or deleted unless locked. A user created input that is locked only allows for copying. Inputs 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 inputs.



No Elements Found.
Please select an element folder.

Element Report Not Available.
Please select an element for preview.

Figure 74 Selecting Inputs

System generated inputs are grayed out and cannot be edited, copied or deleted. The auto line of business has 24 system created inputs that are needed for driver assignment. For example, VehPrincipallyDriven (XML ID=299) is a system input. These inputs cannot be edited, copied, or deleted. It is recommended that you do not create another input using a system input name.

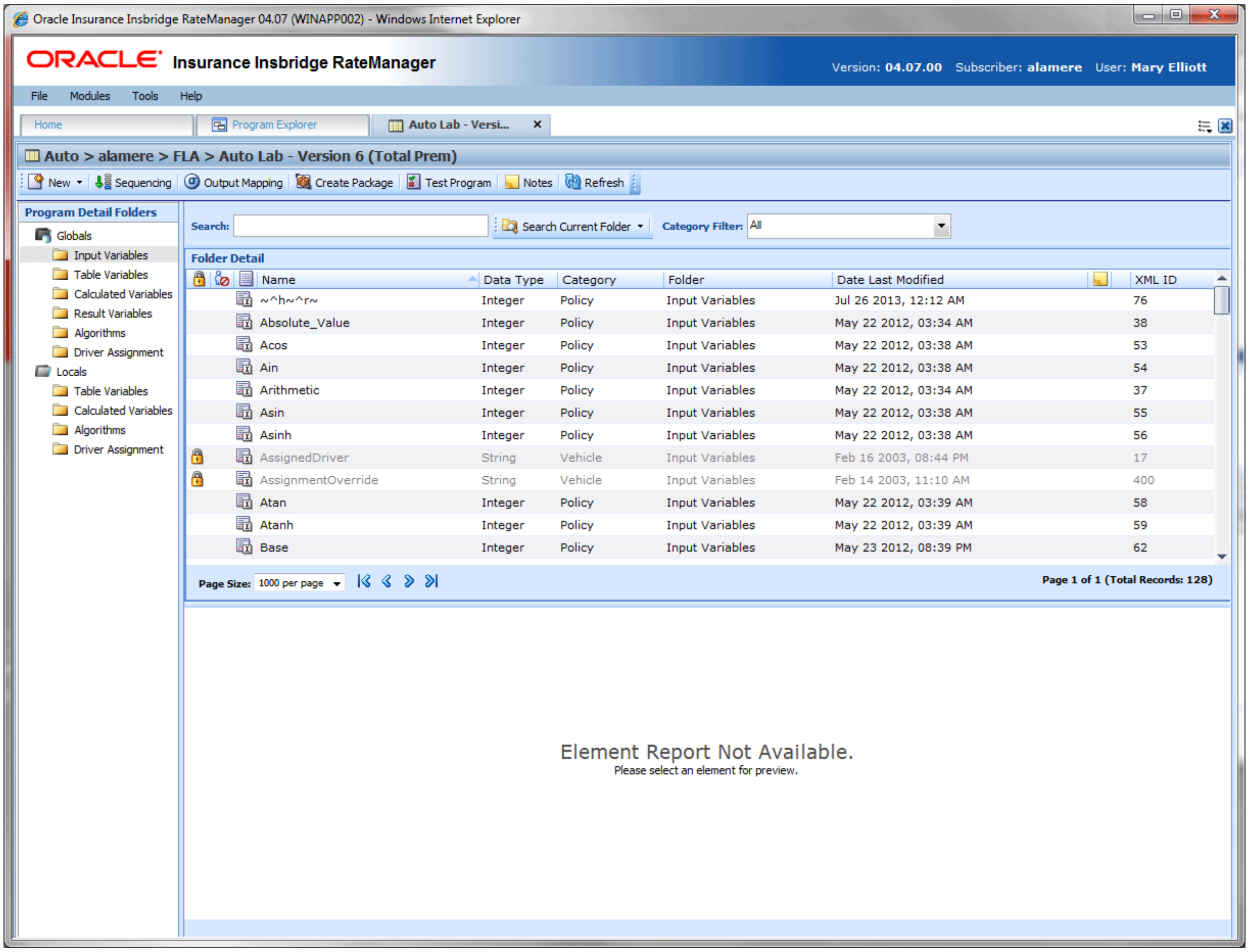
Navigating to Inputs:

Inputs are not an option on the Program Explorer tab.

1. Modules→Program Explorer. This opens the Program Explorer tab.

2. Select the Line of Business→Subline→Folder→Program → Program Version. You also can access Inputs from the globals tab. Line of Business→Subline→Globals.

3. **Open** the program version to get the Program Details tab. Inputs are available from the Program Detail Folders area on the left hand side of the tab.



🔒			Name	Data Type	Category	Folder	Date Last Modified		XML ID
			~^h~^r~	Integer	Policy	Input Variables	Jul 26 2013, 12:12 AM		76
			Absolute_Value	Integer	Policy	Input Variables	May 22 2012, 03:34 AM		38
			Acos	Integer	Policy	Input Variables	May 22 2012, 03:38 AM		53
			Ain	Integer	Policy	Input Variables	May 22 2012, 03:38 AM		54
			Arithmetic	Integer	Policy	Input Variables	May 22 2012, 03:34 AM		37
			Asin	Integer	Policy	Input Variables	May 22 2012, 03:38 AM		55
			Asinh	Integer	Policy	Input Variables	May 22 2012, 03:38 AM		56
🔒			AssignedDriver	String	Vehicle	Input Variables	Feb 16 2003, 08:44 PM		17
🔒			AssignmentOverride	String	Vehicle	Input Variables	Feb 14 2003, 11:10 AM		400
			Atan	Integer	Policy	Input Variables	May 22 2012, 03:39 AM		58
			Atanh	Integer	Policy	Input Variables	May 22 2012, 03:39 AM		59
			Base	Integer	Policy	Input Variables	May 23 2012, 08:39 PM		62

Element Report Not Available.
Please select an element for preview.

Figure 75 Input Listing

INPUTS FUNCTIONALITY

Functions can be accessed from the menu bar located at the top of the screen and from the right click menu of any input. The functions you can perform depends on the status of the input. Locked inputs have fewer options. There are icons that can tell you at a glance what the status of the input is.

Icon	What it Indicates
📄	**Input.** Inputs are subline specific and can be created at the global level, from a program version, or from another program element. All inputs are displayed on the Program Details→Input tab. The same input listing also is available for Subline→Globals tab. The same options, new, edit, copy and delete are available on either tab.
🔒	**Locked Input.** Locked inputs are not open for editing. A lock icon indicates that this input is locked by either a program or is system generated. System generated inputs are not open for copying, editing, or deleting. Program locked inputs can be copied. If you need to edit a program locked input, you must return to the program and unlock it. This unlocks the elements and changes can be made. If the input is locked because it is system generated, you cannot unlock the input.
🔒	**Template Lock.** If the input is locked because it came from a template, you cannot unlock the input. You can copy the input or create a new input for use in the program if necessary.
🚫	**In-Use.** Inputs that are in use are currently being worked on by another user in the system are not open for editing, copying, or deleting. The input will be available when the user exits the input.

INPUTS FIELDS

The fields are the same for the Global Tab and Program Details. For more on field descriptions, please see Program Details→Fields.

Right Click Menu Options:

To view right click menu options, highlight the input you want to view and right click. Unlocked input variables will have six right click menu options:

- **Edit:** Edits the input.
- **Copy:** Brings up the copy screen for the input.
- **Delete:** Removes the input 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:** Allows you to enter, edit, and view notes for the input.

Template locked input variables and program locked inputs have four options, Copy Variable, View Where Used, View History, and Notes.

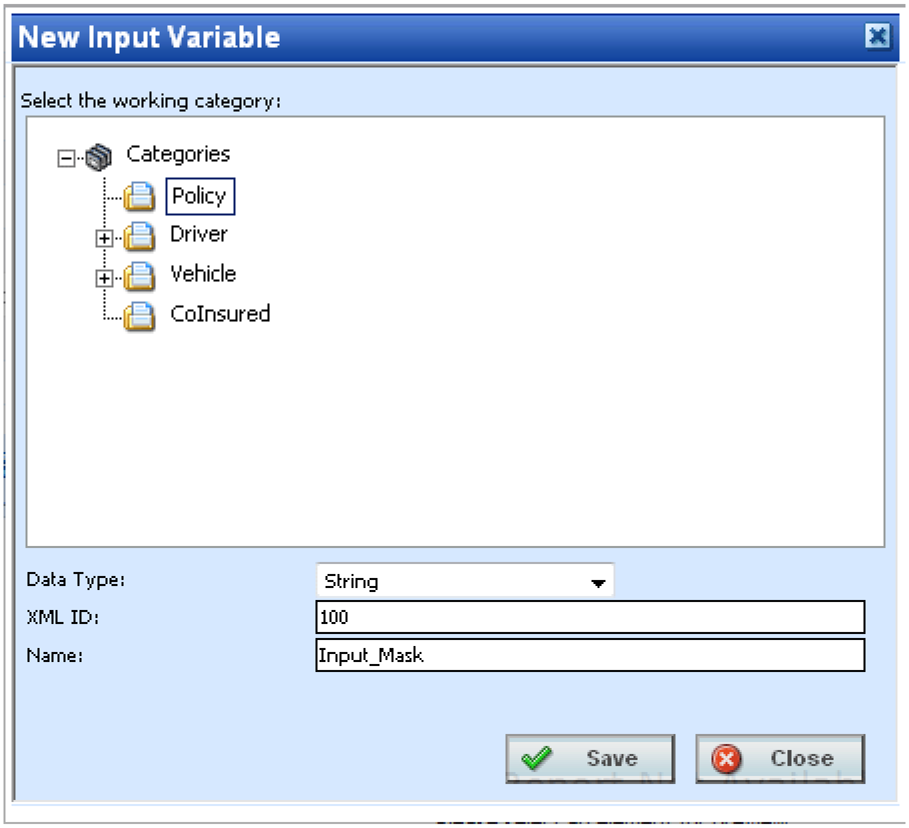
CREATING AN INPUT

In order to create a new input, you must have your categories set up. RateManager automatically has a Policy category for each subline. Inputs can be template generated or created by you.

Template generated inputs are locked and not open for editing, or deletion. A user created input can be added to a template generated subline if needed. The new variables are open for editing, copying, or deletion.

Inputs should be unique at the category level. This means that you should use unique input names in every category. For example, if you need a BaseRate input for Policy and Driver, you can create a BaseRate input for the policy category and a BaseRateDR input for the Driver category.

1. On the Program Details tab, select Inputs from the Program Detail Folders. Click **New** in the top bar menu. This opens the **New Input Variable** popup.



New Input Variable

Select the working category:

Data Type: String

XML ID: 100

Name: Input_Mask

Figure 76 Global Input Variable Popup

2. Double click the category where you want the new input 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 input will be placed will have a box around it.

3. Select a **Data Type** from the menu.

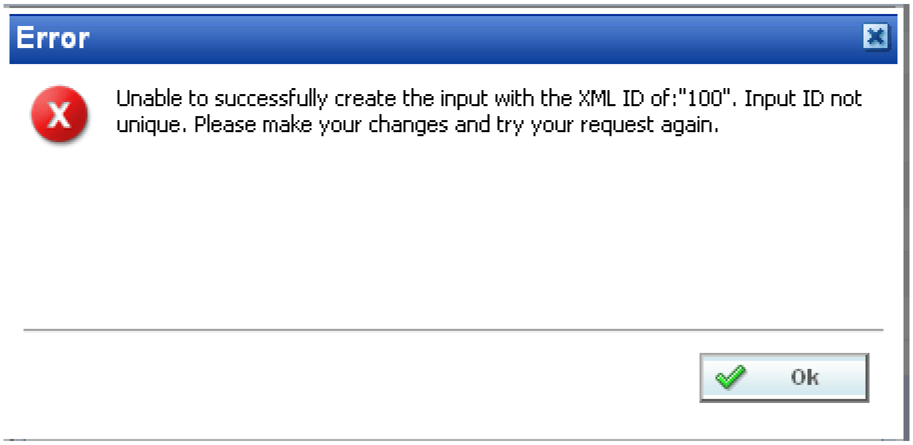
4. **OPTIONAL:** If you need to create an input with a specific XML ID, you can enter it. If the number is already in use, you will receive an error message. If you do not enter a number, RateManager will automatically assign one.

5. Enter in a **Name**.

6. Click **Save** to save your input. The screen refreshes and the new input is displayed on the Program Details→Input tab. An XML ID is automatically assigned for any new input variable if a number is not entered.

Custom XML ID

Custom XML IDs are an optional feature that allows you to 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 subline but do not have to be sequential. You can enter an ID of 97 for an input and then enter 530 for the next input. You do not have to create the next input as ID 98. If an ID is being used, you will receive an error message.



Unable to successfully create the input with the XML ID of:"100". Input ID not unique. Please make your changes and try your request again.

Figure 77 Custom XML ID Error

Custom XML IDs allow you to match inputs across sublines. For example, if you have an input that every subline is going to use, such as Effective Date, you can create the same input using the same ID in every subline. This can be useful when creating programs for crossline P2P.

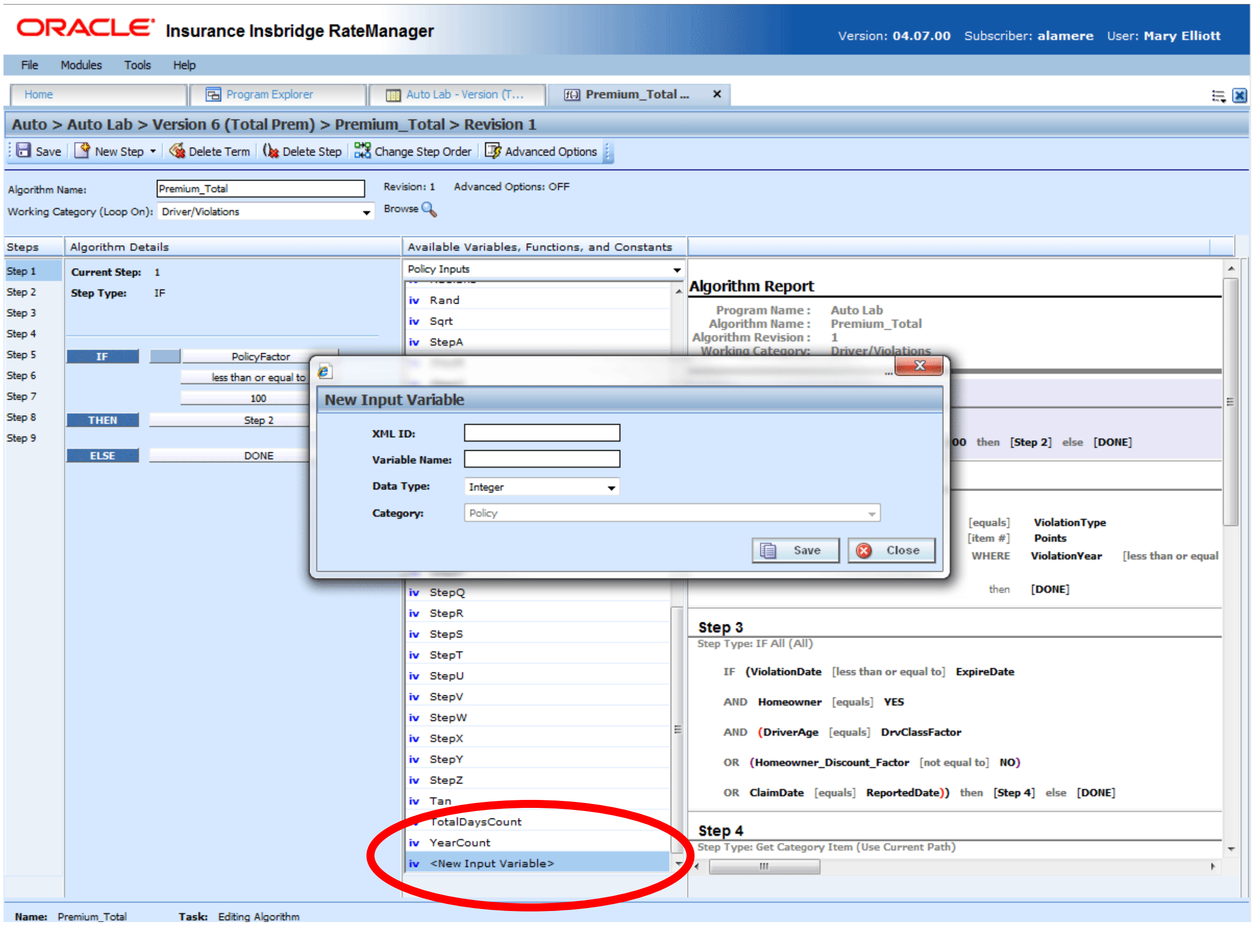
When entering a custom XML ID, please verify your work prior to saving. XML IDs cannot be edited. If you have made a mistake, you will have to recreate the category or input. The incorrect entry can be deleted if necessary. Deleting a category or input releases the XML ID to be used again.

Custom XML IDs cannot be created on copied inputs. Copied inputs have system generated XML IDs.

Custom XML IDs are available in Categories and Inputs.

Adding an Input from another Area

A new input 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 input to is determined by the Available Variables, Functions and Constants selection. Except for the XML ID, the new input can be edited on the Program Details→Input tab, if needed.



Auto > Auto Lab > Version 6 (Total Prem) > Premium_Total > Revision 1

Algorithm Name: Premium_Total
Revision: 1 Advanced Options: OFF
Working Category (Loop On): Driver/Violations

Current Step: 1
Step Type: IF

IF PolicyFactor less than or equal to 100
THEN Step 2
ELSE DONE

Policy Inputs

New Input Variable

XML ID:
Variable Name:
Data Type: Integer
Category: Policy

Algorithm Report

Program Name : Auto Lab
Algorithm Name : Premium_Total
Algorithm Revision : 1
Working Category: Driver/Violations

00 then [Step 2] else [DONE]

[equals] ViolationType
[item #] Points
WHERE ViolationYear [less than or equal

then [DONE]

Step 3
Step Type: IF All (All)

IF (ViolationDate [less than or equal to] ExpireDate
AND Homeowner [equals] YES
AND (DriverAge [equals] DrvClassFactor
OR (Homeowner_Discount_Factor [not equal to] NO)
OR ClaimDate [equals] ReportedDate)) then [Step 4] else [DONE]

Step 4
Step Type: Get Category Item (Use Current Path)

Name: Premium_Total Task: Editing Algorithm

Figure 78 Adding an Input Variable

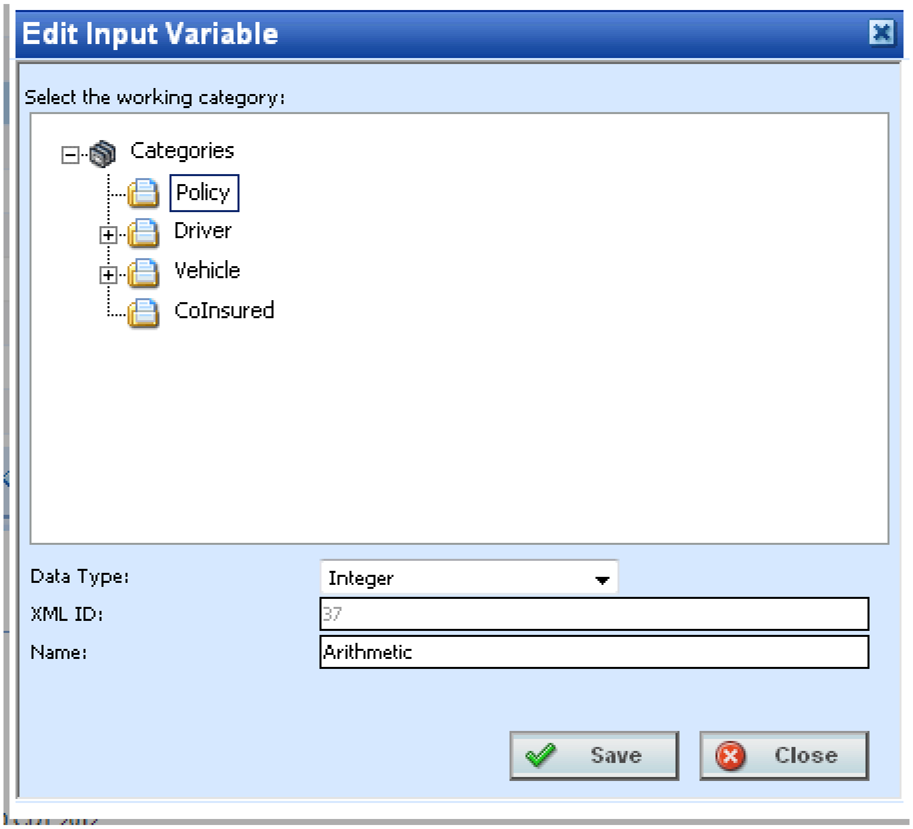
1. Select the Input category you need from the Available Variables, Functions and Constants.
2. Scroll down to the bottom of the inputs list. Select **<New Input Variable>**. The New Input Variable screen will be displayed.
3. **OPTIONAL:** If you need to create an input with a specific XML ID, you can enter it. If the number is already in use, you will receive an error message. If you do not enter a number, RateManager will automatically assign one.
4. Enter in a **Variable Name**.
5. Select the **Data Type**.
6. Click **Save** to save your entry. The popup window closes. Your new input variable is available for selection.

EDITING AN INPUT

Unlocked inputs can be edited at any time. If the input is locked because the program is locked, you must return to the program and unlock it. Once unlocked, changes can be made. If the input is locked because it came from a template, you cannot unlock it. System generated inputs are grayed out and are not available for editing at any time. XML IDs cannot be edited.

NOTE: *If a change is made to an input, the change affects all programs that use that input.*

1. Either double click the input you want to edit or right click and select **Edit**. This opens the **Edit Input Variable** popup.



Data Type: Integer
XML ID: 37
Name: Arithmetic

Figure 79 Editing Global Input Variable

2. To change the **Category**, double click the category where you want the input to be located. On the tree view, the final destination where the input will be placed will have a box around it.

3. Make any changes to the **Data Type** and **Name**.

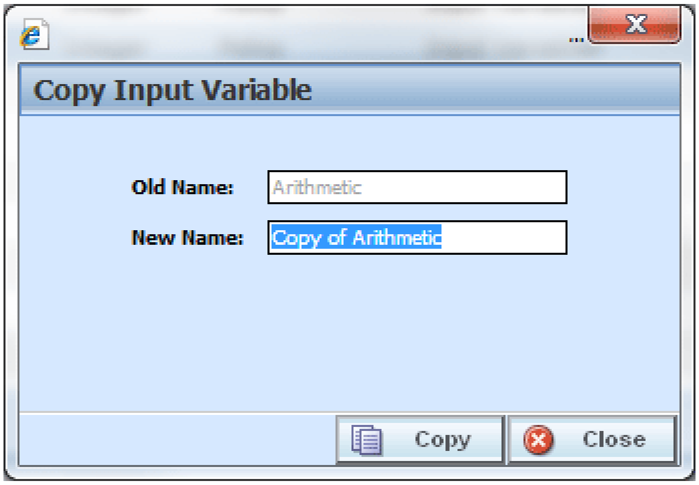
4. Click **Save** to save your changes.

Copying an Input

Inputs can be copied at any time. The input is copied with the exact same data type and category. To make a change to the input, you must go into the edit screen. System generated inputs cannot be copied. Names cannot be duplicated within the same category.

Copied inputs will receive an auto generated XML ID from the system. You cannot enter an XML ID for a copied input. If you need to enter an XML ID, you must create a new input.

1. Right click the input you want to copy. Select **Copy**. The Copy Input Variable popup is shown.



Copy Input Variable

Old Name: Arithmetic

New Name: Copy of Arithmetic

Figure 80 Copying Global Input Variable

2. Enter in a **New Name**.

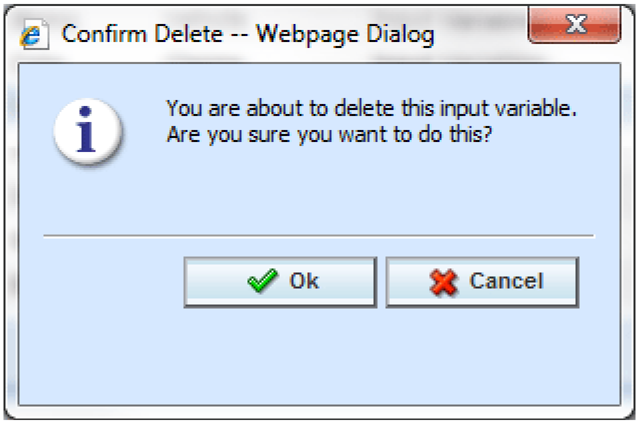
3. Click **Copy** to copy your input. The screen will refresh and the copied input is shown on the Program Details→Input tab. An XML ID is automatically assigned for the input.

DELETING AN INPUT

Any unlocked input variable that is not in use can be deleted. Inputs that are in use, locked, or system generated cannot be deleted. If you try to delete an input that is unavailable for deletion, an error message is displayed informing you where the inputs are being used. System generated inputs are grayed out.

NOTE: *Take care when deleting Inputs, this action cannot be undone.*

1. Global inputs can be deleted by right clicking the input and selecting **Delete**.



You are about to delete this input variable.
Are you sure you want to do this?

Figure 81 Deleting an Input

2. A confirmation message is displayed.
3. Click **OK** to delete or **Cancel** to cancel. If the input is being used, you will not be able to delete. A message will be displayed informing you where the input is being used.

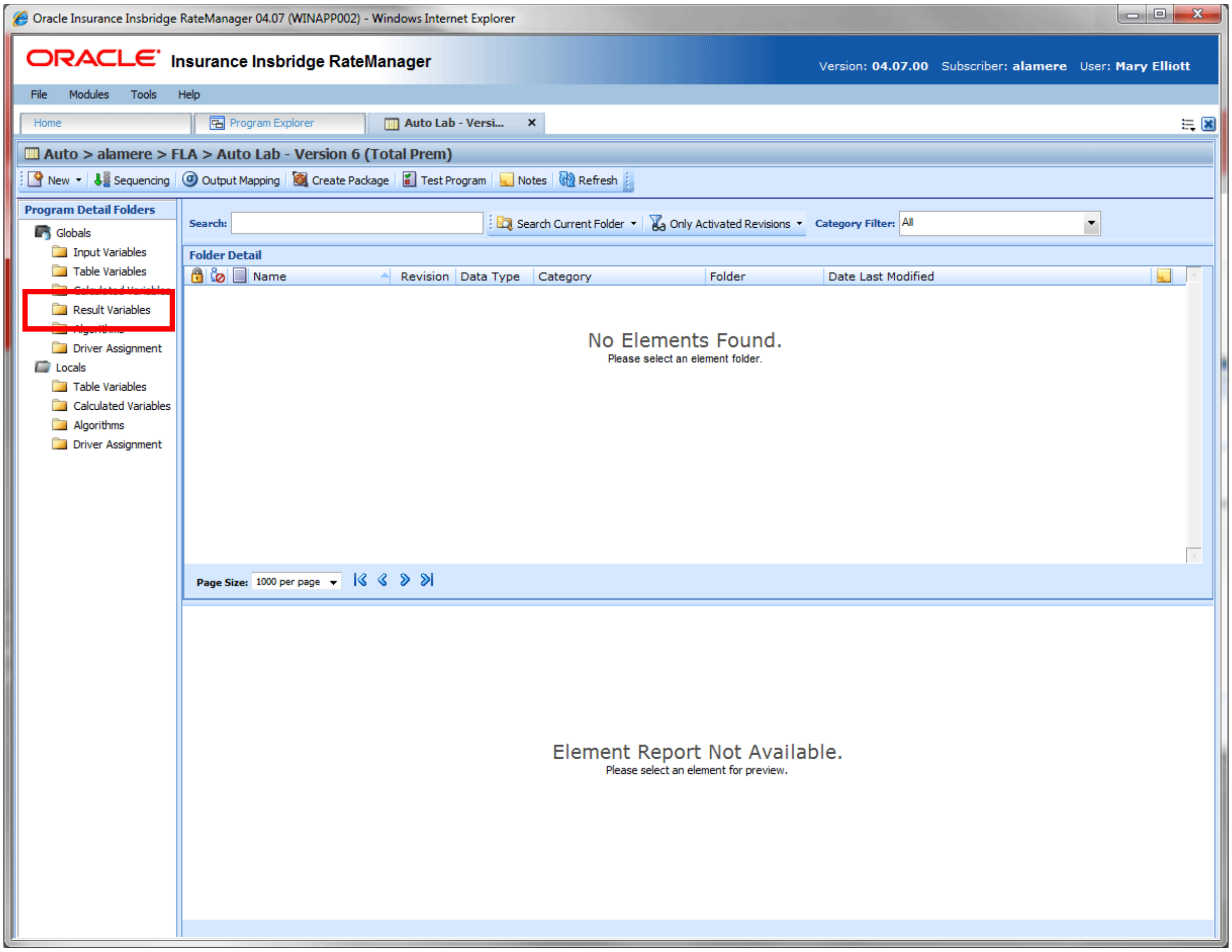
Chapter 12

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 subline, it is referred to as a global variable. Result variables can only be created at the global 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 will be in place for selection from the drop down listings when you need them in your calculation sequence.



No Elements Found.
Please select an element folder.

Element Report Not Available.
Please select an element for preview.

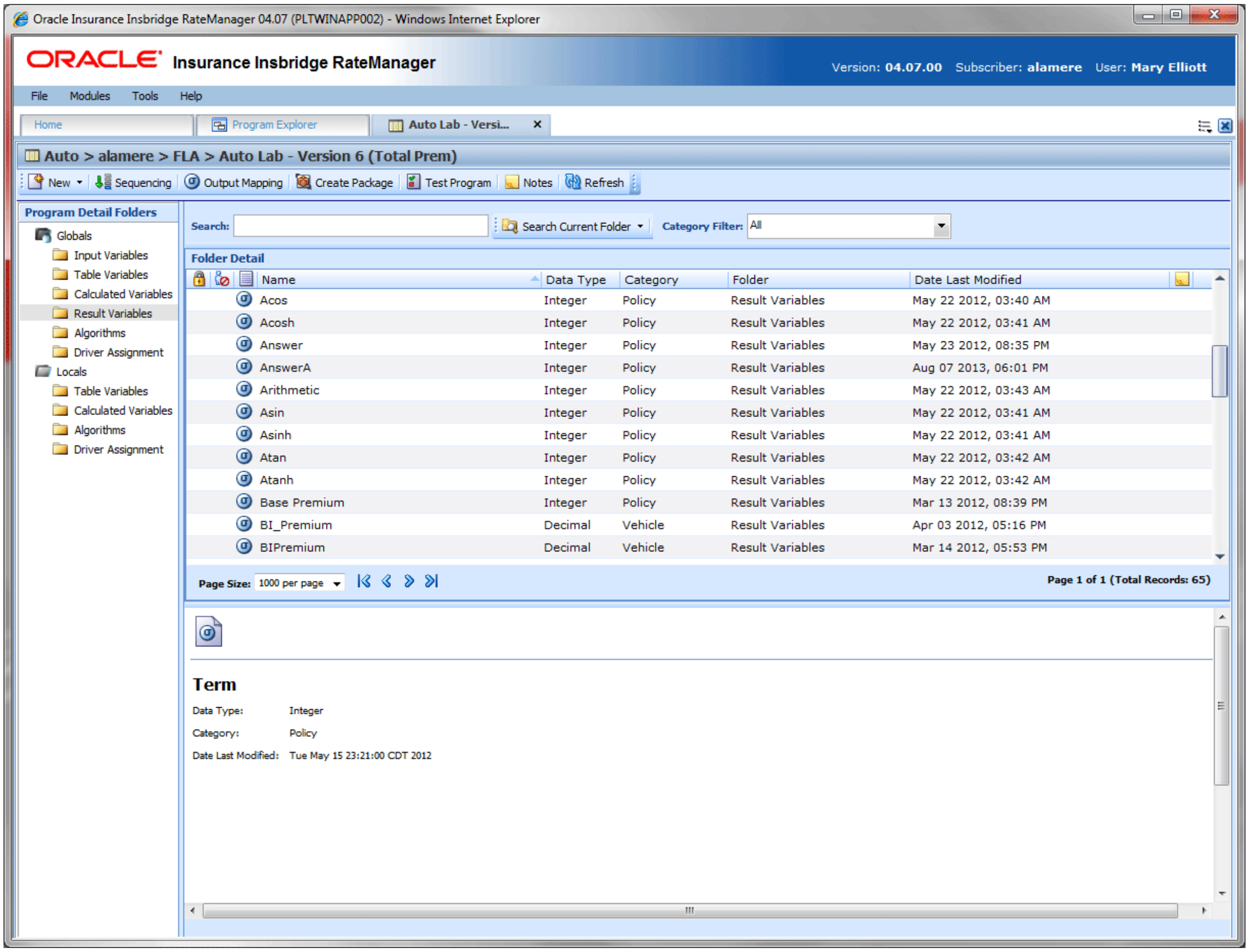
Figure 82 Selecting Result Variables

Result variables can be created by you or brought in from a template. Variables that are user created will be 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 will be 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 will be 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. This opens the Program Explorer tab.
2. Select the Line of Business→Subline→Folder→Program → Program Version. You also can access Result Variables from the globals tab. Line of Business→Subline→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 hand side of the tab.



Name	Data Type	Category	Folder	Date Last Modified
Acos	Integer	Policy	Result Variables	May 22 2012, 03:40 AM
Acosh	Integer	Policy	Result Variables	May 22 2012, 03:41 AM
Answer	Integer	Policy	Result Variables	May 23 2012, 08:35 PM
AnswerA	Integer	Policy	Result Variables	Aug 07 2013, 06:01 PM
Arithmetic	Integer	Policy	Result Variables	May 22 2012, 03:43 AM
Asin	Integer	Policy	Result Variables	May 22 2012, 03:41 AM
Asinh	Integer	Policy	Result Variables	May 22 2012, 03:41 AM
Atan	Integer	Policy	Result Variables	May 22 2012, 03:42 AM
Atanh	Integer	Policy	Result Variables	May 22 2012, 03:42 AM
Base Premium	Integer	Policy	Result Variables	Mar 13 2012, 08:39 PM
BI_Premium	Decimal	Vehicle	Result Variables	Apr 03 2012, 05:16 PM
BIPremium	Decimal	Vehicle	Result Variables	Mar 14 2012, 05:53 PM

Term

Data Type: Integer

Category: Policy

Date Last Modified: Tue May 15 23:21:00 CDT 2012

Figure 83 Result Variables Tab

RESULT VARIABLES FUNCTIONALITY

Functions can be accessed from the menu bar located at the top of the screen and from the right click menu of any unlocked result variable. The functions you can perform depends 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
σ	**Result Variable.** Result variables are subline specific and can be created at the global level, from a program version, or from another program element. All result variables are displayed on the Program Details→Result Variables tab. The same result variable listing is also available for Subline→Globals tab. The same options, new, edit, copy and delete are available on either tab.
🔒	**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 will not be able to 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 FIELDS

The fields are the same for the Global Tab and Program Details. For more on field descriptions, please see Program Details→Fields.

Right Click Menu Options:

To view right click menu options, highlight the result variable you want to view and right click. Unlocked result variables will 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:** Allows you to enter, edit and view notes for the result variable.

Template Locked result variables has four options, Copy, View Where Used, View History and Notes.

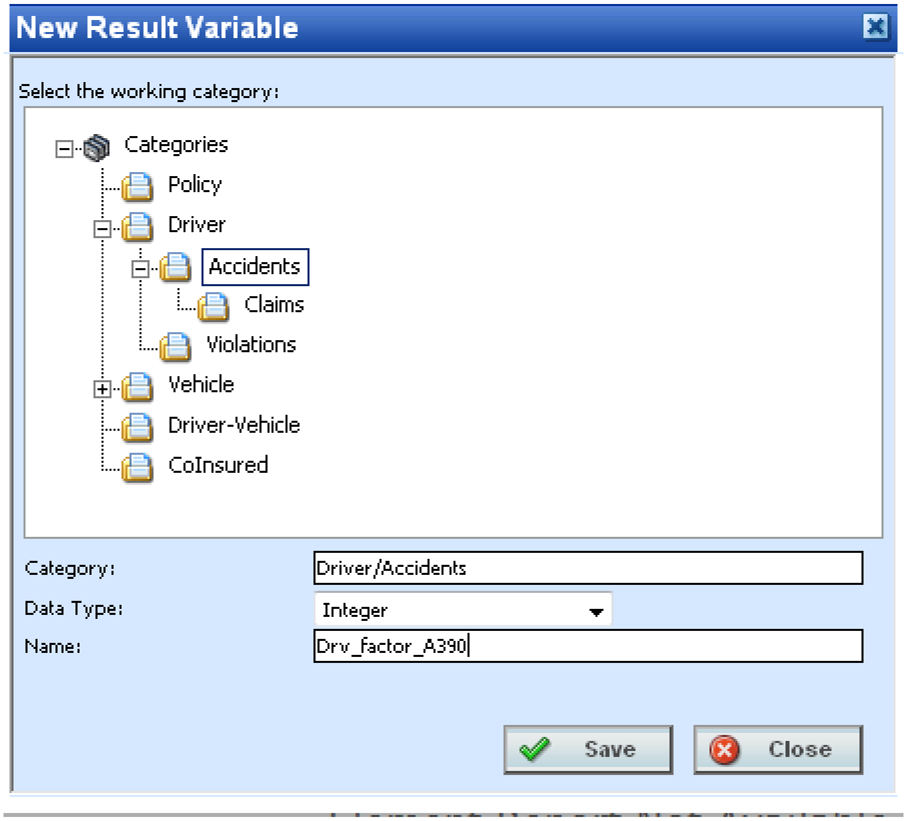
Program Locked result variables has three options, View Where Used, View History and Notes.

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 need 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.

1. On the Program Details tab, select Result Variables from the Program Detail Folders. Click **New** in the top bar menu. This will open the **New Result Variable** popup.



Category: Driver/Accidents

Data Type: Integer

Name: Drv_factor_A390

Figure 84 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.

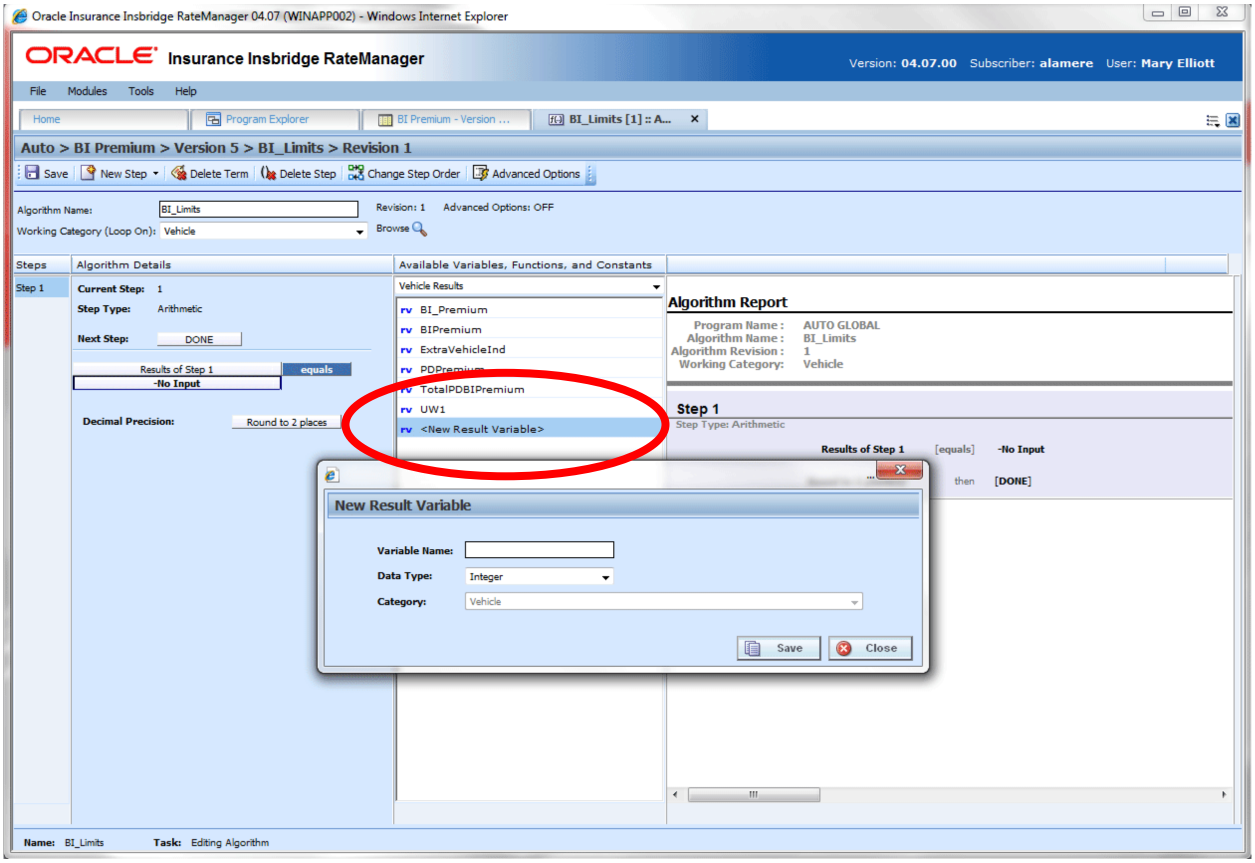
4. Enter a **Name** for the new result variable.

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 will be 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 Details→Globals tab and then return to the Edit tab or the Program Details tab to create your result variable.



Version: 04.07.00 Subscriber: alamere User: Mary Elliott

Auto > BI Premium > Version 5 > BI_Limits > Revision 1

Algorithm Name: BI_Limits
Revision: 1
Advanced Options: OFF
Working Category (Loop On): Vehicle

Current Step: 1
Step Type: Arithmetic
Next Step: DONE
Decimal Precision: Round to 2 places

Program Name : AUTO GLOBAL
Algorithm Name : BI_Limits
Algorithm Revision : 1
Working Category: Vehicle

Variable Name:
Data Type: Integer
Category: Vehicle

Name: BI_Limits
Task: Editing Algorithm

Figure 85 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 popup will be displayed.

2. Enter in a **Variable Name**.

3. Select the **Data Type**.

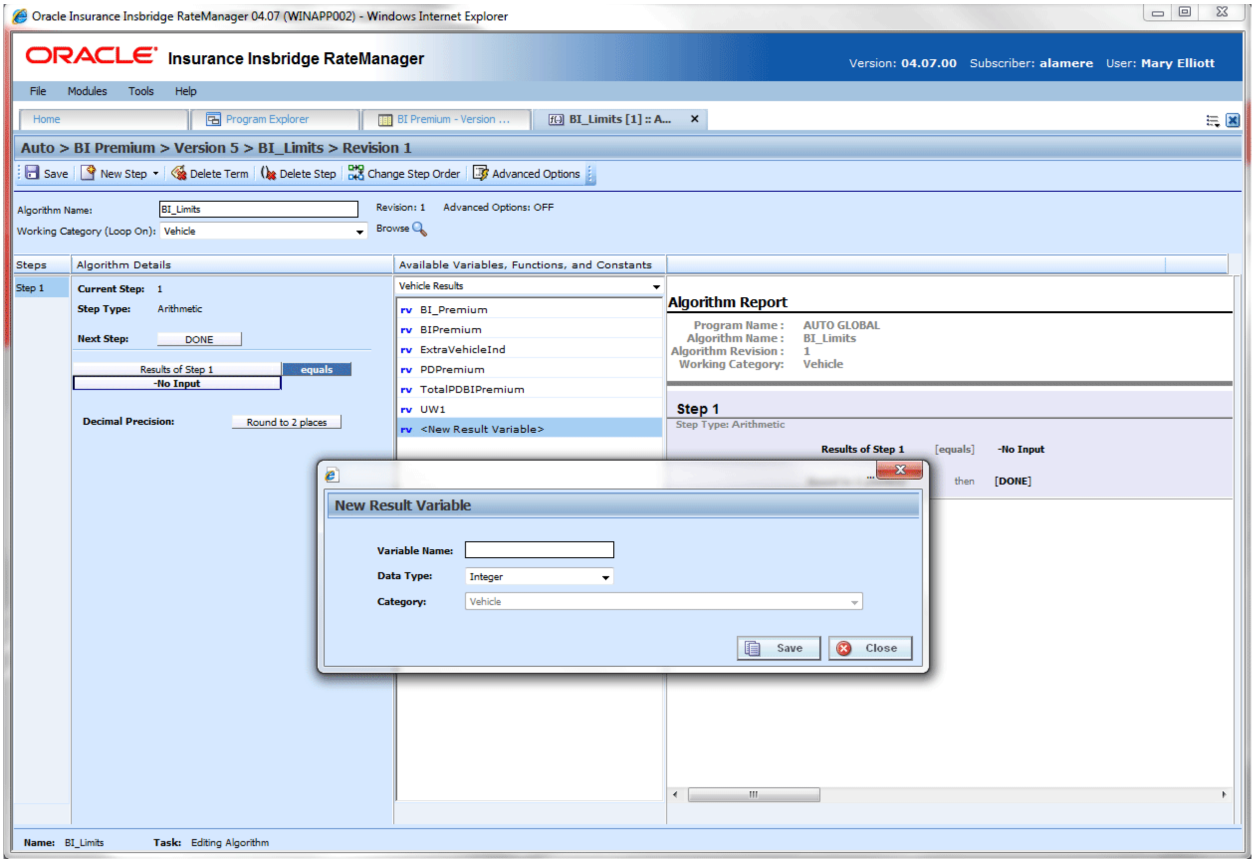
4. Click **Save** to save your entry. The new result is now available for use.

Editing a Result Variable

Result variables can be edited from the Result Variable Tab or from a program variable listing screen and are edited like their local counterparts. 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 will be locked at all times and cannot be unlocked. A new 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** popup.



Algorithm Name: BI_Limits

Revision: 1 Advanced Options: OFF

Working Category (Loop On): Vehicle

Current Step: 1

Step Type: Arithmetic

Next Step: DONE

Decimal Precision: Round to 2 places

Program Name : AUTO GLOBAL
Algorithm Name : BI_Limits
Algorithm Revision : 1
Working Category: Vehicle

Variable Name:

Data Type: Integer

Category: Vehicle

Name: BI_Limits Task: Editing Algorithm

Figure 86 Editing Result Variable

2. To change the **Category**, double click the category where you want the variable to fall under. The pathway will be displayed in the Category field. On the tree view, the final destination where the variable will be placed will have a box around it.

3. Make any changes to the **Data Type** and **Name**.

4. Click **Save** to save your changes.

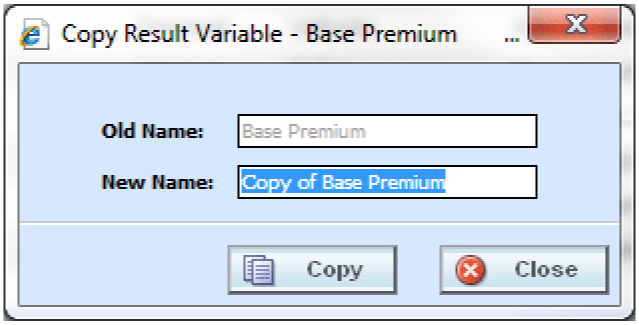
Copy a Result Variable

Any unlocked result variable can be copied at any time. The result variable will be copied with the exact same data type and category. The copied variable will be open for editing. To make a change to the variable, you must go into the edit screen.

A copied result variable will require a new name. No other information will be required.

To Copy a Result Variable

1. Right click the result variable you want to copy. Select **Copy**. The Copy Result Variable popup is displayed.



Copy Result Variable - Base Premium ...

Old Name: Base Premium

New Name: Copy of Base Premium

Figure 87 Copying Result Variable

2. Enter in a **New Name**.

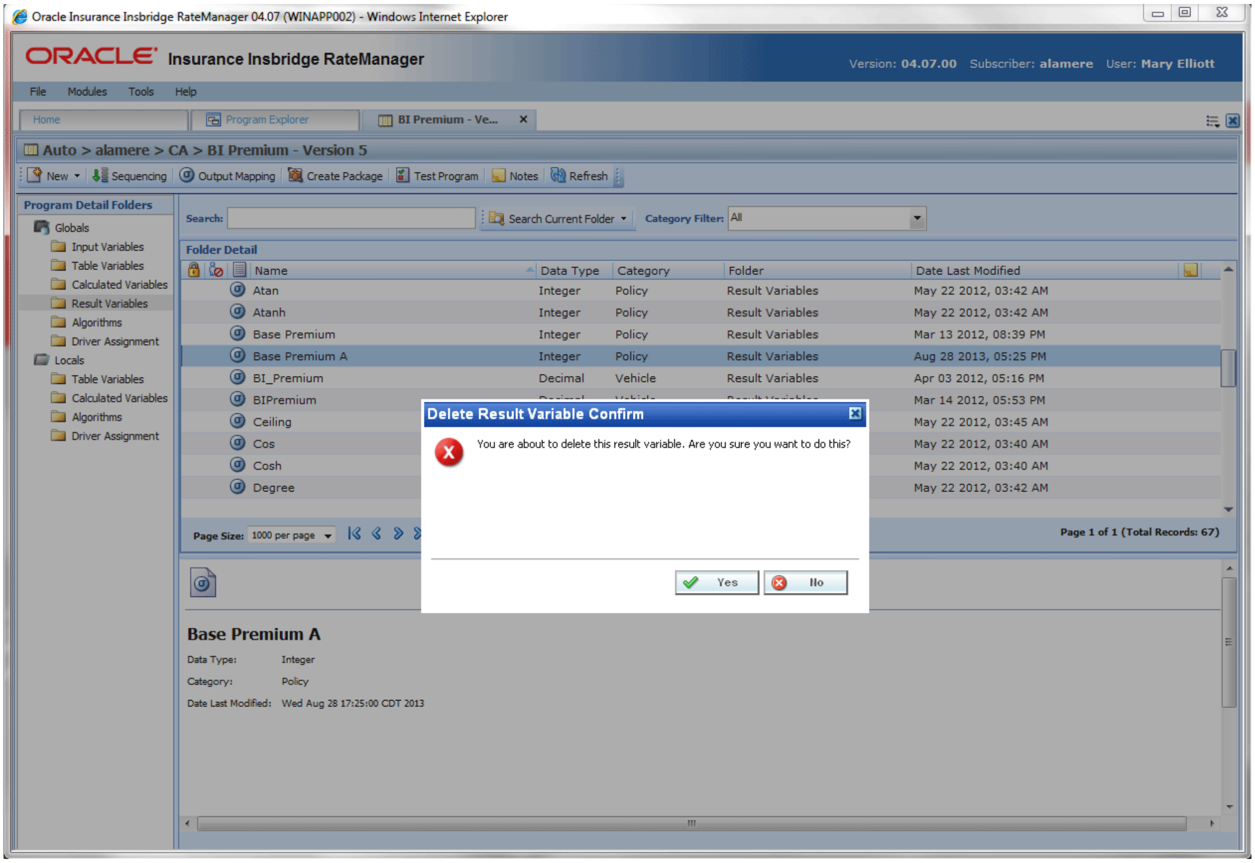
3. Click **Copy** to copy your result variable. The screen refreshes and the copied result variable is displayed on the Program Details→Result Variable tab.

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 will be 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.



Name	Data Type	Category	Folder	Date Last Modified
Atan	Integer	Policy	Result Variables	May 22 2012, 03:42 AM
Atanh	Integer	Policy	Result Variables	May 22 2012, 03:42 AM
Base Premium	Integer	Policy	Result Variables	Mar 13 2012, 08:39 PM
Base Premium A	Integer	Policy	Result Variables	Aug 28 2013, 05:25 PM
BI_Premium	Decimal	Vehicle	Result Variables	Apr 03 2012, 05:16 PM
BIPremium	Decimal	Vehicle	Result Variables	Mar 14 2012, 05:53 PM
Ceiling				May 22 2012, 03:45 AM
Cos				May 22 2012, 03:40 AM
Cosh				May 22 2012, 03:40 AM
Degree				May 22 2012, 03:42 AM

Delete Result Variable Confirm

You are about to delete this result variable. Are you sure you want to do this?

Base Premium A

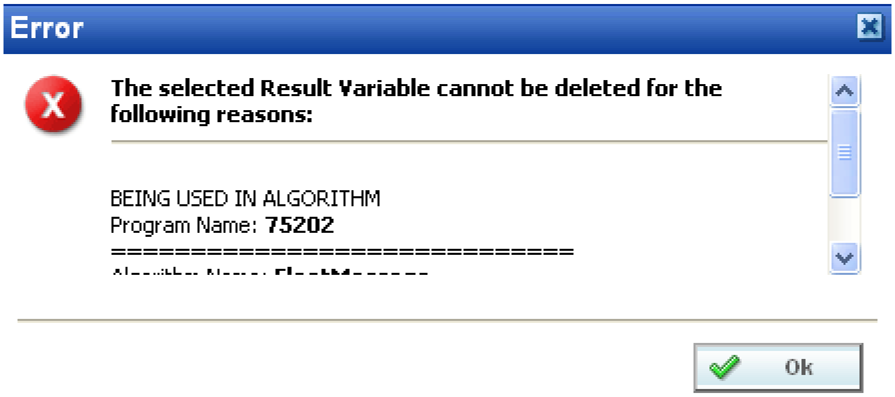
Data Type: Integer

Category: Policy

Date Last Modified: Wed Aug 28 17:25:00 CDT 2013

Figure 88 Deleting a Result Variable

2. A confirmation message will be displayed. Click **Yes** to delete or **No** to cancel. If the result variable is being used, you cannot delete it. A message will be displayed informing you where the variable is being used.



Error

The selected Result Variable cannot be deleted for the following reasons:

BEING USED IN ALGORITHM
Program Name: **75202**
==============================

Figure 89 Result Variable Error Message

Chapter 13

TABLE VARIABLES

Variables are placeholders that get filled in during rating, based on inputs 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 managed at two levels:

Global: Global variables are created and managed at the global level and are subline specific. Global variables can be accessed from the Global tab or from the Program Details tab. When selected, variables will be listed in the Folder Details area.

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. When selected, variables will be listed in the Folder Details area.

There are a couple of differences between globals and locals. One difference between them is that global variables are available to all programs within the subline while program level variables are available only to the program where they were created. Another is that Global variables can use only global elements. Local variables can use either local or global elements.

Both Global and Local calculated variables can be template generated or created by you. Variables that are user created will be 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 will be 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 will be 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 two paths to view and manage variables, either from the Globals tab or from the Program Details tab. The Global tab will display global variables only. You cannot view or manage local variables on the Global 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.

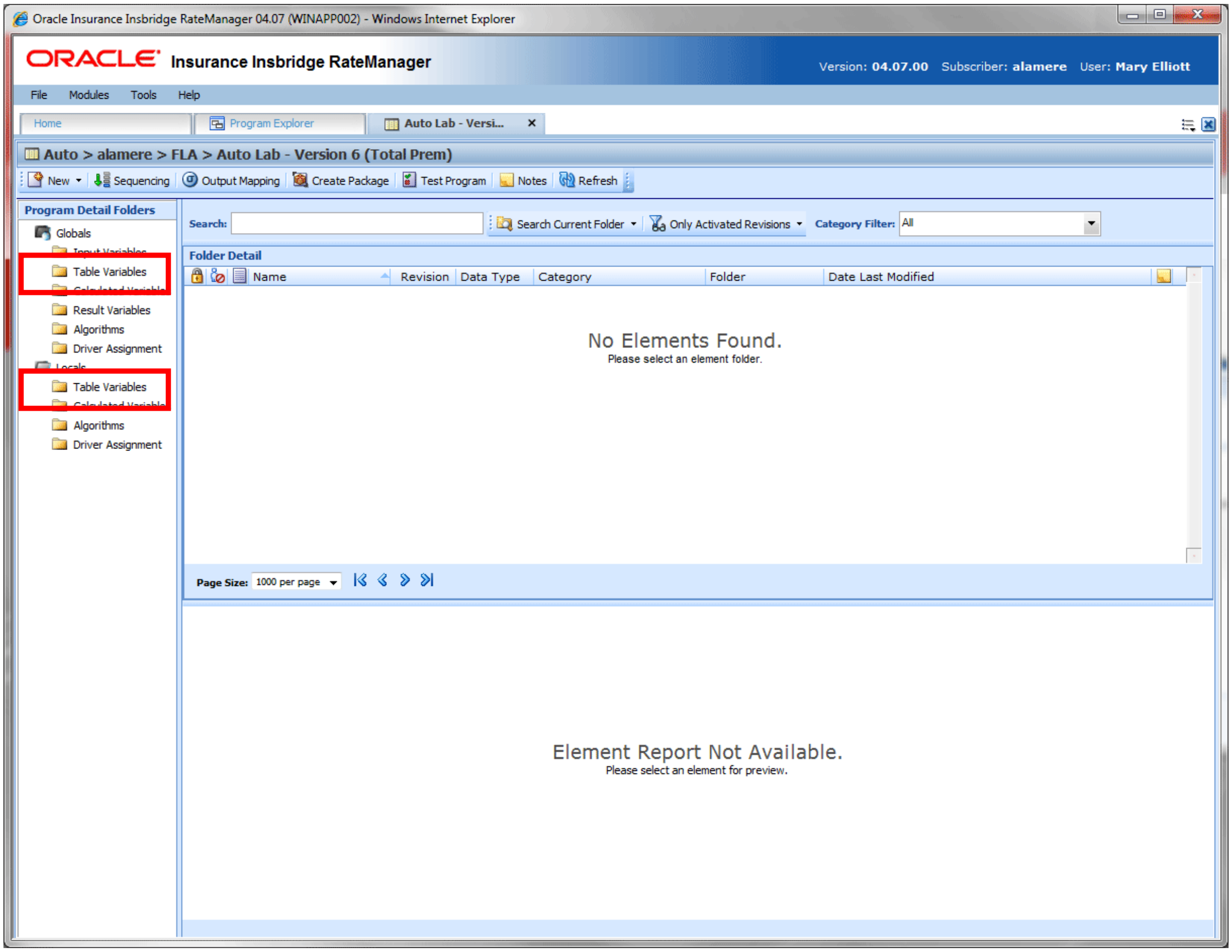
From either tab, when you select Table Variables in the Program Details Folder, the current table variables will be listed in the Folder Details area.

Table variables is not an option on the Program Explorer tab.

1. Modules→Program Explorer. This opens the Program Explorer tab.

2. Select the Line of Business→Subline→Folder→Program → Program Version. You also can access table variables from the globals tab. Line of Business→Subline→Globals.

3. Open the program version to get the Program Details tab. Variables are available from the Program Details Folders area on the left hand side of the tab.



No Elements Found.
Please select an element folder.

Element Report Not Available.
Please select an element for preview.

Figure 90 Table Variable Navigation

TABLE VARIABLES FUNCTIONALITY ON PROGRAM DETAILS TAB

On the 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 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 will be able to 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 will be placed on the Table Variable tab.

The functions you can perform depends on the status of the table variable. Locked table variables will 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.** Global Table Variables are subline 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. Local Table Variables will be displayed on the Program Details→Local Table Variables. The same variable listing will also be available for Subline→Globals tab. The same options, new, edit, copy and delete are available on either tab.
🔒	**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.

Program Details Table Variables Fields:

The fields are the same for the Global Tab and Program Details. For more on field descriptions, please see Program Details→Fields.

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.

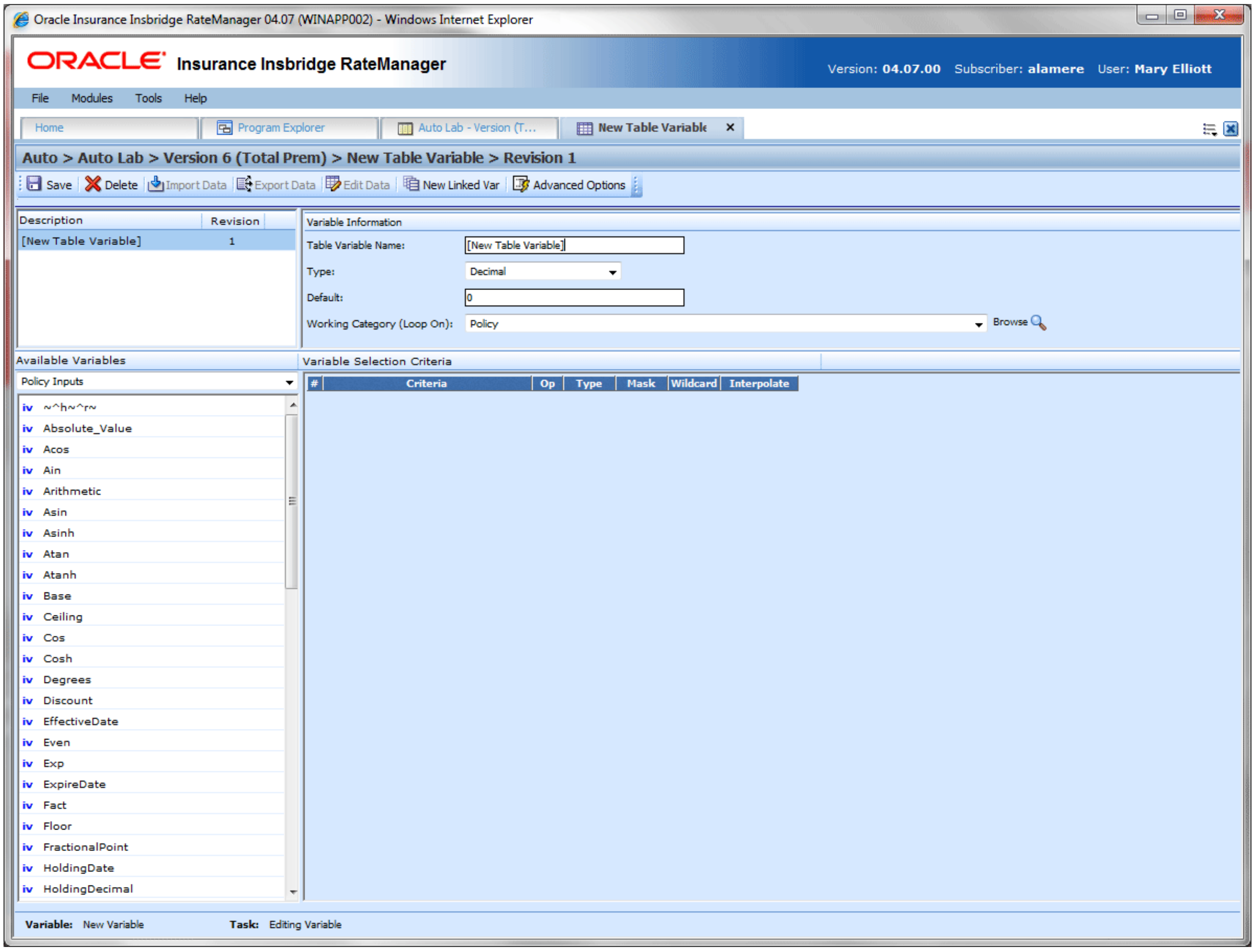


Table Variable Name: [New Table Variable]

Type: Decimal

Default: 0

Working Category (Loop On): Policy

Variable: New Variable Task: Editing Variable

Figure 91 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	**Save:** Saves the table variable.
Delete	**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	**Import Data:** Allows the user to import a tab-delimited data file. See Importing a Table for more information.
Export Data	**Export Data:** Allows the user to export the contents of the current table variable or linked variables to a tab-delimited file. See Exporting Data for more information.
Edit Data	**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	**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	**Advanced Options:** Allows you to change 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.*

TABLE VARIABLE TAB FIELDS

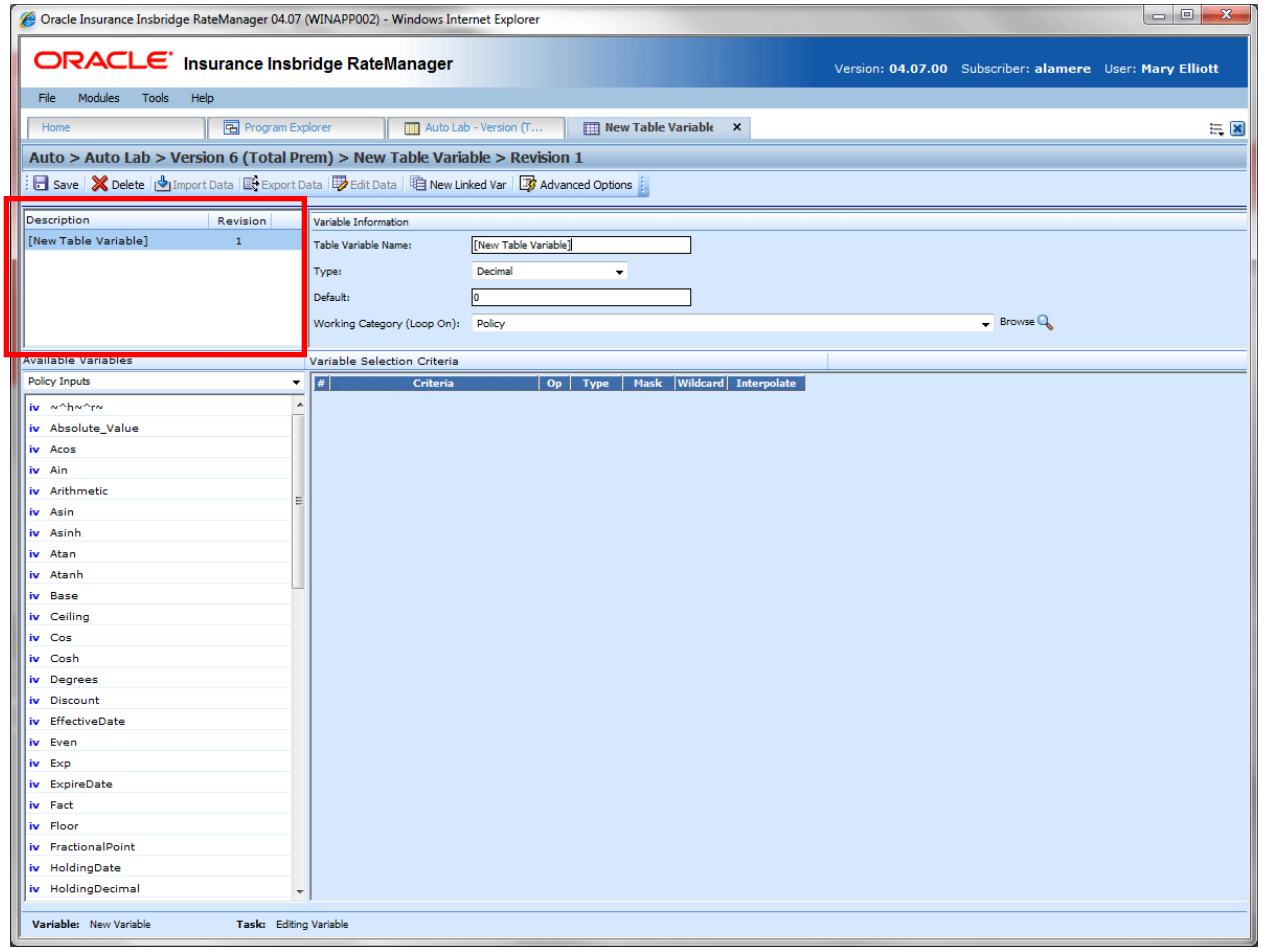
There are four areas on the Table Variable Tab.

- **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.
- **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. All linked variables use the same criteria.


Oracle Insurance Insbridge RateManager 04.07 (WINAPP002) - Windows Internet Explorer
ORACLE Insurance Insbridge RateManager
Version: 04.07.00 Subscriber: alamere User: Mary Elliott
File Modules Tools Help
Home
Program Explorer
Auto Lab - Version (T...
New Table Variable
Auto > Auto Lab > Version 6 (Total Prem) > New Table Variable > Revision 1
Save
Delete
Import Data
Export Data
Edit Data
New Linked Var
Advanced Options
Browse
Available Variables
Variable Selection Criteria
#
Criteria
Op
Type
Mask
Wildcard
Interpolate

Description	Revision
[New Table Variable]	1

Variable Information

Table Variable Name: [New Table Variable]

Type: Decimal

Default: 0

Working Category (Loop On): Policy

Policy Inputs

iv ~^h~^r~
iv Absolute_Value
iv Acos
iv Ain
iv Arithmetic
iv Asin
iv Asinh
iv Atan
iv Atanh
iv Base
iv Ceiling
iv Cos
iv Cosh
iv Degrees
iv Discount
iv EffectiveDate
iv Even
iv Exp
iv ExpireDate
iv Fact
iv Floor
iv FractionalPoint
iv HoldingDate
iv HoldingDecimal

Variable: New Variable   Task: Editing Variable

Figure 92 Table Linked Variable Field

Fields

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.

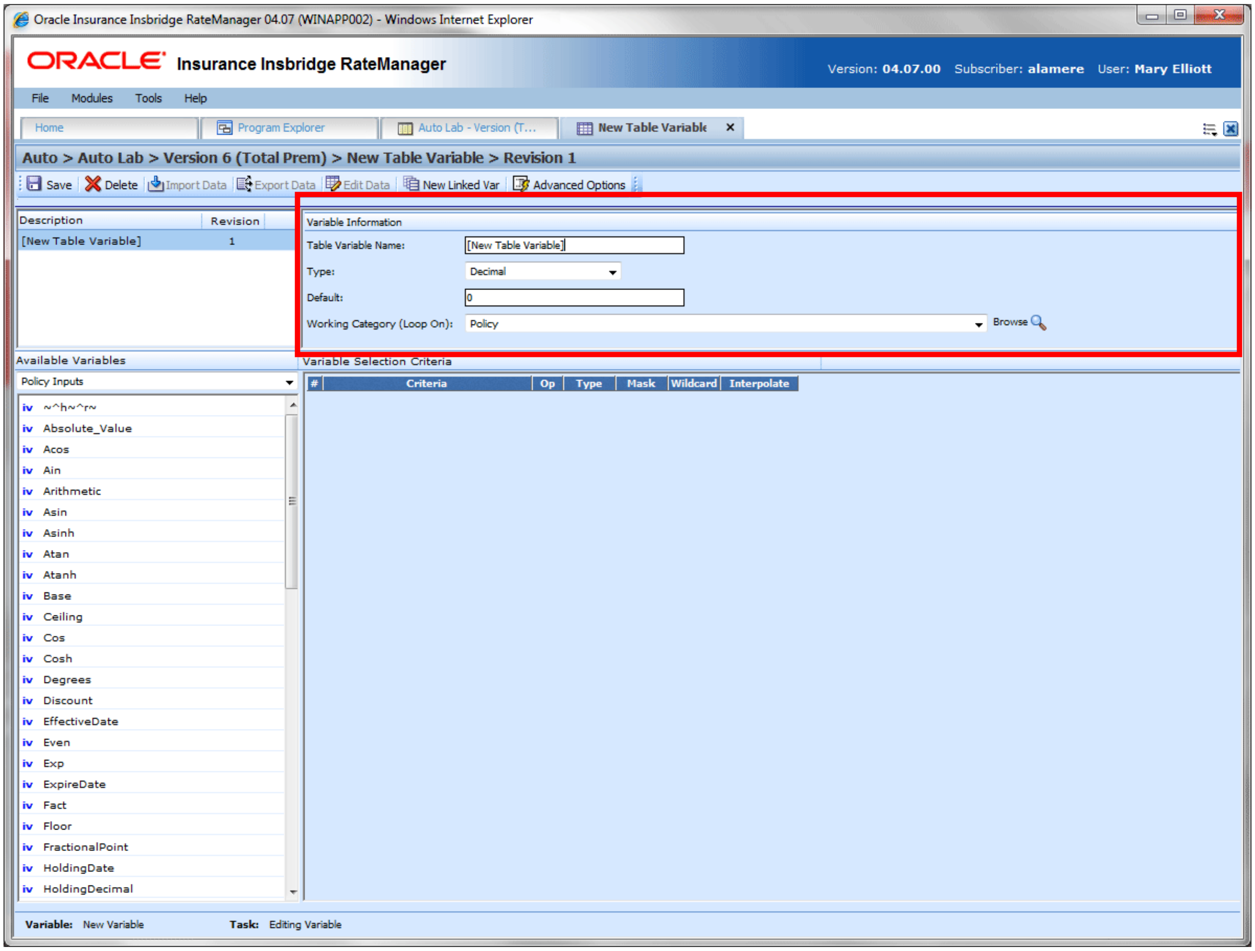


Table Variable Name: [New Table Variable]

Type: Decimal

Default: 0

Working Category (Loop On): Policy

Variable: New Variable

Task: Editing Variable

Figure 93 Table Variable Information Field

Fields:

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.

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 will be 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.

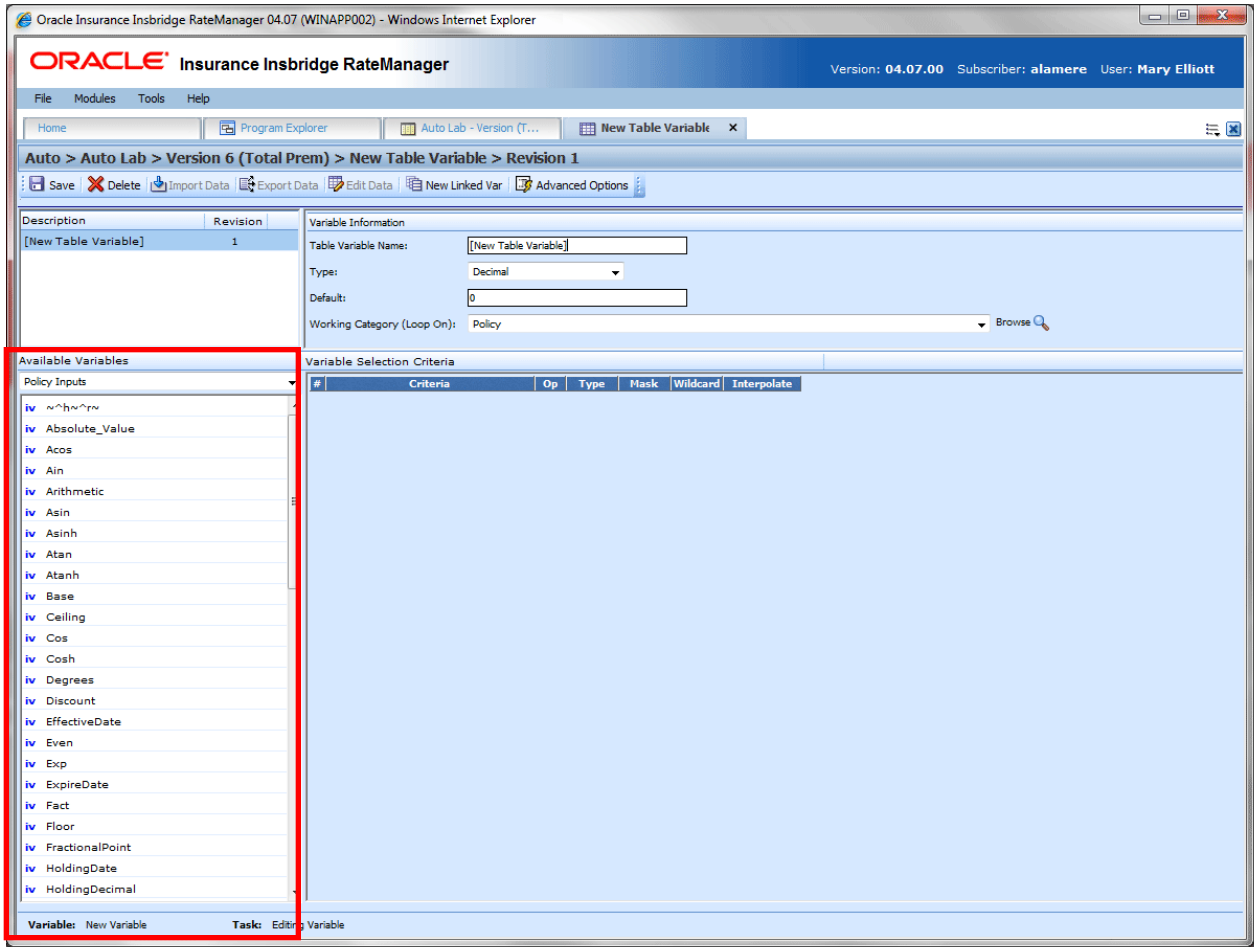
You can select the working category from the drop down menu or you can browse for a working category by clicking **BROWSE**. This brings up the Browse Category popup. From here you can navigate to the Category you want. Once you have selected the category, click OK. The category will auto fill in the 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, Policy Inputs, Policy Variables and Policy Results. If you select Vehicle, the options will now include Vehicle Inputs, Vehicle Variable and Vehicle Results along with Constants, Policy Inputs, Policy Variables and Policy Results.

Table Variables Available Variables

The Available Variables area contains a listing of available inputs, variables, and results, sorted by category and system constants that can be used as criteria. For global variables, only other global variables or inputs 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 Inputs, you will be able to add a new input variable without leaving the Edit Table Variable tab. Scroll to the bottom of the available variables list and select <New Input Variable>. A popup is displayed that allows you to enter a new input variable.



Auto > Auto Lab > Version 6 (Total Prem) > New Table Variable > Revision 1

Description	Revision
[New Table Variable]	1

Table Variable Name: [New Table Variable]

Type: Decimal

Default: 0

Working Category (Loop On): Policy

Available Variables: Policy Inputs

Variable: New Variable   Task: Editing Variable

Figure 94 Table Available Variable Field

Fields:

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 Variable Information area.

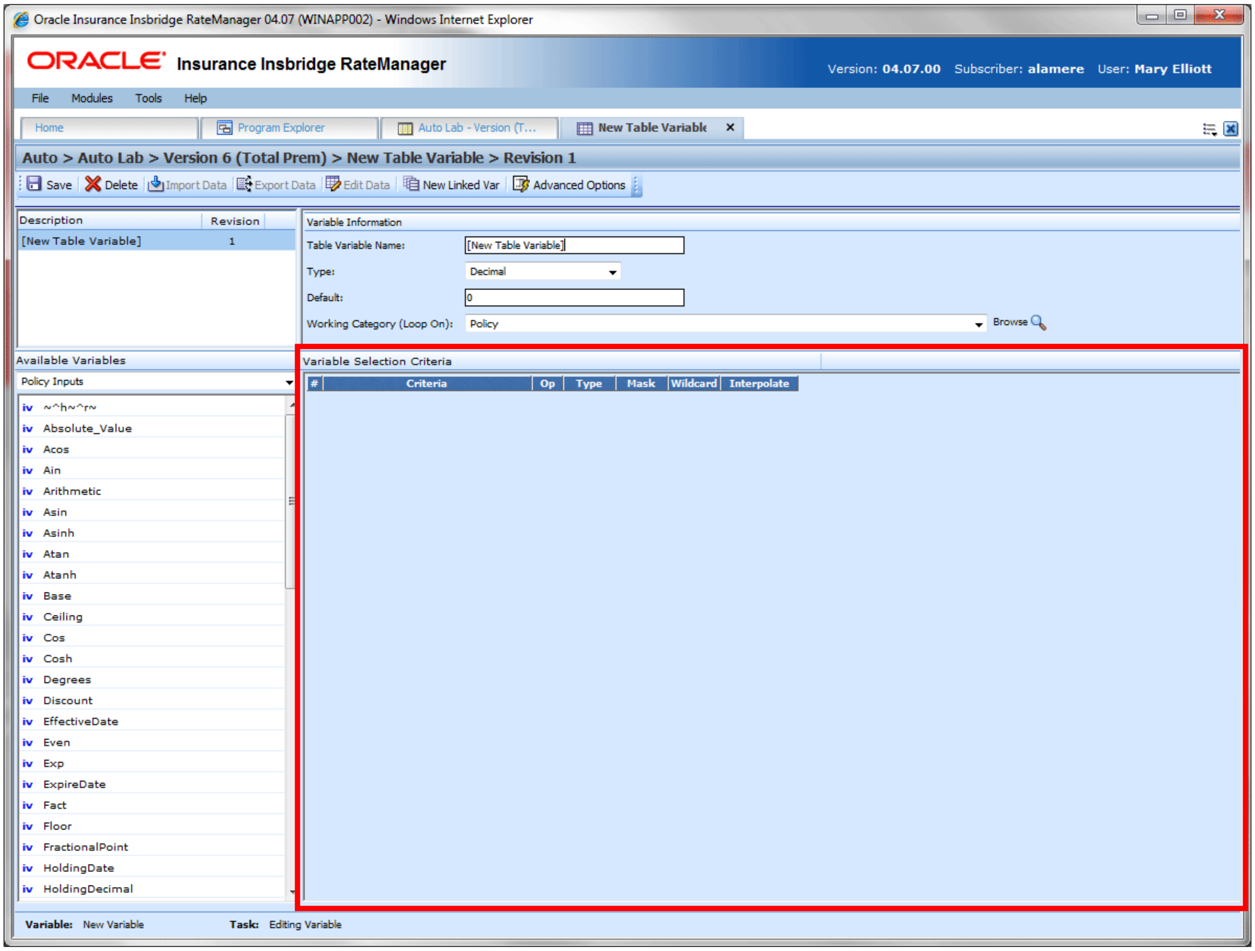
Selection: The available variables will be listed based on the option you select from the menu.

Table Variables Variable Selection Criteria

The criteria for the variable needs to be defined in this section. The Variable Selection Criteria determines whether or not this variable is used. The criteria available will be listed in the Available Variables area. You can select inputs, variables or results from a selected category or a system constant.

The order number, criteria, operator, and type will be 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 ten criteria can be added to a table variable.*



Auto > Auto Lab > Version 6 (Total Prem) > New Table Variable > Revision 1

Description	Revision
[New Table Variable]	1

Table Variable Name: [New Table Variable]

Type: Decimal

Default: 0

Working Category (Loop On): Policy

Policy Inputs

- iv ~^h~^r~
- iv Absolute_Value
- iv Acos
- iv Ain
- iv Arithmetic
- iv Asin
- iv Asinh
- iv Atan
- iv Atanh
- iv Base
- iv Ceiling
- iv Cos
- iv Cosh
- iv Degrees
- iv Discount
- iv EffectiveDate
- iv Even
- iv Exp
- iv ExpireDate
- iv Fact
- iv Floor
- iv FractionalPoint
- iv HoldingDate
- iv HoldingDecimal

#	Criteria	Op	Type	Mask	Wildcard	Interpolate

Figure 95 Table Variable Selection Criteria Field

Fields:

#: 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 input 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.

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** >=

VARIABLE SELECTION CRITERIA

The variable selection criteria can have one or more options applied to it.

Table Variables Masking

Masking allows you to determine 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 will bring up the Mask Value popup.
2. Enter in the Mask Value.

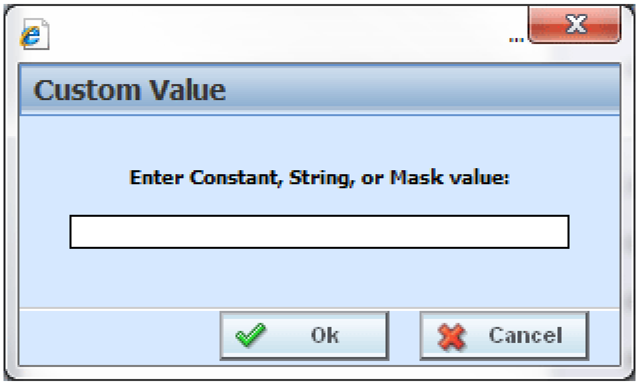


Figure 96 Mask Value Dialogue Box

3. Click **OK** to save the entry. You will be returned to the Edit Variable tab. The mask icon will now reflect 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**

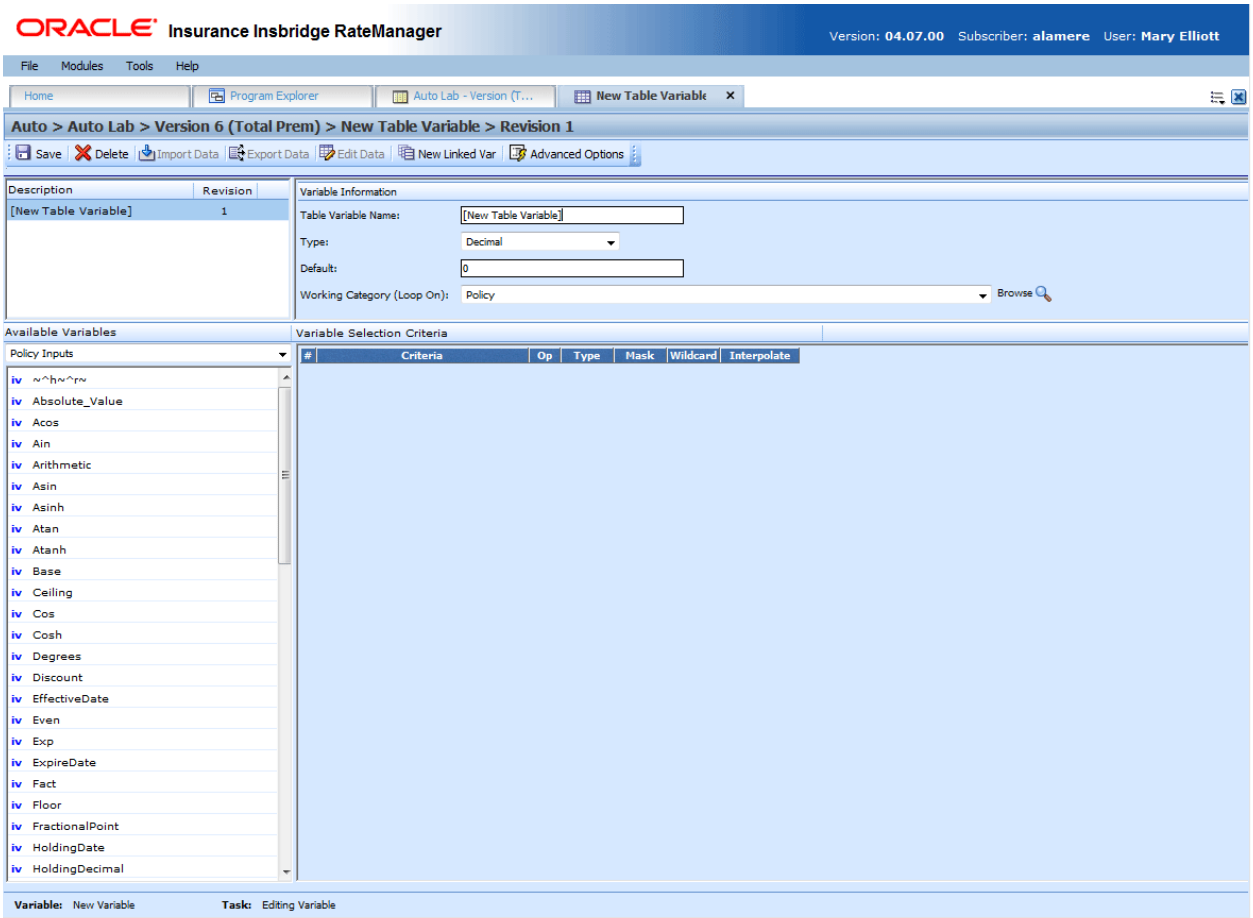
Usage:

- Interpolations should be applied to non-linked variables only. Linked variables may not function as expected.
- Interpolate should not be used in combination with wildcard. Conflicts may arise when combining a wildcard with an interpolated value.
- Only one variable selection criteria element should have interpolation applied.
- Interpolation will only work with integers and decimals. The criteria has 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. On the Program Details tab, Select **New** from the menu bar and then select **Table Variable** for the type of variable you would like to create, Global or Local. This opens the Table Variable tab.



Auto > Auto Lab > Version 6 (Total Prem) > New Table Variable > Revision 1

Description	Revision
[New Table Variable]	1

Variable Information

Table Variable Name: [New Table Variable]

Type: Decimal

Default: 0

Working Category (Loop On): Policy

#	Criteria	Op	Type	Mask	Wildcard	Interpolate

Variable: New Variable Task: Editing Variable

Figure 97 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 will have a unique name.

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 10 criteria.

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. Click **Save** to save your work.

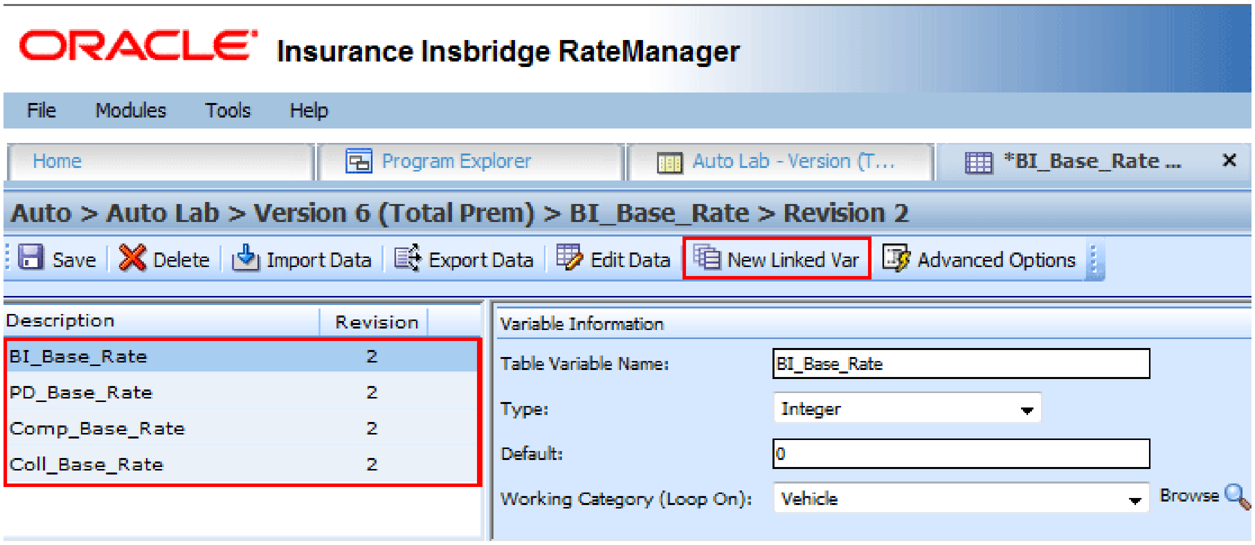
LINKED VARIABLES

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.



Description	Revision
BI_Base_Rate	2
PD_Base_Rate	2
Comp_Base_Rate	2
Coll_Base_Rate	2

Variable Information

Table Variable Name: BI_Base_Rate

Type: Integer

Default: 0

Working Category (Loop On): Vehicle

Figure 98 Creating a Linked Table Variable

2. Click **New Linked Var** from the top bar menu. The Variable Information area at the top right hand portion of the tab will refresh. Another variable will be added in the Linked Variable Listing at the top left hand 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 allows you to change information about the variable such as the name, data type, default value, working category and variables used as selection criteria. It also allows you to 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.

To Edit a Variable

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.

COPYING A TABLE VARIABLE

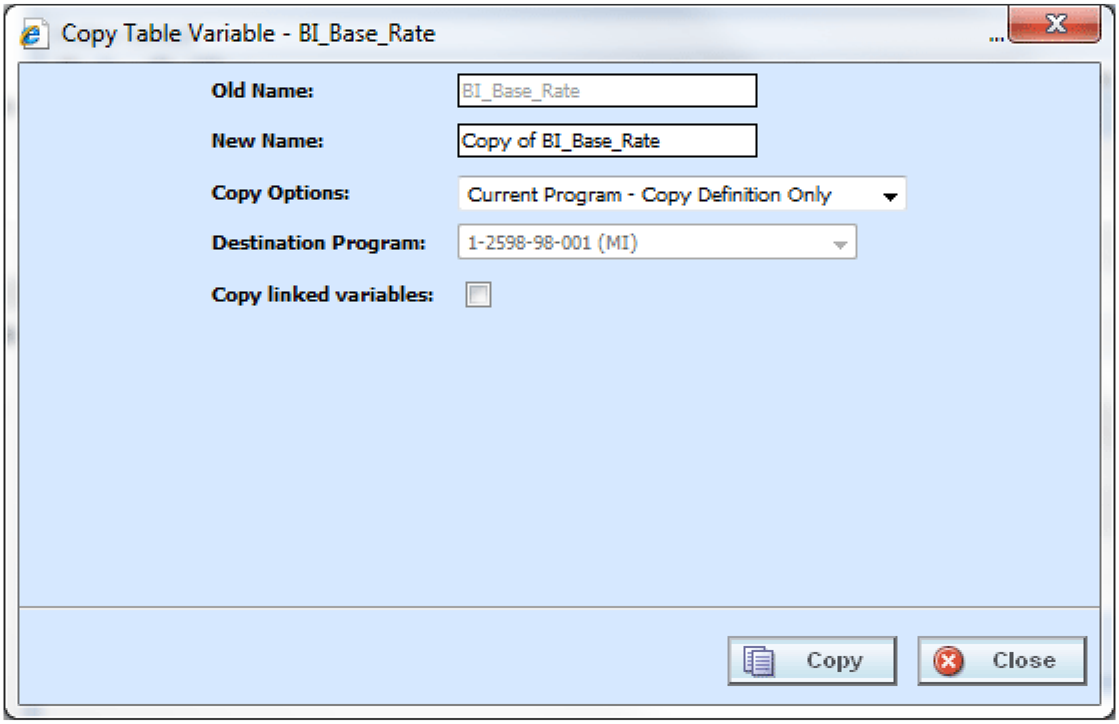
Table Variables may be copied within any program. In your default subline you can copy a local table variable to another program within the subline. Template sublines allow you to copy within the same program only. When copying a table variable, there are four possible copy options, depending upon your selection:

- **Current Program: Copy Definition Only:** Only the variable definition, the structure of the table variable (including the data type, default value, working category and criteria definitions) will be copied, but not the data, within the current program.
- **Current Program: Copy Definition & All Data:** The table variable and all associated table data will be copied to the new variable within the current program.
- **Other Program: Copy Definition Only:** Only the variable definition, the structure of the table variable (including the data type, default value, working category and criteria definitions) will be copied, but not the data, into another program within the same subline that you chose. This option is only for local table variables within your default subline.
- **Other Program: Copy Definition & All Data:** The table variable and all associated table data will be copied to the new variable into another program within the same subline that you choose. This option is only for local table variables within your default subline.

All linked variables associated with the variable may be copied by checking the **Copy linked variables** check box on the copy table variable popup. All linked variables will be copied with the name COPY OF… You will not be allowed to change the copied variable name on the copy table variable popup. You can rename the variables using the edit option.

If you do not choose to copy all linked variables, the system will only copy the linked variable that is being used in the variable you are copying. All other associated variables will not be copied.

1. Select the variable you want to copy and right click it. Select **Copy Variable** and a separate popup window is displayed.



Copy Table Variable - BI_Base_Rate

Old Name: BI_Base_Rate

New Name: Copy of BI_Base_Rate

Copy Options: Current Program - Copy Definition Only

Destination Program: 1-2598-98-001 (MI)

Copy linked variables: □

Figure 99 Copy Table Variable Popup

2. Select the **Copy Options**. It is recommended that you change the copy options first. Changing options will reset the name field.

3. If you are copying a local table variable within your default subline, you can select to copy the variable to another program within the same subline. Select the **Destination Program** from the drop down menu.

   If you are in a template generated subline, this option will be grayed out.

   If you are in a global table variable, this option will not be displayed.

4. 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.

5. Enter in a **New Name**. The name can be entered first. It is recommended that you change the copy options first. Changing options will reset the name field.

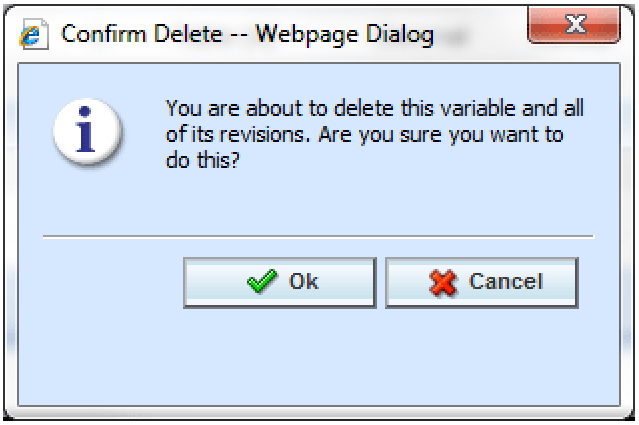
6. Click **Copy** to copy the variable.

7. If the copy was successful, you will be returned to the previous tab. Your variables will have been copied. The last updated date will be 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 will not slow down rating.

1. Select the variable you want to delete. Right click it and select **Delete**.

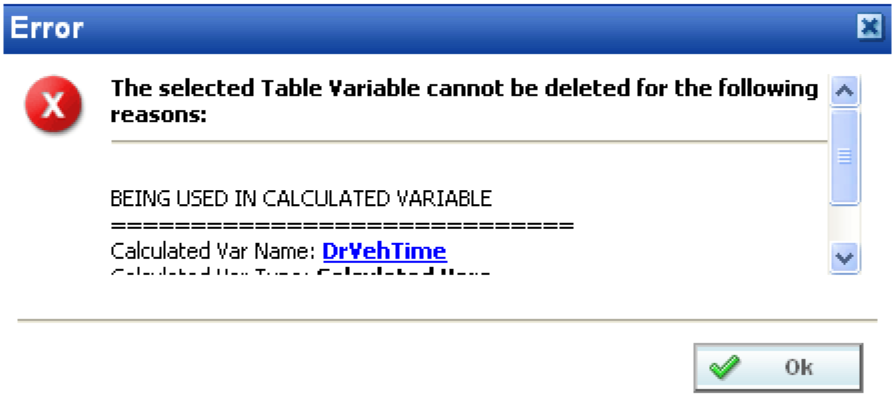


You are about to delete this variable and all of its revisions. Are you sure you want to do this?

Figure 100 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.



The selected Table Variable cannot be deleted for the following reasons:

BEING USED IN CALCULATED VARIABLE
==============================
Calculated Var Name: **DrVehTime**

Figure 101 Error when Deleting a Variable

If you delete a linked variable, only the variable you have selected will be deleted. The other variables that are linked to it will not be removed.

WORKING WITH TABLES

Data can be imported or entered in directly into an unlocked table variable. Data also can be exported out of a table variable at any time. Data is entered and edited on the Table Variable tab. Data can be exported from either the Table Variable tab or by right clicking a table variable on the Program Details tab and selecting 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

In order to import data into RateManager, you need an import file. Import files are tab-delimited text documents, i.e. columns 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.

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:

- The file must be saved in tab-delimited format using a .txt extension.
- 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.

NOTE: *The columns of data do not have to be in any particular order. During the import process, you will be able to match the columns in the input file with the variables and criteria in RateManager.*

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 will be 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	'
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 will display as blank.
- Quotation marks may be placed around commas. This depends upon the source where your data was created. You can preview your data before importing to visually inspect it. RateManager will not remove quotation marks prior to importing.
- Quotation marks (“) and ticks (`) may not always import properly. It is recommended that you avoid using these characters if possible or only use these characters inside a string (an`s).

Data can be imported into any unlocked table variable at any time.

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 inputs 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 input (TerCode). Or you could create a single table variable (RateA) the uses selection criteria input (TerCode) and another separate table variable (RateB) that also uses criteria input

(TerCode). In the case where you create two separate table variables, you would have to import your .txt file twice; once to input the data for RateA and a second time to input 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 inputs (TerCodeA and TerCodeB).

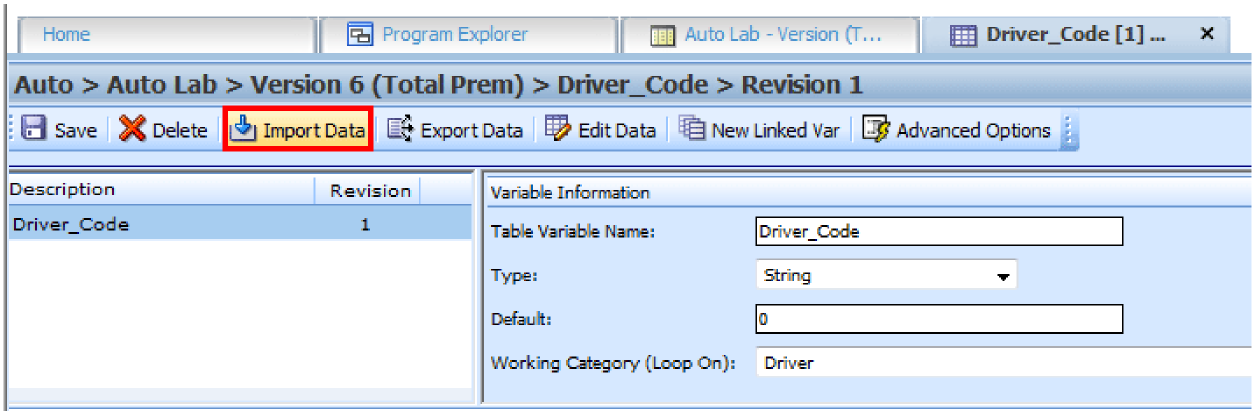
Each column in your table that you want to bring in must be assigned to table variable, linked variable or variable selection criteria. If you do not have a table variable, linked variable or a variable selection criteria, the data will not be brought into the table.

Importing a File in to a Table Variable

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.

2. Click **Import Data** on the navigation bar to open the Import Data File screen.



Auto > Auto Lab > Version 6 (Total Prem) > Driver_Code > Revision 1

Description	Revision
Driver_Code	1

Table Variable Name: Driver_Code

Type: String

Default: 0

Working Category (Loop On): Driver

Figure 102 Selecting Data File

3. If you know the name and location of the file, you can type it into the text box. Otherwise click **BROWSE…** to locate your file. Once the data file has been selected, click **Import**.

   The file will be uploaded.

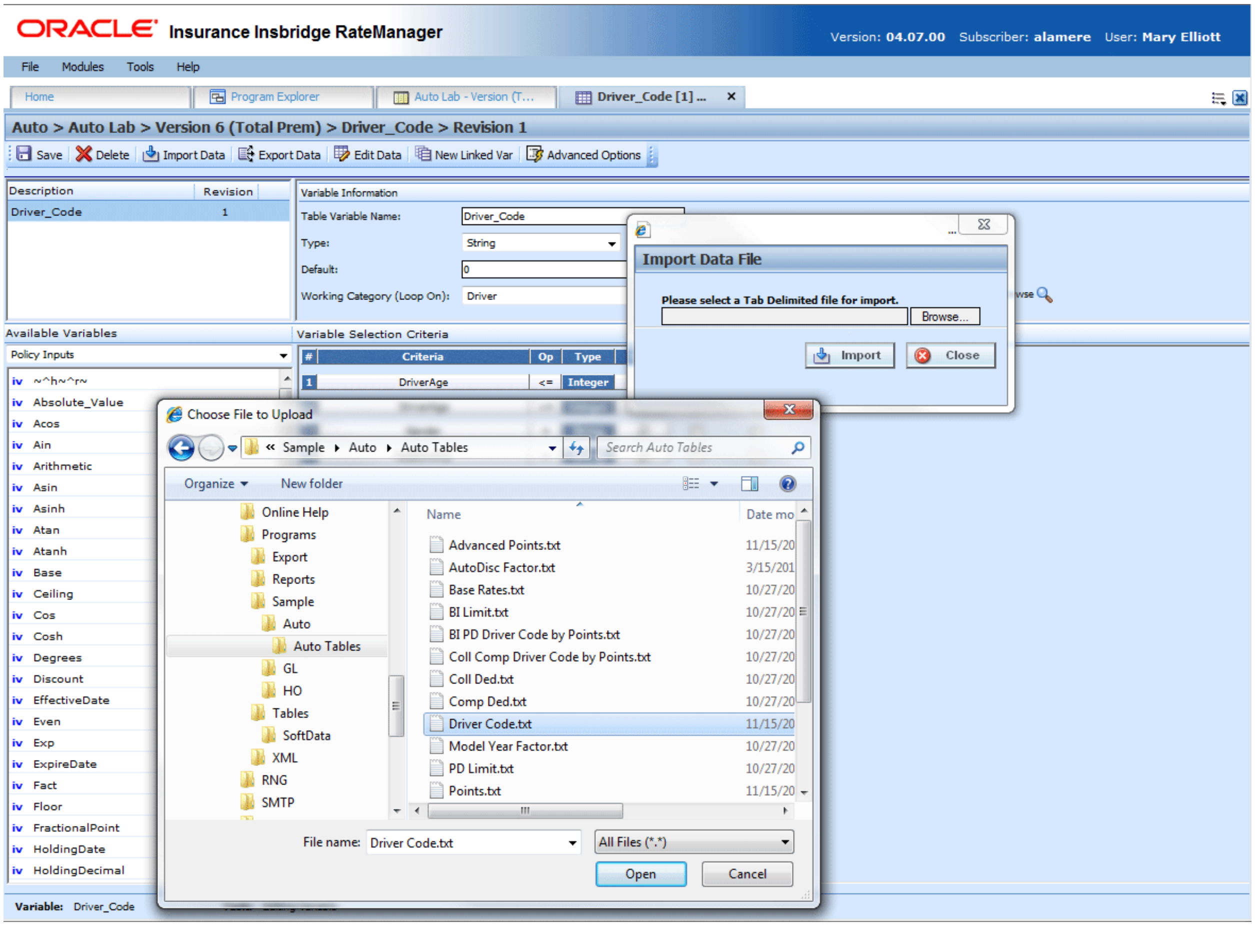
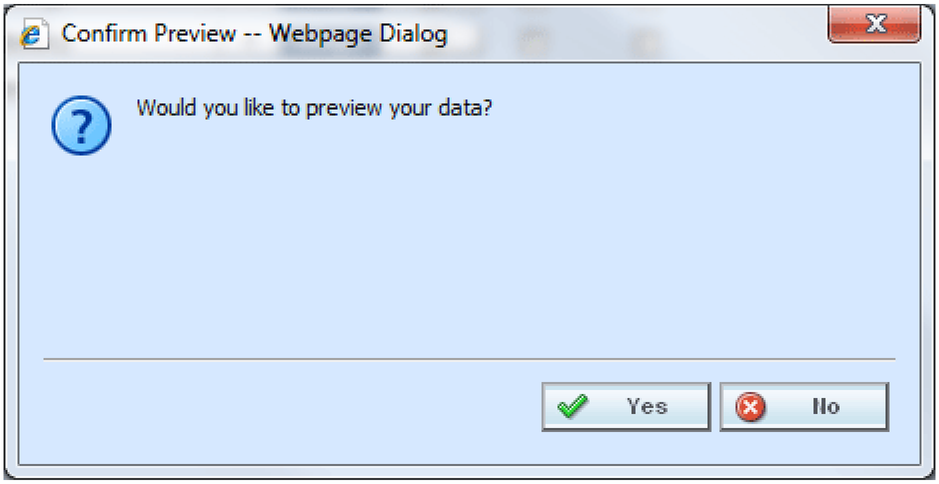


Figure 103 Importing a Table Variable

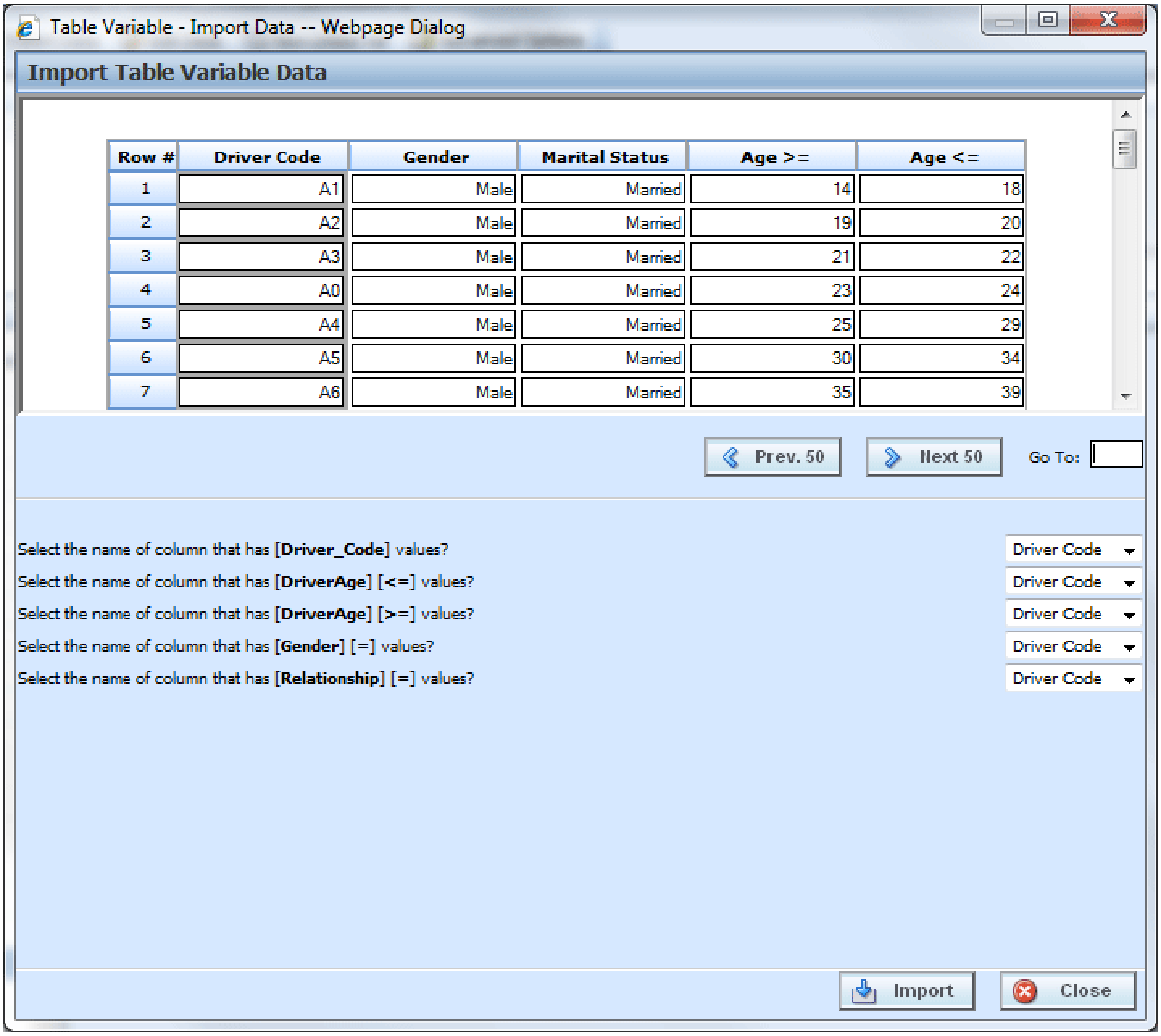
4. You will be asked if you would like to preview your data.



Would you like to preview your data?

Figure 104 Confirming Preview of Data

5. Clicking **Yes** will allow you to see your import file. This view allows you to match the column headings to your variable(s) and variable criteria.



Import Table Variable Data

Row #	Driver Code	Gender	Marital Status	Age >=	Age <=
1	A1	Male	Married	14	18
2	A2	Male	Married	19	20
3	A3	Male	Married	21	22
4	A0	Male	Married	23	24
5	A4	Male	Married	25	29
6	A5	Male	Married	30	34
7	A6	Male	Married	35	39

Select the name of column that has **[Driver_Code]** values? Driver Code

Select the name of column that has **[DriverAge]** [<=] values? Driver Code

Select the name of column that has **[DriverAge]** [>=] values? Driver Code

Select the name of column that has **[Gender]** [=] values? Driver Code

Select the name of column that has **[Relationship]** [=] values? Driver Code

Figure 105 Previewing Data File

The import popup screen 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 with a drop down box where you can match up each element with the correct column in the import file.

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.*

6. Clicking **No** allows you to match the column headings to your variable(s) and variable criteria without viewing the import file. This saves time when trying to import a large table.

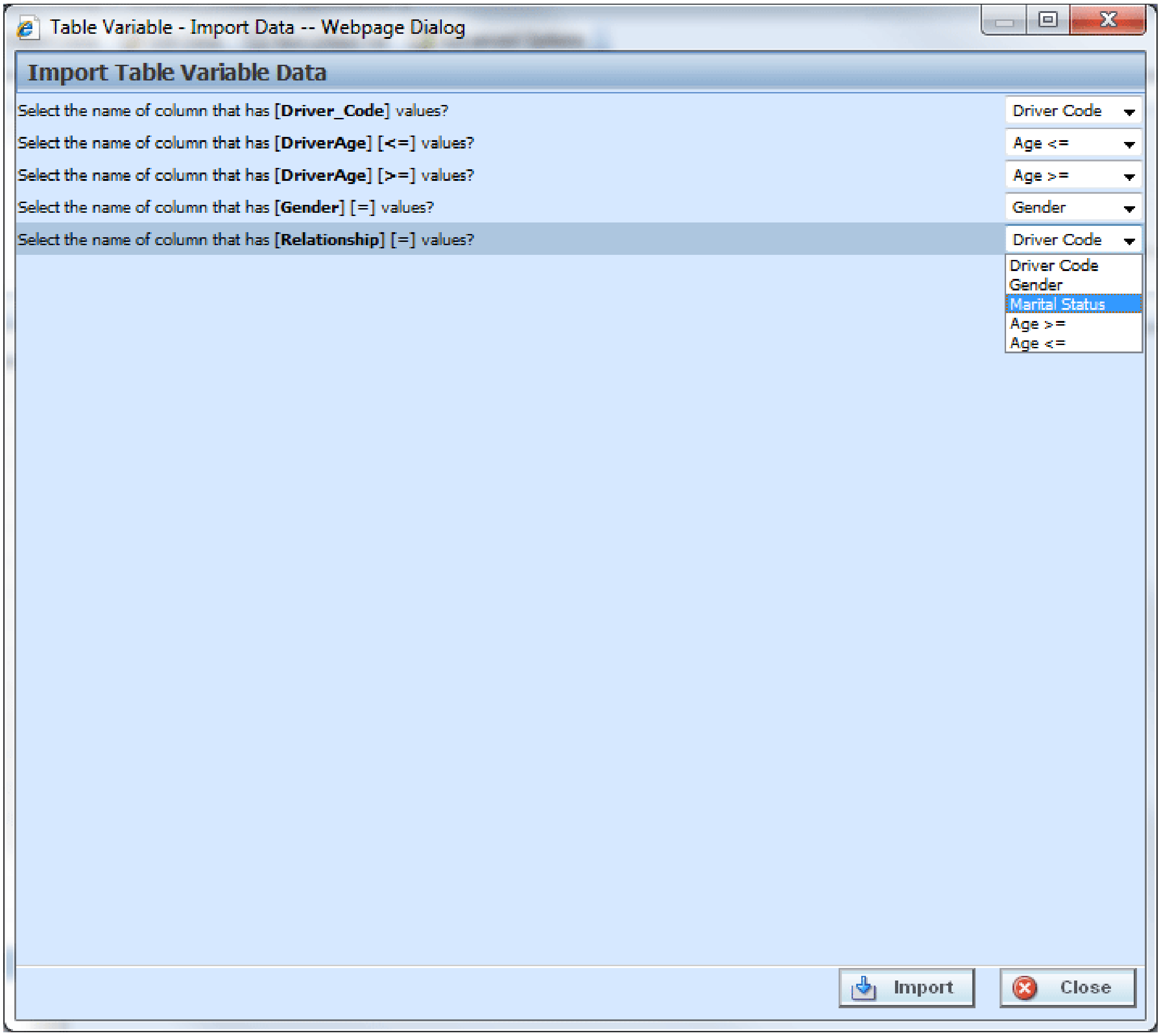


Table Variable - Import Data -- Webpage Dialog

Import Table Variable Data

Select the name of column that has **[Driver_Code]** values?	Driver Code
Select the name of column that has **[DriverAge]** [<=] values?	Age <=
Select the name of column that has **[DriverAge]** [>=] values?	Age >=
Select the name of column that has **[Gender]** [=] values?	Gender
Select the name of column that has **[Relationship]** [=] values?	Driver Code

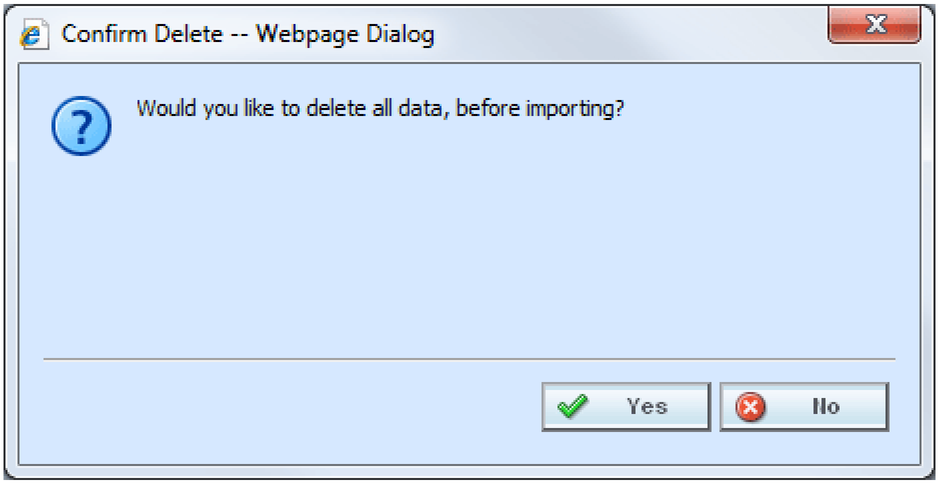
Figure 106 Not Previewing Data File

7. Select the drop down next to a variable or criteria to 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.

8. If the table variable already contains data, you will be asked if you would like to delete the data first.



Confirm Delete -- Webpage Dialog

Would you like to delete all data, before importing?

Figure 107 Confirming Delete All Data

9. 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.

10. When finished, a confirmation box will display, telling you how many rows were imported.

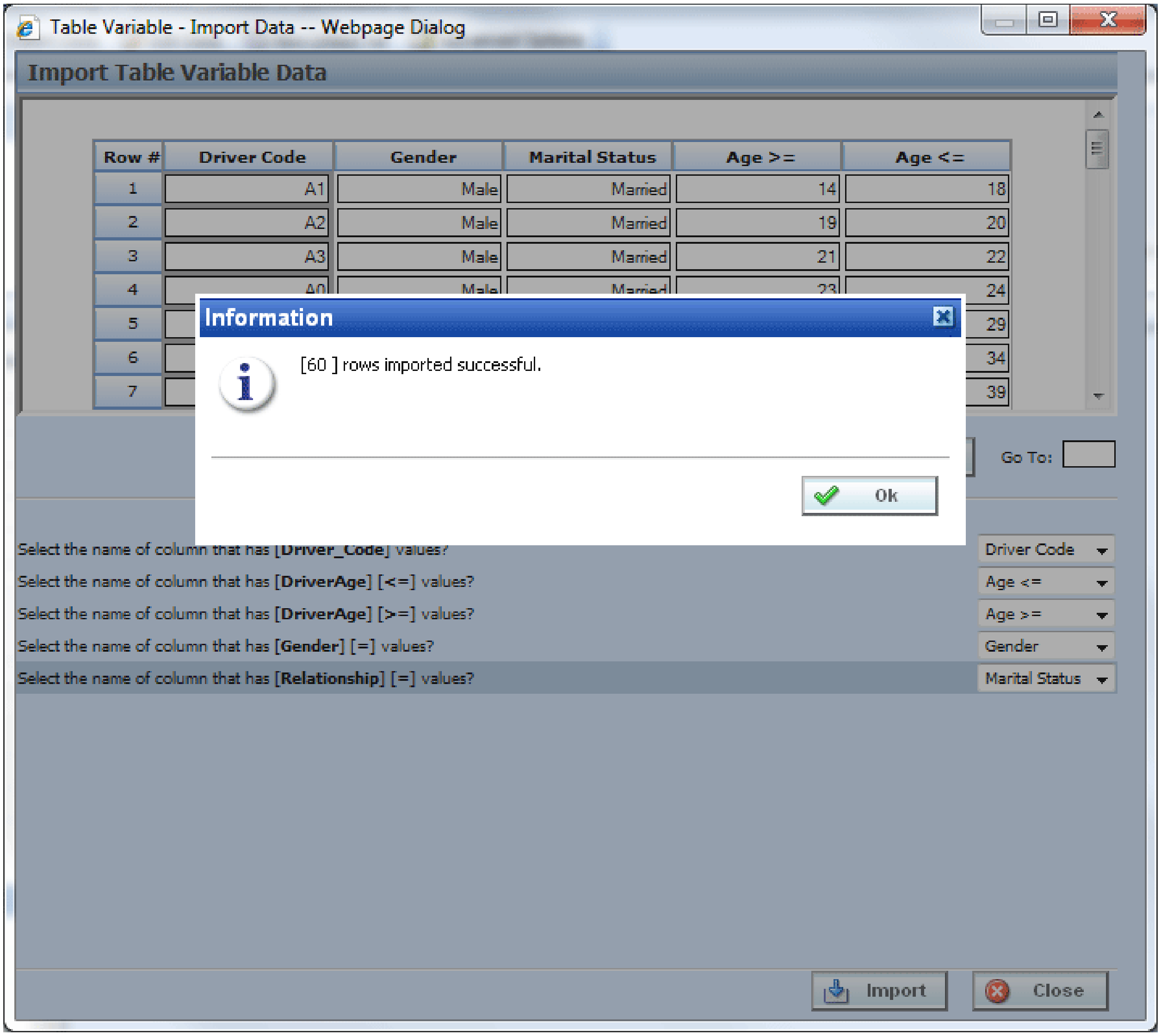


Table Variable - Import Data -- Webpage Dialog
Import Table Variable Data

Row #	Driver Code	Gender	Marital Status	Age >=	Age <=
1	A1	Male	Married	14	18
2	A2	Male	Married	19	20
3	A3	Male	Married	21	22
4	A0	Male	Married	23	24
5					29
6					34
7					39

Information

[60] rows imported successful.

Select the name of column that has [Driver_Code] values? Driver Code

Select the name of column that has [DriverAge] [<=] values? Age <=

Select the name of column that has [DriverAge] [>=] values? Age >=

Select the name of column that has [Gender] [=] values? Gender

Select the name of column that has [Relationship] [=] values? Marital Status

Figure 108 Successful Import Message

Using a Date Mask:

If enabled, a date mask (format) for ‘Date’ data types will be displayed. This date mask will allow 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 to the format defined at the program level.

Date masks can be set at the line of business level and at the program level. At the line of business level, the date mask will apply to all programs in that line of business. At the program level, the date mask will apply to all versions of the program. A program level date mask will override a line of business date mask. Wherever the date format is defined, either at the line of business 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.

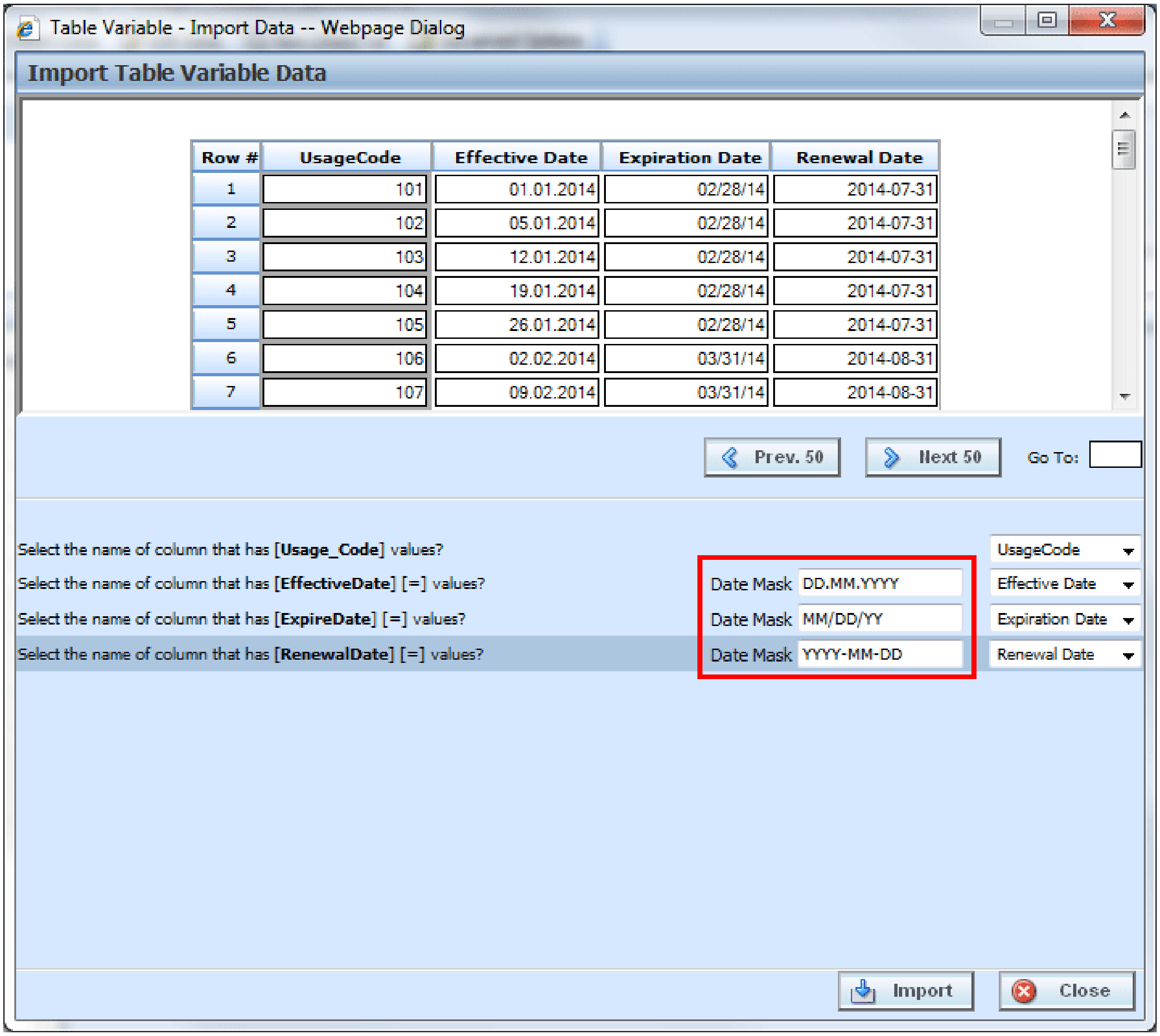


Table Variable - Import Data -- Webpage Dialog
Import Table Variable Data
Prev. 50
Next 50
Go To:
Import
Close

Row #	UsageCode	Effective Date	Expiration Date	Renewal Date
1	101	01.01.2014	02/28/14	2014-07-31
2	102	05.01.2014	02/28/14	2014-07-31
3	103	12.01.2014	02/28/14	2014-07-31
4	104	19.01.2014	02/28/14	2014-07-31
5	105	26.01.2014	02/28/14	2014-07-31
6	106	02.02.2014	03/31/14	2014-08-31
7	107	09.02.2014	03/31/14	2014-08-31

Select the name of column that has **[Usage_Code]** values? UsageCode

Select the name of column that has **[EffectiveDate]** [=] values? Date Mask DD.MM.YYYY Effective Date

Select the name of column that has **[ExpireDate]** [=] values? Date Mask MM/DD/YY Expiration Date

Select the name of column that has **[RenewalDate]** [=] values? Date Mask YYYY-MM-DD Renewal Date

Figure 109 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**.

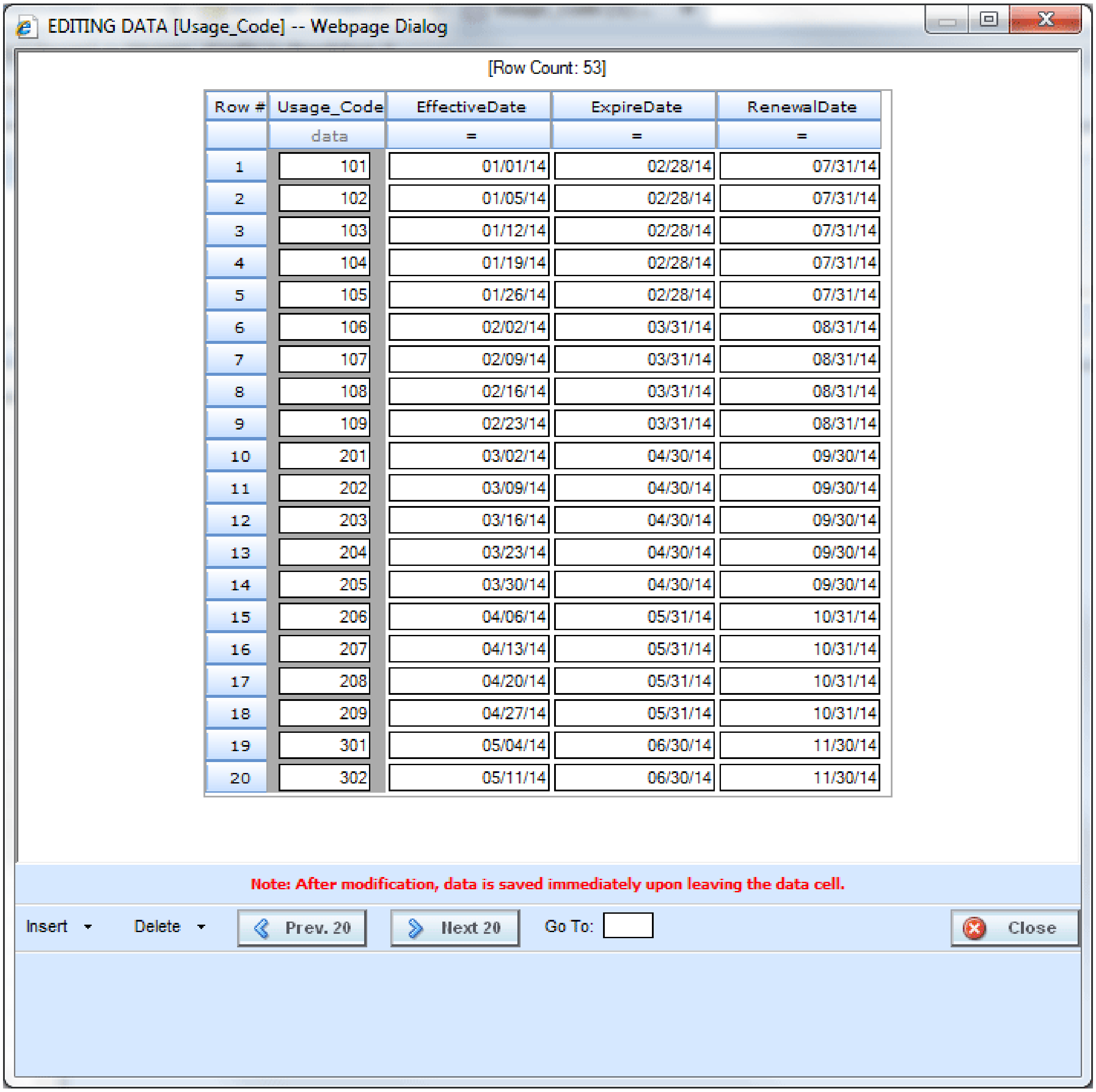
6. If the table variable already contains data, you will be asked if you would like to delete the data first. Select **Ok** to have existing data in the table variable deleted before the new data is imported.

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.



EDITING DATA [Usage_Code] -- Webpage Dialog

[Row Count: 53]

Row #	Usage_Code	EffectiveDate	ExpireDate	RenewalDate
	data	=	=	=
1	101	01/01/14	02/28/14	07/31/14
2	102	01/05/14	02/28/14	07/31/14
3	103	01/12/14	02/28/14	07/31/14
4	104	01/19/14	02/28/14	07/31/14
5	105	01/26/14	02/28/14	07/31/14
6	106	02/02/14	03/31/14	08/31/14
7	107	02/09/14	03/31/14	08/31/14
8	108	02/16/14	03/31/14	08/31/14
9	109	02/23/14	03/31/14	08/31/14
10	201	03/02/14	04/30/14	09/30/14
11	202	03/09/14	04/30/14	09/30/14
12	203	03/16/14	04/30/14	09/30/14
13	204	03/23/14	04/30/14	09/30/14
14	205	03/30/14	04/30/14	09/30/14
15	206	04/06/14	05/31/14	10/31/14
16	207	04/13/14	05/31/14	10/31/14
17	208	04/20/14	05/31/14	10/31/14
18	209	04/27/14	05/31/14	10/31/14
19	301	05/04/14	06/30/14	11/30/14
20	302	05/11/14	06/30/14	11/30/14

Note: After modification, data is saved immediately upon leaving the data cell.

Figure 110 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:

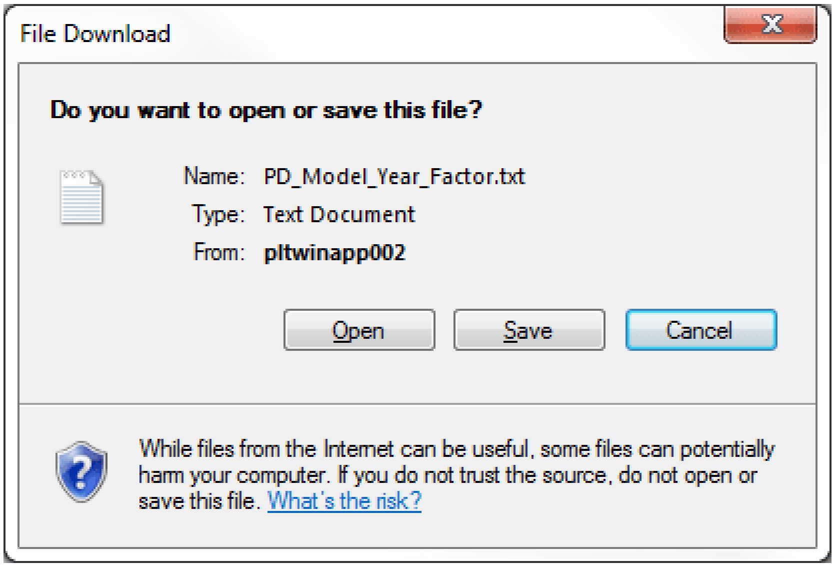
- The date mask applies only to the data you are importing.
- Date formats must match exactly, including the separators being used.
- Day and month formats must be two digits. For example January 7 must have a zero proceeding the month (01) and proceeding the day (07).
- Months must be represented in 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.

EXPORT DATA

Exporting a table places all the data from a table variables data table 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.

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 popup menu.

2. Click Export Data. A File Download popup will be displayed. You will be prompted by your browser to confirm downloading the file.



File Download

Do you want to open or save this file?

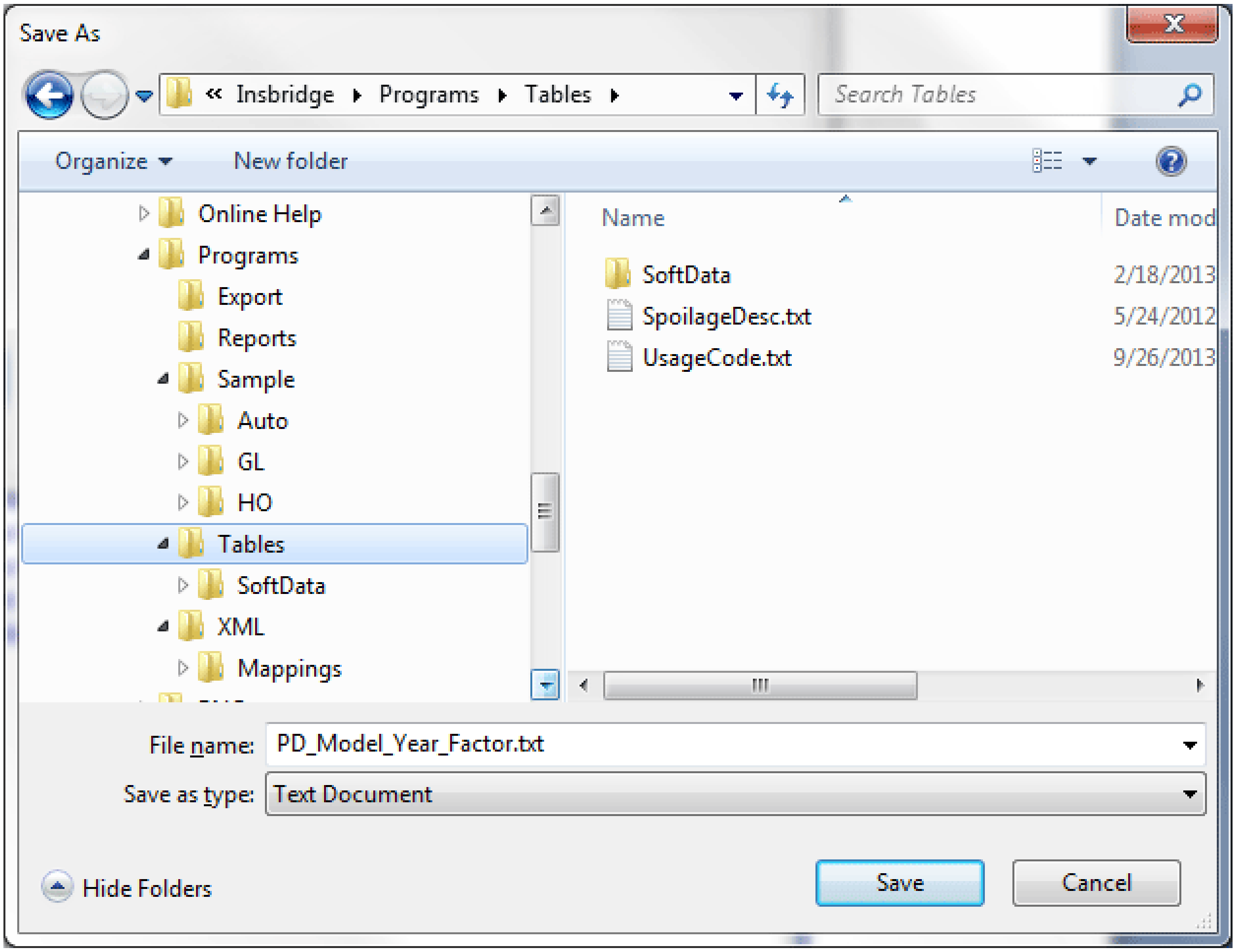
Name: PD_Model_Year_Factor.txt
Type: Text Document
From: pltwinapp002

While files from the Internet can be useful, some files can potentially harm your computer. If you do not trust the source, do not open or save this file. What's the risk?

Figure 111 Exporting a Table

3. Click **Save** to save the file to your local or network hard drive or **Open** to open the file with your default text editor.

4. If you chose to save the file, you will be asked to choose a location to save the file to.



File name: PD_Model_Year_Factor.txt

Save as type: Text Document

Figure 112 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 will open your computers explorer.
   c. **Close**: this will close the dialog box.

   You can now edit the file with the text editor of your choice.

Exported 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.

Using Open Office

Exporting and modifying RateManager table data in Open Office Calc application. When opening a text file, this application allows you to select the File Origin (encoding). If you select UTF-8, the file will display properly.

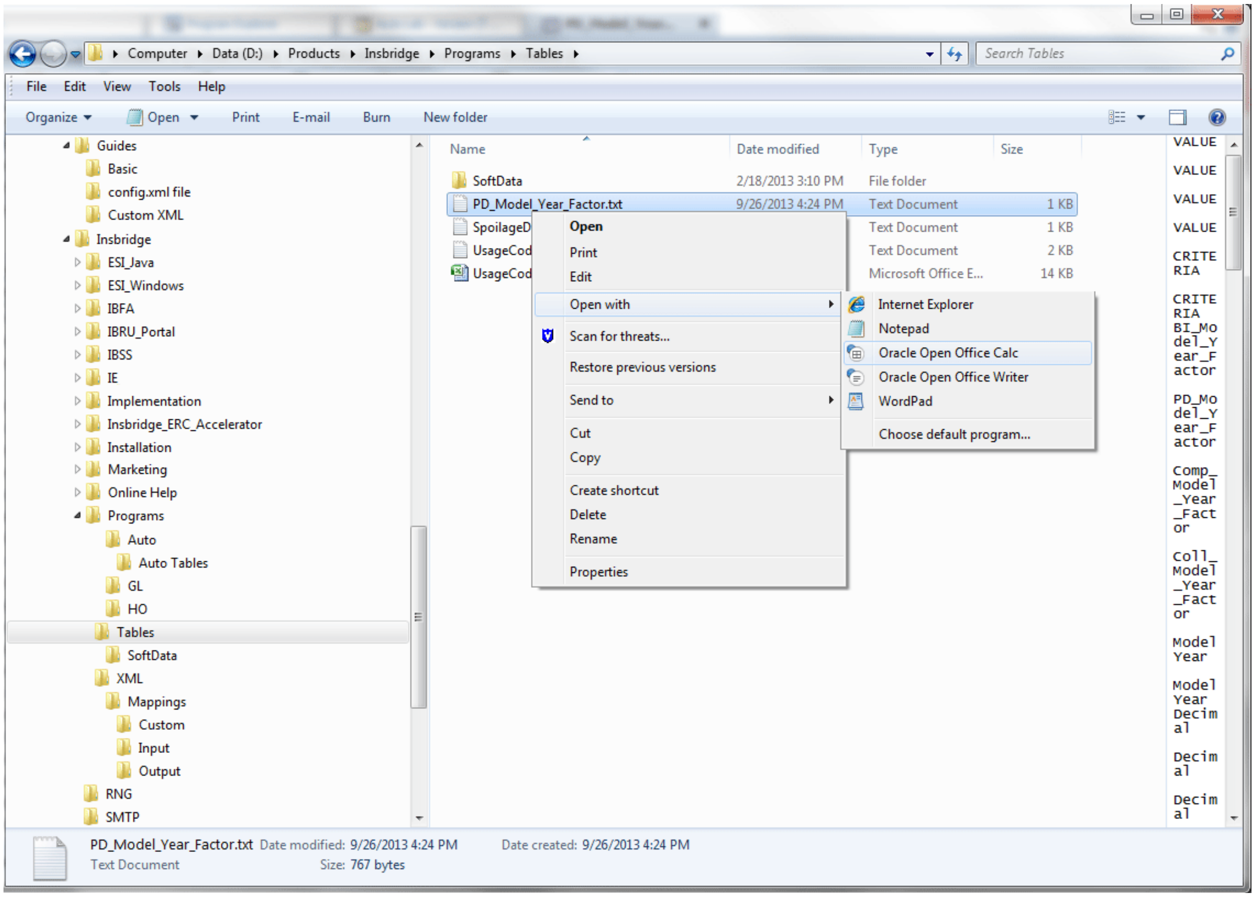

Computer ▸ Data (D:) ▸ Products ▸ Insbridge ▸ Programs ▸ Tables ▸
Search Tables
File Edit View Tools Help
Organize ▾ Open ▾ Print E-mail Burn New folder
Guides
Basic
config.xml file
Custom XML
Insbridge
ESI_Java
ESI_Windows
IBFA
IBRU_Portal
IBSS
IE
Implementation
Insbridge_ERC_Accelerator
Installation
Marketing
Online Help
Programs
Auto
Auto Tables
GL
HO
Tables
SoftData
XML
Mappings
Custom
Input
Output
RNG
SMTP
Name | Date modified | Type | Size
SoftData | 2/18/2013 3:10 PM | File folder
PD_Model_Year_Factor.txt | 9/26/2013 4:24 PM | Text Document | 1 KB
SpoilageD | Text Document | 1 KB
UsageCod | Text Document | 2 KB
UsageCod | Microsoft Office E... | 14 KB
Open
Print
Edit
Open with
Scan for threats...
Restore previous versions
Send to
Cut
Copy
Create shortcut
Delete
Rename
Properties
Internet Explorer
Notepad
Oracle Open Office Calc
Oracle Open Office Writer
WordPad
Choose default program...
VALUE
VALUE
VALUE
VALUE
CRITERIA
CRITERIA
BI_Model_Year_Factor
PD_Model_Year_Factor
Comp_Model_Year_Factor
Coll_Model_Year_Factor
Model Year
Model Year Decimal
Decimal
Decimal
PD_Model_Year_Factor.txt Date modified: 9/26/2013 4:24 PM Date created: 9/26/2013 4:24 PM
Text Document Size: 767 bytes

Figure 113 Using Oracle Open Office for Editing Table Data

1. In RateManager, export your table variable. This will create a tab delimited, .txt, file.

2. Right click the .txt file and choose to Open With Oracle Open Office Calc.

3. Oracle Open Office Calc will present a screen with Import options. The options you select will persist so you will only have to do this one time.

4. Select your Import Settings:

    a. The Import Character set needs to be Unicode, UTF-8.
    b. Separated by option should be Tab.

c. Text delimiter should be blank. Delete the value in the drop-down.

5. Open the file.

When you open your file, the data will be displayed as expected.

Oracle Open Office Calc is available at OpenOffice.org.

Using Other Editing Applications:

If you attempt to open an exported text file, .txt, with another 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.

EDITING TABLE DATA

Edit data in a table when you want to add, delete or change the value for a row. For variables that are locked, such as template generated variables, you will be able to view the data in the data preview area of the Program Details tab but not make any changes. If you need to make changes to a locked variable, you will have to create a new revision. The new revision will be open for editing.

If you need to make a lot of changes to the table, you may want to Export the table, edit it, and then re-import the table.

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.*

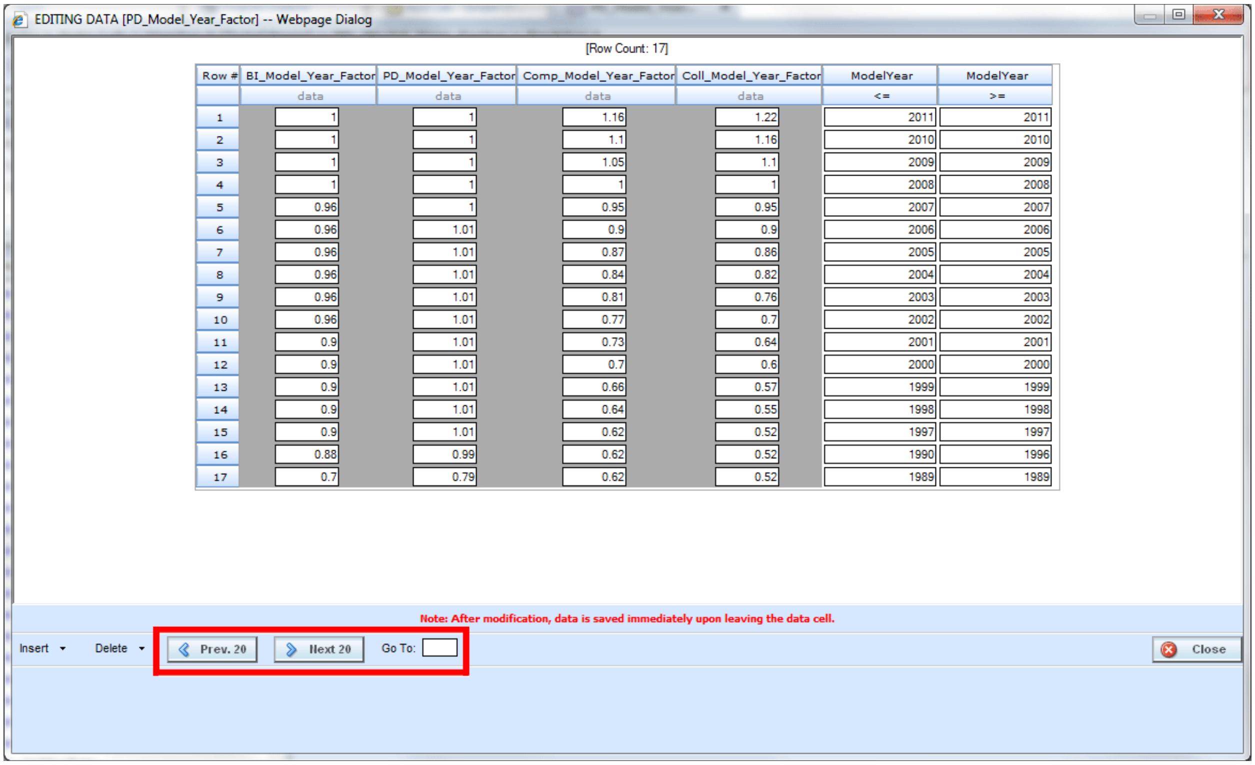
Right Click Menu:

Data tables 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.

To Edit Data in a Table:

1. Navigate to the **Edit Variable** tab for the table variable from where you want to edit the data.

2. This opens the **Edit Data** screen. This screen displays the table data and lists the row count at the top of the screen.



EDITING DATA [PD_Model_Year_Factor] -- Webpage Dialog

[Row Count: 17]

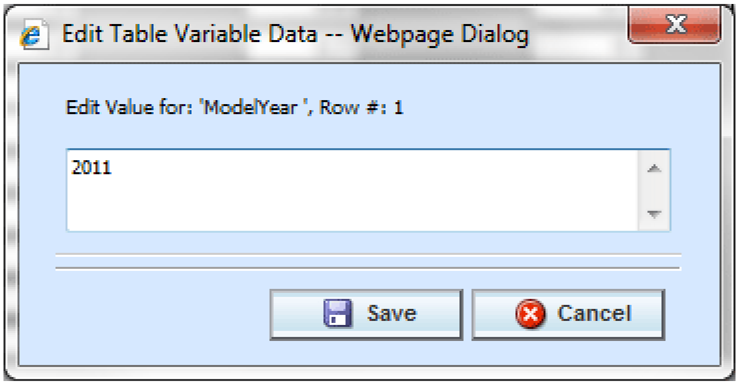
Row #	BI_Model_Year_Factor	PD_Model_Year_Factor	Comp_Model_Year_Factor	Coll_Model_Year_Factor	ModelYear	ModelYear
	data	data	data	data	<=	>=
1	1	1	1.16	1.22	2011	2011
2	1	1	1.1	1.16	2010	2010
3	1	1	1.05	1.1	2009	2009
4	1	1	1	1	2008	2008
5	0.96	1	0.95	0.95	2007	2007
6	0.96	1.01	0.9	0.9	2006	2006
7	0.96	1.01	0.87	0.86	2005	2005
8	0.96	1.01	0.84	0.82	2004	2004
9	0.96	1.01	0.81	0.76	2003	2003
10	0.96	1.01	0.77	0.7	2002	2002
11	0.9	1.01	0.73	0.64	2001	2001
12	0.9	1.01	0.7	0.6	2000	2000
13	0.9	1.01	0.66	0.57	1999	1999
14	0.9	1.01	0.64	0.55	1998	1998
15	0.9	1.01	0.62	0.52	1997	1997
16	0.88	0.99	0.62	0.52	1990	1996
17	0.7	0.79	0.62	0.52	1989	1989

Note: After modification, data is saved immediately upon leaving the data cell.

Figure 114 Editing Table Data

3. If the table contains more than twenty rows, you can 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. To edit a cell, first select it by either tabbing to it or using your mouse to select it and then entering the new data. If you double click you can edit in a separate window.



Edit Table Variable Data -- Webpage Dialog

Edit Value for: 'ModelYear ', Row #: 1

2011

Figure 115 Edit Mapped Variable Data

NOTE: *When you make a change to a cell in the data table, it is immediately updated in the database. Therefore, you cannot undo changes to the data table by clicking **Cancel** on the Edit Data screen. You must re-enter the data if you inadvertently change a value.*

5. To insert a new row into the table, select a cell in the table where you want the new row to be inserted. Click **Insert** and then select either **Above Row** or **Below Row** to insert the new row

either above or below the current row. If no cell is selected, the new row will be added to the end of the table. The Insert and Delete options are on the lower left portion of the screen, next to the Previous 20 button.

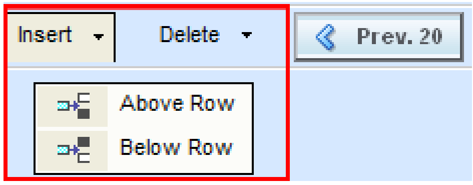


Figure 116 Inserting Row in Table

6. To delete a row from the table, select a cell in the row you want to delete. Click **Delete** and then select **Current Row**. You will not be prompted for confirmation before deleting a single row. The Insert and Delete options are on the lower left portion of the screen, next to the Previous 20 button.

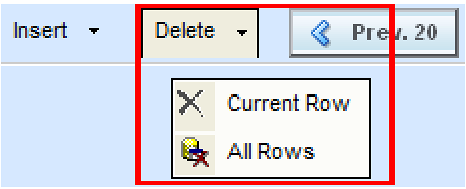
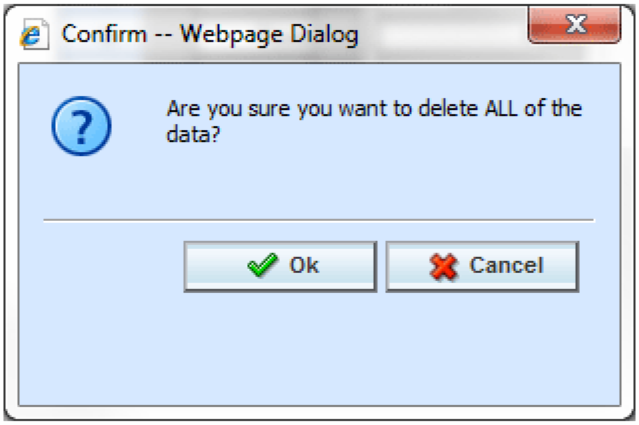


Figure 117 Deleting All Rows in Table

7. To delete all rows from the table, click **Delete** and then select **All Rows**. You will be asked to confirm deleting all rows.



Are you sure you want to delete ALL of the data?

Figure 118 Confirming All Rows Delete

8. Select **Ok** to delete all rows and return to the Edit Data screen or **Cancel** to return to the Edit Data screen.

9. 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 will be updated.

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.

WORKING WITH STEPS

Steps are used use in algorithms, calculated variables, driver assignment algorithms or driver assignment scenarios. 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
  - Exp
  - Rand
  - Fact
  - Sqrt
  - Ceiling
  - Floor
  - Even
  - Odd
- Trig Functions:
  - Cos
  - Cosh
  - Acos
  - Acosh
  - Sin
  - Sinh
  - Asin
  - Asinh
  - Tan
  - Tanh
  - Atan
  - Atanh
  - Degrees
  - Radians
- Ranking:
  - Clear Ranking

    - Rank Across Category
- String Functions:
    - Get Length
    - Set Message
    - String Addition
- Sum Across

The following step types are 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 types are available only for use in ISO ERC template programs:

- 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 and driver assignment scenarios.*

Entering a New Step

New steps can be entered for any unlocked calculated variable, algorithm or driver assignment in a standard program. In a template generated program, any calculated variable, algorithm or driver assignment that is open for editing can have steps added. Whether you are editing an existing program or creating a new one, using a global element of a local 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 will open 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.

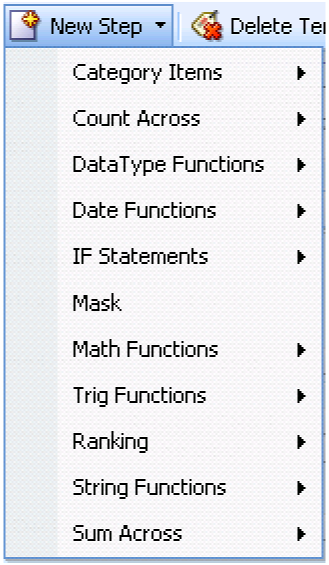


Figure 119 Entering a New Step

5. The new step type will be added in the variable details area. Depending upon the step type, you will have to select the appropriate variables, constants, inputs. The values you need to add 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 hand 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.

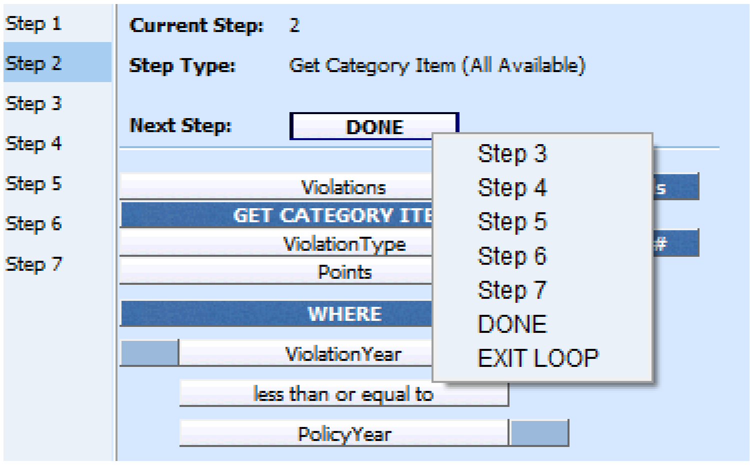
EDITING STEPS

Any step in the Steps list is open for editing. Steps can have Next Steps, Result of Steps, terms, rounding and comparison operators defined or changed. Where clauses, and conditions can also be added 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 will assist with setting the Next Step and Results of Step values.

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.



Current Step: 2

Step Type: Get Category Item (All Available)

Next Step: DONE

Figure 120 Setting the Next Step

The available choices for the Next Step are:

- **STEP #:** Selecting one of the steps listed will set that step as the next step to be processed. Only steps that come after the selected step will be available. You will not be able to go backwards to a previous step or set the step to go to itself. The last step in the list will not have this option.
- **DONE:** Selecting **Done** will exit the algorithm after the step is finished.
- **EXIT LOOP:** Selecting **Exit Loop** will exit the algorithm after the step is finished and prevent 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.

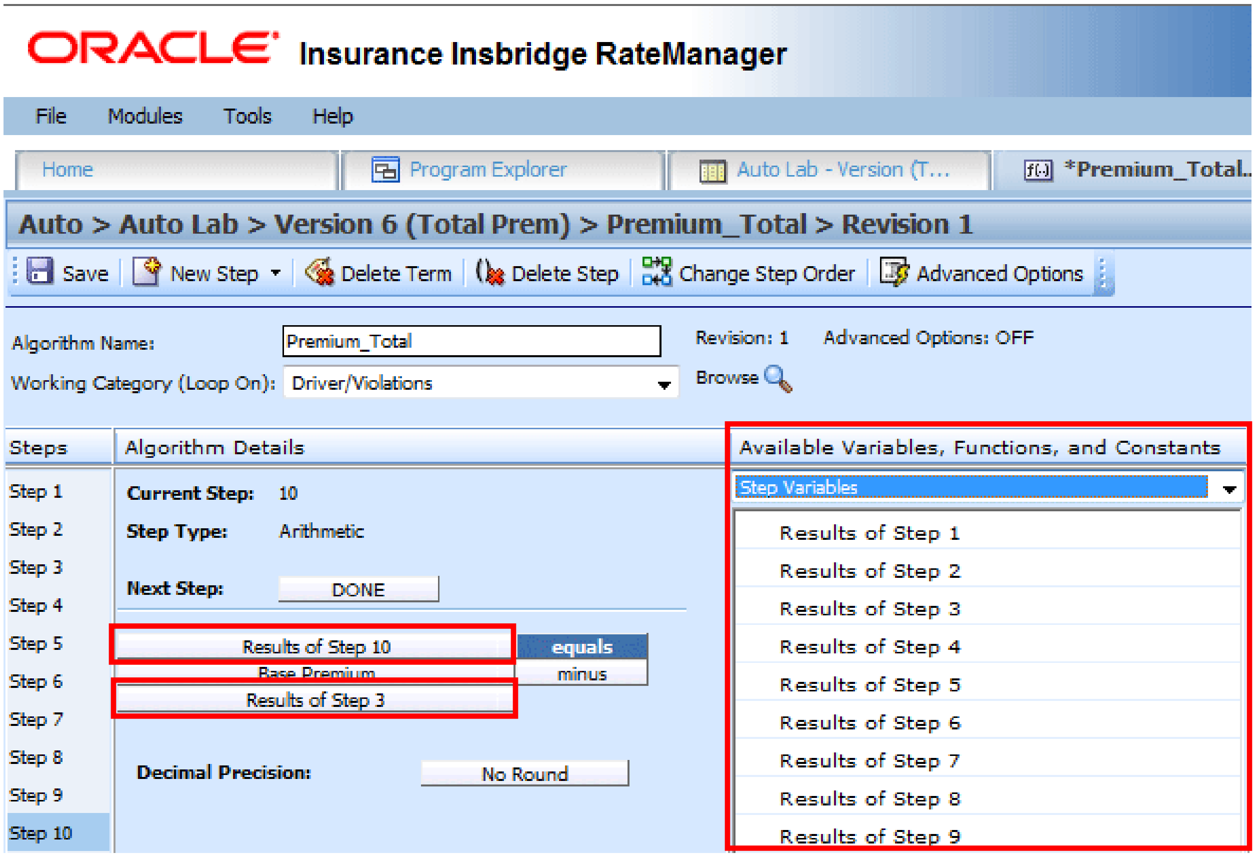
Using Results of Step # Logic

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 need to 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 110 Results of Step, the first field in Step 10 is Results of Step 10. This value will hold whatever result comes from the execution of this step.
- **Second, as a term in a step:** Once you have set a value in a Results of Step #, it can be used as a term in a step further down the steps list. In Figure 110 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 will have 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.



Auto > Auto Lab > Version 6 (Total Prem) > Premium_Total > Revision 1

Algorithm Name: Premium_Total

Revision: 1 Advanced Options: OFF

Working Category (Loop On): Driver/Violations

Current Step: 10

Step Type: Arithmetic

Next Step: DONE

Results of Step 10 equals

Base Premium minus

Results of Step 3

Decimal Precision: No Round

Step Variables

Results of Step 1
Results of Step 2
Results of Step 3
Results of Step 4
Results of Step 5
Results of Step 6
Results of Step 7
Results of Step 8
Results of Step 9

Figure 121 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.

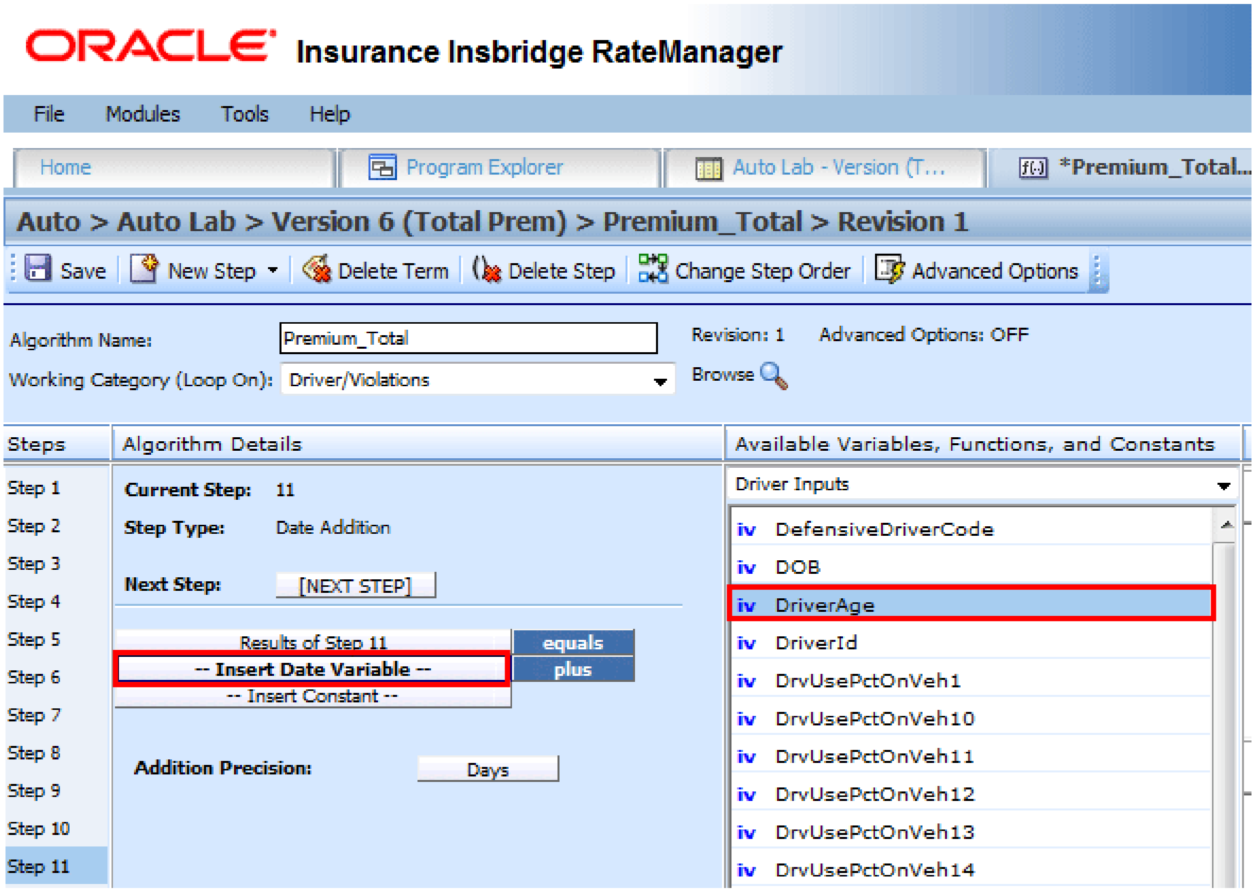
> **Step 1***:* *Results of Step 1 = Base Rate x Territory Factor*
> **Step 2***:* *Results of Step 2 = Results of Step 1 x Limit Factor*
> **Step 3***:* *Results of Step 3 = Results of Step 2 x Credit A*
> **Step 4**: *PremiumX = Results of Step 3 x Credit B*

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: *Results of Step # are* ***NOT*** *updated when steps are moved or deleted. If steps are moved or deleted in an algorithm or calculated variable using Results of Step #, you must reset the 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 will be populated in the term.



Algorithm Name: Premium_Total

Revision: 1

Advanced Options: OFF

Working Category (Loop On): Driver/Violations

Current Step: 11

Step Type: Date Addition

Next Step: [NEXT STEP]

Addition Precision: Days

Figure 122 Editing Steps

NOTE: *Steps are not validated for content. If you do not select a value for a term no error message will be displayed.*

Right Click Menus

Most terms will have a right click menu.

- **View Report:** Brings an information screen containing XML input 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. Allows you to remove 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.

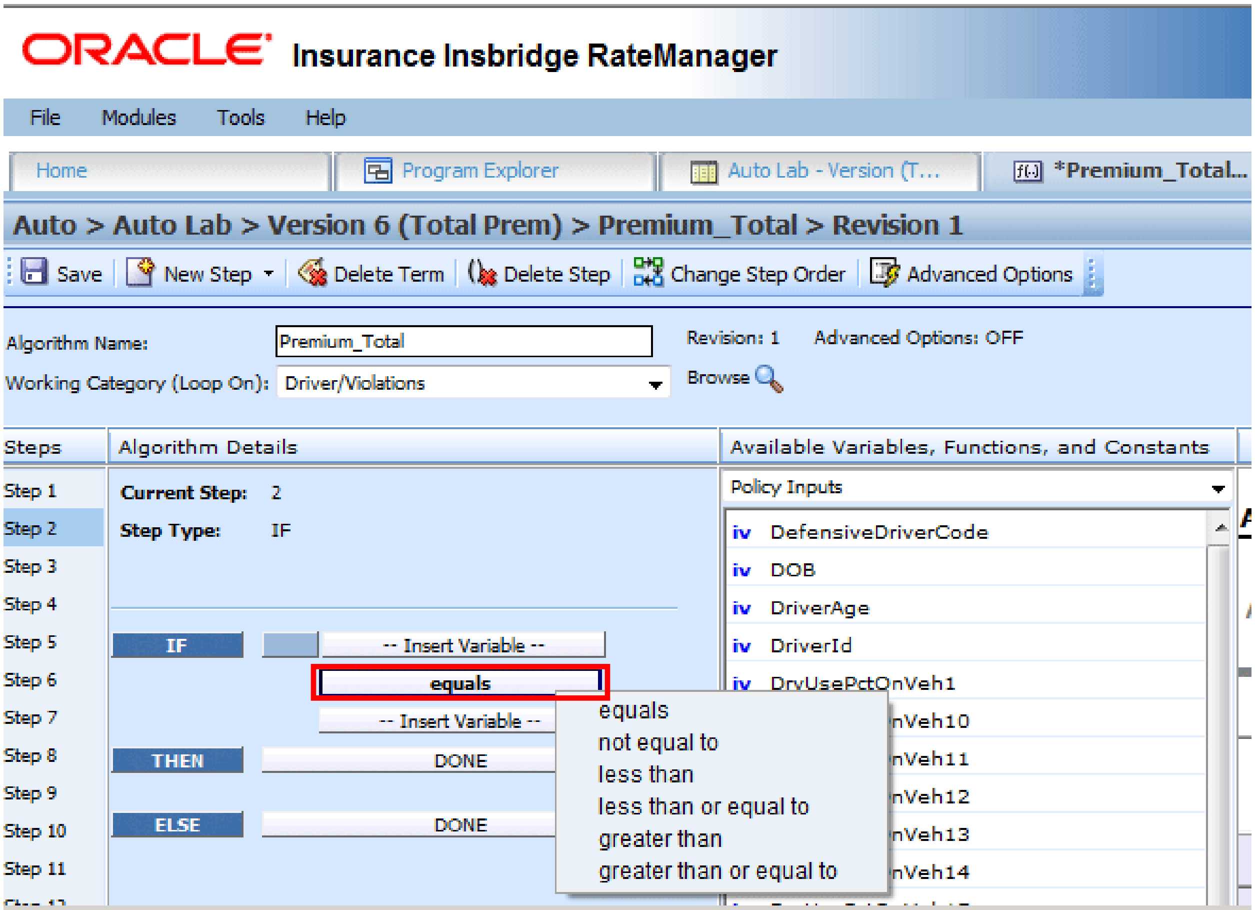

Algorithm Name: Premium_Total
Working Category (Loop On): Driver/Violations
Revision: 1
Advanced Options: OFF
Current Step: 2
Step Type: IF

Figure 123 Setting the Step Operator

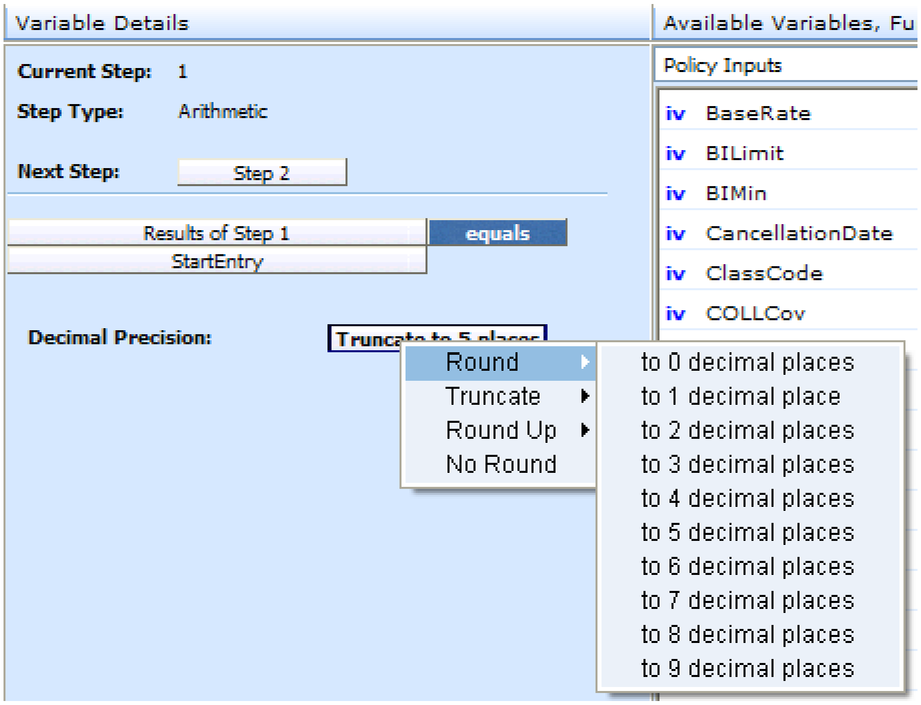
The available comparison operators are:

=	Equals
<>	Not Equals
<	Less Than
<=	Less Than Or Equal To
>	Greater Than
>=	Greater Than Or Equal To

Figure 124 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 popup menu. Rounding can be to 9 decimal places.



Variable Details

Current Step: 1

Step Type: Arithmetic

Next Step: Step 2

Results of Step 1 equals

StartEntry

Decimal Precision: Truncate to 5 places

Figure 125 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 9 digits.
- **Truncate:** All or a portion of the decimal will be dropped. Can be set between 0 and 9 digits.
- **Round Up:** Will round the number, or a portion of it, to the next higher number. Can be set between 0 and 9 digits.
- **No Round:** No rounding will occur on the step.

In RateManager, rounding is always performed Round Half Up. Halfway values, .5 and over, will be rounded up. .4 and lower will be rounded down.

WHERE CLAUSE

The Where clause allows you to specify 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 allows you to run 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 allows you to run 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.

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 IF 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 will be 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 will evaluate to be true and the Then segment will be processed.
- If Group 1 and Group 2 both fail, the entire statement will evaluate to false and the Else segment will be 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 will be validated for pairing. No validation will be 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:

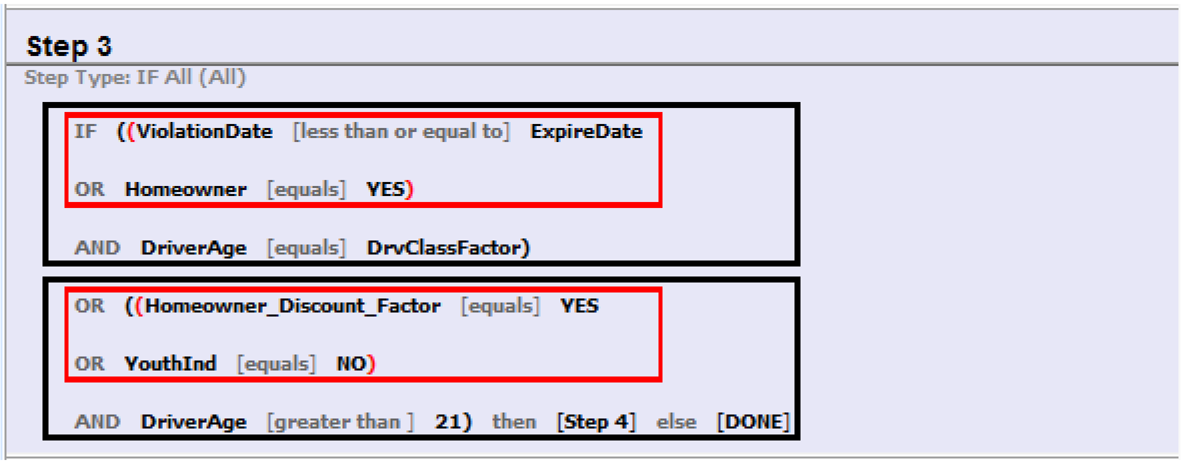
- If Group 1 or Group 2 returns a true value, then the entire statement will evaluate to be true and the Then segment will be processed.

- If Group 1 or Group 2 both fail, the entire statement will evaluate to false and the Else segment will be 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.



Step 3

Step Type: IF All (All)

```
IF   ((ViolationDate   [less than or equal to]   ExpireDate

OR   Homeowner   [equals]   YES)

AND   DriverAge   [equals]   DrvClassFactor)

OR   ((Homeowner_Discount_Factor   [equals]   YES

OR   YouthInd   [equals]   NO)

AND   DriverAge   [greater than ]   21)   then   [Step 4]   else   [DONE]
```

Figure 126 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.

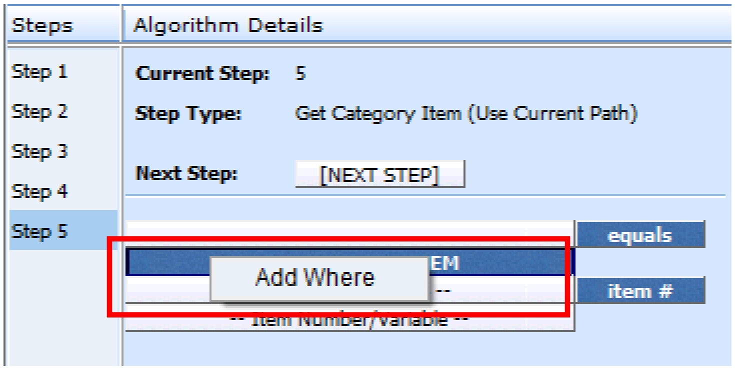
- **Black** is the outermost level
- **Red**
- **Purple**
- **Green**
- **Brown**
- **Blue** is the inner most level

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.



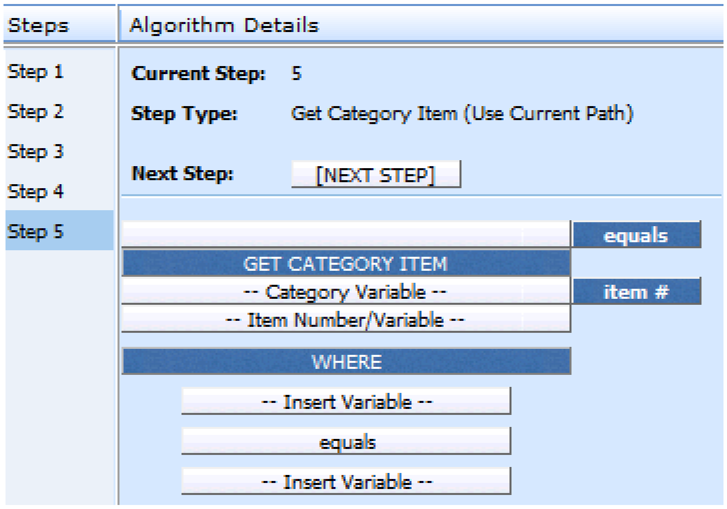
Current Step: 5

Step Type: Get Category Item (Use Current Path)

Next Step: [NEXT STEP]

Figure 127 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.



Current Step: 5

Step Type: Get Category Item (Use Current Path)

Next Step: [NEXT STEP]

Figure 128 Completing the Where Clause

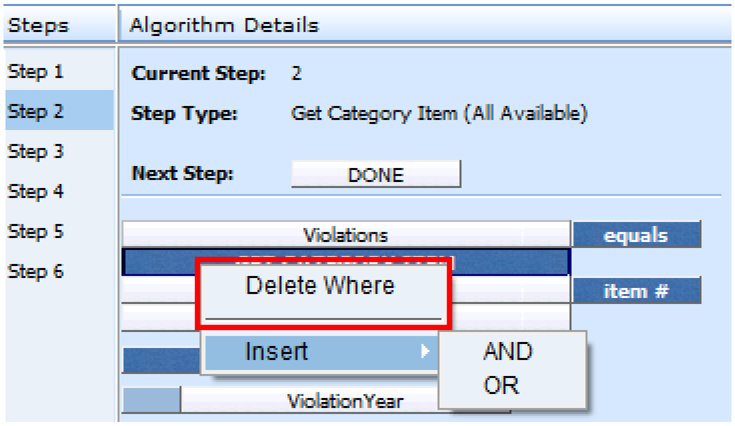
5. When complete, save your work.

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.



Current Step: 2

Step Type: Get Category Item (All Available)

Next Step: DONE

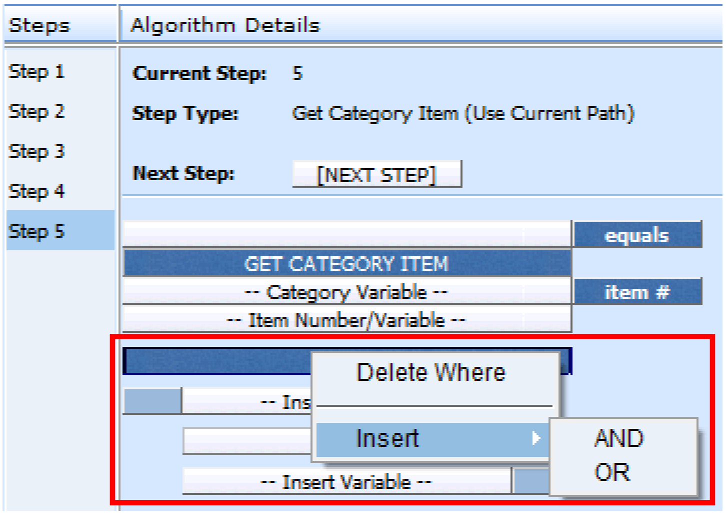
Figure 129 Deleting a Where Clause

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.



Current Step: 5

Step Type: Get Category Item (Use Current Path)

Next Step: [NEXT STEP]

Figure 130 Adding Conditions to a Where Clause

You also can click the step type name to view the menu options. Options are:

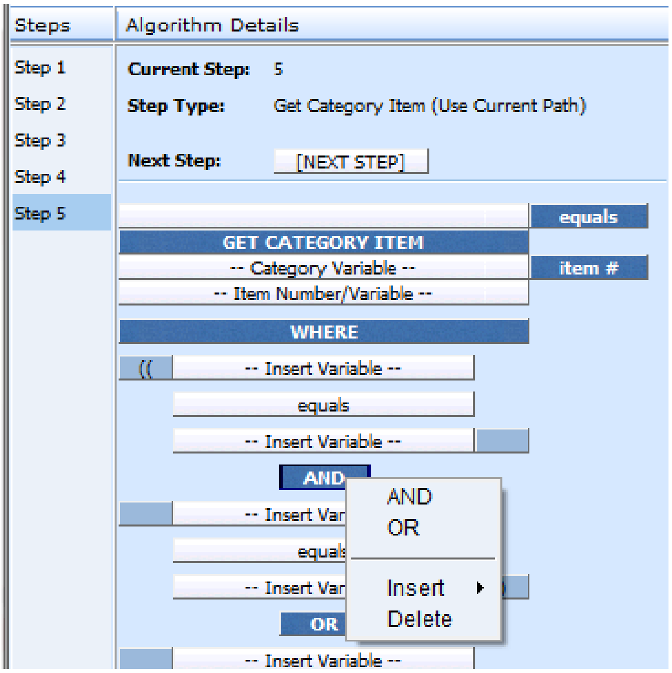
- **Delete Where:** Deletes the entire Where clause, including and conditions. No warning is given. The action is immediate.
- **Insert AND:** Inserts and AND condition under the Where clause.
- **Insert OR:** Inserts an OR condition under the Where clause.

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.



Steps | Algorithm Details

Step 1
Step 2
Step 3
Step 4
Step 5

Current Step: 5

Step Type: Get Category Item (Use Current Path)

Next Step: [NEXT STEP]

Figure 131 Editing a Where Clause

Deleting Conditions

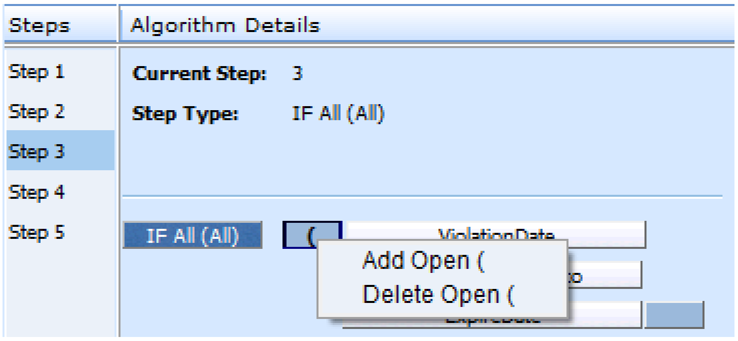
You can delete a condition you do not need. Right click on the AND or the OR blue box of the condition you want to delete. A menu will be displayed. Click **Delete**. The screen will refresh and the condition will be removed.

Conditions are deleted one at a time.

Adding Parentheses

Parentheses can be added to IF 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.



Current Step: 3

Step Type: IF All (All)

Figure 132 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 (.

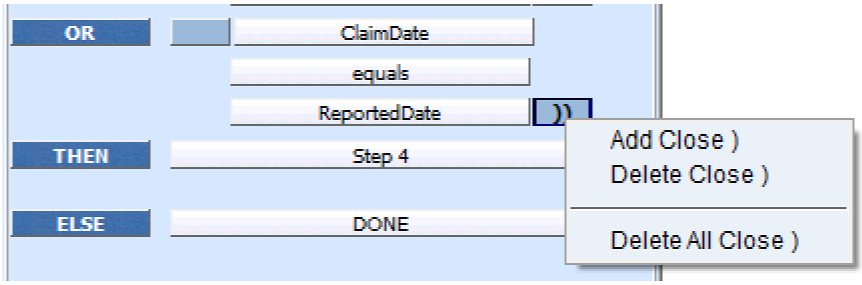


Figure 133 Adding a Close Parentheses

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 will depend upon 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.

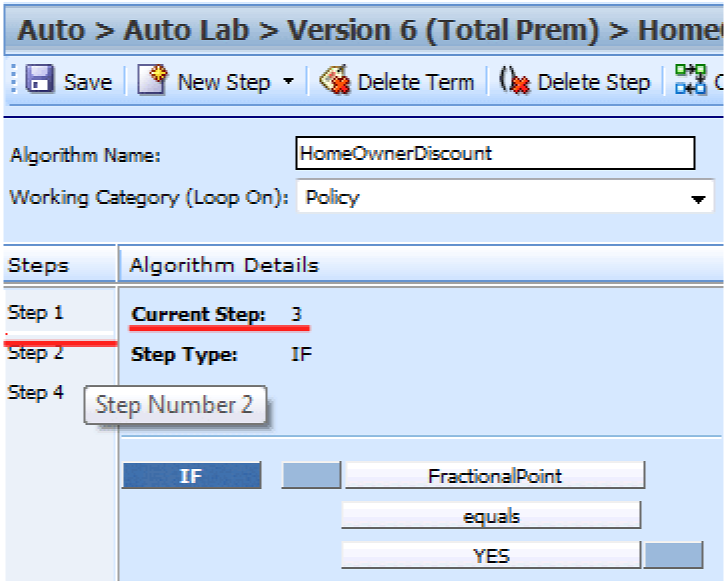
You can review the parentheses pairings in the Algorithm Report area of the tab.

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 will be positioned in the steps column showing where the step will be dropped.



Algorithm Name: HomeOwnerDiscount

Working Category (Loop On): Policy

Current Step: 3

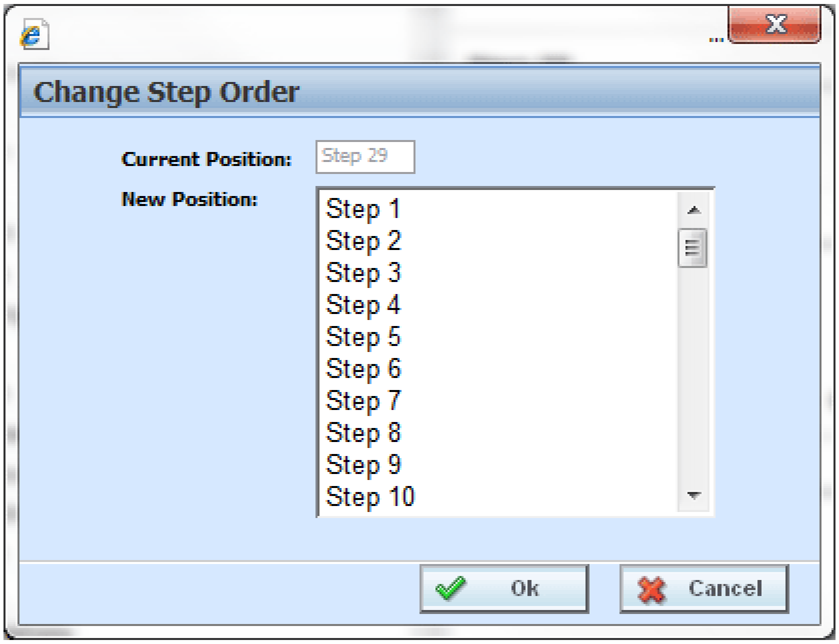
Step Type: IF

Figure 134 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. It allows you to select the exact location you want from a list of steps.



Change Step Order

Current Position: Step 29

New Position:

Figure 135 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 popup window will be displayed.

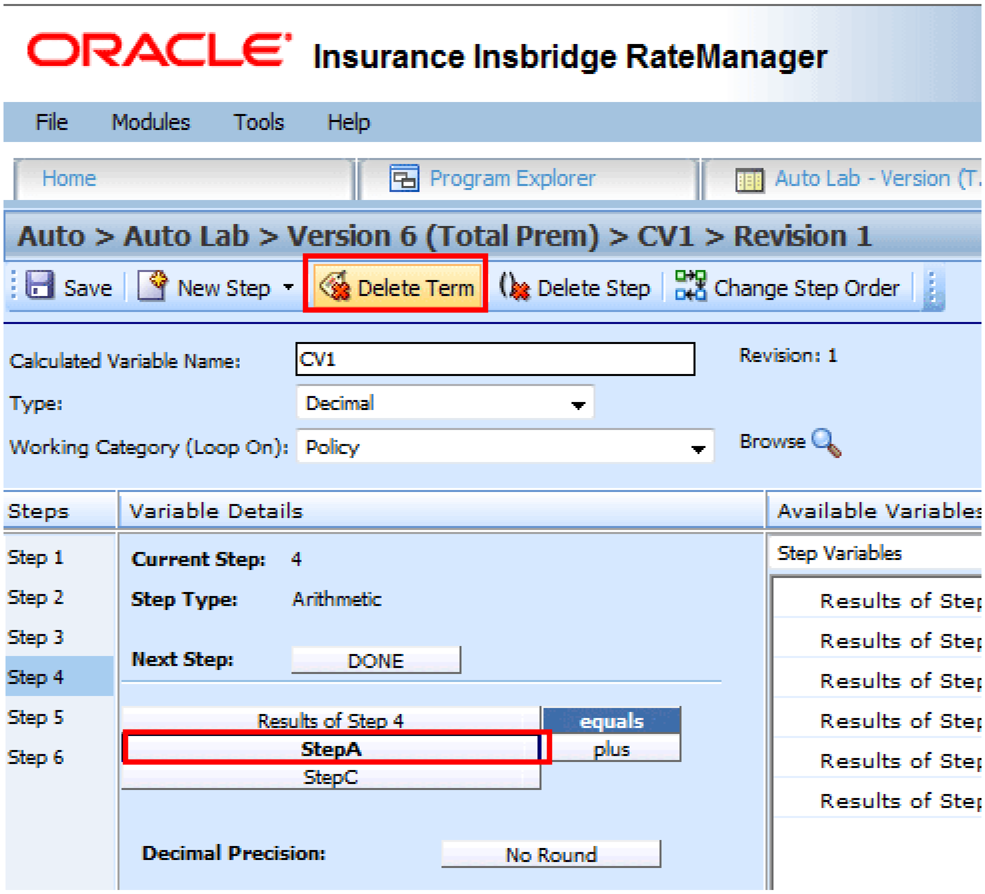
3. Select the new position where you want the step.

4. Click **Ok**. The popup will close and the step will be 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**.



Auto > Auto Lab > Version 6 (Total Prem) > CV1 > Revision 1

Calculated Variable Name: CV1

Type: Decimal

Working Category (Loop On): Policy

Revision: 1

Current Step: 4

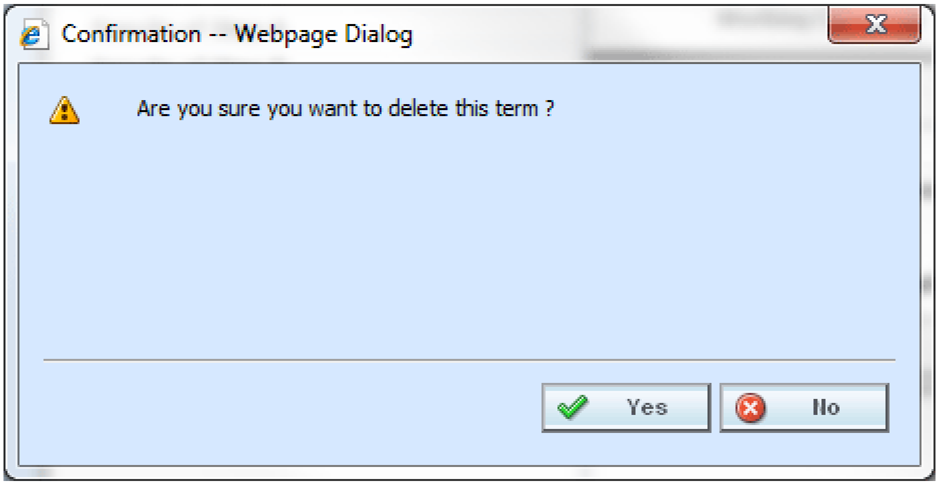
Step Type: Arithmetic

Next Step: DONE

Decimal Precision: No Round

Figure 136 Deleting a Step Term

2. Click **Delete Term** in the top bar menu.



Are you sure you want to delete this term ?

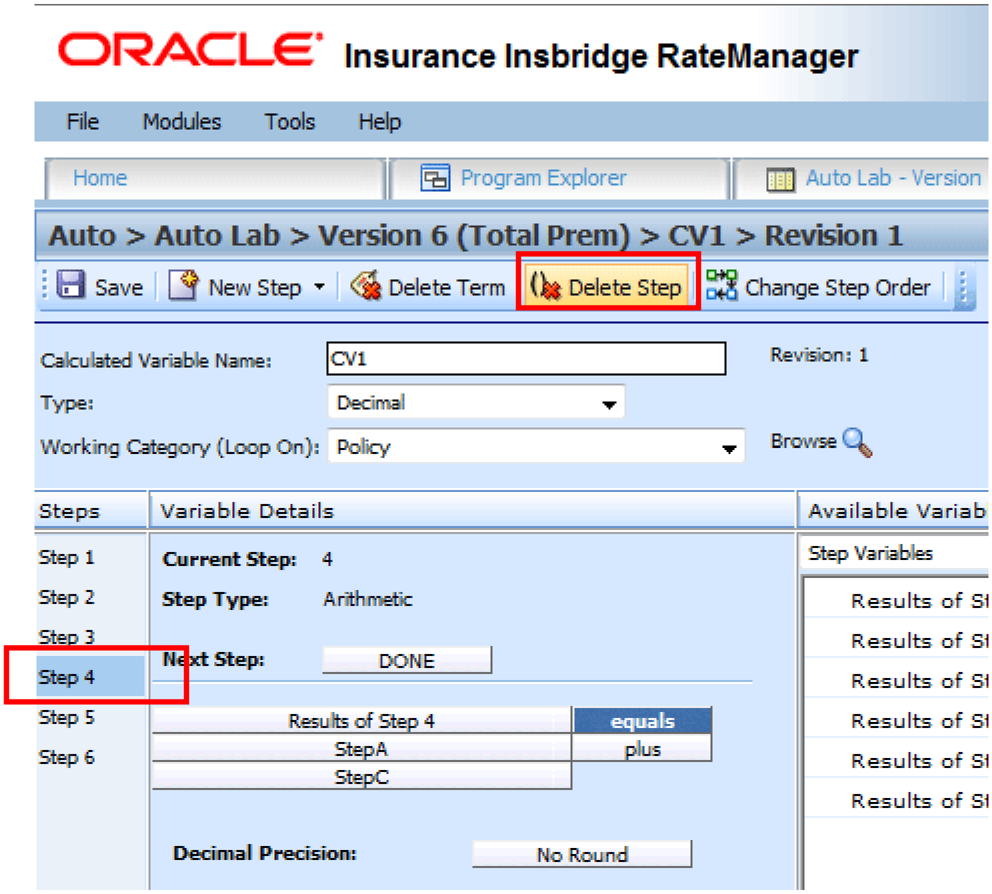
Figure 137 Delete Step Term Warning

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 and drivers assignments must contain at least one step.

1. To delete a step, select the step in the **Steps** column.



Auto > Auto Lab > Version 6 (Total Prem) > CV1 > Revision 1

Calculated Variable Name: CV1

Revision: 1

Type: Decimal

Working Category (Loop On): Policy

Current Step: 4

Step Type: Arithmetic

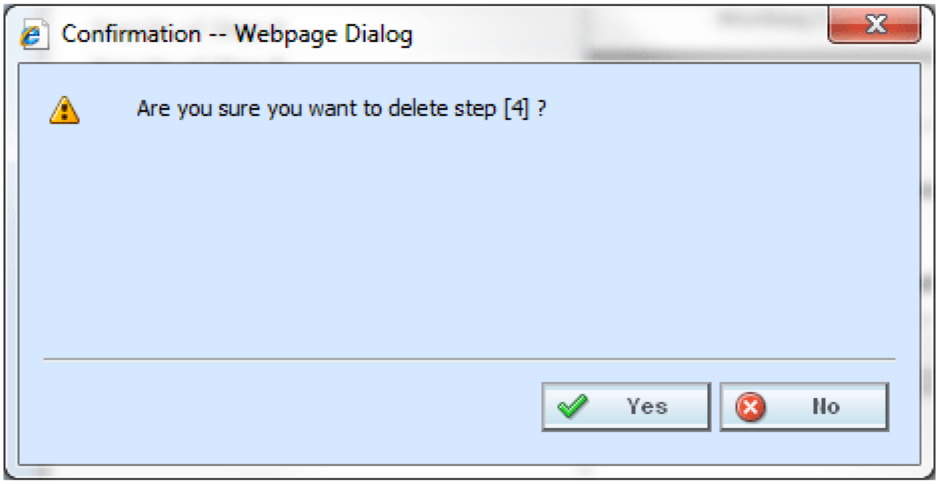
Next Step: DONE

Results of Step 4	equals
StepA	plus
StepC	

Decimal Precision: No Round

Figure 138 Deleting a Step

2. Click **Delete Step** in the top bar menu.



Are you sure you want to delete step [4] ?

Figure 139 Delete Step Warning

3. You will be asked to confirm your deletion. Select **Yes** to delete and **No** to return to the tab.

Category Items

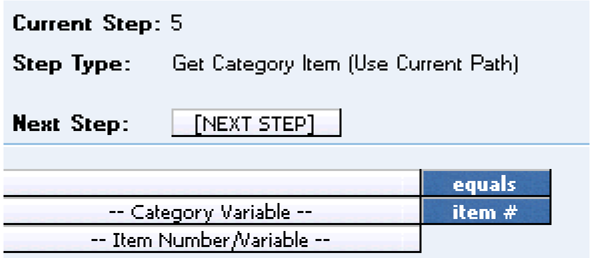
Category Items are used to get or set a specific instance of a variable. There are six Get/Set Category Items:

- 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.



Current Step: 5

Step Type: Get Category Item (Use Current Path)

Next Step: [NEXT STEP]

Figure 140 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: 01/04/2009
               Violation Points: 2
          </Violation>
          <Violation>
               Violation Type: Driving without a license
               Violation Date: 05/30/2009
               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: 12/02/2009
            Violation Points: 1
        </Violation>
        <Violation>
            Violation Type: Speeding
            Violation Date: 01/21/2010
            Violation Points: 2
        </Violation>
        <Violation>
            Violation Type: Running a stop sign
            Violation Date: 11/14/2010
            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: 01/04/2009
            Violation Points: 2
        </Violation>
        <Violation>
            Violation Type: Driving without a license
            Violation Date: 05/30/2009
            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: 12/02/2009
            Violation Points: 1
        </Violation>
        <Violation>
            Violation Type: Speeding
            Violation Date: 01/21/2010
            Violation Points: 2
        </Violation>
        <Violation>
            Violation Type: Running a stop sign
            Violation Date: 11/14/2010
            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.

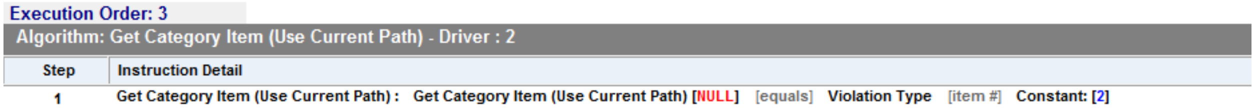
Policy Fourth Violation Type [=] Violation Type [Item #] 4

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.



Execution Order: 3

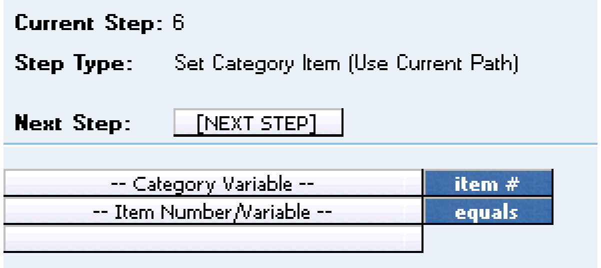
Algorithm: Get Category Item (Use Current Path) - Driver : 2

Step	Instruction Detail
1	Get Category Item (Use Current Path) : Get Category Item (Use Current Path) [NULL] [equals] Violation Type [item #] Constant: [2]

This will be 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.



Current Step: 6

Step Type: Set Category Item (Use Current Path)

Next Step: [NEXT STEP]

Figure 141 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: 01/04/2009
               Violation Points: 2
          </Violation>
          <Violation>
               Violation Type: Driving without a license
               Violation Date: 05/30/2010
               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: 12/02/2009
               Violation Points: 1
          </Violation>
        <Violation>
            Violation Type: Speeding
            Violation Date: 01/21/2010
            Violation Points: 2
        </Violation>
        <Violation>
            Violation Type: Running a stop sign
            Violation Date: 11/14/2010
            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: 01/04/2009
            Violation Points: 2
        </Violation>
        <Violation>
            Violation Type: Driving without a license
            Violation Date: 05/30/2010
            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: 12/02/2009
            Violation Points: 1
        </Violation>
        <Violation>
            Violation Type: Speeding
            Violation Date: 01/21/2010
            Violation Points: 2
        </Violation>
        <Violation>
            Violation Type: Running a stop sign
            Violation Date: 11/14/2010
            Violation Points: 2
        </Violation>
</Driver>
```

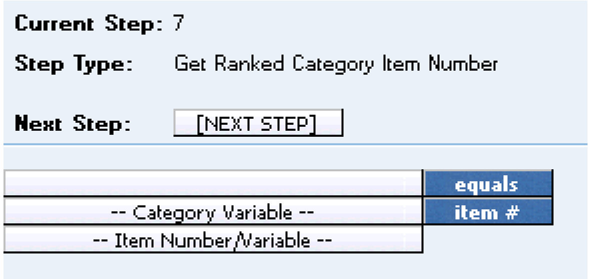
If a **Set Category Item (All Available)** step, used at the Policy level, is used to set the points for the 4th violation on the policy.

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.



Current Step: 7

Step Type: Get Ranked Category Item Number

Next Step: [NEXT STEP]

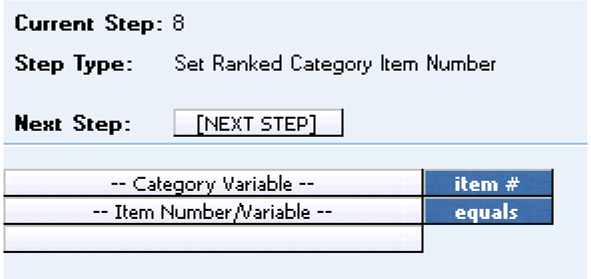
-- Category Variable --

-- Item Number/Variable --

Figure 142 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.



Current Step: 8

Step Type: Set Ranked Category Item Number

Next Step: [NEXT STEP]

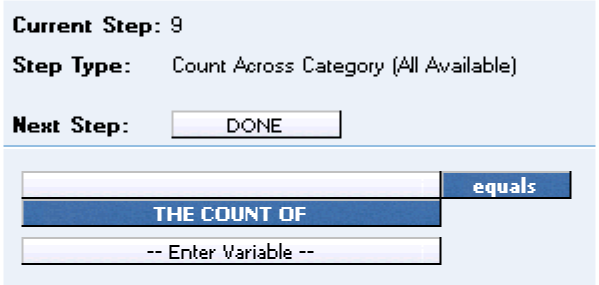
Figure 143 Setting Ranked Category Item

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 (input, variable, result).

There are two available in RateManager:

- Count Across Category (All Available)
- Count Across Category (Use Current Path)



Current Step: 9

Step Type: Count Across Category (All Available)

Next Step: DONE

Figure 144 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: 01/04/2009
               Violation Points: 2
          </Violation>
          <Violation>
               Violation Type: Driving without a license
               Violation Date: 05/30/2010
```

```
            Violation Points: 1
        </Violation>
</Driver>

<Driver>
       Driver #2
        <Violation>
            Violation Type: At fault accident
            Violation Date: 09/19/2008
            Violation Points: 5
        </Violation>
</Driver>

<Driver>
       Driver #3
        <Violation>
            Violation Type: Failure to signal
            Violation Date: 12/02/2009
            Violation Points: 1
        </Violation>
        <Violation>
            Violation Type: Speeding
            Violation Date: 01/21/2010
            Violation Points: 2
        </Violation>
        <Violation>
            Violation Type: Running a stop sign
            Violation Date: 11/14/2011
            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: 01/04/2009
            Violation Points: 2
        </Violation>
        <Violation>
```

```
                Violation Type: Driving without a license
                Violation Date: 05/30/2010
                Violation Points: 1
            </Violation>
</Driver>

<Driver>
        Driver #2
            <Violation>
                Violation Type: At fault accident
                Violation Date: 09/19/2008
                Violation Points: 5
            </Violation>
</Driver>

<Driver>
        Driver #3
            <Violation>
                Violation Type: Failure to signal
                Violation Date: 12/02/2009
                Violation Points: 1
            </Violation>
            <Violation>
                Violation Type: Speeding
                Violation Date: 01/21/2010
                Violation Points: 2
            </Violation>
            <Violation>
                Violation Type: Running a stop sign
                Violation Date: 11/14/2011
                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.

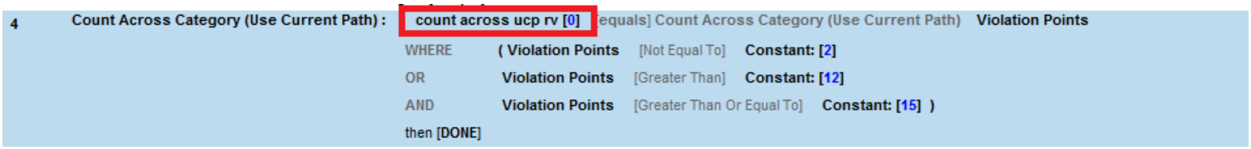
Total Policy Violations [=] The Count (Across All) of Violation Type

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.



```
4   Count Across Category (Use Current Path) :   count across ucp rv [0]   [equals] Count Across Category (Use Current Path)   Violation Points
                                                 WHERE      ( Violation Points   [Not Equal To]   Constant: [2]
                                                 OR           Violation Points   [Greater Than]   Constant: [12]
                                                 AND          Violation Points   [Greater Than Or Equal To]   Constant: [15]  )
                                                 then [DONE]
```

The zero value is used in calculations.

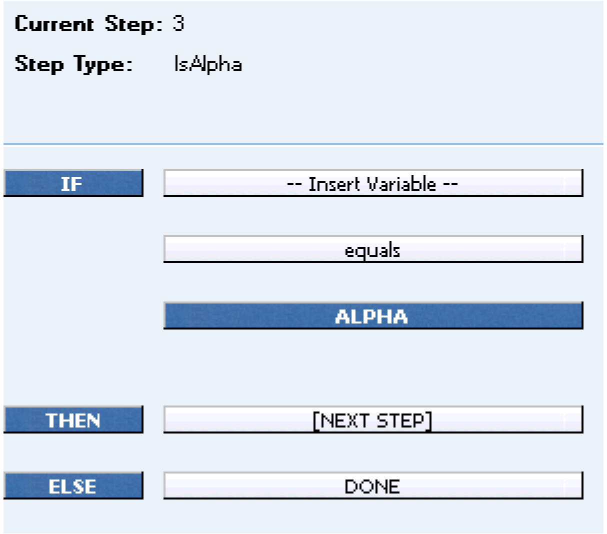
Data Type Functions

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 allows you to check if a variable holds an alpha character value. Operators for this option are equals or not equals.



Current Step: 3

Step Type: IsAlpha

IF -- Insert Variable --

equals

ALPHA

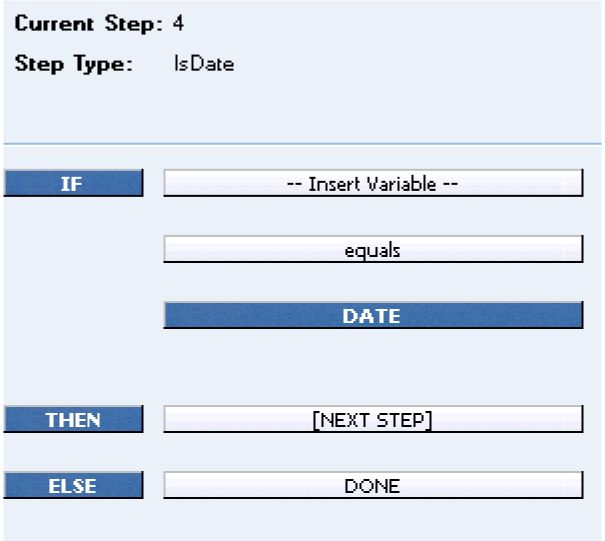
THEN [NEXT STEP]

ELSE DONE

Figure 145 IsAlpha Step

IsDate Step

The IsDate step allows you to check 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.



Current Step: 4

Step Type: IsDate

IF -- Insert Variable --

equals

DATE

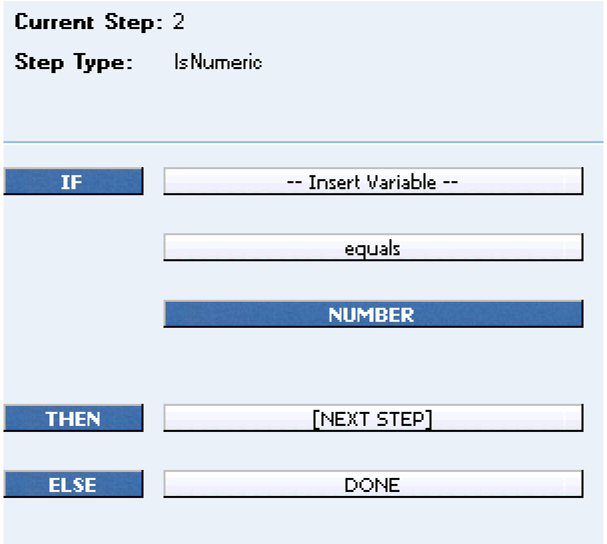
THEN [NEXT STEP]

ELSE DONE

Figure 146 IsDate Step

IsNumeric Step

The IsNumeric step allows you to check 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.



Current Step: 2

Step Type: IsNumeric

IF -- Insert Variable --

equals

NUMBER

THEN [NEXT STEP]

ELSE DONE

Figure 147 IsNumeric Step

Date Functions

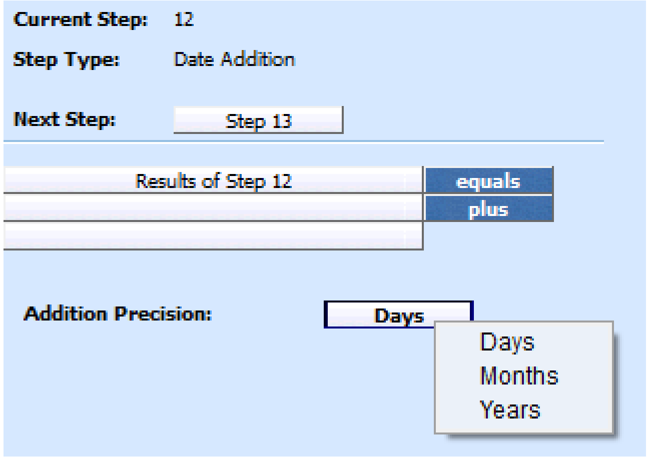
A Date Function step is a special date calculation that calculates the difference between two dates in days, months or years, or adds to a date. There are two types of date functions in RateManager:

- 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 Results variable (the first field in the equation) must be a date data type.



Current Step: 12

Step Type: Date Addition

Next Step: Step 13

Results of Step 12

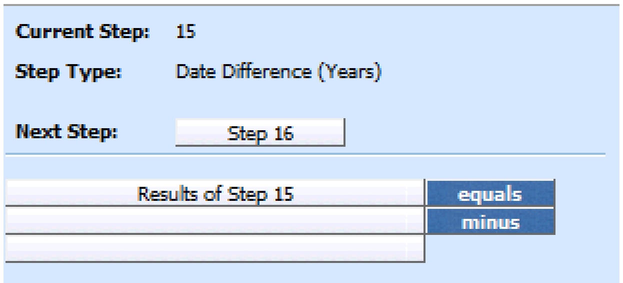
Addition Precision: Days

Figure 148 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 popup menu.

Date Difference Step

A date difference step is used to determine the difference between two dates. 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.



Current Step: 15

Step Type: Date Difference (Years)

Next Step: Step 16

Results of Step 15

Figure 149 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.

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.

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

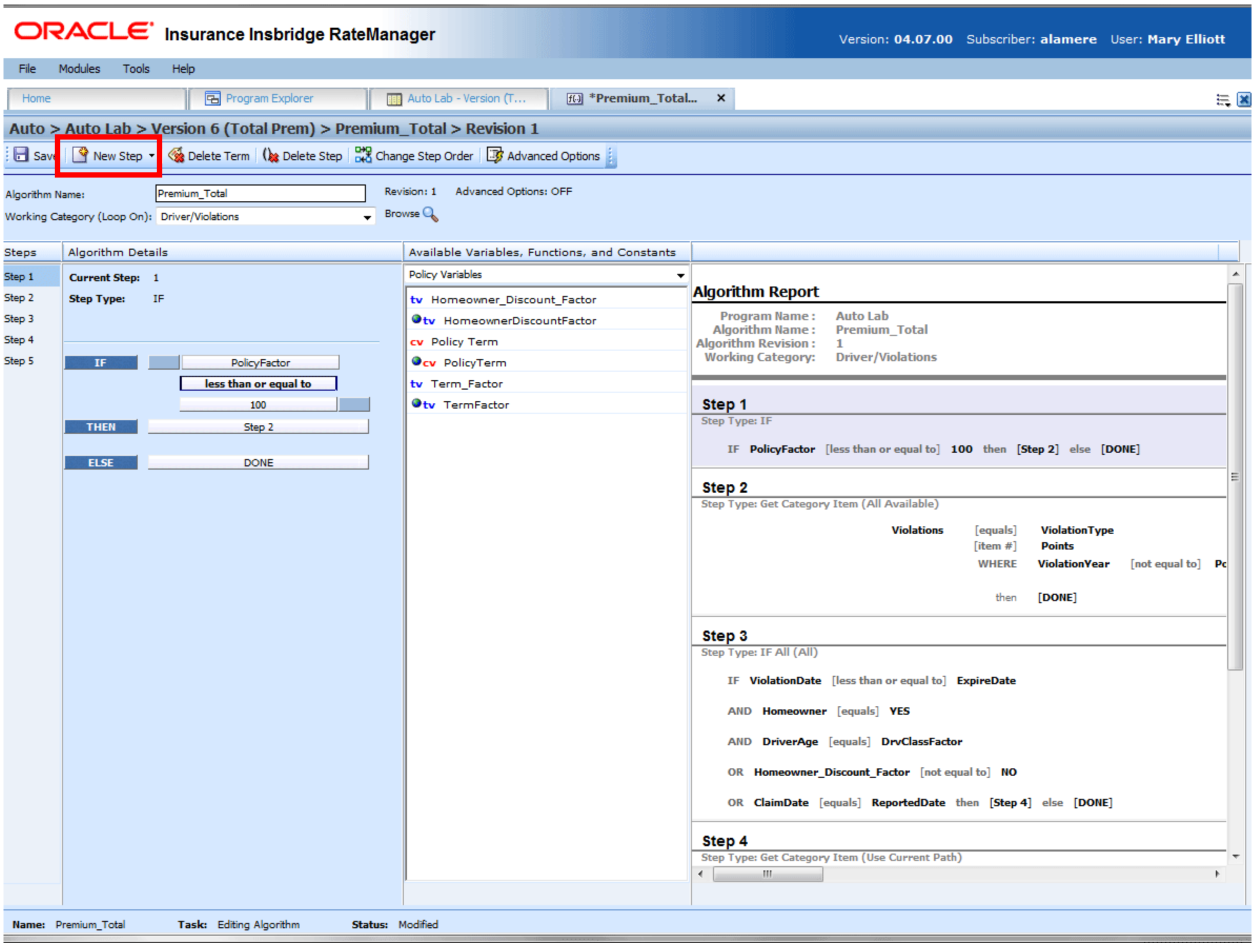
(If the OR condition is **false** and the AND condition is **true**) ELSE Done

(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 and Algorithms.



Auto > Auto Lab > Version 6 (Total Prem) > Premium_Total > Revision 1

Algorithm Name: Premium_Total
Working Category (Loop On): Driver/Violations
Revision: 1
Advanced Options: OFF

Current Step: 1
Step Type: IF

IF PolicyFactor
less than or equal to
100
THEN Step 2
ELSE DONE

Algorithm Report

Program Name : Auto Lab
Algorithm Name : Premium_Total
Algorithm Revision : 1
Working Category: Driver/Violations

Step 1
Step Type: IF

IF **PolicyFactor** [less than or equal to] **100** then **[Step 2]** else **[DONE]**

Step 2
Step Type: Get Category Item (All Available)

Violations [equals] **ViolationType**
[item #] **Points**
WHERE **ViolationYear** [not equal to] **Pc**

then **[DONE]**

Step 3
Step Type: IF All (All)

IF **ViolationDate** [less than or equal to] **ExpireDate**

AND **Homeowner** [equals] **YES**

AND **DriverAge** [equals] **DrvClassFactor**

OR **Homeowner_Discount_Factor** [not equal to] **NO**

OR **ClaimDate** [equals] **ReportedDate** then **[Step 4]** else **[DONE]**

Step 4
Step Type: Get Category Item (Use Current Path)

Name: Premium_Total Task: Editing Algorithm Status: Modified

Figure 150 Creating a Simple IF Statement

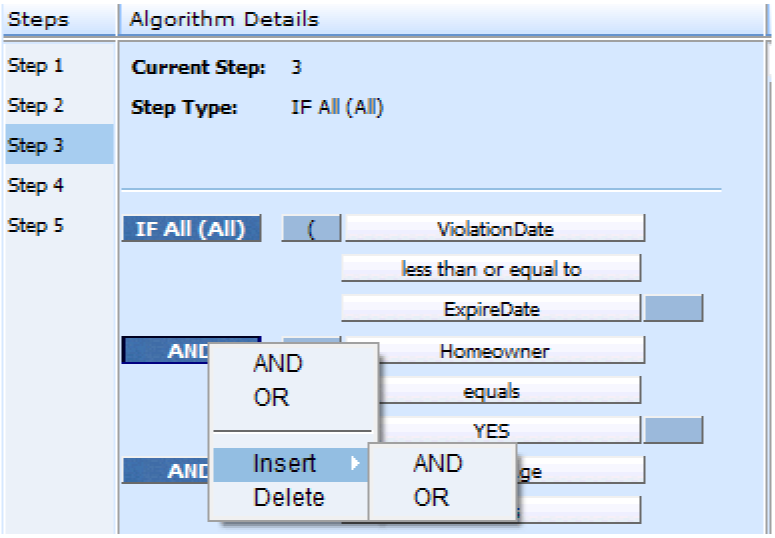
1. Select your New **IF Step Type** from the New Step option at the top of the screen. The screen refreshes with a simple IF statement.

2. Select the term you want to complete. The field will be highlighted.

3. Select the term you want in the field from the Available Variables, Functions and Constants. The field will be 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 will refresh 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 will push 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 will refresh with your selection. Enter in your criteria.



Current Step: 3

Step Type: IF All (All)

ViolationDate
less than or equal to
ExpireDate

Homeowner
equals
YES

Figure 151 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.

6. Continue until you have all your conditions entered. Complete the THEN and ELSE statements. And save your work

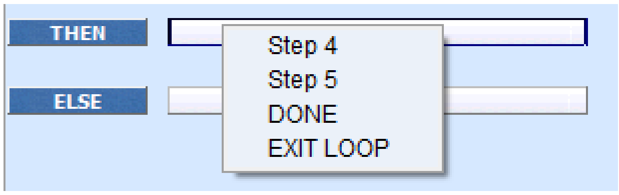
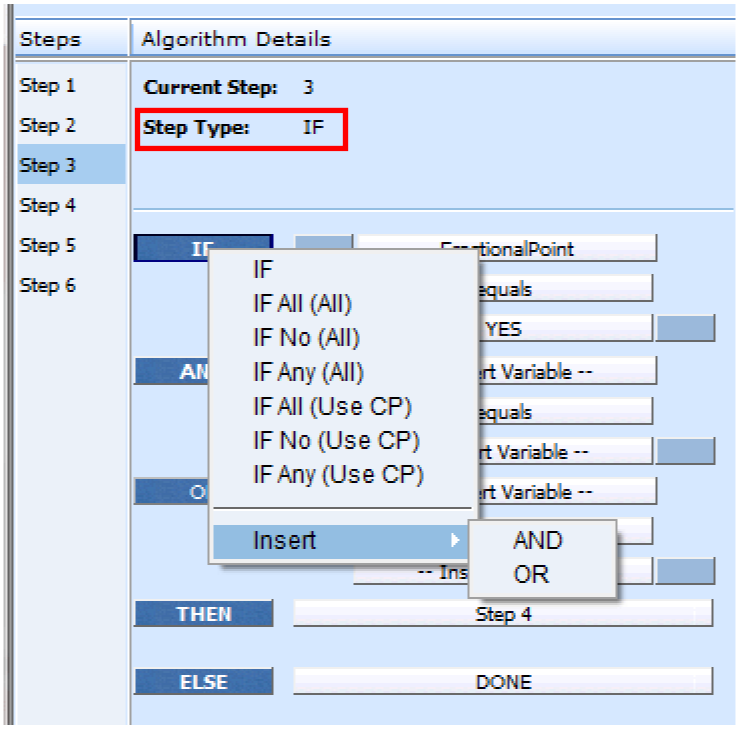


Figure 152 Then and Else Segments of an IF Statement

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 will refresh and the updated step type will be listed at the top of the step.



Current Step: 3

Step Type: IF

Figure 153 Editing a Multiple If Statement

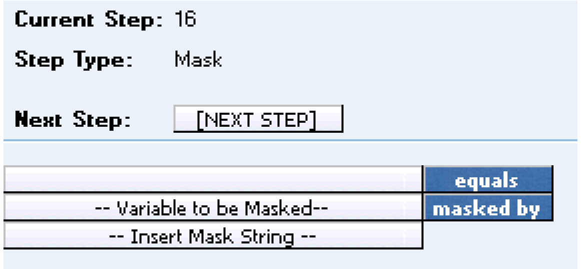
You also can add conditions from the same menu or from the menu on existing terms. Conditions will be populated directly under the statement. The Then and Else segment will be pushed down the list.

Deleting a Condition from an IF Statement

You can delete a condition you do not need. Right click on 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 will be 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.



Current Step: 16

Step Type: Mask

Next Step: [NEXT STEP]

Figure 154 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 155 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 156 Mask Option Examples Table

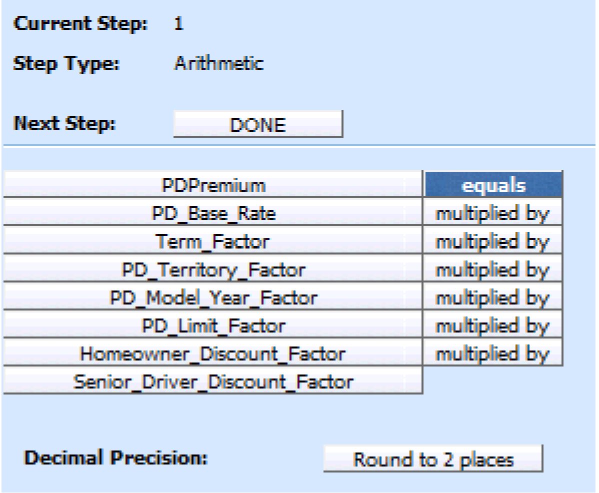
Math Functions

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
- 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.



Current Step: 1

Step Type: Arithmetic

Next Step: DONE

PDPremium	equals
PD_Base_Rate	multiplied by
Term_Factor	multiplied by
PD_Territory_Factor	multiplied by
PD_Model_Year_Factor	multiplied by
PD_Limit_Factor	multiplied by
Homeowner_Discount_Factor	multiplied by
Senior_Driver_Discount_Factor	

Decimal Precision: Round to 2 places

Figure 157 Arithmetic Step

Adding a New Element

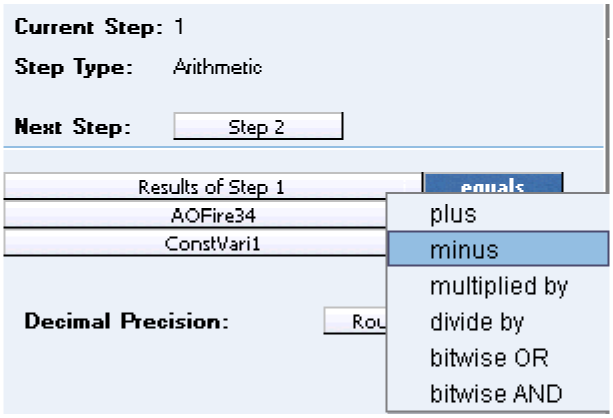
To add a new element, a variable, input, result or constant, to the step, find the item you would like to add from the listing of **Available Variables, Functions and Constants.** Once you have located the correct item, double-click it to add it as the next operand.

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 popup menu.



Current Step: 1

Step Type: Arithmetic

Next Step: Step 2

Figure 158 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

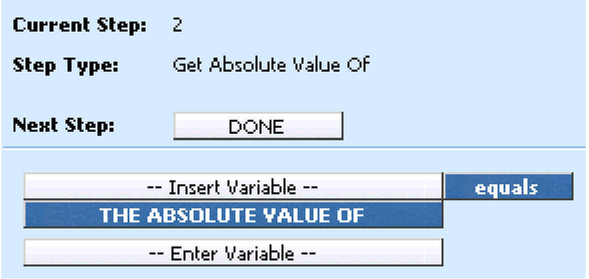
Bitwise AND is determined by converting each numbers to binary form and then performing an AND, bit by bit, where the value returns as a 0 or 1. A 0 value is returned in the result if a 1 is not in every row of the column being evaluated. A 1 is returned if a 1 is in every row of the column being evaluated. 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.



Current Step: 2

Step Type: Get Absolute Value Of

Next Step: DONE

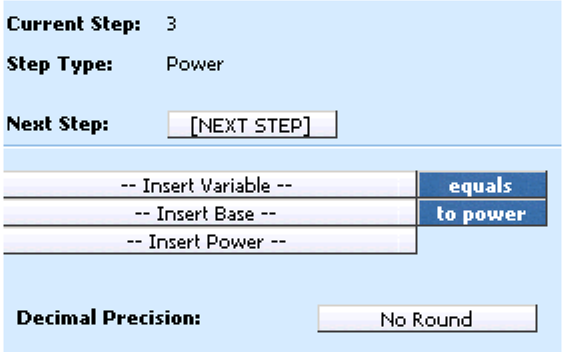
Figure 159 Absolute Value

Power Step

A power step is used when you need 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.



Current Step: 3

Step Type: Power

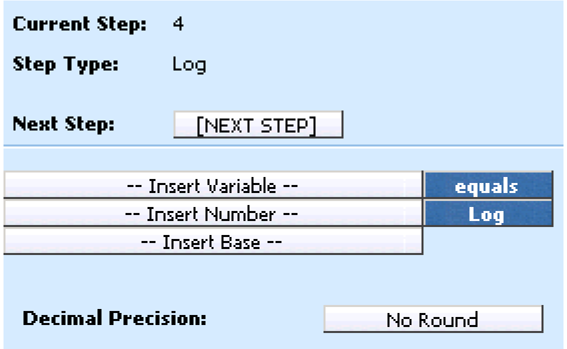
Next Step: [NEXT STEP]

Decimal Precision: No Round

Figure 160 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.



Current Step: 4

Step Type: Log

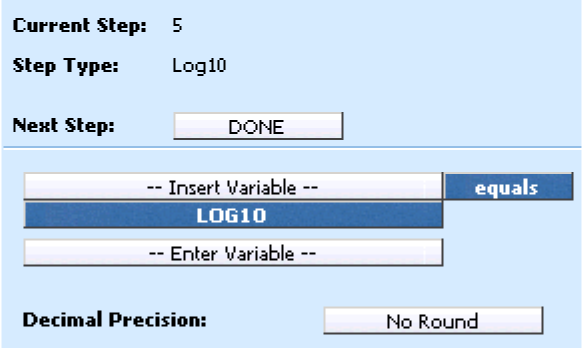
Next Step: [NEXT STEP]

Decimal Precision: No Round

Figure 161 Log Step

Log10 Step

The log10 step returns the base-10 logarithm of a number.



Current Step: 5

Step Type: Log10

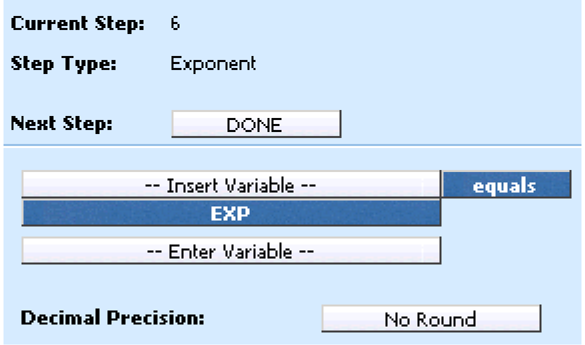
Next Step: DONE

Decimal Precision: No Round

Figure 162 Log10 Step

Exp 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.



Current Step: 6

Step Type: Exponent

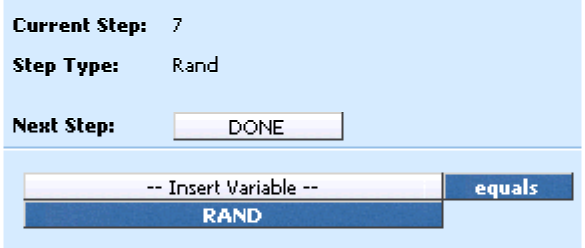
Next Step: DONE

Decimal Precision: No Round

Figure 163 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.



Current Step: 7

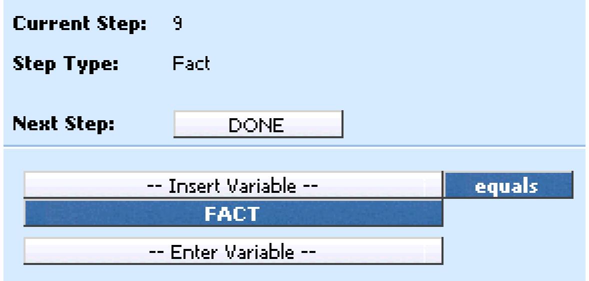
Step Type: Rand

Next Step: DONE

Figure 164 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.



Current Step: 9

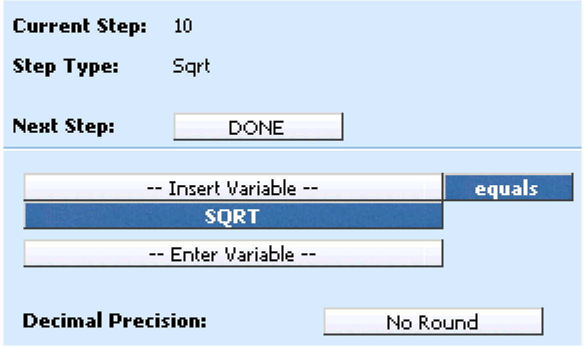
Step Type: Fact

Next Step: DONE

Figure 165 Factorial Step

Sqrt 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.



Current Step: 10

Step Type: Sqrt

Next Step: DONE

Decimal Precision: No Round

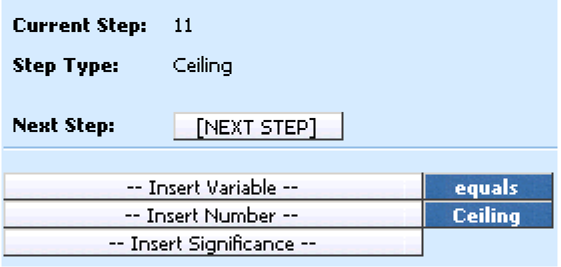
Figure 166 Square Root Step

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.



Current Step: 11

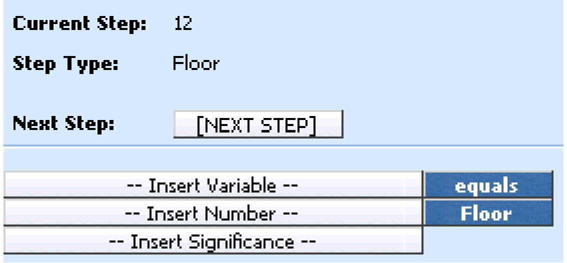
Step Type: Ceiling

Next Step: [NEXT STEP]

Figure 167 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.



Current Step: 12

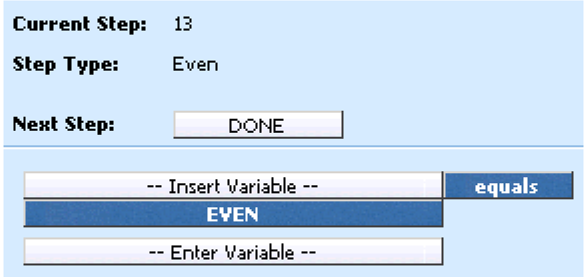
Step Type: Floor

Next Step: [NEXT STEP]

Figure 168 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.



Current Step: 13

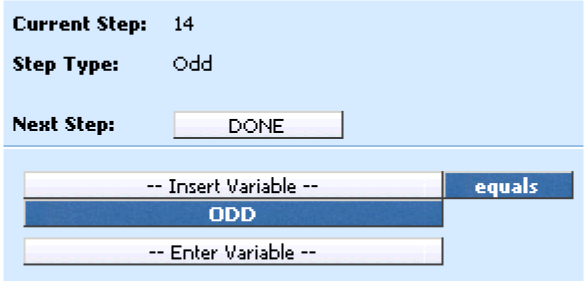
Step Type: Even

Next Step: DONE

Figure 169 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.



Current Step: 14

Step Type: Odd

Next Step: DONE

Figure 170 Odd Step

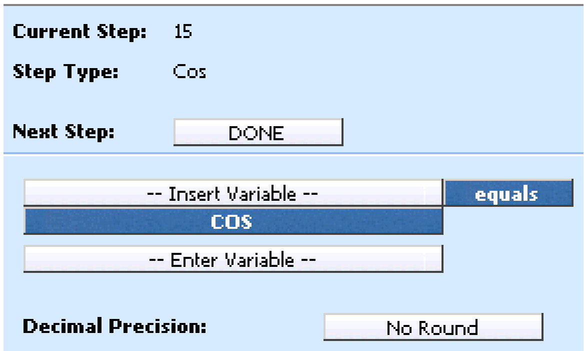
Trig Functions

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.



Current Step: 15

Step Type: Cos

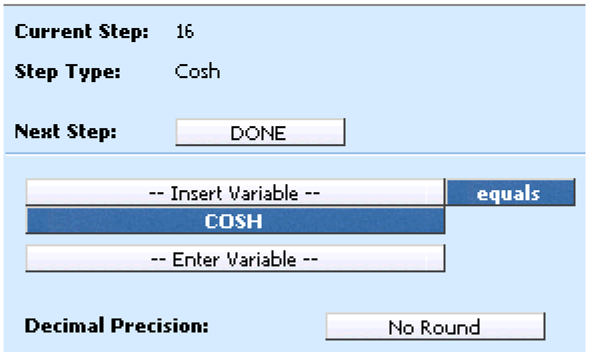
Next Step: DONE

Decimal Precision: No Round

Figure 171 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.



Current Step: 16

Step Type: Cosh

Next Step: DONE

-- Insert Variable -- equals

COSH

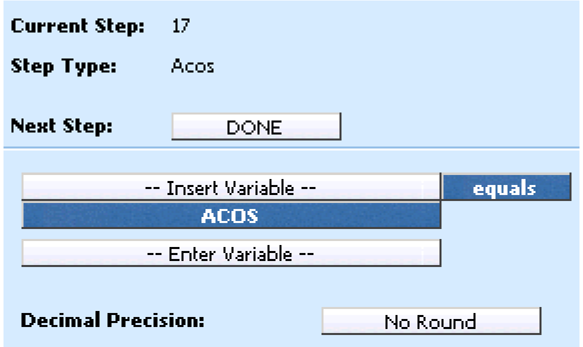
-- Enter Variable --

Decimal Precision: No Round

Figure 172 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.



Current Step: 17

Step Type: Acos

Next Step: DONE

-- Insert Variable -- equals

ACOS

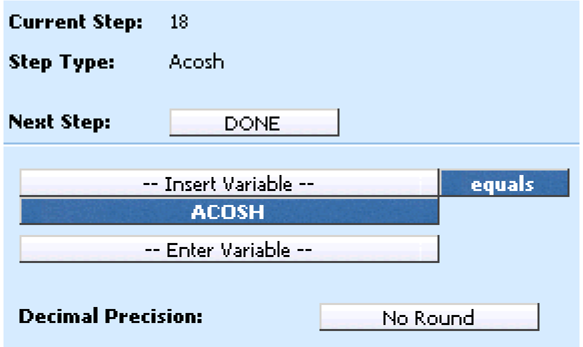
-- Enter Variable --

Decimal Precision: No Round

Figure 173 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.



Current Step: 18

Step Type: Acosh

Next Step: DONE

-- Insert Variable -- equals

ACOSH

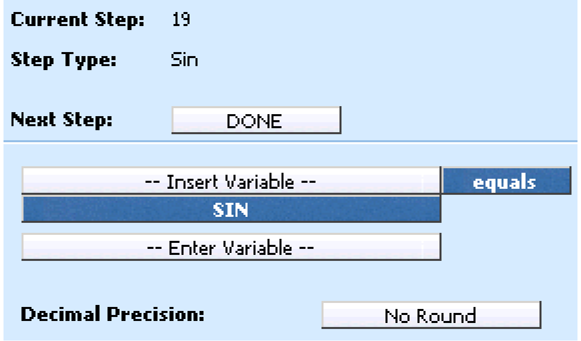
-- Enter Variable --

Decimal Precision: No Round

Figure 174 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.



Current Step: 19

Step Type: Sin

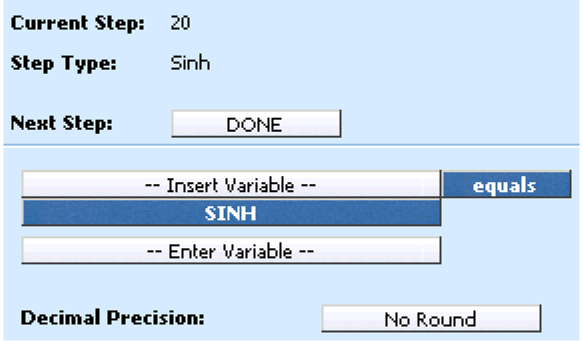
Next Step: DONE

Decimal Precision: No Round

Figure 175 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.



Current Step: 20

Step Type: Sinh

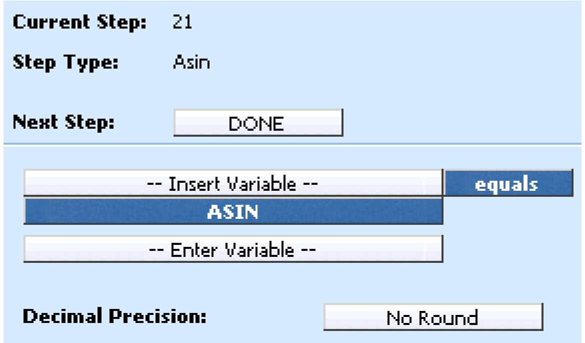
Next Step: DONE

Decimal Precision: No Round

Figure 176 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.



Current Step: 21

Step Type: Asin

Next Step: DONE

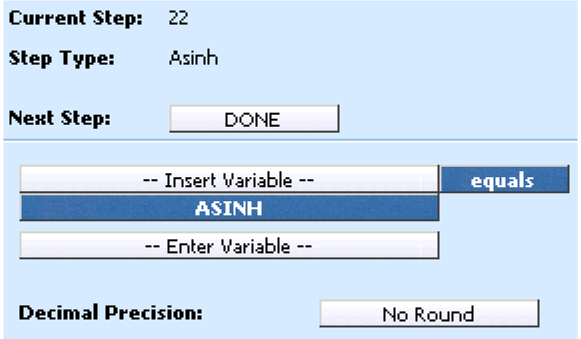
-- Insert Variable -- equals ASIN -- Enter Variable --

Decimal Precision: No Round

Figure 177 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.



Current Step: 22

Step Type: Asinh

Next Step: DONE

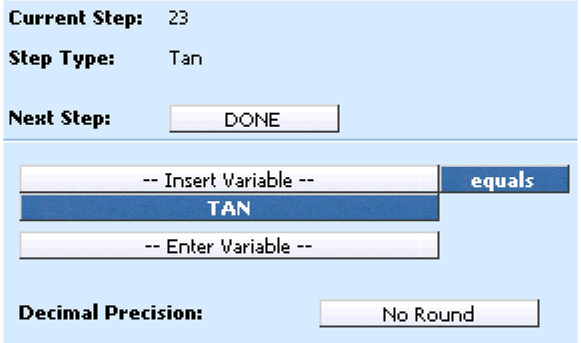
-- Insert Variable -- equals ASINH -- Enter Variable --

Decimal Precision: No Round

Figure 178 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.



Current Step: 23

Step Type: Tan

Next Step: DONE

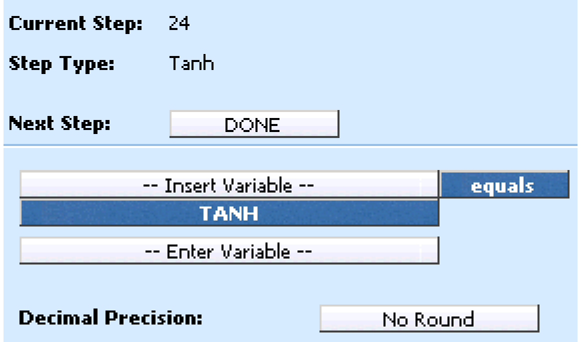
-- Insert Variable -- equals TAN -- Enter Variable --

Decimal Precision: No Round

Figure 179 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.



Current Step: 24
Step Type: Tanh
Next Step: DONE

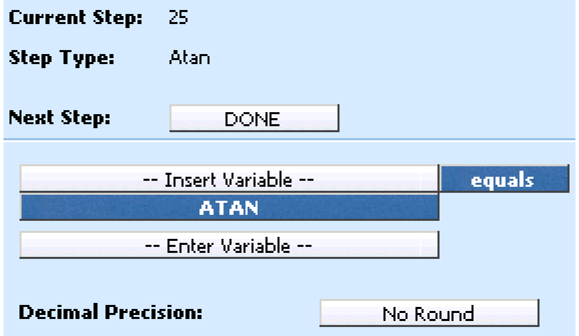
-- Insert Variable -- equals
TANH
-- Enter Variable --

Decimal Precision: No Round

Figure 180 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.



Current Step: 25
Step Type: Atan
Next Step: DONE

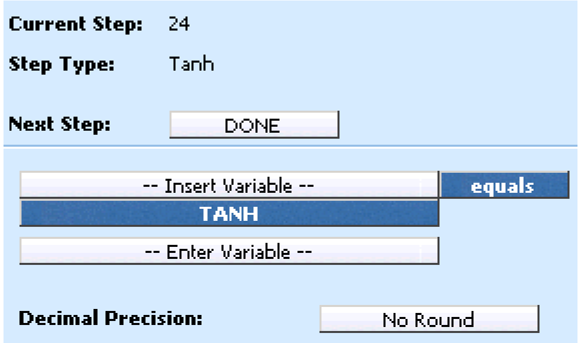
-- Insert Variable -- equals
ATAN
-- Enter Variable --

Decimal Precision: No Round

Figure 181 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.



Current Step: 24

Step Type: Tanh

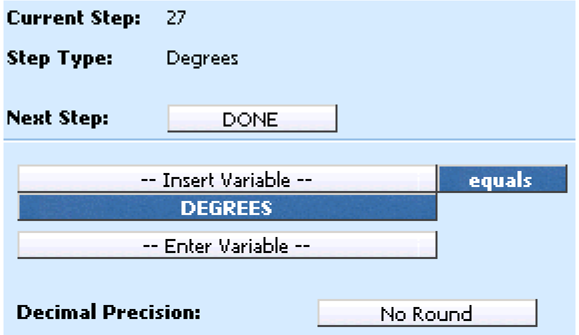
Next Step: DONE

Decimal Precision: No Round

Figure 182 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.



Current Step: 27

Step Type: Degrees

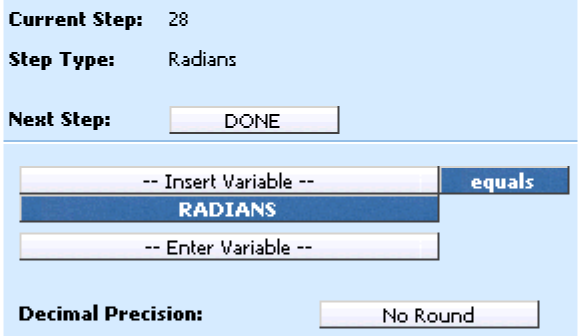
Next Step: DONE

Decimal Precision: No Round

Figure 183 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.



Current Step: 28

Step Type: Radians

Next Step: DONE

Decimal Precision: No Round

Figure 184 Radians Step

Ranking

A Rank Across category step allows you to rank 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:

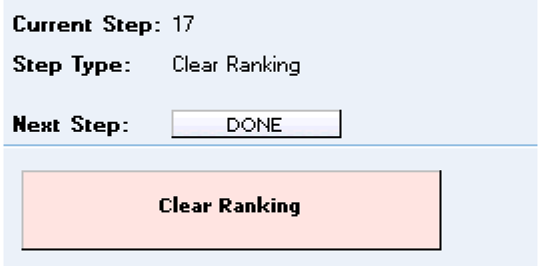
- **Clear Ranking**
- **Rank Across Category**

RateManager provides four types of the Rank across Category Step.

- **Rank across Category (All Available) (Low-to-High):** Allows you to rank elements in any category, lowest to highest.
- **Rank across Category (High-to-Low):** Allows you to rank 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):** Allows you to rank 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):** Allows you to rank 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.



Current Step: 17

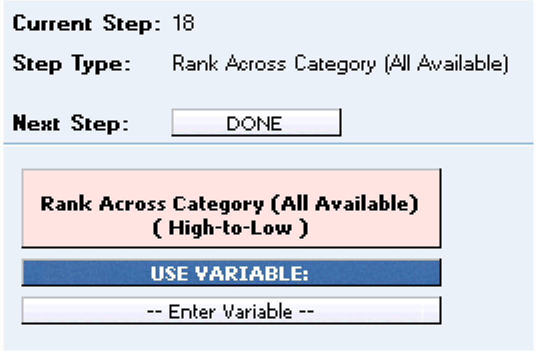
Step Type: Clear Ranking

Next Step: DONE

Figure 185 Clear Ranking Step

Rank across Category Step

A rank across category step allows you to rank all instances of a specific element, either highest to lowest or lowest to highest.



Current Step: 18

Step Type: Rank Across Category (All Available)

Next Step: DONE

Rank Across Category (All Available) (High-to-Low)

USE VARIABLE:

-- Enter Variable --

Figure 186 Rank Across Category Step

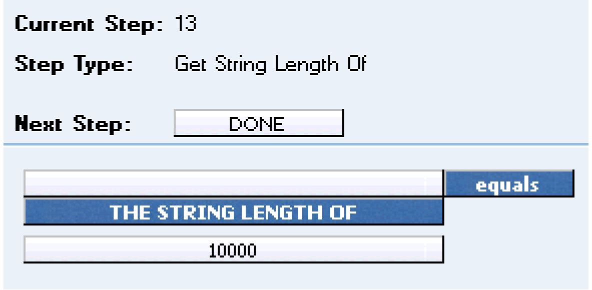
String Functions

String functions can be used any time you need to 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 164, the **Results of the Step** will equal 5, since the constant value of "10000" is 5 characters.



Current Step: 13

Step Type: Get String Length Of

Next Step: DONE

equals

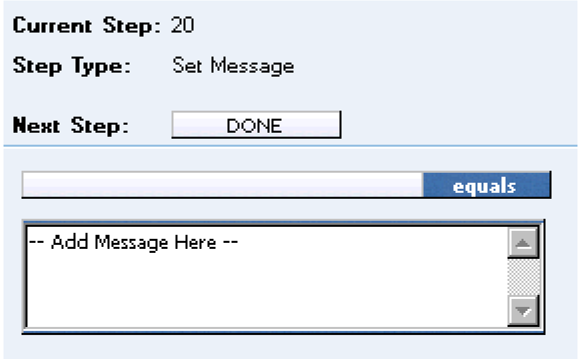
THE STRING LENGTH OF

10000

Figure 187 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 input must be a string data type.



Current Step: 20

Step Type: Set Message

Next Step: DONE

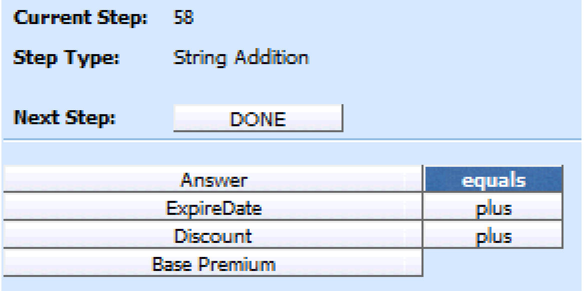
-- Add Message Here --

Figure 188 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 input strings are allowed. Select the term you want to add and double click. The only operator will be plus.



Current Step: 58

Step Type: String Addition

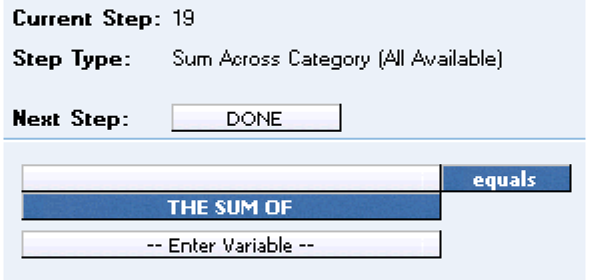
Next Step: DONE

Answer	equals
ExpireDate	plus
Discount	plus
Base Premium	

Figure 189 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 (input, 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.



Current Step: 19

Step Type: Sum Across Category (All Available)

Next Step: DONE

-- Enter Variable --

Figure 190 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: 01/04/2008
               Violation Points: 2
           </Violation>
           <Violation>
               Violation Type: Driving without a license
               Violation Date: 05/30/2008
               Violation Points: 1
           </Violation>
  </Driver>

  <Driver>
         Driver #2
           <Violation>
               Violation Type: At fault accident
               Violation Date: 09/19/2008
               Violation Points: 5
           </Violation>
  </Driver>

  <Driver>
         Driver #3
           <Violation>
               Violation Type: Failure to signal
               Violation Date: 12/02/2009
               Violation Points: 1
           </Violation>
           <Violation>
               Violation Type: Speeding
               Violation Date: 01/21/2008
               Violation Points: 2
           </Violation>
           <Violation>
               Violation Type: Running a stop sign
               Violation Date: 11/14/2008
               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: 01/04/2008
               Violation Points: 2
          </Violation>
          <Violation>
               Violation Type: Driving without a license
               Violation Date: 05/30/2008
               Violation Points: 1
          </Violation>
</Driver>

<Driver>
        Driver #2
          <Violation>
               Violation Type: At fault accident
               Violation Date: 09/19/2008
               Violation Points: 5
          </Violation>
</Driver>

<Driver>
        Driver #3
          <Violation>
               Violation Type: Failure to signal
               Violation Date: 12/02/2009
               Violation Points: 1
          </Violation>
          <Violation>
               Violation Type: Speeding
               Violation Date: 01/21/2008
               Violation Points: 2
          </Violation>
          <Violation>
               Violation Type: Running a stop sign
               Violation Date: 11/14/2008
               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.

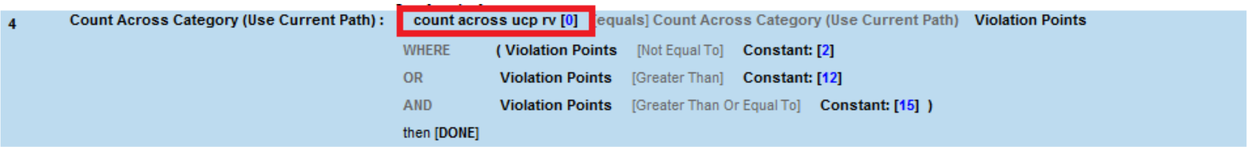
Total Policy Violation Points [=] The Sum (Across All) of Violation Points

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.



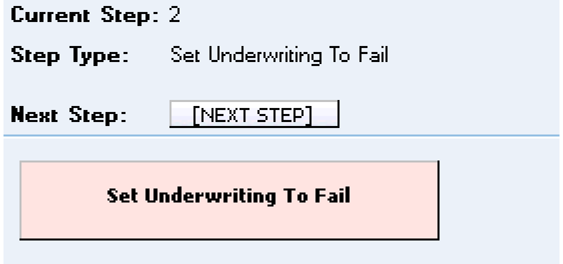
```
4    Count Across Category (Use Current Path) :  count across ucp rv [0]  equals] Count Across Category (Use Current Path)   Violation Points
                                                 WHERE   ( Violation Points   [Not Equal To]              Constant: [2]
                                                 OR        Violation Points   [Greater Than]              Constant: [12]
                                                 AND       Violation Points   [Greater Than Or Equal To]  Constant: [15]  )
                                                 then [DONE]
```

The zero value is used in calculations.

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 line of business, 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.



Current Step: 2

Step Type: Set Underwriting To Fail

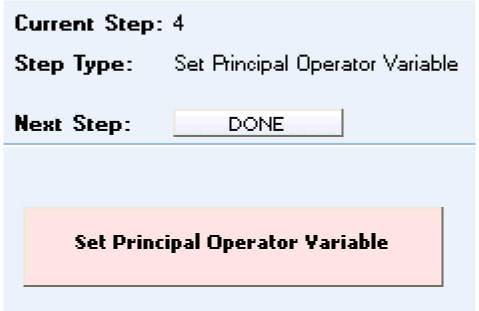
Next Step: [NEXT STEP]

Figure 191 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.

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.



Current Step: 4

Step Type: Set Principal Operator Variable

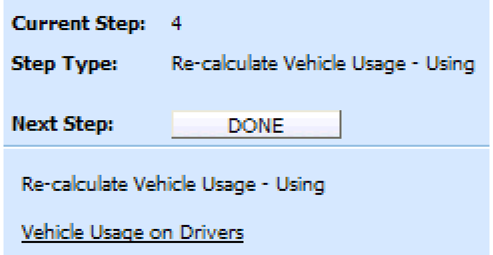
Next Step: DONE

Figure 192 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.



Current Step: 4

Step Type: Re-calculate Vehicle Usage - Using

Next Step: DONE

Re-calculate Vehicle Usage - Using

Vehicle Usage on Drivers

Figure 193 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 (Input: 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 (Inputs: 299/368):** Vehicle usage is designated as the actual vehicle a driver principally operates, based on the vehicle number specified in the input file.

Chapter 15

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 inputs 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 two levels:

Global: Global variables are created and managed at the global level and are subline specific. Global variables can be accessed from the Global tab or from the Program Details tab. When selected, variables will be listed in the Folder Details area.

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. When selected, variables will be listed in the Folder Details area.

There are a couple of differences between globals and locals. One difference between them is that global variables are available to all programs within the subline while program level variables are available only to the program where they were created. Another is that Global variables can use only global elements. Local variables can use either local or global elements.

Both Global and Local calculated variables can be template generated or created by you. Variables that are user created will be 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 will be 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 will be 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 two paths to view and manage variables, either from the Globals tab or from the Program Details tab. The Global tab will display global variables only. You cannot view or manage local variables on the Global 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 will be listed in the Folder Details area.

Calculated variables are not an option on the Program Explorer tab.

1. Modules→Program Explorer. This opens the Program Explorer tab.

2. Select the Line of Business→Subline→Folder→Program → Program Version. You also can access calculated variables from the globals tab. Line of Business→Subline→Globals.

3. Open the program version to get the Program Details tab. Calculated Variables are available from the program details folders area on the left hand side of the tab.

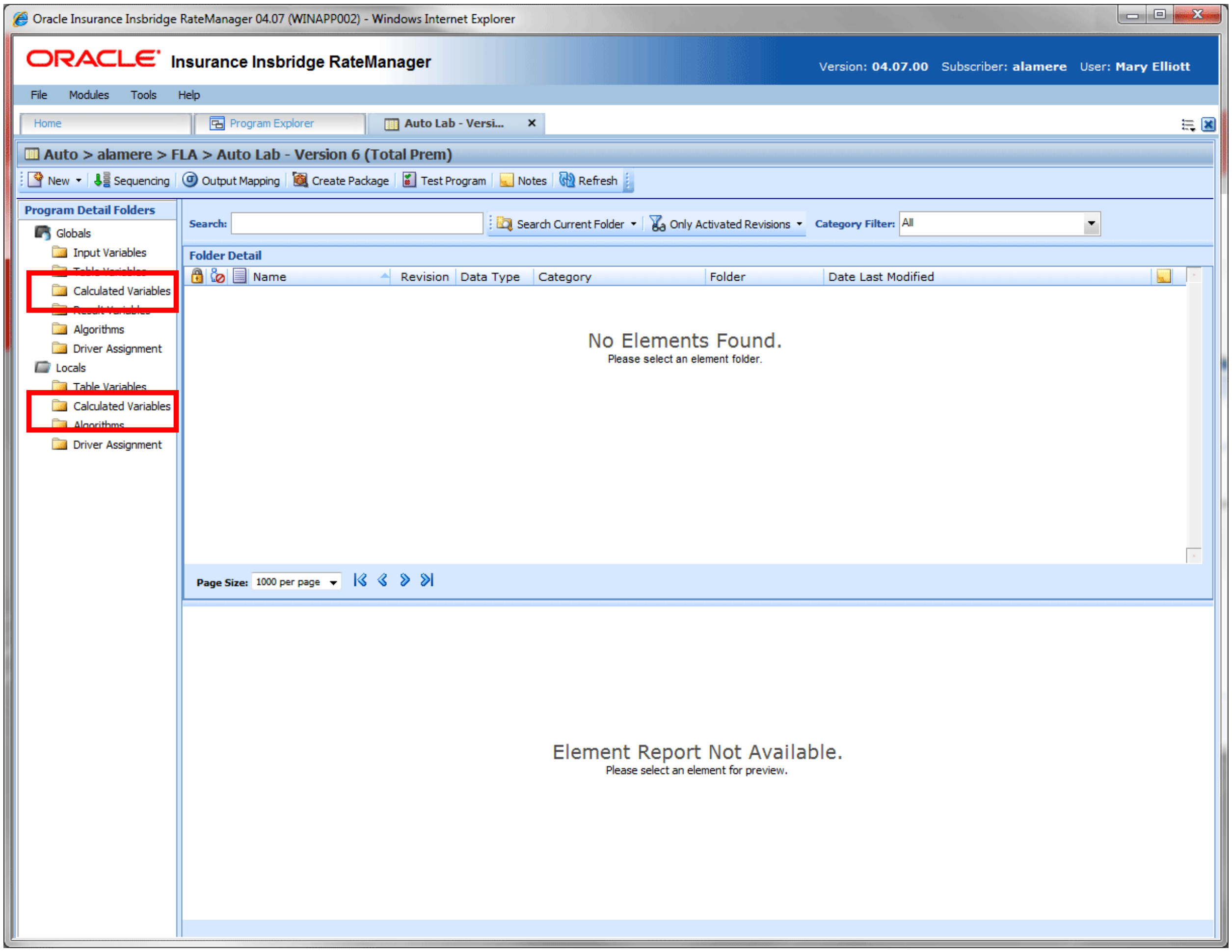

Oracle Insurance Insbridge RateManager 04.07 (WINAPP002) - Windows Internet Explorer
ORACLE Insurance Insbridge RateManager
Version: 04.07.00 Subscriber: alamere User: Mary Elliott
File Modules Tools Help
Home
Program Explorer
Auto Lab - Versi...
Auto > alamere > FLA > Auto Lab - Version 6 (Total Prem)
New Sequencing Output Mapping Create Package Test Program Notes Refresh
Program Detail Folders
Globals
Input Variables
Table Variables
Calculated Variables
Result Variables
Algorithms
Driver Assignment
Locals
Table Variables
Calculated Variables
Algorithms
Driver Assignment
Search:
Search Current Folder
Only Activated Revisions
Category Filter: All
Folder Detail
Name Revision Data Type Category Folder Date Last Modified
No Elements Found.
Please select an element folder.
Page Size: 1000 per page
Element Report Not Available.
Please select an element for preview.

Figure 194 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 Program Details tab using a right click menu. Reports also can be generated using the right click menu on the Program Details tab.
- **Calculated Variable Revision Functions:** You will be able to view, copy, and delete calculated variable revisions. Active revisions can be set on the Program Details tab as well.
- **Calculated Variable Tab:** When you create a new calculated variable, edit an existing variable, or work with revisions, you will be placed on the Calculated Variable tab.

The functions you can perform will depend on the status of the calculated variable. Locked calculated variables will have fewer options. There are icons that can tell you at a glance what the status of the calculated variable is.

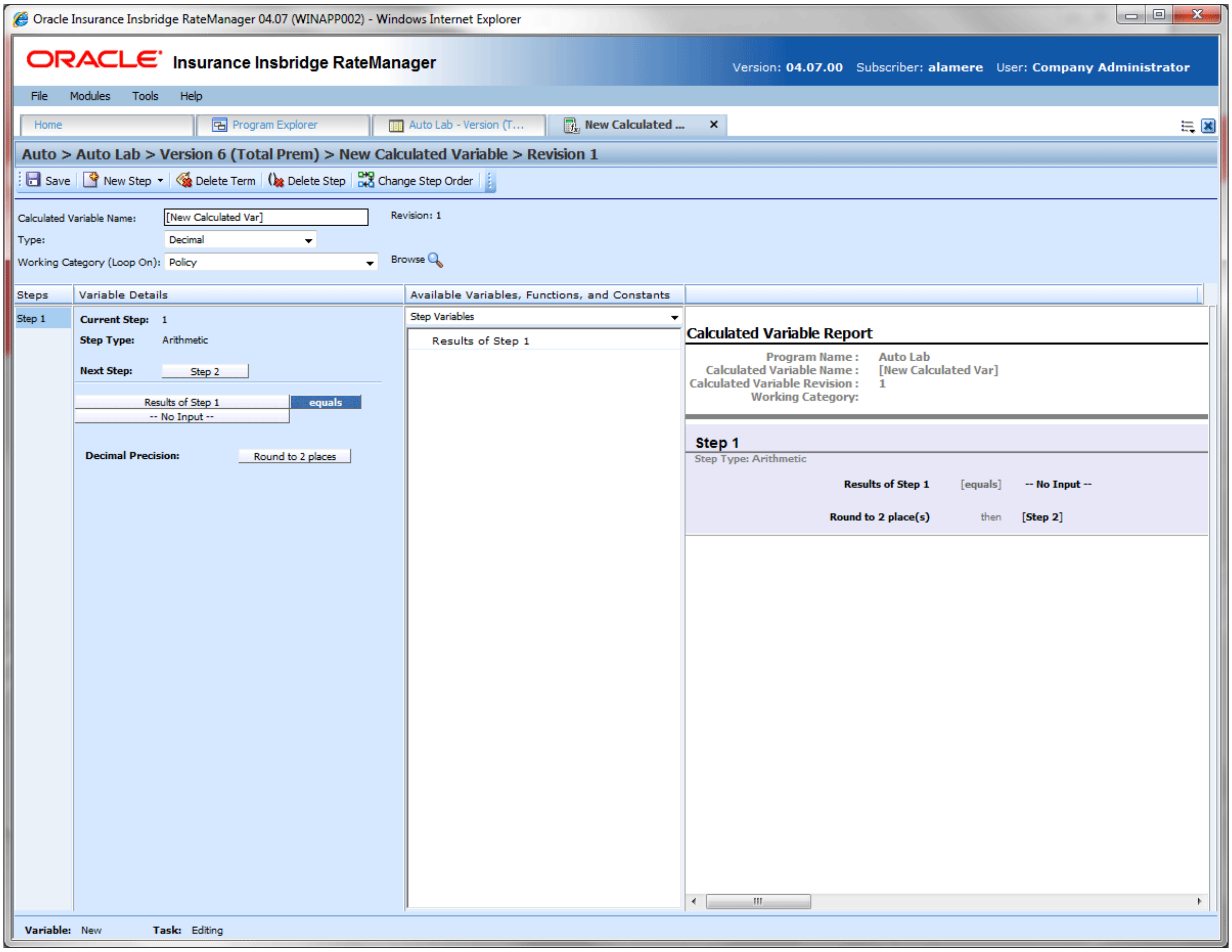
Icon	What it Indicates
fx	**Calculated Variable.** Global Calculated Variables are subline specific and can be created at the global level, or from a program version. Local Calculated Variables are program specific and can be created at the program version level. Local Calculated Variables are displayed on the Program Details→Local Calculated Variables. The same variable listings are also available for Subline→Globals tab. The same options, new, edit, copy and delete are available on either tab.
🔒	**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 will unlock the calculated variable and changes can be made.
🔒	**Template Lock.** If the calculated variable is locked because it came from a template, you will not be able to 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 Fields on Program Details Tab:

The fields are the same for the Global Tab and Program Details. For more on field descriptions, please see Program Details→Fields.

CALCULATED VARIABLE TAB

The Calculated Variable tab will be 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.


Oracle Insurance Insbridge RateManager 04.07 (WINAPP002) - Windows Internet Explorer
ORACLE Insurance Insbridge RateManager
Version: 04.07.00 Subscriber: alamere User: Company Administrator
File
Modules
Tools
Help
Home
Program Explorer
Auto Lab - Version (T...
New Calculated ...
Auto > Auto Lab > Version 6 (Total Prem) > New Calculated Variable > Revision 1
Save
New Step
Delete Term
Delete Step
Change Step Order
Browse
Steps
Variable Details
Available Variables, Functions, and Constants
Step 1
Step Variables
Results of Step 1

Calculated Variable Name: [New Calculated Var]
Revision: 1
Type: Decimal
Working Category (Loop On): Policy

Current Step: 1
Step Type: Arithmetic
Next Step: Step 2

Results of Step 1 equals
-- No Input --

Decimal Precision: Round to 2 places

Calculated Variable Report

Program Name : Auto Lab
Calculated Variable Name : [New Calculated Var]
Calculated Variable Revision : 1
Working Category:

Step 1
Step Type: Arithmetic

Results of Step 1 [equals] -- No Input --
Round to 2 place(s) then [Step 2]

Variable: New Task: Editing

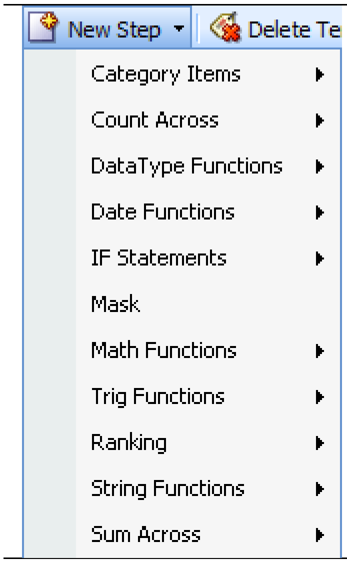
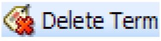
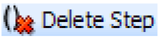
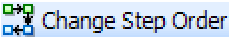
Figure 195 Calculated Variable Tab

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 will always be at least one step. You will not be able to 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 Delete Te Category Items Count Across DataType Functions Date Functions IF Statements Mask Math Functions Trig Functions Ranking String Functions Sum Across	**New Step:** Creates a new step in the calculated variable. For more information, see Working with Steps.
	**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.

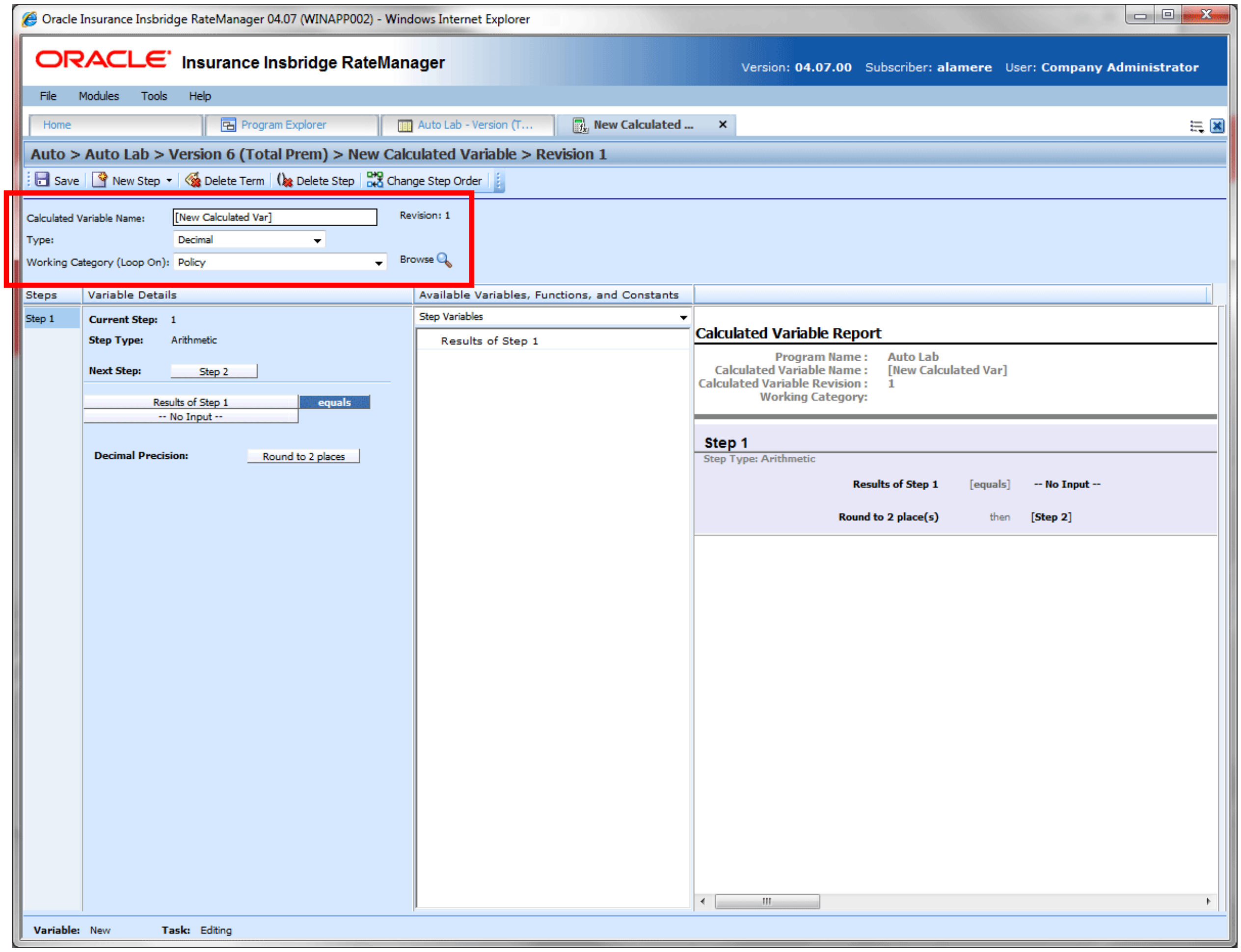
CALCULATED VARIABLE FIELDS

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.


Oracle Insurance Insbridge RateManager 04.07 (WINAPP002) - Windows Internet Explorer
ORACLE Insurance Insbridge RateManager
Version: 04.07.00 Subscriber: alamere User: Company Administrator
File Modules Tools Help
Home
Program Explorer
Auto Lab - Version (T...
New Calculated ...
Auto > Auto Lab > Version 6 (Total Prem) > New Calculated Variable > Revision 1
Save
New Step
Delete Term
Delete Step
Change Step Order
Browse
Steps
Variable Details
Available Variables, Functions, and Constants
Step 1
Step Variables
Results of Step 1
Variable: New Task: Editing

Calculated Variable Name: [New Calculated Var]
Type: Decimal
Working Category (Loop On): Policy
Revision: 1

Current Step: 1
Step Type: Arithmetic
Next Step: Step 2
Results of Step 1 equals -- No Input --
Decimal Precision: Round to 2 places

Calculated Variable Report

Program Name : Auto Lab
Calculated Variable Name : [New Calculated Var]
Calculated Variable Revision : 1
Working Category:

Step 1
Step Type: Arithmetic

Results of Step 1 [equals] -- No Input --
Round to 2 place(s) then [Step 2]

Figure 196 Calculated Variable Information Field

Fields:

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.

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.

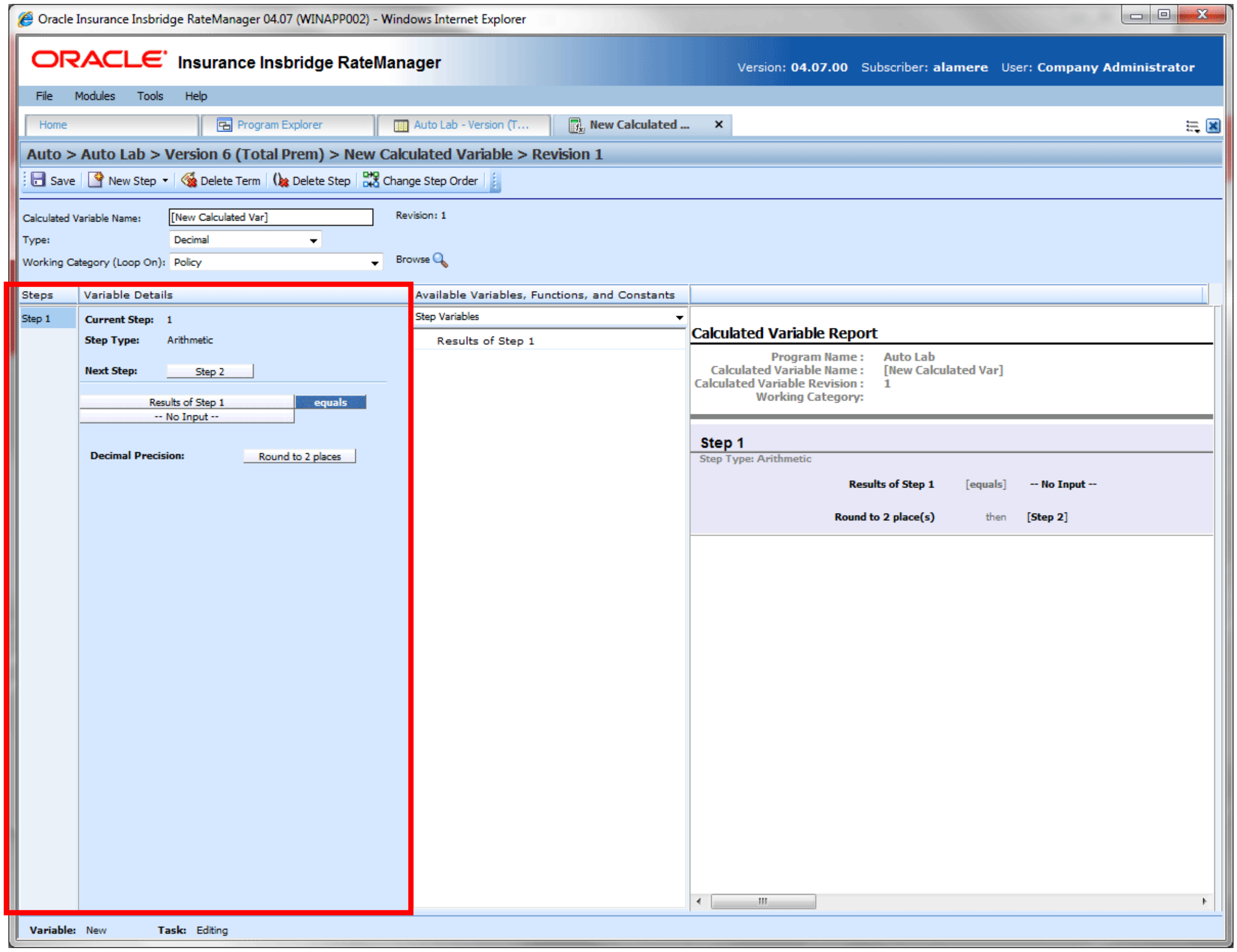
You can select the working category from the drop down menu or you can browse for a working category by clicking **BROWSE**. This will bring up the Browse Category popup. From here you can navigate to the Category you want. Once you have selected the category, click OK. The category will auto fill in 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 Inputs, Policy Variables and Policy Results. If you select Vehicle, the options will now include Vehicle Inputs, Vehicle Variable and Vehicle Results along with Constants, Step Variables, Policy Inputs, Policy Variables and Policy Results.

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.


Oracle Insurance Insbridge RateManager 04.07 (WINAPP002) - Windows Internet Explorer
ORACLE Insurance Insbridge RateManager
Version: 04.07.00 Subscriber: alamere User: Company Administrator
File
Modules
Tools
Help
Home
Program Explorer
Auto Lab - Version (T...
New Calculated ...
Save
New Step
Delete Term
Delete Step
Change Step Order
Browse
Steps
Variable Details
Available Variables, Functions, and Constants
Step Variables
Results of Step 1

Auto > Auto Lab > Version 6 (Total Prem) > New Calculated Variable > Revision 1

Calculated Variable Name: [New Calculated Var]
Revision: 1
Type: Decimal
Working Category (Loop On): Policy

Step 1

Current Step: 1
Step Type: Arithmetic
Next Step: Step 2

Results of Step 1 equals
-- No Input --

Decimal Precision: Round to 2 places

Calculated Variable Report

Program Name : Auto Lab
Calculated Variable Name : [New Calculated Var]
Calculated Variable Revision : 1
Working Category:

Step 1
Step Type: Arithmetic

Results of Step 1 [equals] -- No Input --
Round to 2 place(s) then [Step 2]

Variable: New Task: Editing

Figure 197 Calculated Variable Selection Criteria Field

Fields

Steps: Sequential order of steps created with drag-and-drop capabilities for moving a selected step up or down in the sequence.

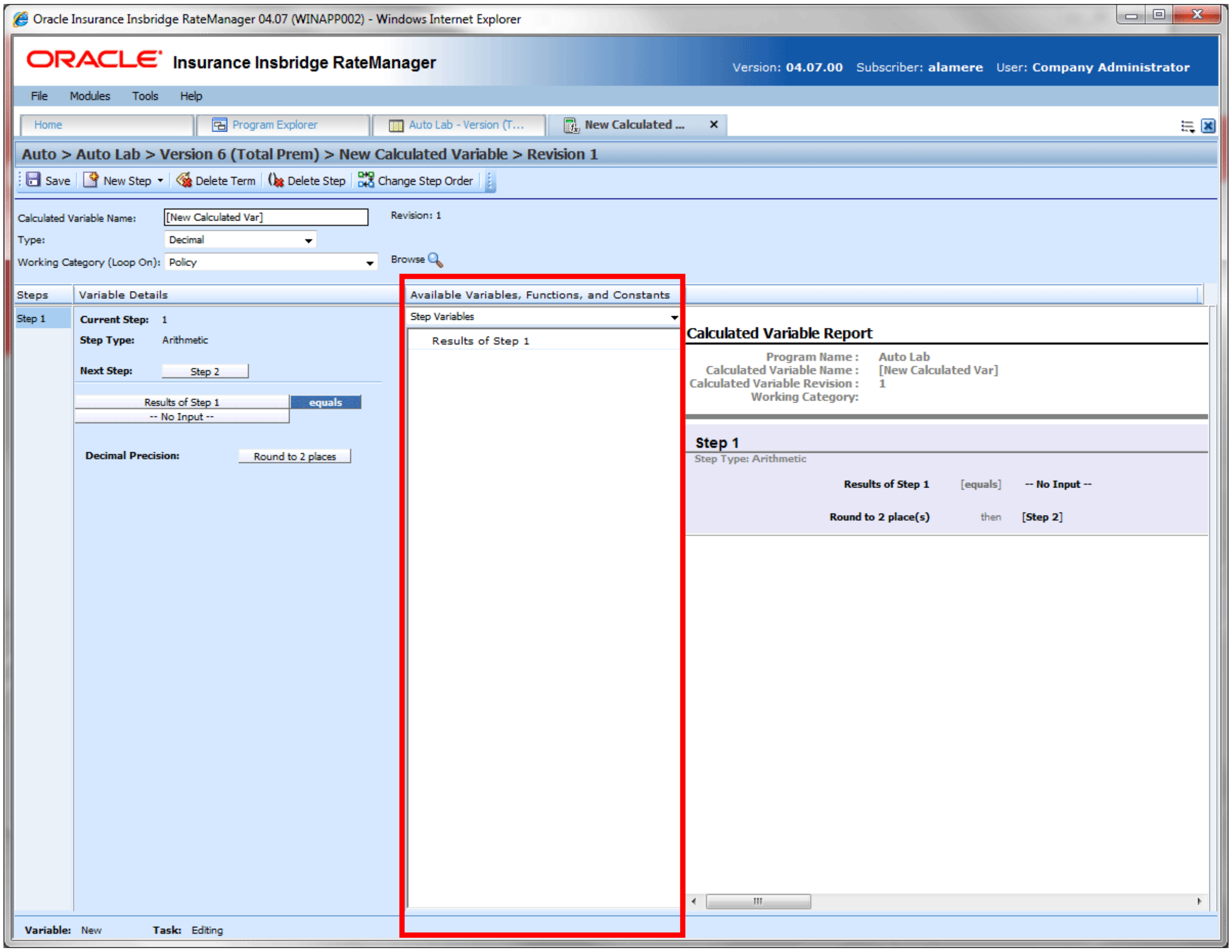
Variable Details: Shows the selected step number, step type selected, next step to follow and inputs/variables used for calculation. Rounding is set by clicking on the text box next to **Decimal Precision**. You can select a rounding precision from 0 to 9 digits.

Inputs 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 inputs, variables, and results, sorted by category and system constants that can be used as criteria. For global variables, only other global variables or inputs 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 Inputs, you will be able to add a new input variable without leaving the Edit Calculated Variable tab. Scroll to the bottom of the available variables list and select <New Input Variable>. A popup will be displayed that will allow you to enter a new input variable.


Oracle Insurance Insbridge RateManager 04.07 (WINAPP002) - Windows Internet Explorer
ORACLE Insurance Insbridge RateManager
Version: 04.07.00 Subscriber: alamere User: Company Administrator
File
Modules
Tools
Help
Home
Program Explorer
Auto Lab - Version (T...
New Calculated ...
Auto > Auto Lab > Version 6 (Total Prem) > New Calculated Variable > Revision 1
Save
New Step
Delete Term
Delete Step
Change Step Order
Browse
Steps
Step 1
Variable Details
Available Variables, Functions, and Constants
Step Variables
Results of Step 1

Calculated Variable Name: [New Calculated Var]

Type: Decimal

Working Category (Loop On): Policy

Revision: 1

Current Step: 1

Step Type: Arithmetic

Next Step: Step 2

Results of Step 1 equals -- No Input --

Decimal Precision: Round to 2 places

Calculated Variable Report

Program Name : Auto Lab
Calculated Variable Name : [New Calculated Var]
Calculated Variable Revision : 1
Working Category:

Step 1
Step Type: Arithmetic

Results of Step 1 [equals] -- No Input --

Round to 2 place(s) then [Step 2]

Variable: New Task: Editing

Figure 198 Available Variables, Functions and Constants Field

Fields

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 Variable Information area.

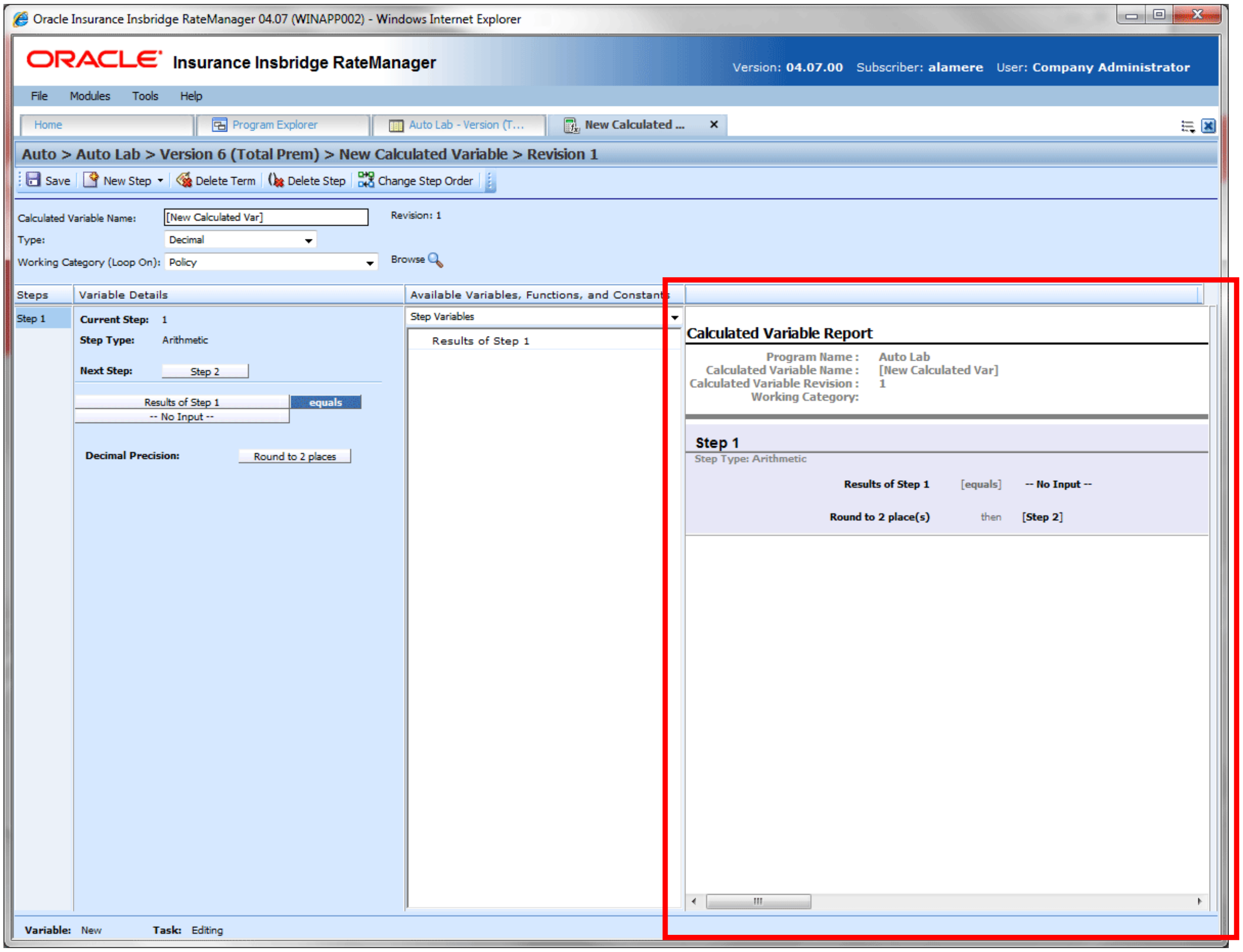
Selection: The available variables, constants and step results will be listed. Variables will be based on the working category you select. Policy level inputs variables and results will always be listed. If you select another category, that categories inputs, variables and results will be listed.

Selection	Description
Constants	Selection of system constants, category instances and system date and time. Or you can create a custom value for use in this program.
Step Variables	Results from the previously entered steps.
Policy Inputs	Inputs 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 inputs, 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.


Oracle Insurance Insbridge RateManager 04.07 (WINAPP002) - Windows Internet Explorer
ORACLE Insurance Insbridge RateManager
Version: 04.07.00 Subscriber: alamere User: Company Administrator
File
Modules
Tools
Help
Home
Program Explorer
Auto Lab - Version (T...
New Calculated ...
Auto > Auto Lab > Version 6 (Total Prem) > New Calculated Variable > Revision 1
Save
New Step
Delete Term
Delete Step
Change Step Order
Browse
Steps
Step 1
Variable Details
Available Variables, Functions, and Constants
Step Variables
Results of Step 1

Calculated Variable Name: [New Calculated Var]

Revision: 1

Type: Decimal

Working Category (Loop On): Policy

Current Step: 1

Step Type: Arithmetic

Next Step: Step 2

Results of Step 1 equals

-- No Input --

Decimal Precision: Round to 2 places

Calculated Variable Report

Program Name :	Auto Lab
Calculated Variable Name :	[New Calculated Var]
Calculated Variable Revision :	1
Working Category:	

Step 1

Step Type: Arithmetic

Results of Step 1	[equals]	-- No Input --
Round to 2 place(s)	then	[Step 2]

Variable: New Task: Editing

Figure 199 Calculated Variable Report

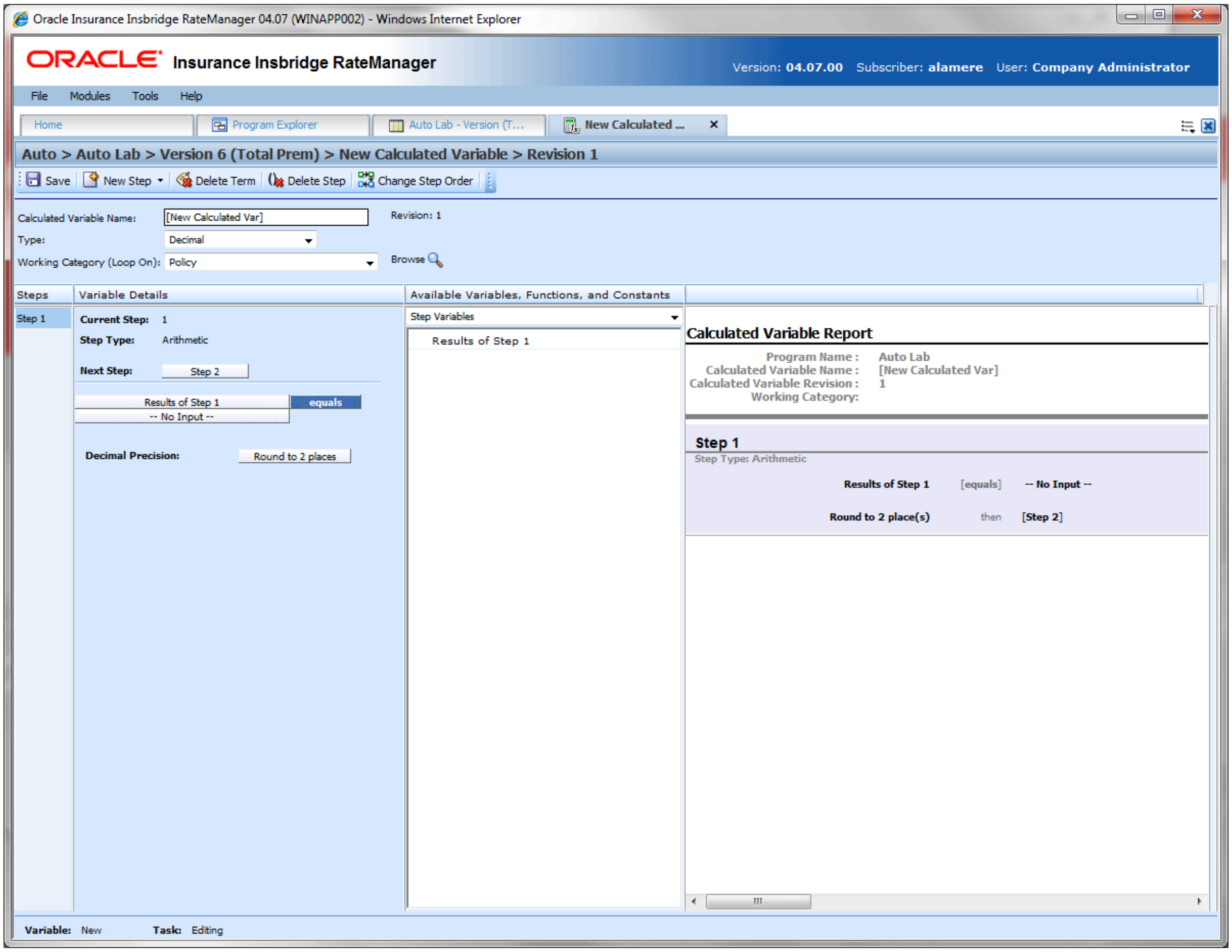
CREATING A CALCULATED VARIABLE

Calculated variables can be created and added to standard programs and template generated programs. Local calculated variables will be available in the program where you created it and can be copied to other programs within the subline. Global calculated variables will be available across the subline.

Local and global calculated variables are created and function in much the same way. One difference is that global calculated variables can use only global table variables, inputs and result variables. Local calculated variables can use both local and global elements. Another difference is that global calculated variables can be created from the globals tab and from the program details tab.

1. On the Program Details tab of the program where you want to create a new variable, select either the Global Calculated Variables folder or the Local Calculated Variables folder.

2. Click **New** in the top bar menu. If you do not select a folder, you can select New→Globals→Calculated Variable or New→Locals→Calculated Variables.


Oracle Insurance Insbridge RateManager 04.07 (WINAPP002) - Windows Internet Explorer
ORACLE Insurance Insbridge RateManager
Version: 04.07.00 Subscriber: alamere User: Company Administrator
File
Modules
Tools
Help
Home
Program Explorer
Auto Lab - Version (T...
New Calculated ...
Auto > Auto Lab > Version 6 (Total Prem) > New Calculated Variable > Revision 1
Save
New Step
Delete Term
Delete Step
Change Step Order
Browse
Steps
Step 1
Variable Details
Available Variables, Functions, and Constants
Step Variables
Results of Step 1

Calculated Variable Name: [New Calculated Var]

Type: Decimal

Working Category (Loop On): Policy

Revision: 1

Current Step: 1

Step Type: Arithmetic

Next Step: Step 2

Results of Step 1 equals -- No Input --

Decimal Precision: Round to 2 places

Calculated Variable Report

Program Name : Auto Lab
Calculated Variable Name : [New Calculated Var]
Calculated Variable Revision : 1
Working Category:

Step 1
Step Type: Arithmetic

Results of Step 1 [equals] -- No Input --

Round to 2 place(s) then [Step 2]

Variable: New Task: Editing

Figure 200 Creating a New Calculated Variable

3. This opens the **Edit Variable** tab for the new variable where you can enter or select the variable elements you need.

4. Enter a **Name** for the calculated variable.

5. Select the **type** of data associated with the variable. Options are:
   - **Integer**
   - **String**
   - **Decimal**
   - **Date**

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 popup.

   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.

   If this is not the screen you want, click **Cancel** to close the popup 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.*

7. The first step will be 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 will refresh with the new step parameters. Depending upon the step type, you will have to select the appropriate variables, functions, constants or inputs. The value you need to select will be 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 will be populated underneath. Click the “Enter Variable” box and then double-click the corresponding variable you want. The box will refresh with your selected value.

10. Continue entering steps until complete. When you are finished making your entries, click **Save**.

Adding an Input Variable or a Custom Value Constant

A new input variable, result variable or custom value constant can be created while entering or editing a calculated variable without leaving the Edit tab.

- To add an **Input Variable**: See Adding a New Input from Another Area
- To add a **Custom Value Constant**: See Constants

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 will be 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 will open 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 will be 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 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.

COPYING A CALCULATED VARIABLE

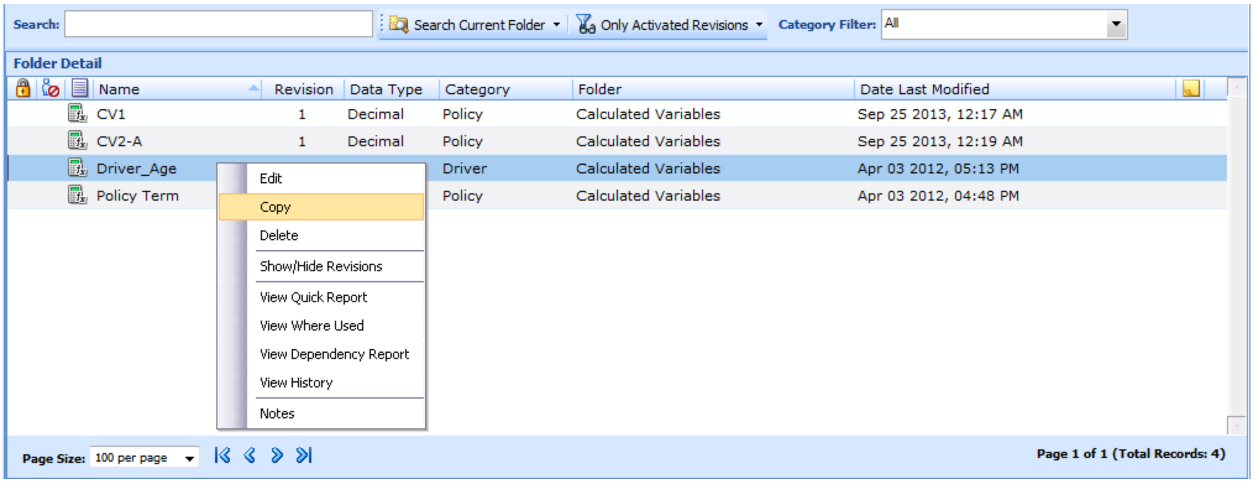
Calculated variables may be copied within any program. In your default subline, you can copy local calculated variables to another program within the same subline. Template sublines will allow you to copy within the same program only. When copying a local calculated variable, there are two copy options:

- **Existing Program:** Copy the variable into the existing program under another name.
- **Other Program:** Copy the variable into another program within the same subline. This option is for calculated variables within your default subline.

Global calculated variables will not have any copy options.

To Copy a Calculated Variable

1. Select the calculated variable you want to copy and right click it. Select **Copy** from the menu.



Search: Search Current Folder Only Activated Revisions Category Filter: All

Folder Detail

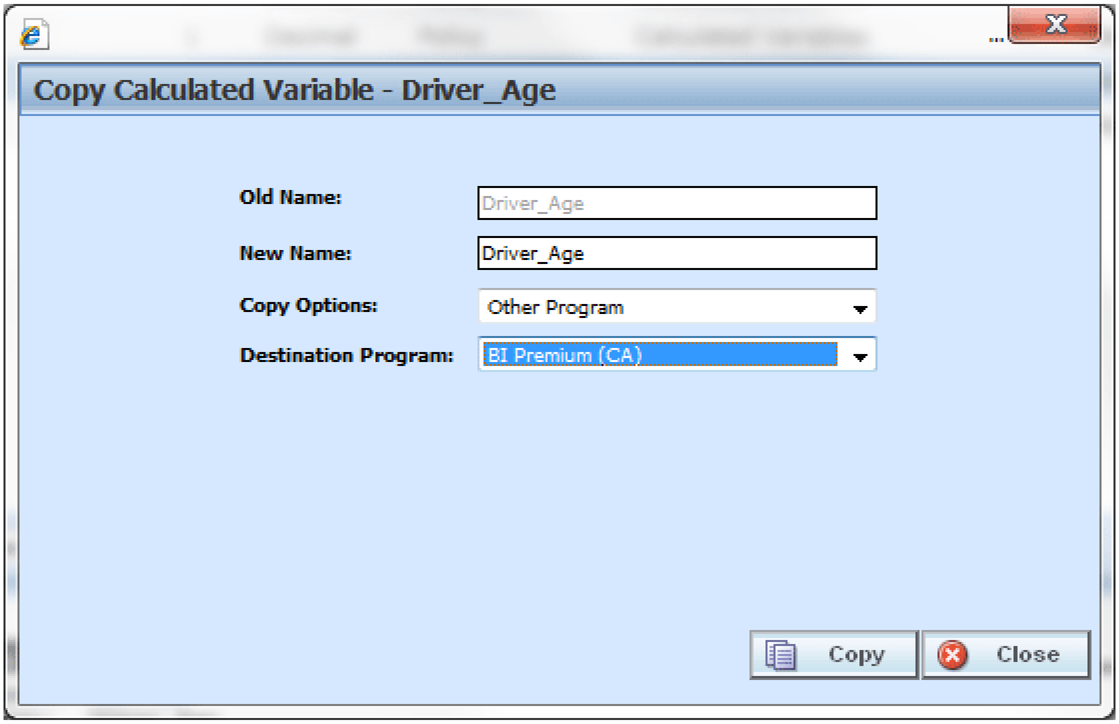
Name	Revision	Data Type	Category	Folder	Date Last Modified
CV1	1	Decimal	Policy	Calculated Variables	Sep 25 2013, 12:17 AM
CV2-A	1	Decimal	Policy	Calculated Variables	Sep 25 2013, 12:19 AM
Driver_Age			Driver	Calculated Variables	Apr 03 2012, 05:13 PM
Policy Term			Policy	Calculated Variables	Apr 03 2012, 04:48 PM

Page Size: 100 per page

Page 1 of 1 (Total Records: 4)

Figure 201 Copy Variable Menu

2. A separate popup window will be displayed.



Copy Calculated Variable - Driver_Age

Old Name: Driver_Age

New Name: Driver_Age

Copy Options: Other Program

Destination Program: BI Premium (CA)

Figure 202 Copy Calculated Variable Popup

3. Select the **Copy Options**. It is recommended that you change the copy options first. Changing options may reset the name field.

4. If you are copying a calculated variable within your default subline, you can select to copy the variable to another program within the same subline. Select the **Destination Program** from the drop down menu.

   If you are in a template generated subline, this option will be grayed out.

   If you are in a global table variable, this option will not be displayed.

5. 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.

6. Click **Copy** to copy the variable. Dependencies will be searched. If there are any dependencies that need to be resolved, a popup window will be displayed. See Resolving Dependencies.

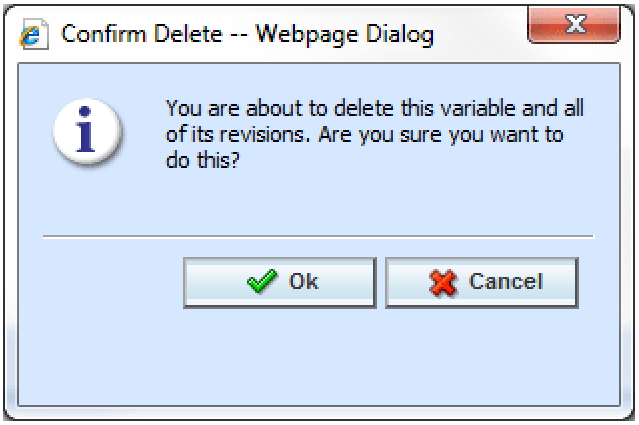
7. If the copy was successful, you will be returned to the previous screen. Your variable will have been copied. The last updated date will be the date the copy was made.

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**.



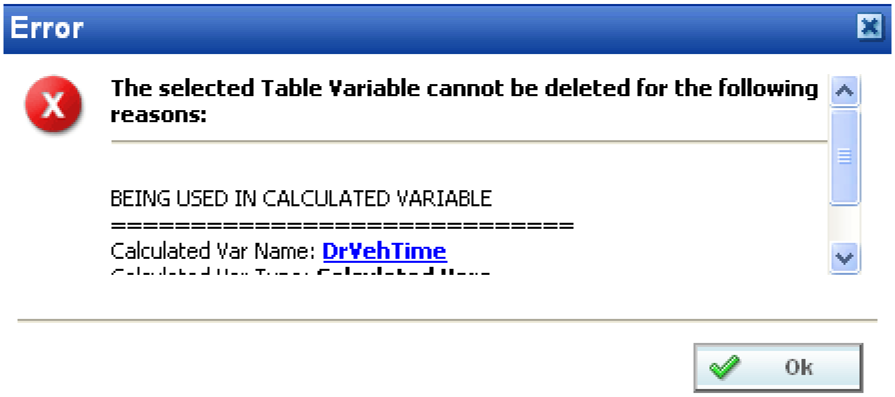
You are about to delete this variable and all of its revisions. Are you sure you want to do this?

Figure 203 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.*



The selected Table Variable cannot be deleted for the following reasons:

```
BEING USED IN CALCULATED VARIABLE
==============================
Calculated Var Name: DrVehTime
```

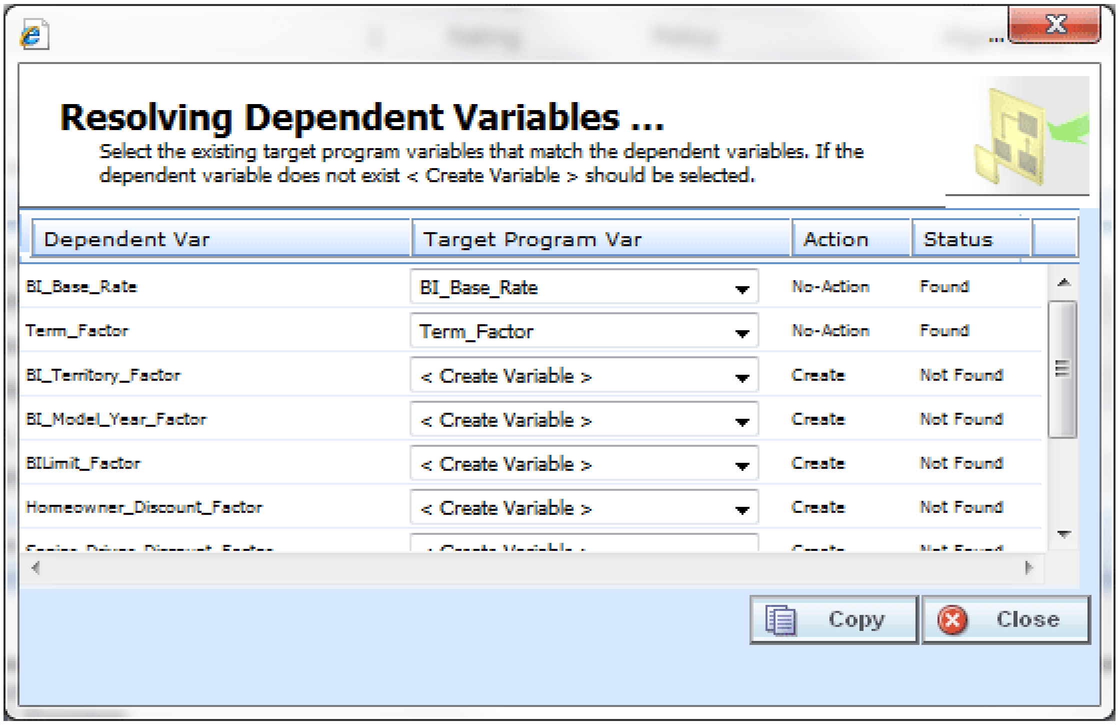
Figure 204 Error Message for Deleting a Variable

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.



Resolving Dependent Variables ...

Select the existing target program variables that match the dependent variables. If the dependent variable does not exist < Create Variable > should be selected.

Dependent Var	Target Program Var	Action	Status
BI_Base_Rate	BI_Base_Rate	No-Action	Found
Term_Factor	Term_Factor	No-Action	Found
BI_Territory_Factor	< Create Variable >	Create	Not Found
BI_Model_Year_Factor	< Create Variable >	Create	Not Found
BILimit_Factor	< Create Variable >	Create	Not Found
Homeowner_Discount_Factor	< Create Variable >	Create	Not Found
Senior_Driver_Discount_Factor	< Create Variable >	Create	Not Found

Figure 205 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 16

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 allows you to 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.

Global and Local Algorithms

Rating Algorithms and Underwriting Algorithms can be created and managed at the global level and at the local level. Global algorithms are like program level algorithms in the way they function and in the way they are created. In global algorithms you can create normal rating algorithms and underwriting algorithms, exactly like you can at the program level.

There are a couple of differences between globals and locals. One difference between them is that global algorithms are available to all programs within the subline while program level algorithms are available only to the program where they were created. Another is that Global algorithms can use only global elements. Local algorithms can use either local or global elements.

Global: Global algorithms are created and managed at the global level and are subline specific. Global algorithms can be accessed from the Global tab or from the Program Details tab. When selected, algorithms will be listed in the Folder Details area.

Local: Local algorithms are created and managed at the program version level and are specific to the program. Local algorithms can be accessed from the Program Details tab. When selected, algorithms will be listed in the Folder Details area.

Both global and local algorithms can be created by the user or brought in from a template. Algorithms that are user created will be open for editing, copying, creating new revisions and deletion if they are not locked.

Algorithms that were brought in from a template will be locked at all times and cannot be unlocked. No editing or deleting will be 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 will be open for editing, copying, creating new revisions 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 subline. Result variables can be used in local algorithms and in global 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 will remain 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 two paths to view and manage algorithms, either from the Globals tab or from the Program Details tab. The Global tab will display global algorithms only. You cannot view or manage local algorithms on the Global tab. On the Program Details tab, you can select to view either global algorithms or local algorithms. If you want to view and manage local algorithms, you must use the Program Details tab. The views and options are the same on either tab.

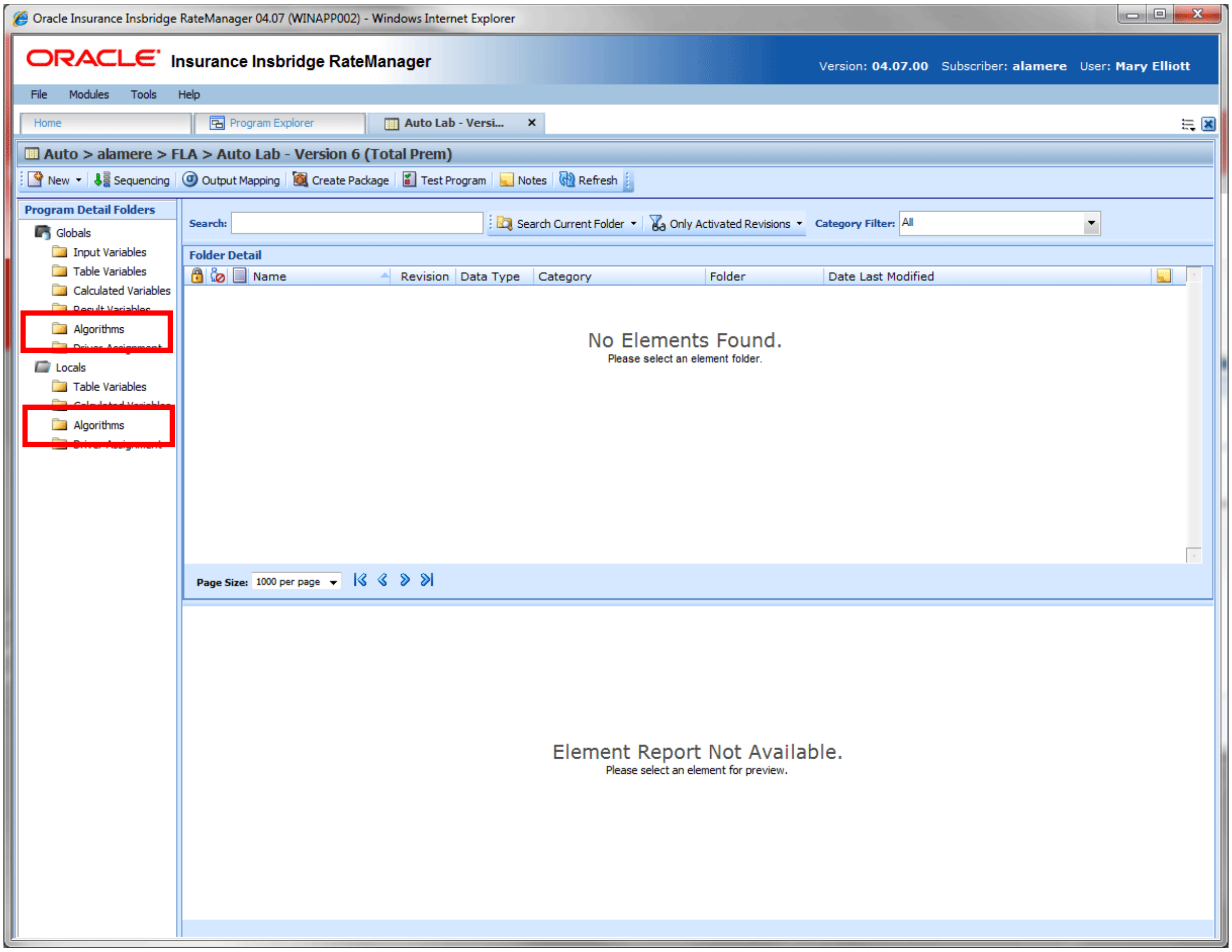
From either tab, when you select Algorithms in the Program Details Folder, the current algorithms will be listed in the Folder Details area.

Algorithms will not be an option on the Program Explorer tab.

1. Modules→Program Explorer. This will open the Program Explorer tab.

2. Select the Line of Business→Subline→Folder→Program → Program Version. You also can access algorithms from the globals tab. Line of Business→Subline→Globals.

3. Open the program version to get the Program Details tab. Algorithms are available from the program details folders area on the left hand side of the tab.



No Elements Found.
Please select an element folder.

Element Report Not Available.
Please select an element for preview.

Figure 206 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 will be able to 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 will have fewer options. There are icons that can tell you at a glance what the status of the algorithm is.

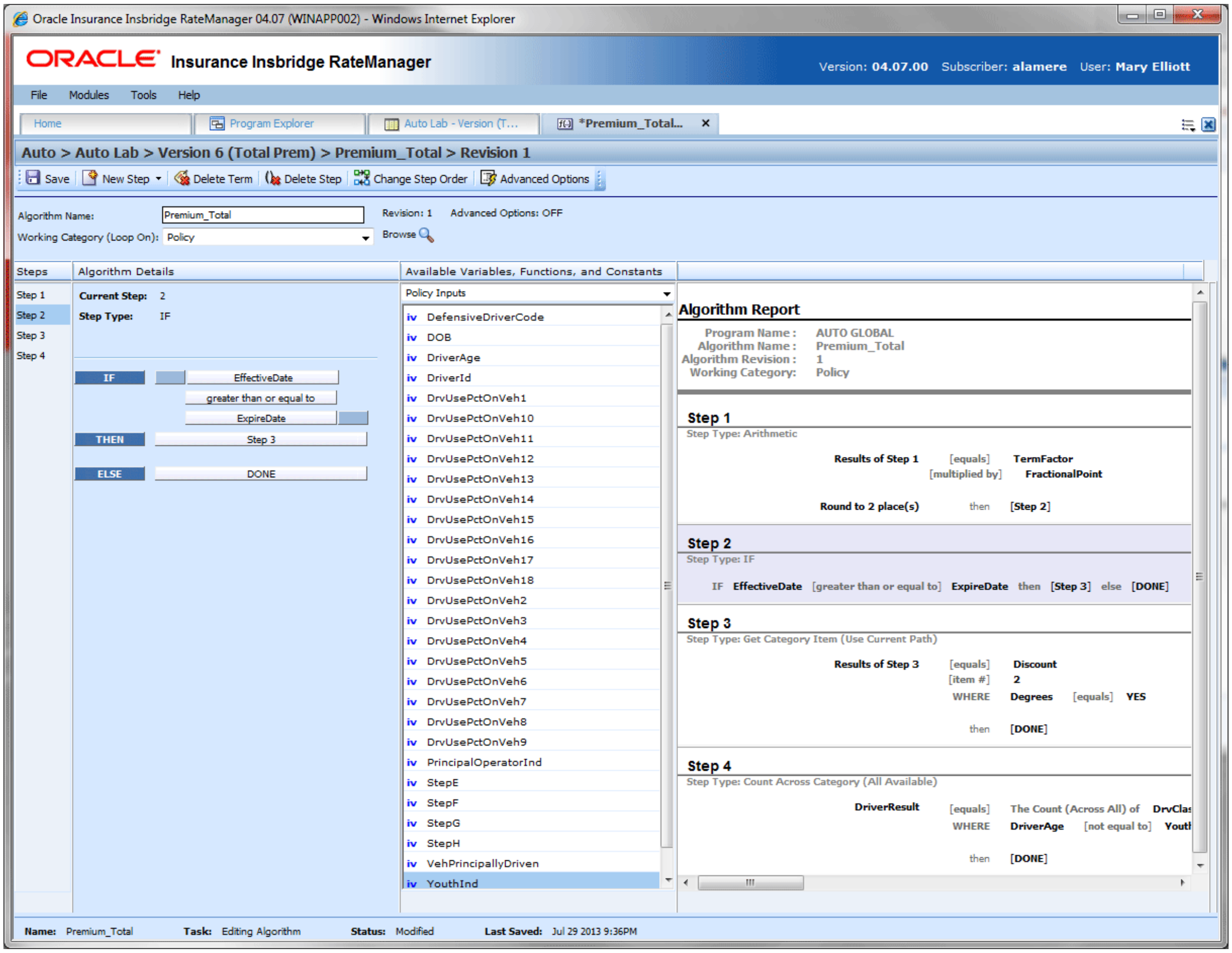
Icon	What it Indicates
f(·)	**Algorithm.** Global Algorithms are subline specific and can be created at the global level, or from a program version. Local Algorithms are program specific and can be created at the program version level. Local Algorithms will be displayed on the Program Details→Local Algorithms. The same algorithm listing will also be available for Subline→Globals tab. The same options, new, edit, copy and delete are available on either tab.
🔒	**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.
✓	**Active Revision.** This algorithm is the current active revision.

Algorithm Fields Program Details

The fields are the same for the Global Tab and Program Details. For more on field descriptions, please see Program Details→Fields.

ALGORITHM TAB

The Algorithm tab will be 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.



Version: **04.07.00** Subscriber: **alamere** User: **Mary Elliott**

Auto > Auto Lab > Version 6 (Total Prem) > Premium_Total > Revision 1

Algorithm Name: Premium_Total

Working Category (Loop On): Policy

Revision: 1 Advanced Options: OFF

Step 1
Step 2
Step 3
Step 4

Current Step: 2
Step Type: IF

IF EffectiveDate greater than or equal to ExpireDate
THEN Step 3
ELSE DONE

Policy Inputs

- iv DefensiveDriverCode
- iv DOB
- iv DriverAge
- iv DriverId
- iv DrvUsePctOnVeh1
- iv DrvUsePctOnVeh10
- iv DrvUsePctOnVeh11
- iv DrvUsePctOnVeh12
- iv DrvUsePctOnVeh13
- iv DrvUsePctOnVeh14
- iv DrvUsePctOnVeh15
- iv DrvUsePctOnVeh16
- iv DrvUsePctOnVeh17
- iv DrvUsePctOnVeh18
- iv DrvUsePctOnVeh2
- iv DrvUsePctOnVeh3
- iv DrvUsePctOnVeh4
- iv DrvUsePctOnVeh5
- iv DrvUsePctOnVeh6
- iv DrvUsePctOnVeh7
- iv DrvUsePctOnVeh8
- iv DrvUsePctOnVeh9
- iv PrincipalOperatorInd
- iv StepE
- iv StepF
- iv StepG
- iv StepH
- iv VehPrincipallyDriven
- iv YouthInd

Algorithm Report

Program Name : AUTO GLOBAL
Algorithm Name : Premium_Total
Algorithm Revision : 1
Working Category: Policy

Step 1
Step Type: Arithmetic

Results of Step 1 [equals] **TermFactor**
[multiplied by] **FractionalPoint**

Round to 2 place(s) then **[Step 2]**

Step 2
Step Type: IF

IF **EffectiveDate** [greater than or equal to] **ExpireDate** then **[Step 3]** else **[DONE]**

Step 3
Step Type: Get Category Item (Use Current Path)

Results of Step 3 [equals] **Discount**
[item #] **2**
WHERE **Degrees** [equals] **YES**

then **[DONE]**

Step 4
Step Type: Count Across Category (All Available)

DriverResult [equals] The Count (Across All) of **DrvClas**
WHERE **DriverAge** [not equal to] **Youtl**

then **[DONE]**

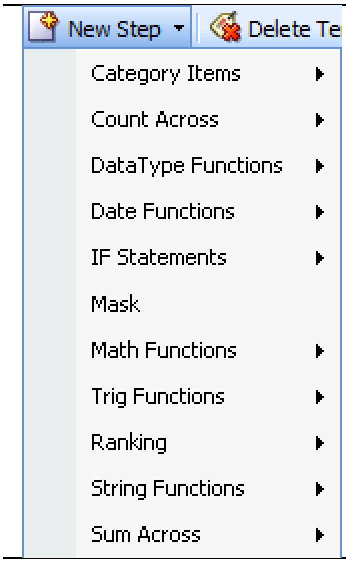
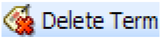
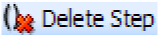
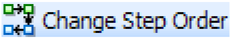
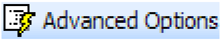
Name: Premium_Total **Task:** Editing Algorithm **Status:** Modified **Last Saved:** Jul 29 2013 9:36PM

Figure 207 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 on this tab.

Save	**Save:** Saves the algorithm.

 New Step Delete Te Category Items Count Across DataType Functions Date Functions IF Statements Mask Math Functions Trig Functions Ranking String Functions Sum Across	**New Step:** Creates a new step in the algorithm. For more information, see Working with Steps. **NOTE:** *Underwriting Algorithms will have a "Set Underwriting to Fail" step type in the list.*
	**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.

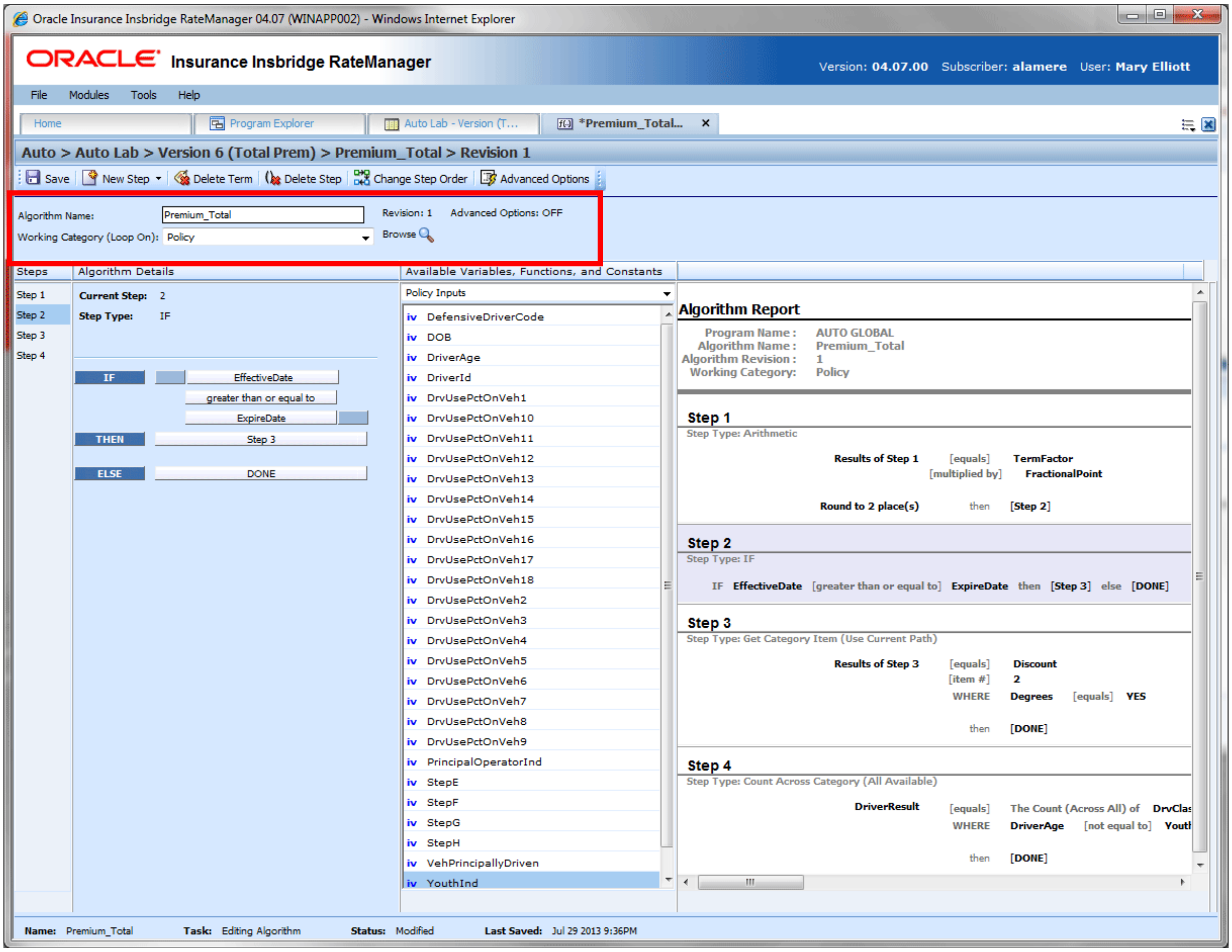
ALGORITHM TAB FIELDS

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.


Oracle Insurance Insbridge RateManager 04.07 (WINAPP002) - Windows Internet Explorer
ORACLE Insurance Insbridge RateManager
Version: 04.07.00 Subscriber: alamere User: Mary Elliott
File
Modules
Tools
Help
Home
Program Explorer
Auto Lab - Version (T...
*Premium_Total...
Auto > Auto Lab > Version 6 (Total Prem) > Premium_Total > Revision 1
Save
New Step
Delete Term
Delete Step
Change Step Order
Advanced Options
Browse
Steps
Algorithm Details
Available Variables, Functions, and Constants
Step 1
Step 2
Step 3
Step 4

Algorithm Name: Premium_Total

Working Category (Loop On): Policy

Revision: 1 Advanced Options: OFF

Current Step: 2

Step Type: IF

IF EffectiveDate greater than or equal to ExpireDate

THEN Step 3

ELSE DONE

Policy Inputs

iv DefensiveDriverCode
iv DOB
iv DriverAge
iv DriverId
iv DrvUsePctOnVeh1
iv DrvUsePctOnVeh10
iv DrvUsePctOnVeh11
iv DrvUsePctOnVeh12
iv DrvUsePctOnVeh13
iv DrvUsePctOnVeh14
iv DrvUsePctOnVeh15
iv DrvUsePctOnVeh16
iv DrvUsePctOnVeh17
iv DrvUsePctOnVeh18
iv DrvUsePctOnVeh2
iv DrvUsePctOnVeh3
iv DrvUsePctOnVeh4
iv DrvUsePctOnVeh5
iv DrvUsePctOnVeh6
iv DrvUsePctOnVeh7
iv DrvUsePctOnVeh8
iv DrvUsePctOnVeh9
iv PrincipalOperatorInd
iv StepE
iv StepF
iv StepG
iv StepH
iv VehPrincipallyDriven
iv YouthInd

Algorithm Report

Program Name : AUTO GLOBAL
Algorithm Name : Premium_Total
Algorithm Revision : 1
Working Category: Policy

Step 1
Step Type: Arithmetic
Results of Step 1 [equals] TermFactor
[multiplied by] FractionalPoint
Round to 2 place(s) then [Step 2]

Step 2
Step Type: IF
IF EffectiveDate [greater than or equal to] ExpireDate then [Step 3] else [DONE]

Step 3
Step Type: Get Category Item (Use Current Path)
Results of Step 3 [equals] Discount
[item #] 2
WHERE Degrees [equals] YES
then [DONE]

Step 4
Step Type: Count Across Category (All Available)
DriverResult [equals] The Count (Across All) of DrvClas
WHERE DriverAge [not equal to] Youth
then [DONE]

Name: Premium_Total Task: Editing Algorithm Status: Modified Last Saved: Jul 29 2013 9:36PM

Figure 208 Algorithm Information Field

Fields:

Algorithm Name: Entry for what you want the algorithm to be called. Alphanumeric characters are permitted, but no special characters (ex: &, *, ", +, @, etc.) are allowed.

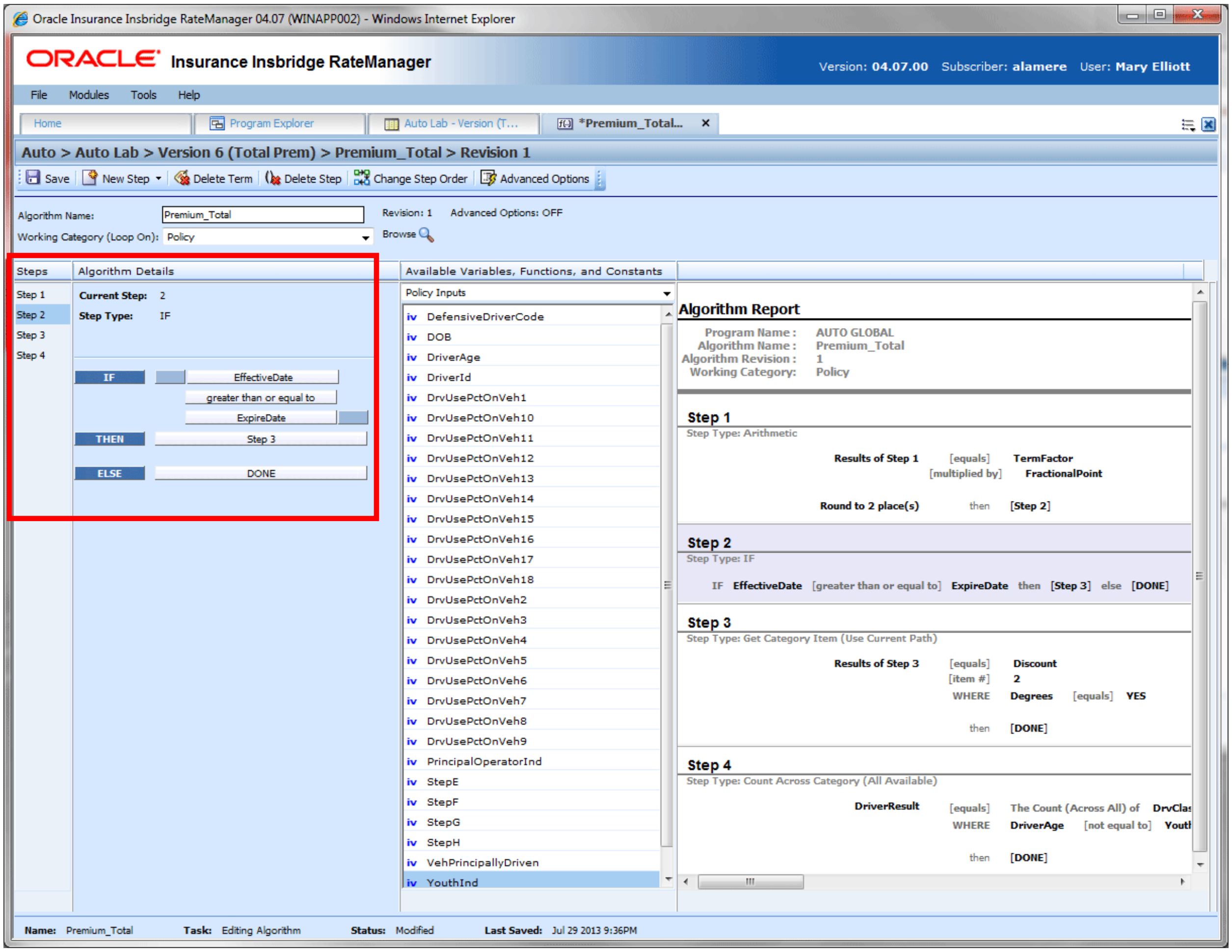
Working Category: Sets the category where the algorithm will be run. You can select the working category from the drop down menu or you can browse for a working category by clicking **BROWSE**. This will bring up the Browse Category popup. From here you can navigate to the Category you want. Once you 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 Inputs, Policy Variables and Policy Results.

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.



Oracle Insurance Insbridge RateManager 04.07 (WINAPP002) - Windows Internet Explorer
ORACLE Insurance Insbridge RateManager
Version: 04.07.00 Subscriber: alamere User: Mary Elliott
File Modules Tools Help
Home | Program Explorer | Auto Lab - Version (T... | *Premium_Total...
Save | New Step | Delete Term | Delete Step | Change Step Order | Advanced Options

Auto > Auto Lab > Version 6 (Total Prem) > Premium_Total > Revision 1

Algorithm Name: Premium_Total
Working Category (Loop On): Policy
Revision: 1 Advanced Options: OFF
Browse

Steps: Step 1, Step 2, Step 3, Step 4

Algorithm Details
Current Step: 2
Step Type: IF

IF EffectiveDate greater than or equal to ExpireDate
THEN Step 3
ELSE DONE

Available Variables, Functions, and Constants
Policy Inputs
iv DefensiveDriverCode
iv DOB
iv DriverAge
iv DriverId
iv DrvUsePctOnVeh1
iv DrvUsePctOnVeh10
iv DrvUsePctOnVeh11
iv DrvUsePctOnVeh12
iv DrvUsePctOnVeh13
iv DrvUsePctOnVeh14
iv DrvUsePctOnVeh15
iv DrvUsePctOnVeh16
iv DrvUsePctOnVeh17
iv DrvUsePctOnVeh18
iv DrvUsePctOnVeh2
iv DrvUsePctOnVeh3
iv DrvUsePctOnVeh4
iv DrvUsePctOnVeh5
iv DrvUsePctOnVeh6
iv DrvUsePctOnVeh7
iv DrvUsePctOnVeh8
iv DrvUsePctOnVeh9
iv PrincipalOperatorInd
iv StepE
iv StepF
iv StepG
iv StepH
iv VehPrincipallyDriven
iv YouthInd

Algorithm Report
Program Name : AUTO GLOBAL
Algorithm Name : Premium_Total
Algorithm Revision : 1
Working Category: Policy

Step 1
Step Type: Arithmetic
Results of Step 1 [equals] TermFactor
[multiplied by] FractionalPoint
Round to 2 place(s) then [Step 2]

Step 2
Step Type: IF
IF EffectiveDate [greater than or equal to] ExpireDate then [Step 3] else [DONE]

Step 3
Step Type: Get Category Item (Use Current Path)
Results of Step 3 [equals] Discount
[item #] 2
WHERE Degrees [equals] YES
then [DONE]

Step 4
Step Type: Count Across Category (All Available)
DriverResult [equals] The Count (Across All) of DrvClas
WHERE DriverAge [not equal to] Youtl
then [DONE]

Name: Premium_Total Task: Editing Algorithm Status: Modified Last Saved: Jul 29 2013 9:36PM

Figure 209 Algorithm Selection Criteria Field

Fields:

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 inputs/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 9 digits.

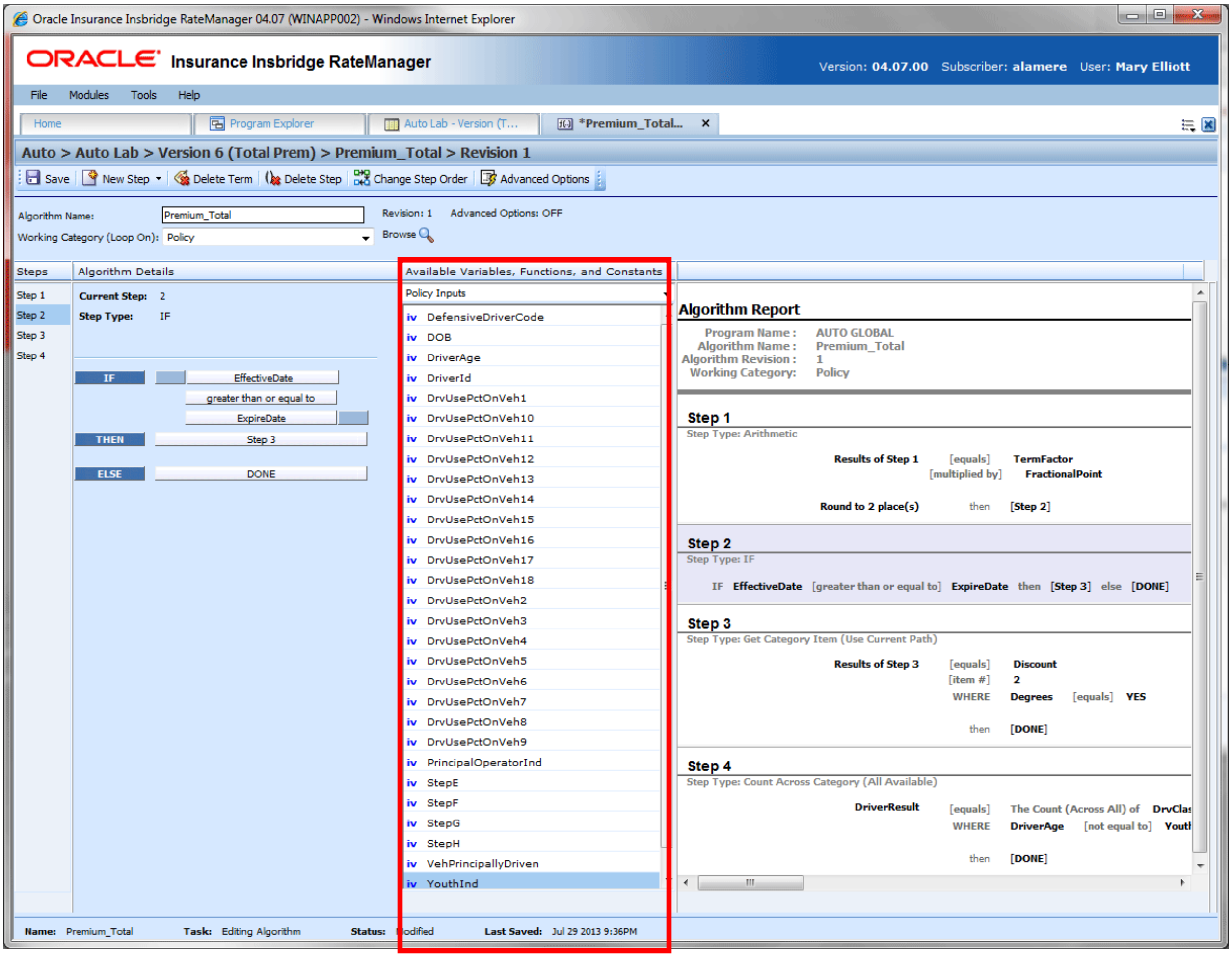
Inputs and variables will have a right click menu available; View Report, View Where Used and Custom Value.

Algorithm Tab Available Variables, Functions and Constants

The Available Variables, Functions and Constants area contains a listing of available inputs, variables, and results, sorted by category and system constants that can be used as criteria. For global algorithms, only global variables or inputs 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 Inputs, you will be able to add a new input variable without leaving the Edit Algorithm tab. Scroll to the bottom of the Available Variables, Functions and Constants list and select <New Input Variable>. A popup will be displayed that will allow you to enter a new input variable.

You also will be able to add new a new result variable without leaving the Edit Algorithm tab.



Version: **04.07.00** Subscriber: **alamere** User: **Mary Elliott**

Auto > Auto Lab > Version 6 (Total Prem) > Premium_Total > Revision 1

Algorithm Name: Premium_Total

Working Category (Loop On): Policy

Revision: 1 Advanced Options: OFF

Current Step: 2

Step Type: IF

Policy Inputs

- iv DefensiveDriverCode
- iv DOB
- iv DriverAge
- iv DriverId
- iv DrvUsePctOnVeh1
- iv DrvUsePctOnVeh10
- iv DrvUsePctOnVeh11
- iv DrvUsePctOnVeh12
- iv DrvUsePctOnVeh13
- iv DrvUsePctOnVeh14
- iv DrvUsePctOnVeh15
- iv DrvUsePctOnVeh16
- iv DrvUsePctOnVeh17
- iv DrvUsePctOnVeh18
- iv DrvUsePctOnVeh2
- iv DrvUsePctOnVeh3
- iv DrvUsePctOnVeh4
- iv DrvUsePctOnVeh5
- iv DrvUsePctOnVeh6
- iv DrvUsePctOnVeh7
- iv DrvUsePctOnVeh8
- iv DrvUsePctOnVeh9
- iv PrincipalOperatorInd
- iv StepE
- iv StepF
- iv StepG
- iv StepH
- iv VehPrincipallyDriven
- iv YouthInd

Algorithm Report

Program Name : AUTO GLOBAL
Algorithm Name : Premium_Total
Algorithm Revision : 1
Working Category: Policy

Step 1
Step Type: Arithmetic

Results of Step 1 [equals] **TermFactor**
[multiplied by] **FractionalPoint**

Round to 2 place(s) then **[Step 2]**

Step 2
Step Type: IF

IF **EffectiveDate** [greater than or equal to] **ExpireDate** then **[Step 3]** else **[DONE]**

Step 3
Step Type: Get Category Item (Use Current Path)

Results of Step 3 [equals] **Discount**
[item #] **2**
WHERE **Degrees** [equals] **YES**

then **[DONE]**

Step 4
Step Type: Count Across Category (All Available)

DriverResult [equals] The Count (Across All) of **DrvClas**
WHERE **DriverAge** [not equal to] **Youtl**

then **[DONE]**

Name: Premium_Total **Task:** Editing Algorithm **Status:** odified **Last Saved:** Jul 29 2013 9:36PM

Figure 210 Available Variables, Function and Constants Field

Fields:

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 results will be listed. Variables will be 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.
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.
Constants	Selection of system constants, category instances and system date and time. Or you can create a custom value for use in this program.
Step Variables	Results from the previously entered steps.
Policy Inputs	Inputs in the Policy category.
Policy Variables	Table and calculated variables in the Policy category.
Policy Results	Result variables in the Policy category.

Using Functions:

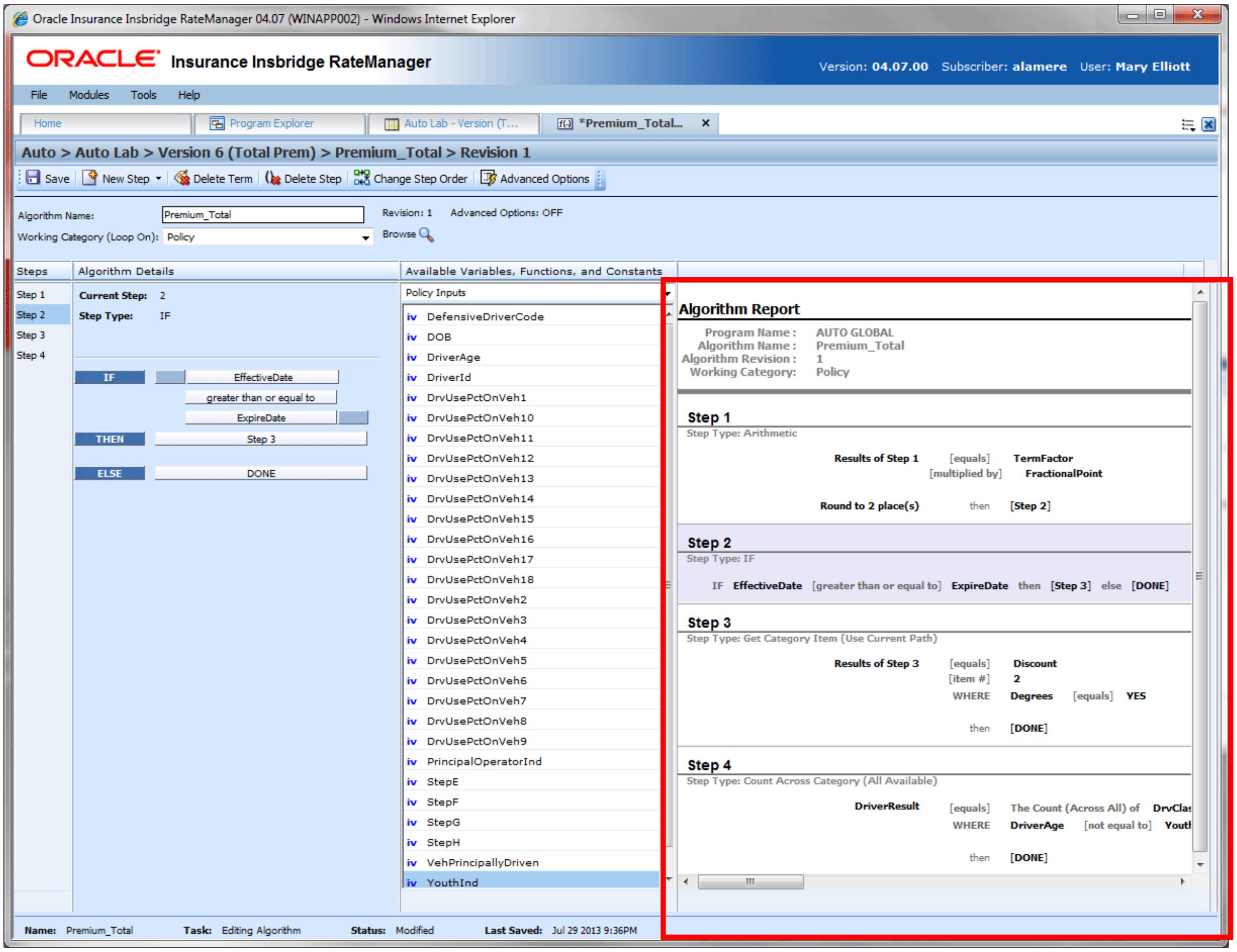
Algorithms and Driver Assignments have functions available on the Available Variable, Functions, and Constants menu.

When a function is selected, the function will be 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 inputs, 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.



Algorithm Name: Premium_Total
Working Category (Loop On): Policy
Revision: 1 Advanced Options: OFF
Browse

Current Step: 2
Step Type: IF

Algorithm Report

Program Name : AUTO GLOBAL
Algorithm Name : Premium_Total
Algorithm Revision : 1
Working Category: Policy

Step 1
Step Type: Arithmetic

Results of Step 1 [equals] **TermFactor**
[multiplied by] **FractionalPoint**

Round to 2 place(s) then **[Step 2]**

Step 2
Step Type: IF

IF **EffectiveDate** [greater than or equal to] **ExpireDate** then **[Step 3]** else **[DONE]**

Step 3
Step Type: Get Category Item (Use Current Path)

Results of Step 3 [equals] **Discount**
[item #] **2**
WHERE **Degrees** [equals] **YES**

then **[DONE]**

Step 4
Step Type: Count Across Category (All Available)

DriverResult [equals] The Count (Across All) of **DrvClas**
WHERE **DriverAge** [not equal to] **Youtl**

then **[DONE]**

Name: Premium_Total **Task:** Editing Algorithm **Status:** Modified **Last Saved:** Jul 29 2013 9:36PM

Figure 211 Algorithm Report

CREATING AN ALGORITHM

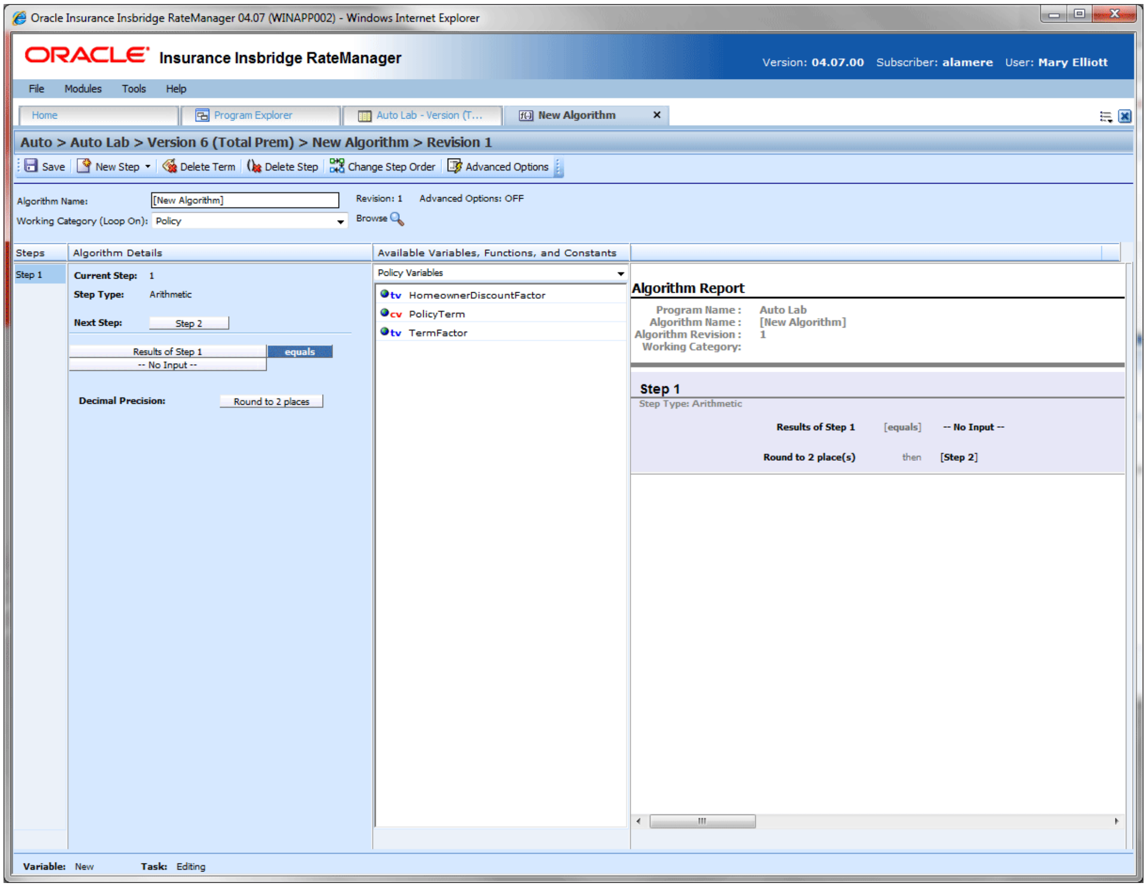
Algorithms can be created and added to standard programs and template generated programs. Local algorithms will be available in the program where you created it and can be copied to other programs within the subline. Global algorithms will be available across the subline.

Local and global algorithms are created and function in much the same way. One difference is that global algorithms can use only global table variables, inputs and result variables. Local algorithms can use both local and global elements. Another difference is that global algorithms can be created from the globals tab and from the program details tab.

1. On the Program Details tab of the program where you want to create a new algorithm, select either the Global Algorithms folder or the Local Algorithms folder.

2. Click **New** in the top bar menu. You then need to select to create a rating algorithm or an underwriting algorithm. If you do not want to select a folder, you can click New and select from there.

   New→Globals→Algorithm→Normal Rating
   New→Globals→Algorithm→Underwriting
   New→Locals→Algorithm→Normal Rating
   New→Locals→Algorithm→Underwriting



Oracle Insurance Insbridge RateManager 04.07 (WINAPP002) - Windows Internet Explorer

ORACLE Insurance Insbridge RateManager Version: **04.07.00** Subscriber: **alamere** User: **Mary Elliott**

Auto > Auto Lab > Version 6 (Total Prem) > New Algorithm > Revision 1

Algorithm Name: [New Algorithm] Revision: 1 Advanced Options: OFF
Working Category (Loop On): Policy

Current Step: 1
Step Type: Arithmetic
Next Step: Step 2

Results of Step 1 equals
-- No Input --

Decimal Precision: Round to 2 places

tv HomeownerDiscountFactor
cv PolicyTerm
tv TermFactor

Algorithm Report

Program Name : Auto Lab
Algorithm Name : [New Algorithm]
Algorithm Revision : 1
Working Category:

Step 1
Step Type: Arithmetic

Results of Step 1 [equals] **-- No Input --**
Round to 2 place(s) then **[Step 2]**

Variable: New **Task:** Editing

Figure 212 Creating a New Algorithm

3. This opens the Algorithm tab for the new algorithm where you can enter or select the variable elements you need.

4. Enter a **NAME** for the algorithm.

5. 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 popup.

   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.

   If this is not the screen you want, click **Cancel** to close the popup and return to the previous screen.

NOTE: *Always save an algorithm after defining the name, and working category. The algorithm will not be available to be used within a step until the algorithm has been saved.*

6. The first step will be 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.

7. To add a new step to the algorithm, click **New Step** and select the type of step you want to add. The screen will refresh with the new step parameters. Depending upon the step type, you will have to select the appropriate variables, functions, constants or inputs. The value you need to select will be 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 will be populated underneath. Click the “Enter Variable” box and then double-click the corresponding variable you want. The box will refresh 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.*

8. Continue entering steps until complete. When you are finished making your entries, click **Save**.

Adding an Input Variable, a Result Variable or a Custom Value Constant

A new input variable, result variable or custom value constant can be created while entering or editing an algorithm, without leaving the Edit screen.

- To add an **Input Variable**: See Adding a New Input from Another Area
- To add a **Custom Value Constant**: See Constants
- 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 will allow 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.

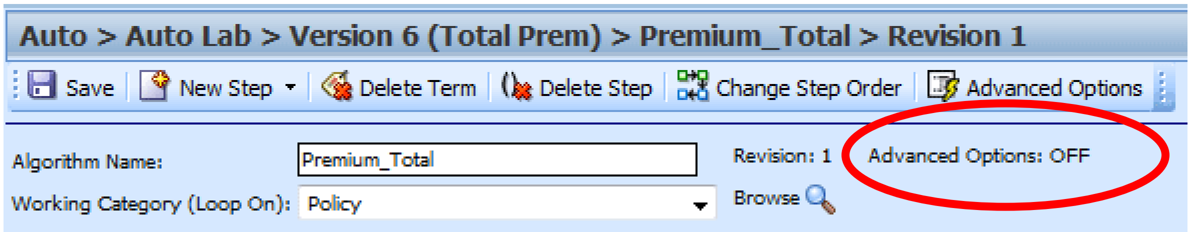
Algorithm Looping is typically used in conjunction with Program to Program (P2P) or Program to Library (P2L) Callouts.

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 will not be 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.



Algorithm Name: Premium_Total

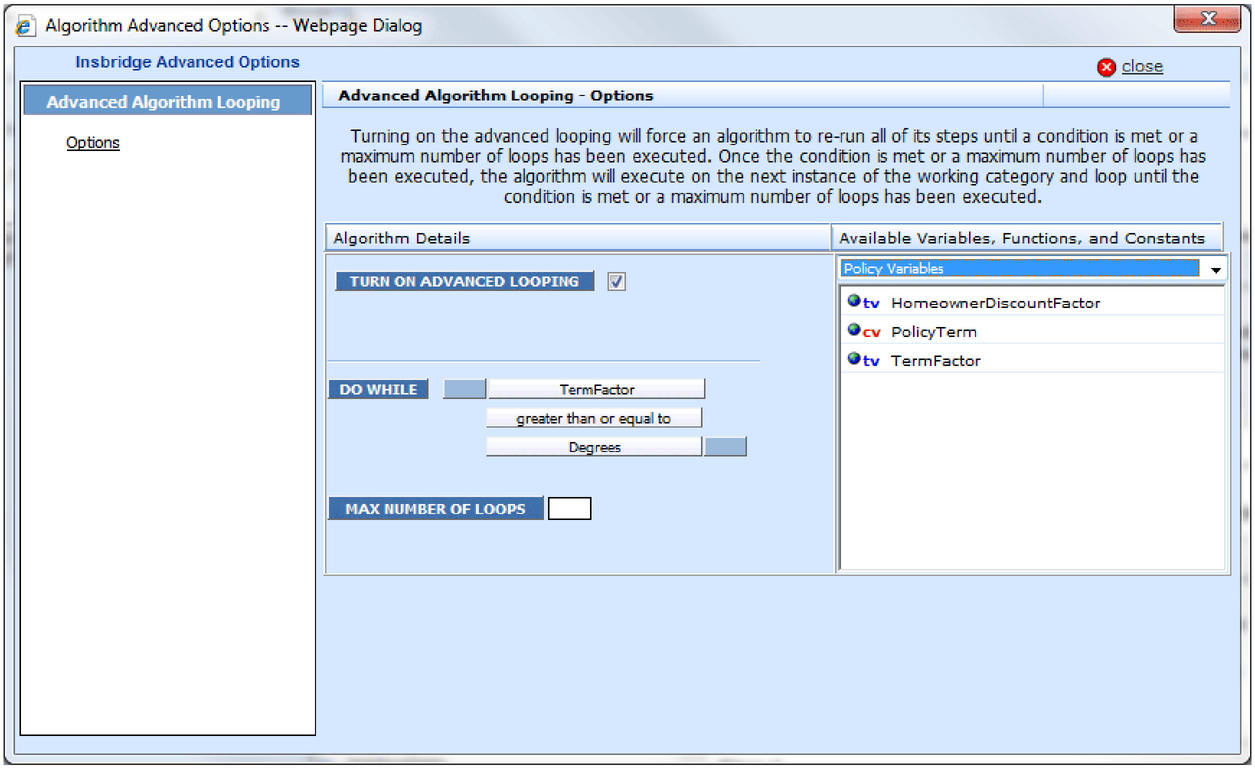
Working Category (Loop On): Policy

Revision: 1

Advanced Options: OFF

Figure 213 Adding an Advanced Option to a Rating Algorithm

2. Click **Advanced Options**. This opens the **Algorithm Advanced Options** screen.



Algorithm Advanced Options -- Webpage Dialog

Insbridge Advanced Options

close

Advanced Algorithm Looping

Options

Advanced Algorithm Looping - Options

Turning on the advanced looping will force an algorithm to re-run all of its steps until a condition is met or a maximum number of loops has been executed. Once the condition is met or a maximum number of loops has been executed, the algorithm will execute on the next instance of the working category and loop until the condition is met or a maximum number of loops has been executed.

Algorithm Details

TURN ON ADVANCED LOOPING ☑

DO WHILE TermFactor greater than or equal to Degrees

MAX NUMBER OF LOOPS

Available Variables, Functions, and Constants

Policy Variables

- tv HomeownerDiscountFactor
- cv PolicyTerm
- tv TermFactor

Figure 214 Algorithm Advanced Options

3. Check the **TURN ON ADVANCED LOOPING** box. This will turn on the algorithm looping.

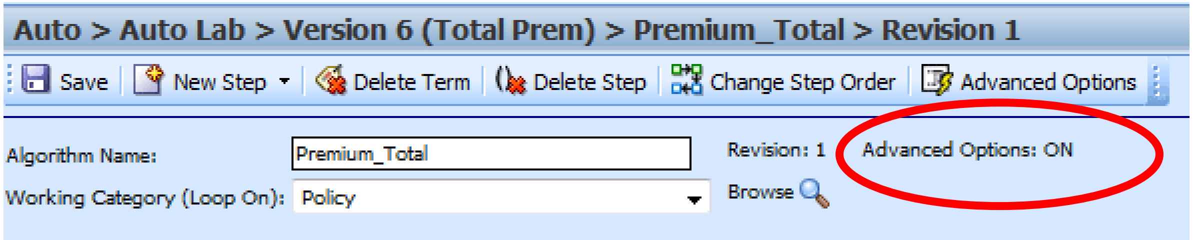
4. The **DO WHILE** section is where you will select the conditions of the loop. The first variable must be a result or input variable. Select the **first variable**.

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.



Auto > Auto Lab > Version 6 (Total Prem) > Premium_Total > Revision 1

Algorithm Name: Premium_Total

Revision: 1

Advanced Options: ON

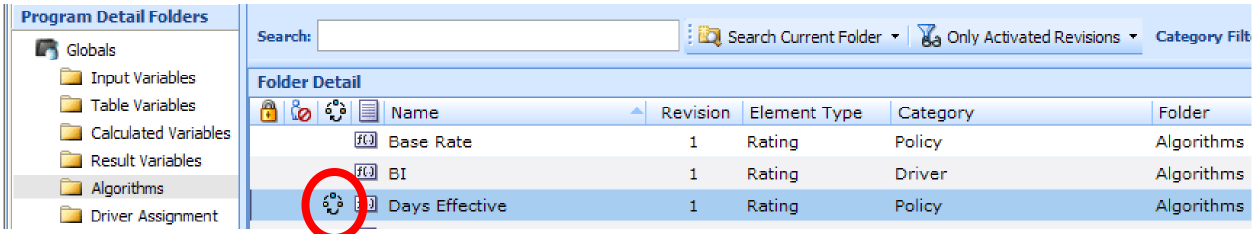
Working Category (Loop On): Policy

Browse

Figure 215 Advanced Options ON

9. On the Algorithm Edit screen, the Advanced Options will show **ON**.

10. Click **Save** to save your work.



Name	Revision	Element Type	Category	Folder
Base Rate	1	Rating	Policy	Algorithms
BI	1	Rating	Driver	Algorithms
Days Effective	1	Rating	Policy	Algorithms

Figure 216 Algorithm With Algorithm Looping Applied

11. On the Program Details tab, the Advance Options Icon will be displayed on the algorithm.

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 will open 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 will be 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.

COPYING AN ALGORITHM

Any algorithm may be copied at any time. In your default subline, local algorithms may be copied to any program within the subline. Global algorithms can be copied within the same subline. Template generated programs will allow you to copy algorithms only within the same program. Any copied algorithm will be open for editing. To make a change to the algorithm, you must go into the edit tab.

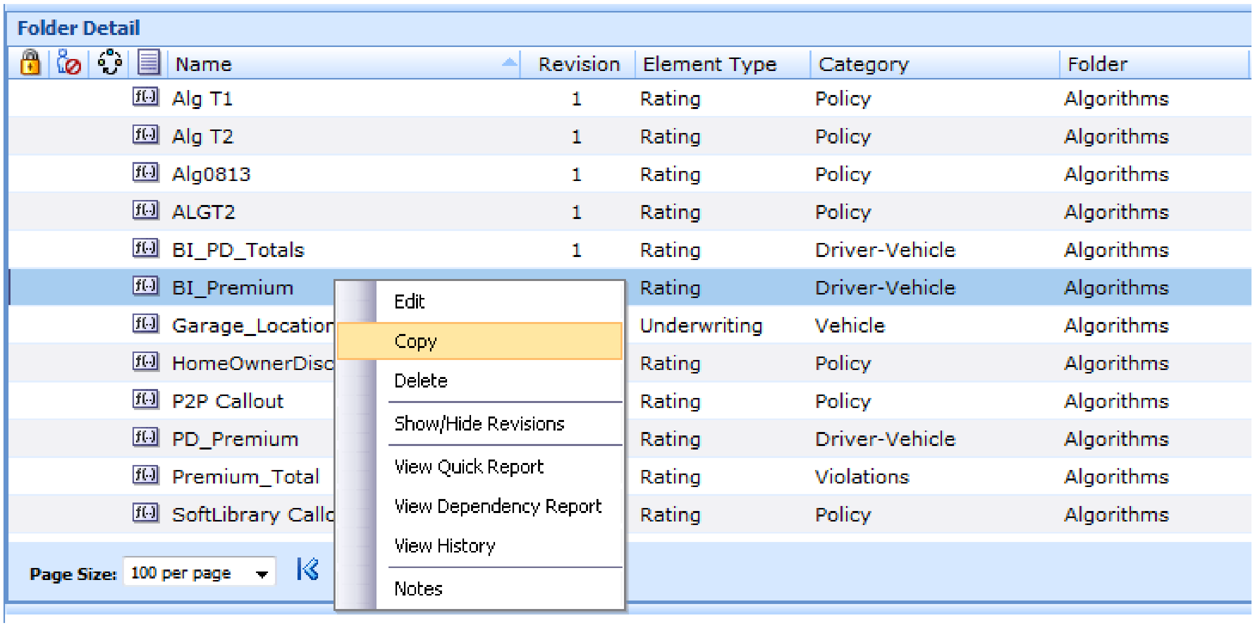
When copying an algorithm, the structure of the algorithm (including the working category and steps) is copied. Only the active revision is copied. If you have an algorithm with 10 revisions, and you copy the algorithm, only the active revision will be copied into the new algorithm. The other 9 revisions will not be copied.

For Local Algorithms, 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 subline. This option is for algorithms within your default subline.

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 it. Select **Copy** from the popup menu.



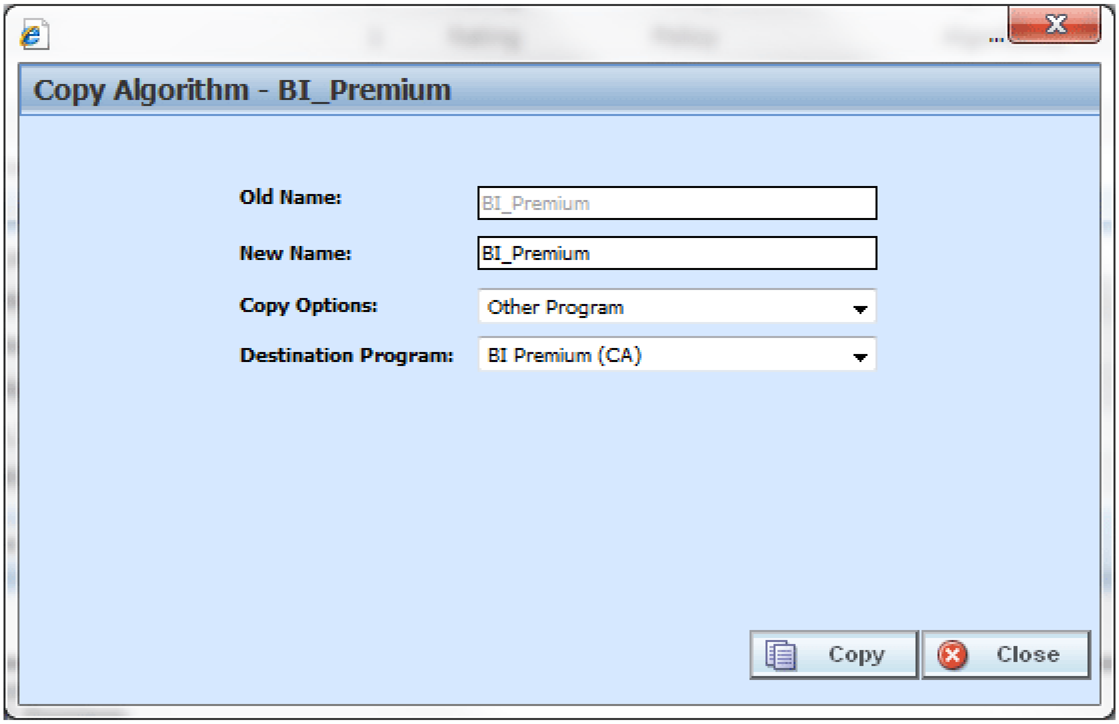
Folder Detail

Name	Revision	Element Type	Category	Folder
Alg T1	1	Rating	Policy	Algorithms
Alg T2	1	Rating	Policy	Algorithms
Alg0813	1	Rating	Policy	Algorithms
ALGT2	1	Rating	Policy	Algorithms
BI_PD_Totals	1	Rating	Driver-Vehicle	Algorithms
BI_Premium		Rating	Driver-Vehicle	Algorithms
Garage_Locatior		Underwriting	Vehicle	Algorithms
HomeOwnerDisc		Rating	Policy	Algorithms
P2P Callout		Rating	Policy	Algorithms
PD_Premium		Rating	Driver-Vehicle	Algorithms
Premium_Total		Rating	Violations	Algorithms
SoftLibrary Callc		Rating	Policy	Algorithms

Page Size: 100 per page

Figure 217 Copy Algorithm Menu

2. A separate popup window will be displayed.



Copy Algorithm - BI_Premium

Old Name: BI_Premium

New Name: BI_Premium

Copy Options: Other Program

Destination Program: BI Premium (CA)

Figure 218 Copy Algorithm Popup

3. Select the **Copy Options**. It is recommended that you change the copy options first. Changing options may reset the name field.

4. If you are copying an algorithm within your default subline, you can select to copy the algorithm to another program within the same subline. Select the **Destination Program** from the drop down menu.

5. 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.

6. Click **Copy** to copy the algorithm. Dependencies will be searched. If there are any dependencies that need to be resolved, a popup window will be displayed. See Resolving Dependencies.

7. If the copy was successful, you will be returned to the previous screen. Your algorithm will have been copied. The last updated date will be 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 will be displayed informing you where the global algorithm is being used.

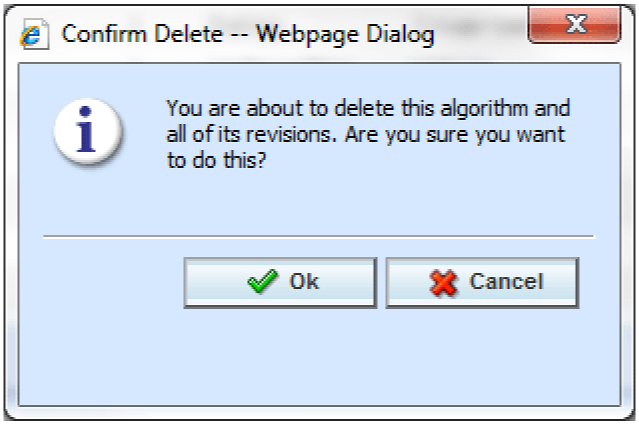
A lock icon indicates that this algorithm is locked by either a program or a template and is not open for editing or deleting.

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 popup menu.



You are about to delete this algorithm and all of its revisions. Are you sure you want to do this?

Figure 219 Deleting an Algorithm

3. You will be 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 17

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. The Driver Assignment option is available in the Auto LOB only. No other line will have this option. Criteria often differ by carrier basis and sometimes even by program level.

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 will not be able to 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. 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 line of business, 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 subline 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 driver assignments are created and managed at the global level and are subline specific. Global driver assignments can be accessed from the Global tab or from the Program Details tab. When selected, driver assignments will be listed in the Folder Details area.

Local: Local driver assignments are created and managed at the program version level and are specific to the program. Local driver assignments can be accessed from the Program Details tab. When selected, driver assignments will be listed in the Folder Details area.

Both global and local driver assignments can be created by the user or brought in from a template. driver assignments that are user created will be open for editing, copying, creating new revisions and deletion if they are not locked.

Driver assignments that were brought in from a template will be locked at all times and cannot be unlocked. No editing or deleting will be 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 will be open for editing, copying, creating new revisions or deletion, unless the program version is locked.

Separating Driver and Vehicle Output (Auto LOB 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 (inputs, 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 input)
Step 2	String Addition	Assigned Driver Number (vehicle result) = DriverID (driver input)

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** will 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 new 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 Assignments:

There are two paths to view and manage driver assignments, either from the Globals tab or from the Program Details tab. The Global tab will display 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.

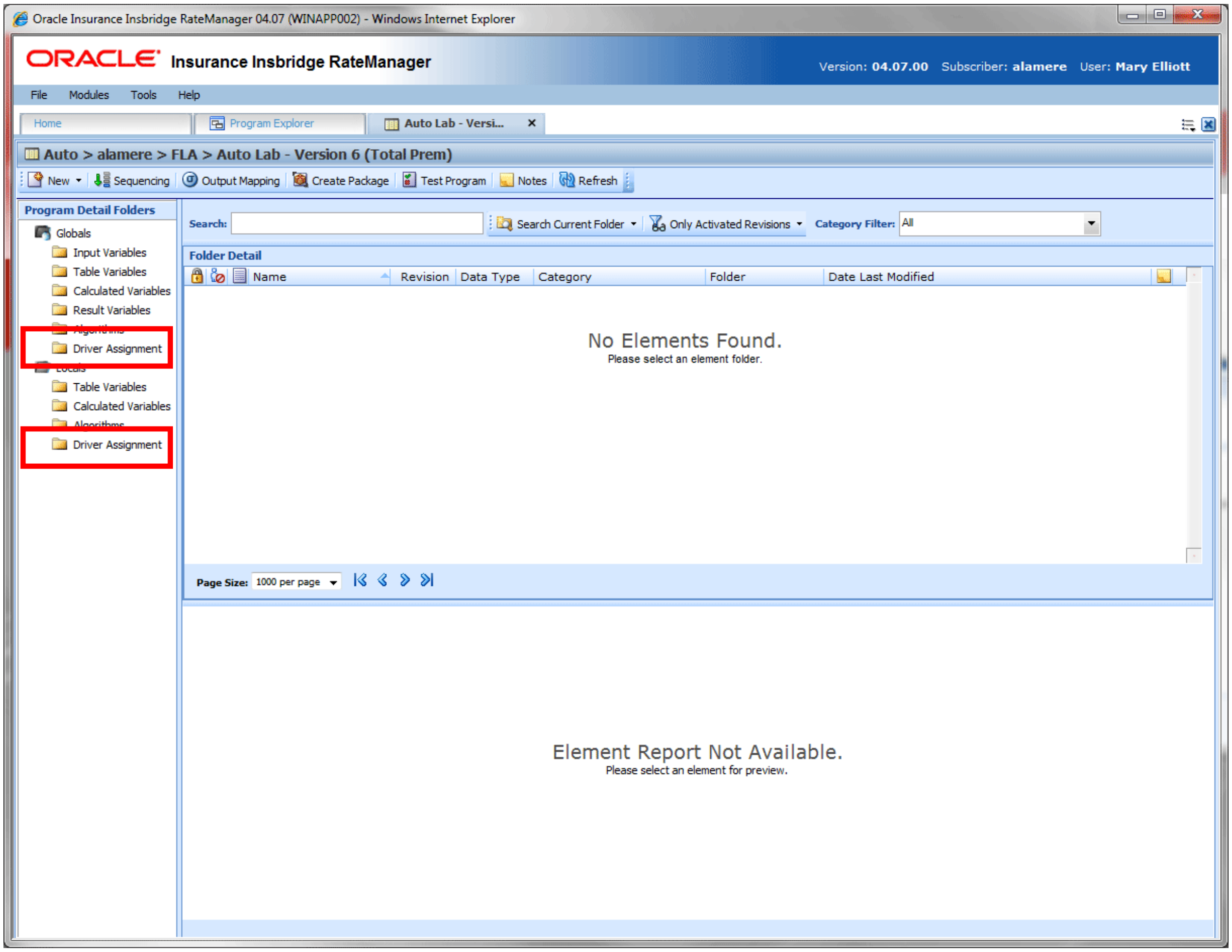
From either tab, when you select Driver Assignments in the Program Details Folder, the current driver assignments will be listed in the Folder Details area.

Driver Assignments is not an option on the Program Explorer tab.

1. Modules→Program Explorer. This will open the Program Explorer tab.

2. Select the Line of Business→Subline→Folder→Program → Program Version. You also can access driver assignments from the globals tab. Line of Business→Subline→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 hand side of the tab.



Search:

Search Current Folder

Only Activated Revisions

Category Filter: All

No Elements Found.
Please select an element folder.

Page Size: 1000 per page

Element Report Not Available.
Please select an element for preview.

Figure 220 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 depends 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.

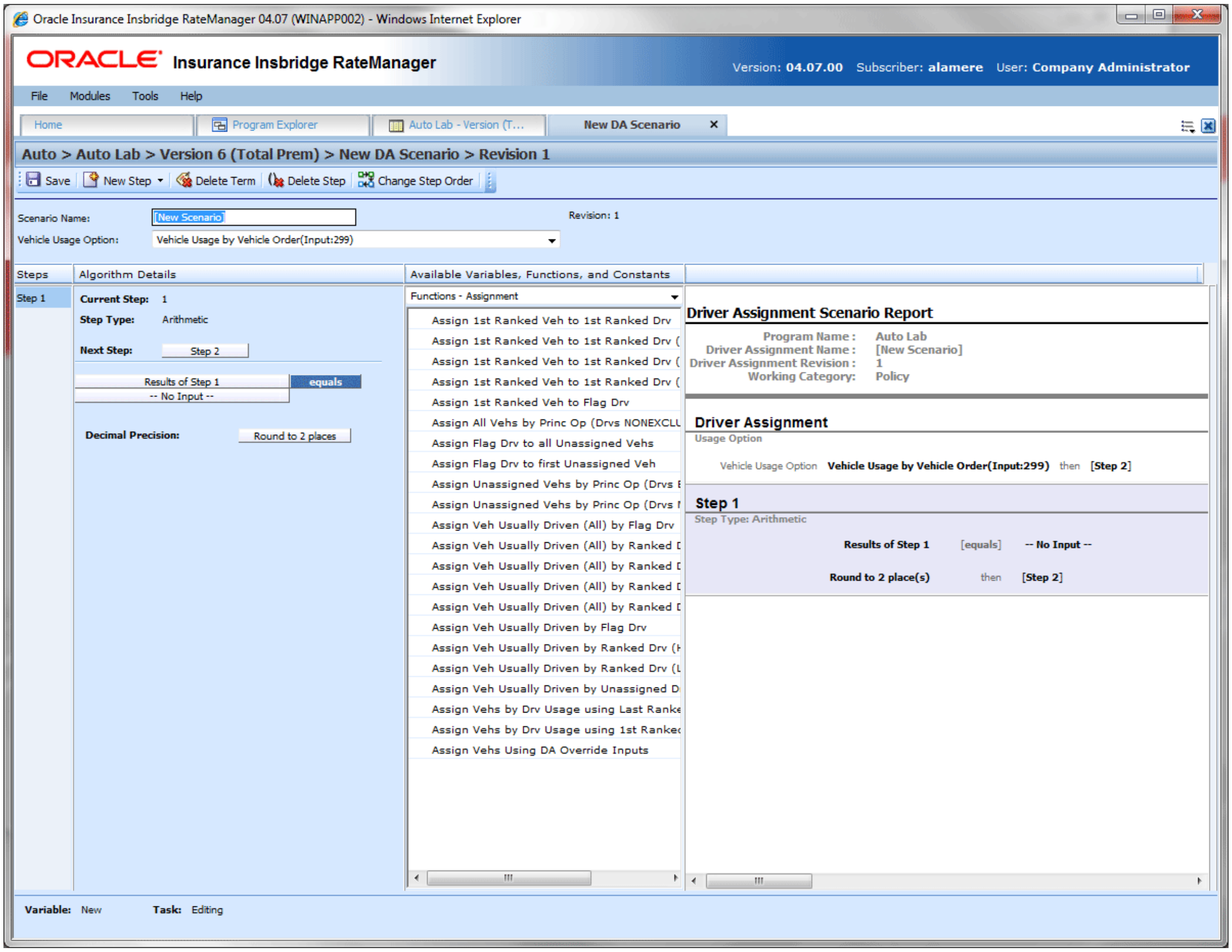
Icon	What it Indicates
f(-)	**Driver Assignment.** Global Driver Assignments are subline specific and can be created at the global level, or from a program version. Local Driver Assignments are program specific and can be created at the program version level. Local Driver Assignments will be displayed on the Program Details→Local Driver Assignments. The same driver assignments listing will also be available for Subline→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 will unlock 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 Fields Program Details:

The fields are the same for the Global Tab and Program Details. For more on field descriptions, please see Program Details→Fields.

Driver Assignment Tab

The Driver Assignment tab will be 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.


Oracle Insurance Insbridge RateManager 04.07 (WINAPP002) - Windows Internet Explorer
ORACLE Insurance Insbridge RateManager
File
Modules
Tools
Help
Home
Program Explorer
Auto Lab - Version (T...
New DA Scenario
Save
New Step
Delete Term
Delete Step
Change Step Order
Steps
Algorithm Details
Available Variables, Functions, and Constants
Functions - Assignment
Assign 1st Ranked Veh to 1st Ranked Drv
Assign 1st Ranked Veh to 1st Ranked Drv (
Assign 1st Ranked Veh to 1st Ranked Drv (
Assign 1st Ranked Veh to 1st Ranked Drv (
Assign 1st Ranked Veh to Flag Drv
Assign All Vehs by Princ Op (Drvs NONEXCLU
Assign Flag Drv to all Unassigned Vehs
Assign Flag Drv to first Unassigned Veh
Assign Unassigned Vehs by Princ Op (Drvs E
Assign Unassigned Vehs by Princ Op (Drvs I
Assign Veh Usually Driven (All) by Flag Drv
Assign Veh Usually Driven (All) by Ranked D
Assign Veh Usually Driven (All) by Ranked D
Assign Veh Usually Driven (All) by Ranked D
Assign Veh Usually Driven (All) by Ranked D
Assign Veh Usually Driven by Flag Drv
Assign Veh Usually Driven by Ranked Drv (H
Assign Veh Usually Driven by Ranked Drv (L
Assign Veh Usually Driven by Unassigned D
Assign Vehs by Drv Usage using Last Ranke
Assign Vehs by Drv Usage using 1st Rankec
Assign Vehs Using DA Override Inputs

Version: 04.07.00 Subscriber: alamere User: Company Administrator

Auto > Auto Lab > Version 6 (Total Prem) > New DA Scenario > Revision 1

Scenario Name: New Scenario

Revision: 1

Vehicle Usage Option: Vehicle Usage by Vehicle Order(Input:299)

Step 1

Current Step: 1

Step Type: Arithmetic

Next Step: Step 2

Results of Step 1 equals

-- No Input --

Decimal Precision: Round to 2 places

Driver Assignment Scenario Report

Program Name : Auto Lab
Driver Assignment Name : [New Scenario]
Driver Assignment Revision : 1
Working Category: Policy

Driver Assignment

Usage Option

Vehicle Usage Option **Vehicle Usage by Vehicle Order(Input:299)** then **[Step 2]**

Step 1

Step Type: Arithmetic

Results of Step 1 [equals] **-- No Input --**

Round to 2 place(s) then **[Step 2]**

Variable: New Task: Editing

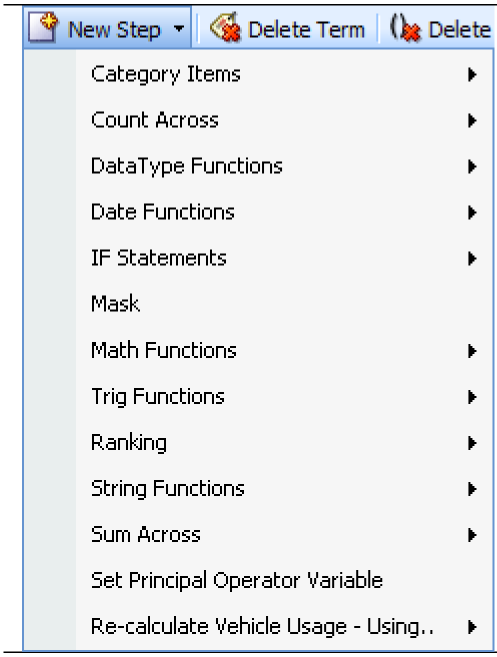
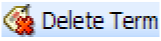
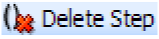
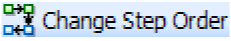
Figure 221 Driver Assignment Tab

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 will always be at least one step. You will not be able to 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 Delete Term Delete Category Items Count Across DataType Functions Date Functions IF Statements Mask Math Functions Trig Functions Ranking String Functions Sum Across Set Principal Operator Variable Re-calculate Vehicle Usage - Using..	**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.*
	**Delete Term:** Deletes the currently selected term from a step in the driver assignment.
	**Delete Step:** Deletes the currently selected step. You cannot delete the last step of a driver assignment.
	**Change Step Order:** Allows users to change the order of the steps without using the drop and drag feature.

DRIVER ASSIGNMENT TAB FIELDS

There are four areas on the Driver Assignment Tab.

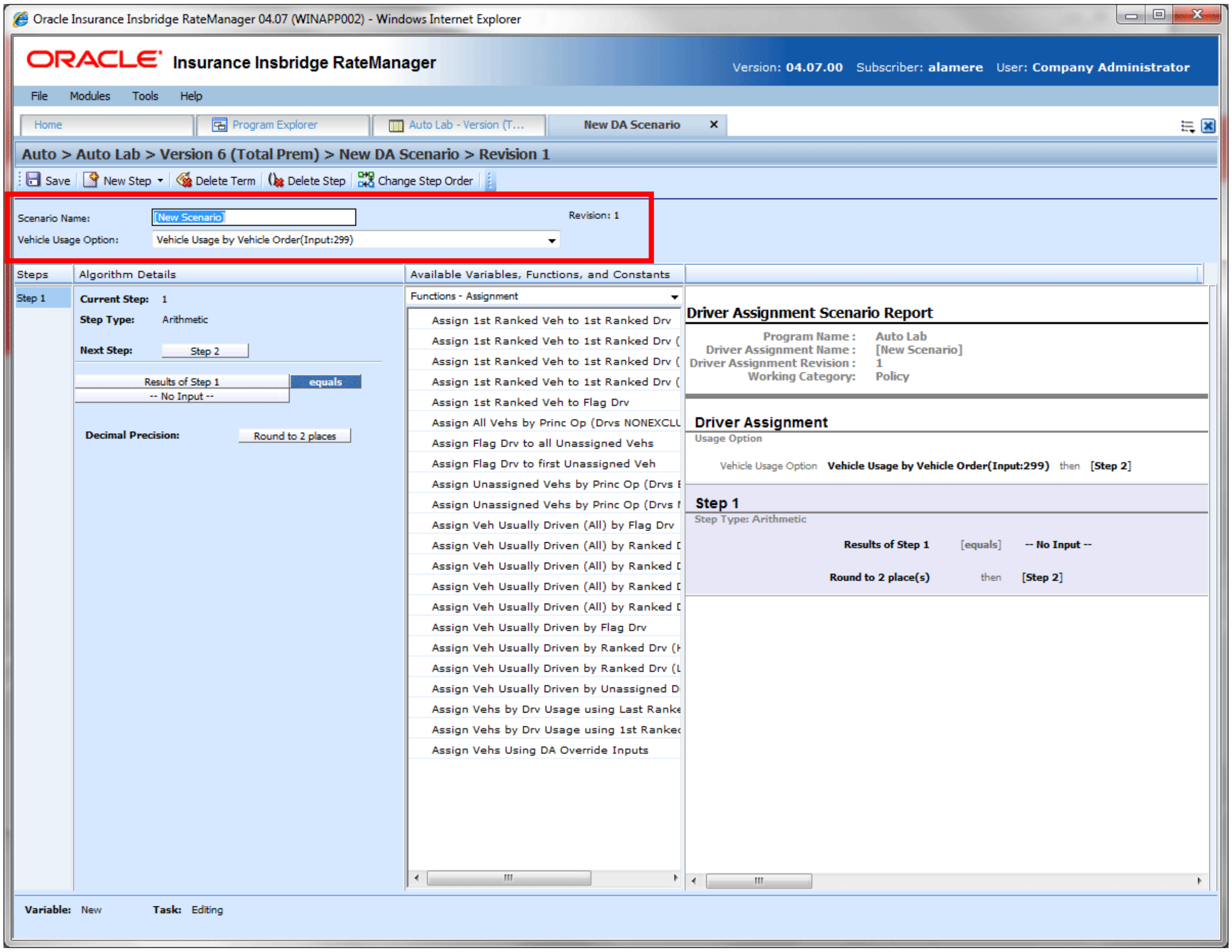
- **Driver Assignment Information:** For a driver assignment scenario, the driver assignment information area contains the name and defines the Vehicle Usage Option of the driver assignment. For a driver assignment algorithm, the driver assignment information area contains the name and defines the working 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.


Oracle Insurance Insbridge RateManager 04.07 (WINAPP002) - Windows Internet Explorer
ORACLE Insurance Insbridge RateManager
Version: 04.07.00 Subscriber: alamere User: Company Administrator
File
Modules
Tools
Help
Home
Program Explorer
Auto Lab - Version (T...
New DA Scenario
Auto > Auto Lab > Version 6 (Total Prem) > New DA Scenario > Revision 1
Save
New Step
Delete Term
Delete Step
Change Step Order
Steps
Algorithm Details
Available Variables, Functions, and Constants
Step 1
Functions - Assignment
Variable: New
Task: Editing

Scenario Name: New Scenario

Revision: 1

Vehicle Usage Option: Vehicle Usage by Vehicle Order(Input:299)

Current Step: 1

Step Type: Arithmetic

Next Step: Step 2

Results of Step 1 equals

-- No Input --

Decimal Precision: Round to 2 places

Assign 1st Ranked Veh to 1st Ranked Drv
Assign 1st Ranked Veh to 1st Ranked Drv (
Assign 1st Ranked Veh to 1st Ranked Drv (
Assign 1st Ranked Veh to 1st Ranked Drv (
Assign 1st Ranked Veh to Flag Drv
Assign All Vehs by Princ Op (Drvs NONEXCLU
Assign Flag Drv to all Unassigned Vehs
Assign Flag Drv to first Unassigned Veh
Assign Unassigned Vehs by Princ Op (Drvs E
Assign Unassigned Vehs by Princ Op (Drvs I
Assign Veh Usually Driven (All) by Flag Drv
Assign Veh Usually Driven (All) by Ranked D
Assign Veh Usually Driven (All) by Ranked D
Assign Veh Usually Driven (All) by Ranked D
Assign Veh Usually Driven (All) by Ranked D
Assign Veh Usually Driven by Flag Drv
Assign Veh Usually Driven by Ranked Drv (H
Assign Veh Usually Driven by Ranked Drv (L
Assign Veh Usually Driven by Unassigned D
Assign Vehs by Drv Usage using Last Ranke
Assign Vehs by Drv Usage using 1st Ranke
Assign Vehs Using DA Override Inputs

Driver Assignment Scenario Report

Program Name : Auto Lab
Driver Assignment Name : [New Scenario]
Driver Assignment Revision : 1
Working Category: Policy

Driver Assignment

Usage Option

Vehicle Usage Option Vehicle Usage by Vehicle Order(Input:299) then [Step 2]

Step 1

Step Type: Arithmetic

Results of Step 1 [equals] -- No Input --

Round to 2 place(s) then [Step 2]

Figure 222 Driver Assignment Information Field

Fields:

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.

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 will bring up the Browse Category popup. 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, DA Results, Constants, Step Variables, Policy Inputs, Policy Variables and Policy Results.

All driver assignment scenarios will be under the Policy working category.

Driver Assignment Algorithms are used in Driver Assignment scenarios only. They will not be 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 will display 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 (Input:299)
- Vehicle Usage on Drivers
- Driver Usage on Vehicles
- Vehicle Usage by VehicleID (Inputs:299/368)

Vehicle Usage by Vehicle Order (Input:299)

Vehicle usage is designated as the actual vehicle a driver principally operates, based on the order of the vehicles in the input file. For example, driver 1 principally operates the second vehicle on the policy. In RateManager, this is accomplished using the Driver input 299 (VehPrincipallyDriven). Specify the vehicle a driver principally operates by setting the VehPrincipallyDriven input equal to the order the vehicle appears in the input 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 inputs 9-16 and 50 (DrvUsePctOnVehX, where X represents an integer 1-9). This method supports an unlimited number of drivers and up to nine 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 inputs 1-8 and 700 (VehUsePctOnDrvX, where X represents an integer 1-9). This method supports an unlimited number of vehicles and up to nine drivers. The percentages should add up to 100%.

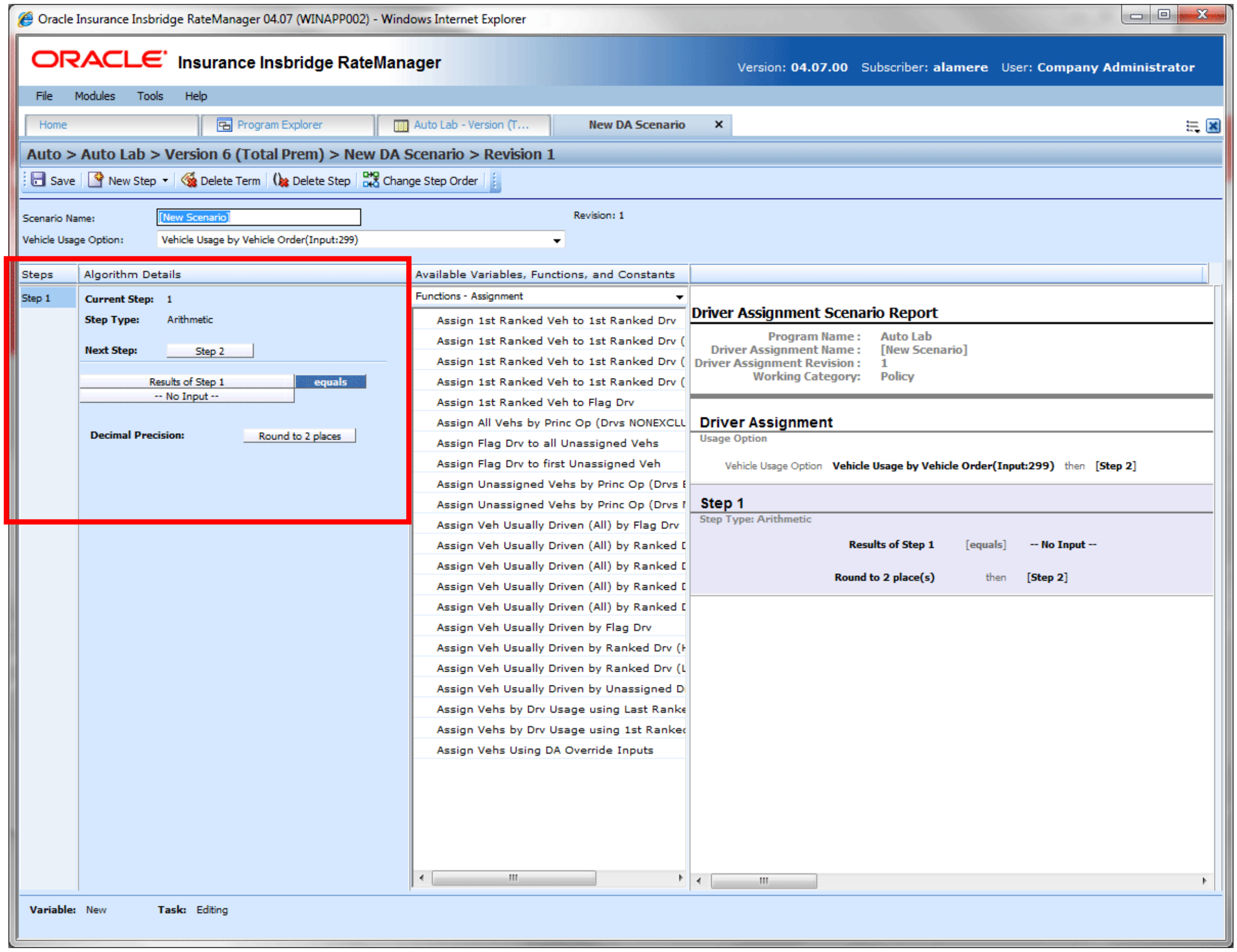
Vehicle Usage by VehicleID (Inputs:299/368)

Vehicle usage is designated as the actual vehicle a driver principally operates, based on the vehicle number specified in the input file. For example, driver 1 principally operates vehicle 3. In RateManager, this is accomplished using the Vehicle input 368 (VehicleID) and Driver input 299 (VehPrincipallyDriven). Specify the vehicle a driver principally operates by setting the VehPrincipallyDriven input 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.


Oracle Insurance Insbridge RateManager 04.07 (WINAPP002) - Windows Internet Explorer
ORACLE Insurance Insbridge RateManager
Version: 04.07.00 Subscriber: alamere User: Company Administrator
File
Modules
Tools
Help
Home
Program Explorer
Auto Lab - Version (T...
New DA Scenario
Save
New Step
Delete Term
Delete Step
Change Step Order
Steps
Step 1
Algorithm Details
Available Variables, Functions, and Constants
Functions - Assignment
Assign 1st Ranked Veh to 1st Ranked Drv
Assign 1st Ranked Veh to 1st Ranked Drv (
Assign 1st Ranked Veh to 1st Ranked Drv (
Assign 1st Ranked Veh to 1st Ranked Drv (
Assign 1st Ranked Veh to Flag Drv
Assign All Vehs by Princ Op (Drvs NONEXCLU
Assign Flag Drv to all Unassigned Vehs
Assign Flag Drv to first Unassigned Veh
Assign Unassigned Vehs by Princ Op (Drvs E
Assign Unassigned Vehs by Princ Op (Drvs N
Assign Veh Usually Driven (All) by Flag Drv
Assign Veh Usually Driven (All) by Ranked D
Assign Veh Usually Driven (All) by Ranked D
Assign Veh Usually Driven (All) by Ranked D
Assign Veh Usually Driven (All) by Ranked D
Assign Veh Usually Driven by Flag Drv
Assign Veh Usually Driven by Ranked Drv (H
Assign Veh Usually Driven by Ranked Drv (L
Assign Veh Usually Driven by Unassigned D
Assign Vehs by Drv Usage using Last Ranke
Assign Vehs by Drv Usage using 1st Ranked
Assign Vehs Using DA Override Inputs

Auto > Auto Lab > Version 6 (Total Prem) > New DA Scenario > Revision 1

Scenario Name: [New Scenario]

Revision: 1

Vehicle Usage Option: Vehicle Usage by Vehicle Order(Input:299)

Current Step: 1

Step Type: Arithmetic

Next Step: Step 2

Results of Step 1 equals
-- No Input --

Decimal Precision: Round to 2 places

Driver Assignment Scenario Report

Program Name : Auto Lab
Driver Assignment Name : [New Scenario]
Driver Assignment Revision : 1
Working Category: Policy

Driver Assignment

Usage Option

Vehicle Usage Option **Vehicle Usage by Vehicle Order(Input:299)** then **[Step 2]**

Step 1

Step Type: Arithmetic

Results of Step 1 [equals] **-- No Input --**

Round to 2 place(s) then **[Step 2]**

Variable: New **Task:** Editing

Figure 223 Algorithm Selection Criteria Field

Fields:

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 inputs/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 9 digits.

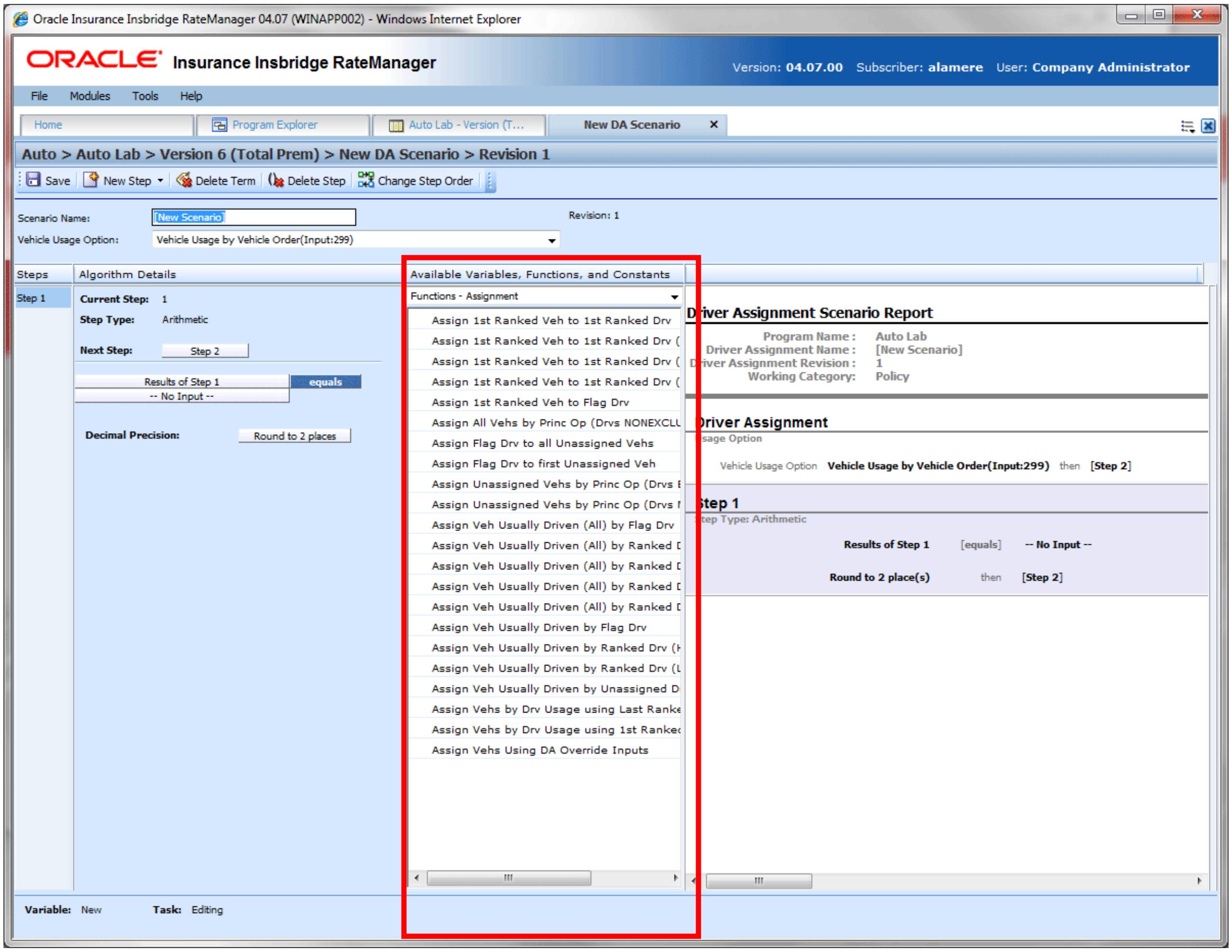
Inputs and variables will have a right click menu available; View Report, View Where Used and Custom Value.

Driver Assignment Tab Available Variables, Functions and Constants

The Available Variables, Functions and Constants area contains a listing of available inputs, variables, and results, sorted by category and system constants that can be used as criteria. For global driver assignments, only global variables or inputs may be used. You cannot use any local variables in the creation of a global driver assignment. A globe icon is used to indicate a global variable.

For Inputs, you will be able to add a new input variable without leaving the Edit Driver Assignment tab. Scroll to the bottom of the Available Variables, Functions and Constants list and select <New Input Variable>. A popup will be displayed that will allow you to enter a new input variable.

You also will be able to add new a new result variable without leaving the Edit Driver Assignment tab.


Oracle Insurance Insbridge RateManager 04.07 (WINAPP002) - Windows Internet Explorer
ORACLE Insurance Insbridge RateManager
Version: 04.07.00 Subscriber: alamere User: Company Administrator
File
Modules
Tools
Help
Home
Program Explorer
Auto Lab - Version (T...
New DA Scenario
Save
New Step
Delete Term
Delete Step
Change Step Order
Steps
Step 1
Algorithm Details
Available Variables, Functions, and Constants

Auto > Auto Lab > Version 6 (Total Prem) > New DA Scenario > Revision 1

Scenario Name: [New Scenario]

Revision: 1

Vehicle Usage Option: Vehicle Usage by Vehicle Order(Input:299)

Current Step: 1

Step Type: Arithmetic

Next Step: Step 2

Results of Step 1 equals

-- No Input --

Decimal Precision: Round to 2 places

Functions - Assignment

Assign 1st Ranked Veh to 1st Ranked Drv
Assign 1st Ranked Veh to 1st Ranked Drv (
Assign 1st Ranked Veh to 1st Ranked Drv (
Assign 1st Ranked Veh to 1st Ranked Drv (
Assign 1st Ranked Veh to Flag Drv
Assign All Vehs by Princ Op (Drvs NONEXCLU
Assign Flag Drv to all Unassigned Vehs
Assign Flag Drv to first Unassigned Veh
Assign Unassigned Vehs by Princ Op (Drvs E
Assign Unassigned Vehs by Princ Op (Drvs N
Assign Veh Usually Driven (All) by Flag Drv
Assign Veh Usually Driven (All) by Ranked D
Assign Veh Usually Driven (All) by Ranked D
Assign Veh Usually Driven (All) by Ranked D
Assign Veh Usually Driven (All) by Ranked D
Assign Veh Usually Driven by Flag Drv
Assign Veh Usually Driven by Ranked Drv (H
Assign Veh Usually Driven by Ranked Drv (L
Assign Veh Usually Driven by Unassigned D
Assign Vehs by Drv Usage using Last Ranke
Assign Vehs by Drv Usage using 1st Ranke
Assign Vehs Using DA Override Inputs

Driver Assignment Scenario Report

Program Name : Auto Lab
Driver Assignment Name : [New Scenario]
Driver Assignment Revision : 1
Working Category: Policy

Driver Assignment

Usage Option

Vehicle Usage Option Vehicle Usage by Vehicle Order(Input:299) then [Step 2]

Step 1

Step Type: Arithmetic

Results of Step 1 [equals] -- No Input --

Round to 2 place(s) then [Step 2]

Variable: New Task: Editing

Figure 224 Available Variables, Function and Constants Field

Fields:

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.

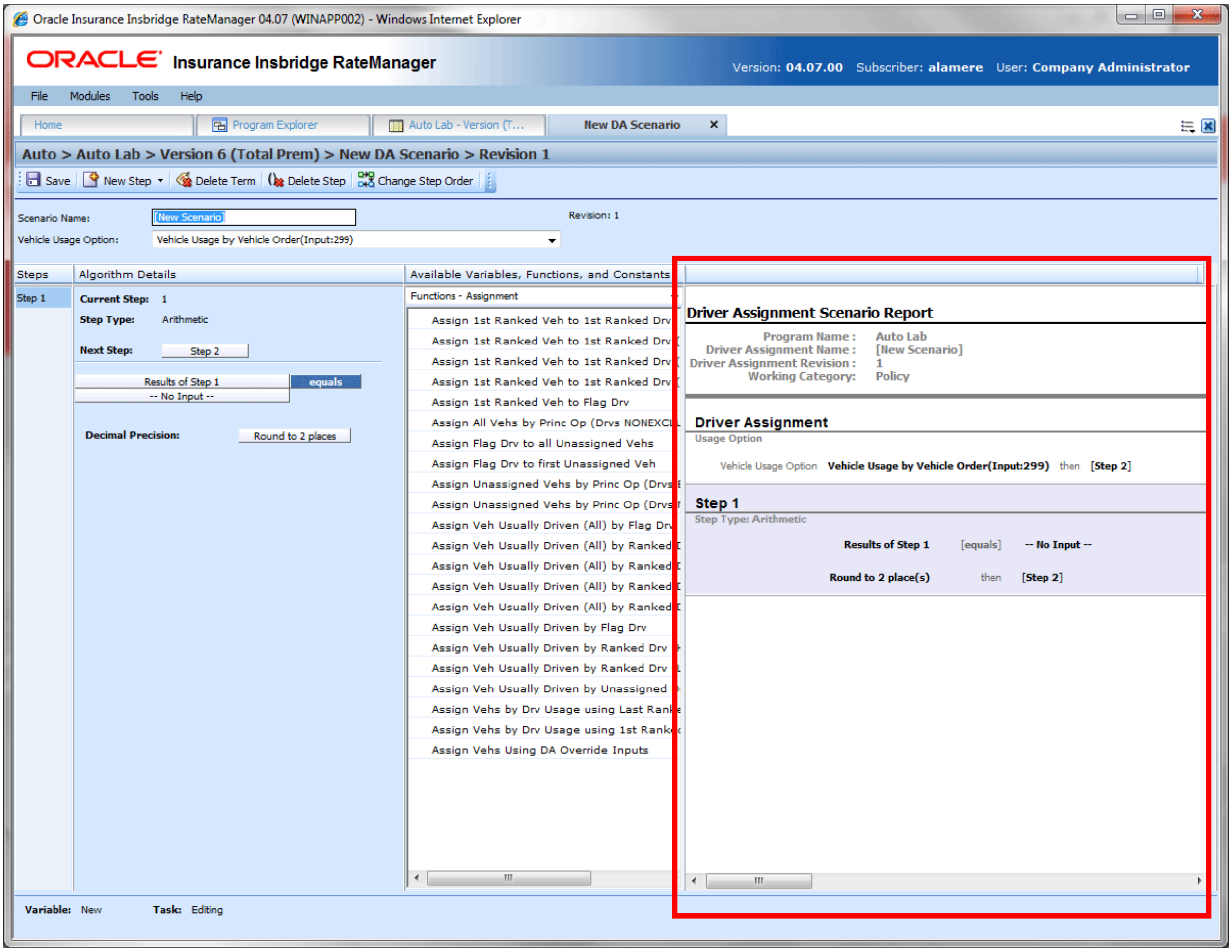
Selection	Description
Functions: Global Versioning	Select the program with the global versioning you want to use.
Functions: Library Callouts	Select a SoftLibrary that has been registered in the IBFA. For callouts, step types do not matter. The entire step will consist of the callout. Callouts are program 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 subline 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	
Constants	Selection of system constants, category instances and system date and time. Or you can create a custom value for use in this program.
Step Variables	Results from the previously entered steps.
Policy Inputs	Inputs 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 will have a scenario report.

Driver assignment algorithms will have an algorithm report.


Oracle Insurance Insbridge RateManager 04.07 (WINAPP002) - Windows Internet Explorer
ORACLE Insurance Insbridge RateManager
Version: 04.07.00 Subscriber: alamere User: Company Administrator
File
Modules
Tools
Help
Home
Program Explorer
Auto Lab - Version (T...
New DA Scenario
Auto > Auto Lab > Version 6 (Total Prem) > New DA Scenario > Revision 1
Save
New Step
Delete Term
Delete Step
Change Step Order
Steps
Step 1
Algorithm Details
Available Variables, Functions, and Constants
Functions - Assignment
Variable: New
Task: Editing

Scenario Name: New Scenario

Revision: 1

Vehicle Usage Option: Vehicle Usage by Vehicle Order(Input:299)

Current Step: 1

Step Type: Arithmetic

Next Step: Step 2

Results of Step 1 equals

-- No Input --

Decimal Precision: Round to 2 places

Assign 1st Ranked Veh to 1st Ranked Drv
Assign 1st Ranked Veh to 1st Ranked Drv
Assign 1st Ranked Veh to 1st Ranked Drv
Assign 1st Ranked Veh to 1st Ranked Drv
Assign 1st Ranked Veh to Flag Drv
Assign All Vehs by Princ Op (Drvs NONEXCL
Assign Flag Drv to all Unassigned Vehs
Assign Flag Drv to first Unassigned Veh
Assign Unassigned Vehs by Princ Op (Drvs
Assign Unassigned Vehs by Princ Op (Drvs
Assign Veh Usually Driven (All) by Flag Drv
Assign Veh Usually Driven (All) by Ranked
Assign Veh Usually Driven (All) by Ranked
Assign Veh Usually Driven (All) by Ranked
Assign Veh Usually Driven (All) by Ranked
Assign Veh Usually Driven by Flag Drv
Assign Veh Usually Driven by Ranked Drv
Assign Veh Usually Driven by Ranked Drv
Assign Veh Usually Driven by Unassigned
Assign Vehs by Drv Usage using Last Rank
Assign Vehs by Drv Usage using 1st Rank
Assign Vehs Using DA Override Inputs

Driver Assignment Scenario Report

Program Name :	Auto Lab
Driver Assignment Name :	[New Scenario]
Driver Assignment Revision :	1
Working Category:	Policy

Driver Assignment

Usage Option

Vehicle Usage Option **Vehicle Usage by Vehicle Order(Input:299)** then **[Step 2]**

Step 1

Step Type: Arithmetic

Results of Step 1 [equals] **-- No Input --**

Round to 2 place(s) then **[Step 2]**

Figure 225 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.

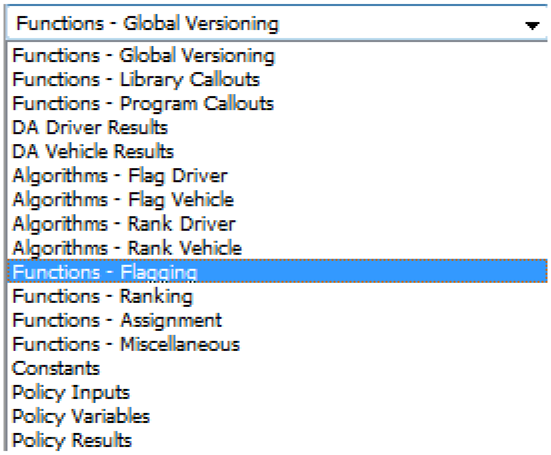


Figure 226 Functions Menu for Driver Assignment Scenario

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.

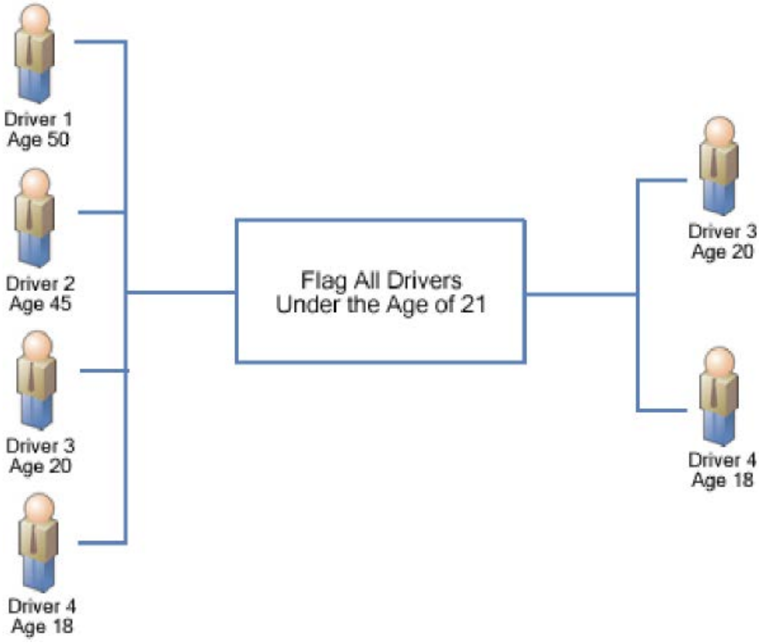
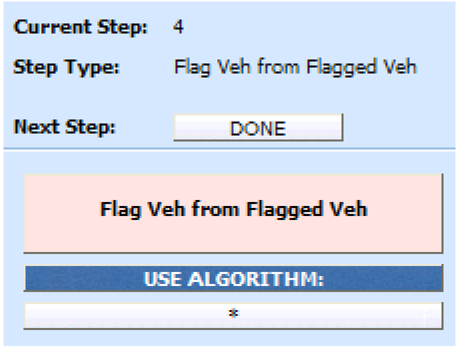


Figure 227 Flagging Functions

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 (*) will appear 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.



Current Step: 4
Step Type: Flag Veh from Flagged Veh
Next Step: DONE

Flag Veh from Flagged Veh

USE ALGORITHM:

*

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 input 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 inputs or variables that can be used in a Rank Driver or Flag Driver algorithms, where vehicle-level inputs 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 input 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.

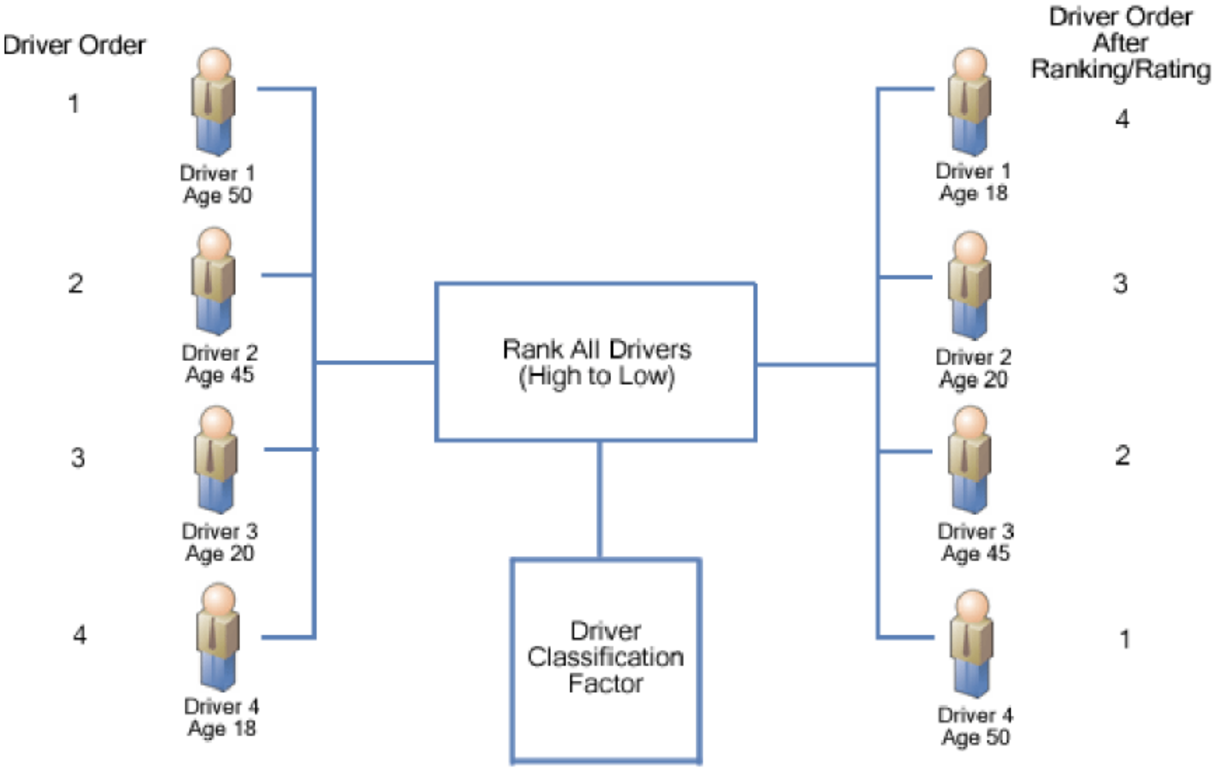
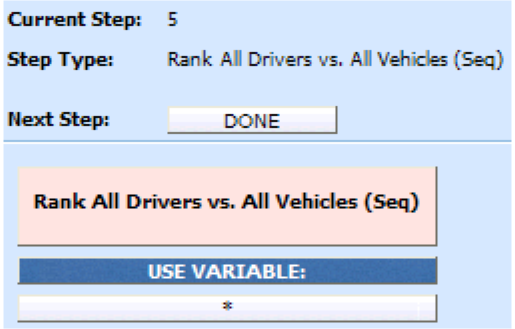


Figure 228 Ranking Functions

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.



Current Step: 5

Step Type: Rank All Drivers vs. All Vehicles (Seq)

Next Step: DONE

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

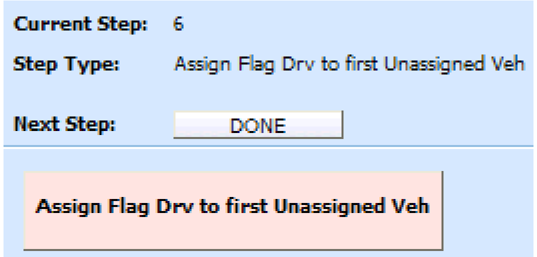
Vehicle Result and **Driver Result** are available from the **Available Variables, Functions, and Constants** drop down under **DA Vehicle Results** and **DA Driver Results,** respectively.

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.



Current Step: 6

Step Type: Assign Flag Drv to first Unassigned Veh

Next Step: DONE

Figure 229 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 an input (input 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 the 1st unassigned vehicle, based on vehicle order. If multiple drivers are flagged, the first driver that was flagged will be assigned to the first unassigned vehicle.

NOTE: *This function is usually used in left over vehicle situations, where there are different criteria for each left over vehicle depending on the number.*

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 an input (input 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 an input (input 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 input 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 inputs 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 inputs 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 input 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 input 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 Inputs

This assignment operation will assign drivers to vehicles based on the information passed in the input file. It requires three inputs:

- The Vehicle input AssignmentOverride (input 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 input DriverID (input ID 367), which contains a driver number for the current driver.
- The Vehicle input AssignedDriver (input 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.

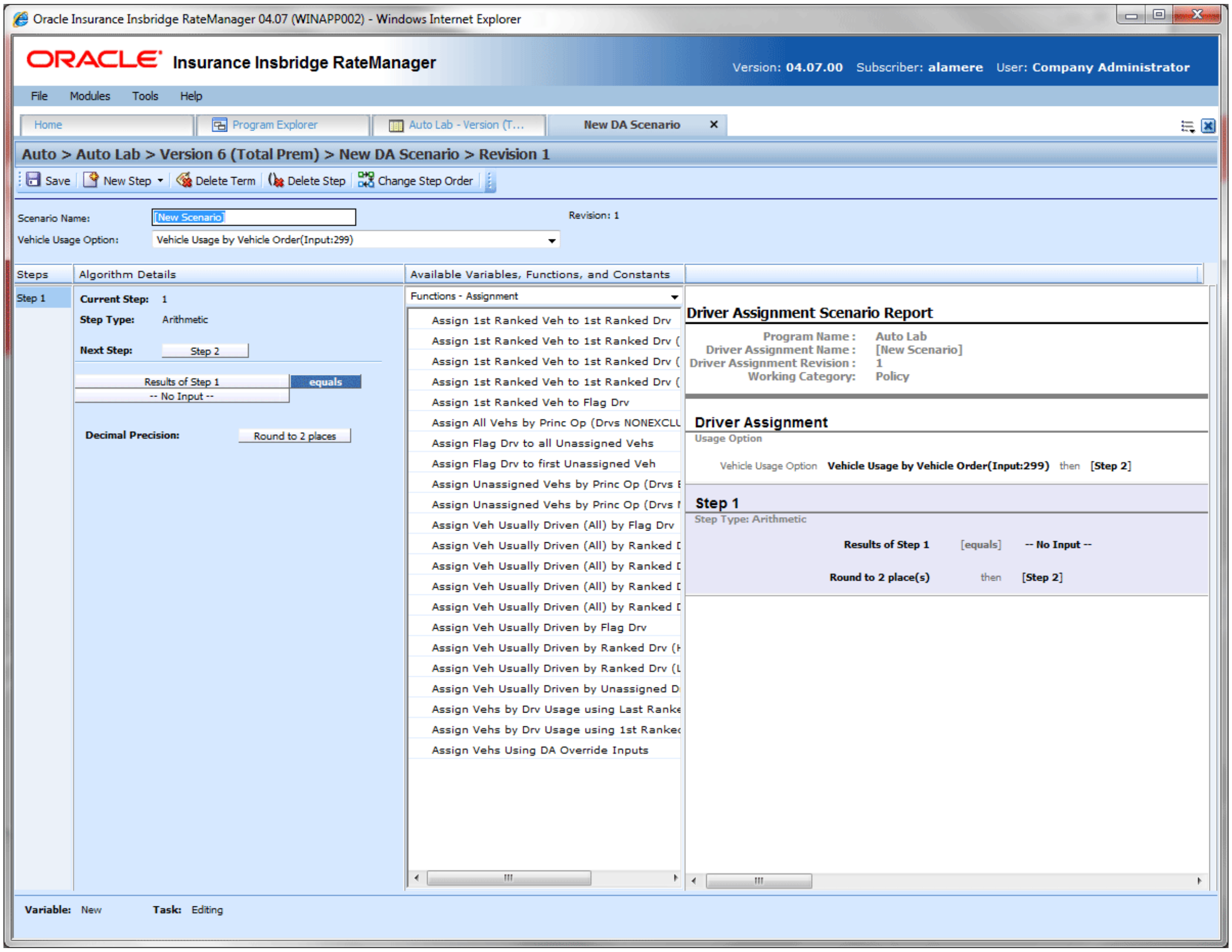
Creating a Driver Assignment

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 subline. Global driver assignments will be available across the subline.

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, inputs 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.


Oracle Insurance Insbridge RateManager 04.07 (WINAPP002) - Windows Internet Explorer
ORACLE Insurance Insbridge RateManager
File
Modules
Tools
Help
Home
Program Explorer
Auto Lab - Version (T...
New DA Scenario
Save
New Step
Delete Term
Delete Step
Change Step Order
Steps
Algorithm Details
Available Variables, Functions, and Constants
Step 1

Version: **04.07.00** Subscriber: **alamere** User: **Company Administrator**

Auto > Auto Lab > Version 6 (Total Prem) > New DA Scenario > Revision 1

Scenario Name: [New Scenario]

Revision: 1

Vehicle Usage Option: Vehicle Usage by Vehicle Order(Input:299)

Current Step: 1

Step Type: Arithmetic

Next Step: Step 2

Results of Step 1 equals

-- No Input --

Decimal Precision: Round to 2 places

Functions - Assignment

- Assign 1st Ranked Veh to 1st Ranked Drv
- Assign 1st Ranked Veh to 1st Ranked Drv (
- Assign 1st Ranked Veh to 1st Ranked Drv (
- Assign 1st Ranked Veh to 1st Ranked Drv (
- Assign 1st Ranked Veh to Flag Drv
- Assign All Vehs by Princ Op (Drvs NONEXCLU
- Assign Flag Drv to all Unassigned Vehs
- Assign Flag Drv to first Unassigned Veh
- Assign Unassigned Vehs by Princ Op (Drvs E
- Assign Unassigned Vehs by Princ Op (Drvs N
- Assign Veh Usually Driven (All) by Flag Drv
- Assign Veh Usually Driven (All) by Ranked D
- Assign Veh Usually Driven (All) by Ranked D
- Assign Veh Usually Driven (All) by Ranked D
- Assign Veh Usually Driven (All) by Ranked D
- Assign Veh Usually Driven by Flag Drv
- Assign Veh Usually Driven by Ranked Drv (H
- Assign Veh Usually Driven by Ranked Drv (L
- Assign Veh Usually Driven by Unassigned D
- Assign Vehs by Drv Usage using Last Ranke
- Assign Vehs by Drv Usage using 1st Ranke
- Assign Vehs Using DA Override Inputs

Driver Assignment Scenario Report

Program Name :	Auto Lab
Driver Assignment Name :	[New Scenario]
Driver Assignment Revision :	1
Working Category:	Policy

Driver Assignment

Usage Option

Vehicle Usage Option **Vehicle Usage by Vehicle Order(Input:299)** then **[Step 2]**

Step 1

Step Type: Arithmetic

Results of Step 1 [equals] **-- No Input --**

Round to 2 place(s) then **[Step 2]**

Variable: New **Task:** Editing

Figure 230 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.

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 popup.

   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. If this is not the screen you want, click **Cancel** to close the popup and return to the previous screen.

NOTE: *Always save a driver assignment algorithm after defining the name, and working category. The driver assignment will not be available to be used within a step until the variable has been saved.*

6. **FOR A SCENARIO:** Choose a **VEHICLE USAGE OPTION**.

NOTE: *Always save a driver assignment scenario after defining the name and vehicle usage option. The driver assignment will not be available to be used within a step until the variable has been saved.*

7. The first step will be 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.

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 will refresh with the new step parameters. Depending upon the step type, you will have to select the appropriate variables, functions, constants or inputs. The value you need to select will be 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 will be populated underneath. Click the “Enter Variable” box and then double-click the corresponding variable you want. The box will refresh 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 an Input Variable, Result Variable or Custom Value Constant

A new input variable, result variable or custom value constant can be created while entering or editing a driver assignment, without leaving the Edit screen.

- To add an **Input Variable**: See Adding a New Input
- To add a **Custom Value Constant**: See Constants
- To add a **Result Variable**: See Adding a Result Variable

EDITING A DRIVER ASSIGNMENT

Editing a driver assignment allows you to change 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 will open 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 will be 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.

COPYING A DRIVER ASSIGNMENT

Any driver assignment may be copied at any time. In your default subline, local driver assignments may be copied to any program within the subline. Global driver assignments can be copied within the same subline. Template generated programs will allow you to copy driver assignments only within the same program. Any copied driver assignment will be open for editing. To make a change to the driver assignment, you must go to the edit tab.

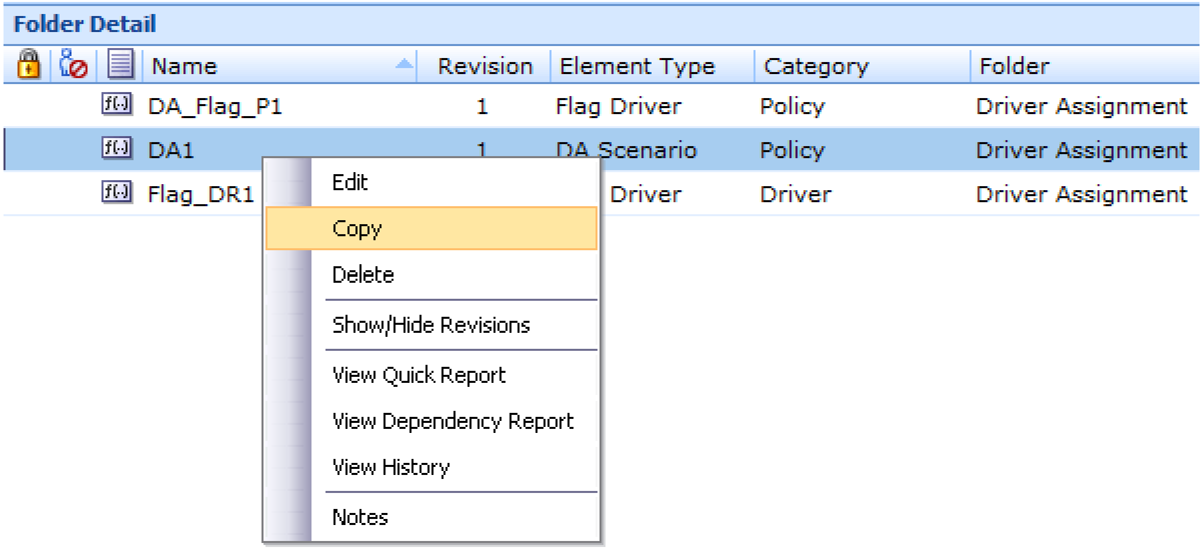
When copying a driver assignment, the structure (including the vehicle usage options/category and steps) is copied. Only the active revision is copied. If you have a driver assignment with 10 revisions, and you copy the driver assignment, only the active revision will be copied into the new drivers assignment. The other 9 revisions will not be copied.

For Local Driver Assignments, 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 subline. This option is for driver assignments within your default subline.

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 it. Select **Copy** from the popup menu.

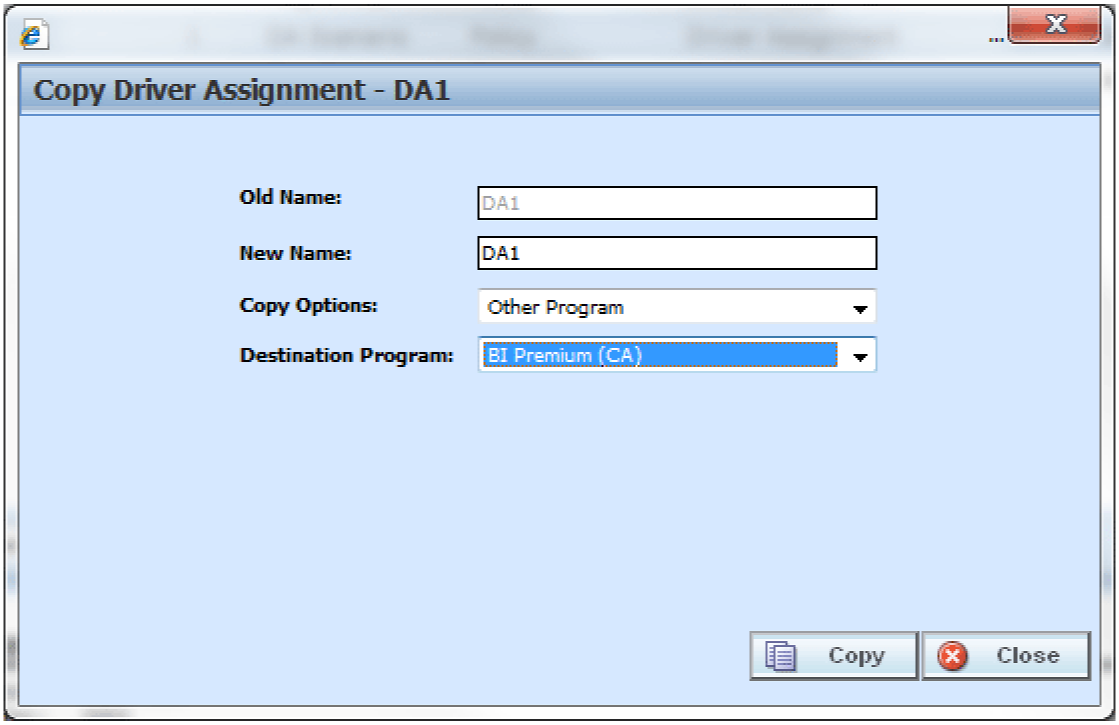


Folder Detail

Name	Revision	Element Type	Category	Folder
DA_Flag_P1	1	Flag Driver	Policy	Driver Assignment
DA1	1	DA Scenario	Policy	Driver Assignment
Flag_DR1		Driver	Driver	Driver Assignment

Figure 231 Copy Driver Assignment Menu

2. A separate popup window is displayed.



Copy Driver Assignment - DA1

Old Name: DA1

New Name: DA1

Copy Options: Other Program

Destination Program: BI Premium (CA)

Figure 232 Copy Driver Assignment Popup

3. Select the **Copy Options**. It is recommended that you change the copy options first. Changing options may reset the name field.

4. If you are copying a driver assignment within your default subline, you can select to copy the driver assignment to another program within the same subline. Select the **Destination Program** from the drop down menu.

5. 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.

6. Click **Copy** to copy the driver assignment. Dependencies will be searched. If there are any dependencies that need to be resolved, a popup window will be displayed. See Resolving Dependencies.

7. 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.

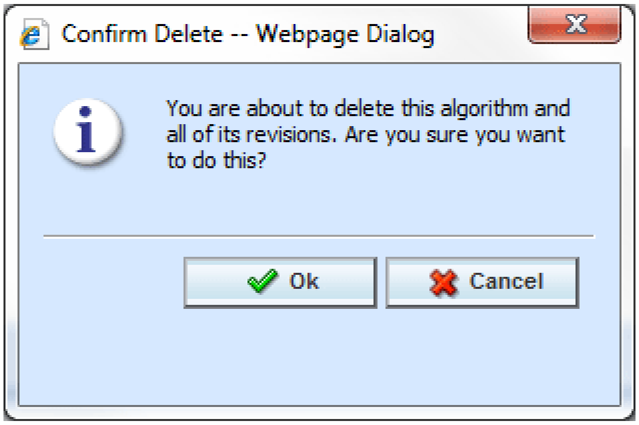
A lock icon indicates that this driver assignment is locked by either a program or a template and is not open for editing or deleting.

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 popup menu.

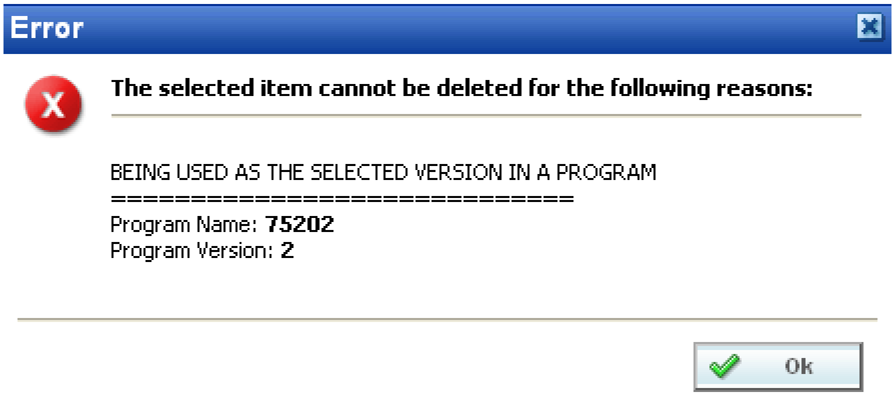


You are about to delete this algorithm and all of its revisions. Are you sure you want to do this?

Figure 233 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.*



Error

The selected item cannot be deleted for the following reasons:

```
BEING USED AS THE SELECTED VERSION IN A PROGRAM
=============================
Program Name: 75202
Program Version: 2
```

Figure 234 Driver Assignment Delete Error Message

Chapter 18

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. When the algorithm or driver assignment is executed, the callout will run the selected program or SoftLibrary and return with results that can be carried further into the rating.

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 subline and LOB, or across an LOB into another subline.

    - **Same Subline:** When a program calls another program within the same subline and LOB.

    - **Crossline:** When a program calls another program in the same or different subline in a different line of business.

- **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. You also can call across one or more LOBs. There is no limit to the number of callouts you can use.

SAME SUBLINE CALLOUTS

Same subline callouts are used to call one or more programs within the same subline.

- This feature is available to any program within the subline.

- No mapping is required. Mapping is automatically matched up. Only exceptions require special mapping.

- Exception mappings are done at the program level and are available to any program version.

For example, you set up smaller programs to handle special functions and then created a master program that calls the special programs when needed. All programs are created in the same LOB under the same subline.

CROSSLINE CALLOUTS

Crossline Callouts allow for a program to callout to another program in the same or different subline in a different line of business. Each subline will be able to set up a single relationship with one other subline per line of business on the Manage Crossline Relationships window.

For example, the Alamere subline in the auto line of business can establish a relationship with the ISO-ERC subline in the BOP line of business. The Alamere subline can set up new relationships with other sublines in other lines of business but cannot set up a second relationship in BOP. If another subline was needed in BOP, then the subline selection would have to be updated.

You can setup a relationship with your own subline in another line of business as well. The Alamere subline in the Home line of business can set up a relationship with the Alamere subline in Auto. This would allow a program in the Alamere subline of Home to make a callout to a program in the Alamere subline of Auto.

You cannot create a relationship within the same line of business. The Alamere subline in the auto line of business cannot create a relationship with the ISO-ERC subline in the auto line of business.

- Relationships are needed if you want to make a callout to a program in another line of business. Relationships must be granted prior to creating any callouts. Until a relationship is granted, the option to callout to another line of business will not be available.

- A crossline callout requires that a relationship be established with the called subline/LOB. Please see Manage Crossline Relationships for more information.

- Crossline callouts must have matching schemas. Categories and inputs must have identical XML IDs for the elements that are to be passed back and forth. If categories are not created identically across the lines, then unexpected results may occur.

Result Variables

When a crossline program is utilized, the called programs result variables are brought in to the calling programs result variables tab. The result variables are displayed with a two character prefix of the LOB where the called program is located followed by and underscore (_). For example, if the called program is in the Home LOB, any result variables listed on the calling programs result tab look like this:

HO_*ResultVariableName*

Where: ResultVariableName is the name of the result variable brought in.

If the LOB does not have a two character abbreviation, then the result variables will begin with an underscore (_).

The result variables from the called program will be available for editing, and usage in the calling program. Any edits will not be applied to the result variable in the called 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 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. 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 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 P2P callout you should have ready:

- **Mapping Exceptions** – If required. A mapping exception is when the input and result variables do not map automatically and require manual mapping or when a specific input or result variable needs to be mapped to another specific input or result variable.

- **Crossline relationships** – If required. A Crossline relationship allows for a program to callout to another program in the same or different subline in a different line of business. If crossline relationships have been created, those will be listed. If only programs from the current subline and LOB are listed, crossline 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.

Creating a P2P Callout

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.

3. Double click the <Select Program> options. A separate window is displayed.

4. Select the callout you want.

5. The callout is listed as a step.

6. If you have a mapping that you want to use, select Program Callout Mappings from the Available Variables, Functions, and Constants.

7. Available mappings will be listed. Select the mapping you need.

8. If you have a program version that you want to use, select the variable that you need.

9. Save your work.

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 select the mapping you want to use. 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 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. Inputs 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

To Create a P2L Callout

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.

3. Available libraries will be listed.

4. Select the library you want.

5. The library is listed as a step.

6. For libraries, you must select the mapping to use. Select Program Callout Mappings from the Available Variables, Functions, and Constants.

7. Available mappings will be listed. Select the mapping you need.

8. Save your work.

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 need 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 will be listed under Functions: Program Callouts or Functions: Library Callouts. Choose the type of callout you want. The available programs or SoftLibraries will be populated underneath. Select the callout you want.

For subline specific callouts, step types will not matter. The entire step will consist of the callout. No other entry can be made. Callout selections will be available under the subline they were created in.

Callout Mappings

Callout mappings map the input and result variables of the calling program to the called program. Callout mappings are needed only for mapping exceptions and for P2L callouts.

When a P2P callout requires a mapping, that is called a mapping exception. A mapping exception is when the input and result variables do not map automatically and require manual mapping or when a specific input or result variable needs to be mapped to another specific input or result variable.

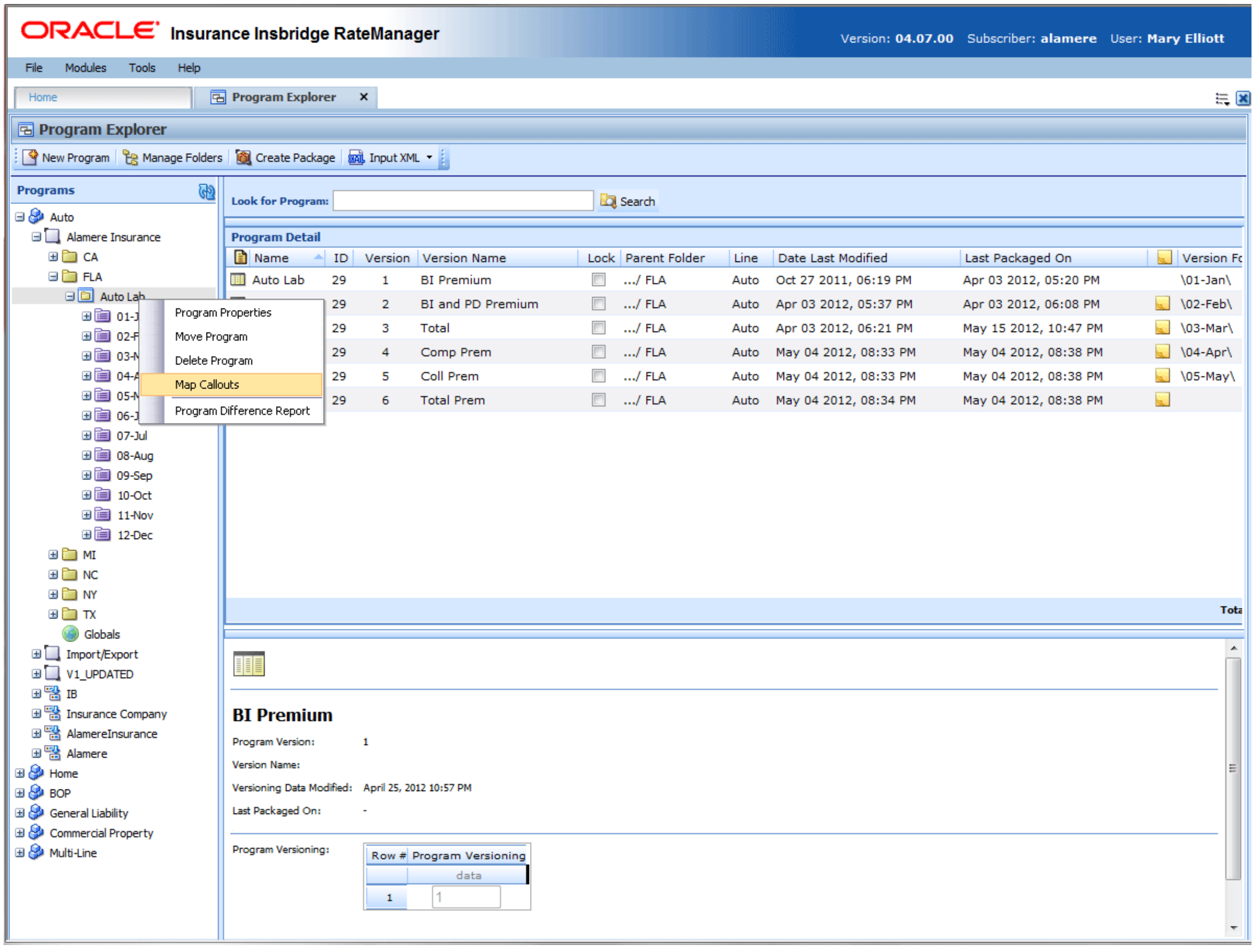
For example, if there is an input variable called garage ZIP in the calling program but in the called program the same input value is required in a variable called parking postal code. If the inputs are not automatically mapped, then a mapping exception is needed.

Callout mappings are done at the program level and are available to every program version.

At this time, the driver/vehicle category found in Drivers Assignments cannot be mapped.

1. Navigate to the LOB→subline→folder→program where you want to map a callout.

2. Right click it and select **Map Callouts** from the popup menu.



Program Detail

Name	ID	Version	Version Name	Lock	Parent Folder	Line	Date Last Modified	Last Packaged On		Version Fc
Auto Lab	29	1	BI Premium	□	.../ FLA	Auto	Oct 27 2011, 06:19 PM	Apr 03 2012, 05:20 PM		\01-Jan\
	29	2	BI and PD Premium	□	.../ FLA	Auto	Apr 03 2012, 05:37 PM	Apr 03 2012, 06:08 PM	🟨	\02-Feb\
	29	3	Total	□	.../ FLA	Auto	Apr 03 2012, 06:21 PM	May 15 2012, 10:47 PM	🟨	\03-Mar\
	29	4	Comp Prem	□	.../ FLA	Auto	May 04 2012, 08:33 PM	May 04 2012, 08:38 PM	🟨	\04-Apr\
	29	5	Coll Prem	□	.../ FLA	Auto	May 04 2012, 08:33 PM	May 04 2012, 08:38 PM	🟨	\05-May\
	29	6	Total Prem	□	.../ FLA	Auto	May 04 2012, 08:34 PM	May 04 2012, 08:38 PM	🟨	

BI Premium

Program Version: 1

Version Name:

Versioning Data Modified: April 25, 2012 10:57 PM

Last Packaged On: -

Program Versioning:

Row #	Program Versioning
	data
1	1

Figure 235 Navigating to Callouts

3. The Map Callouts screen is displayed. If any callouts have been configured, they will be listed.

Mapping Callouts

Mapping must be performed for the callout to run successfully. At this time, the driver/vehicle category found in Drivers Assignments cannot be mapped.

For mapping exceptions, you must set the Input and Output mappings for each program. SoftLibraries require mappings.

FUNCTIONALITY

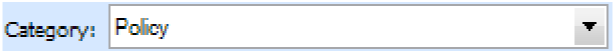
The Menu Bar is located at the top of the tab. The menu bar contains the functions you can perform on this tab.

Functions:

Clear Mappings	**Clear Mappings:**
Save	**Save:** Saves the mapping.
Map Matched Inputs	**Map Matched Inputs:** Creates a new step in the algorithm. For more information, see Working with Steps. **NOTE:** *For Input Mapping Only.*
Map Matched Result Variables	**Map Matched Result Variables:** Creates a new step in the algorithm. For more information, see Working with Steps. **NOTE:** *For Output Mapping Only*.
Refresh	**Refresh:** 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.

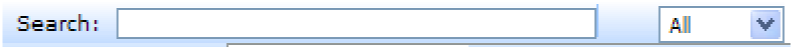
Selecting Categories on Mapping Inputs

You can select the categories you want to map using the category field at the top of the screen.


Category: Policy

Searching on Mapping Outputs

You can search for result variables on the SoftLibrary mapping screens.


Search:
All

Enter in your search criteria in the search window. Select if you want to search all variables or only table variables.

Results will be displayed in the left hand side of the screen.

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. Global and local elements are created, edited, copied, and deleted in the same way. Global and local elements are also revisioned in the same way.

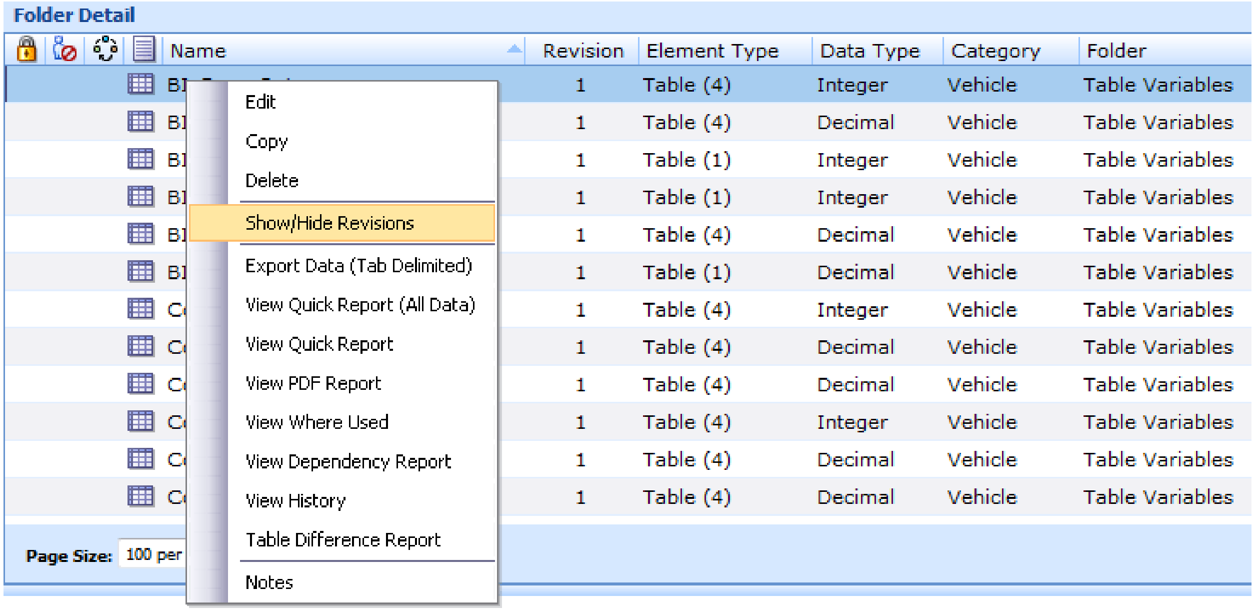
Input and Result variables do not have revisions.

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 may 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** from the popup menu.



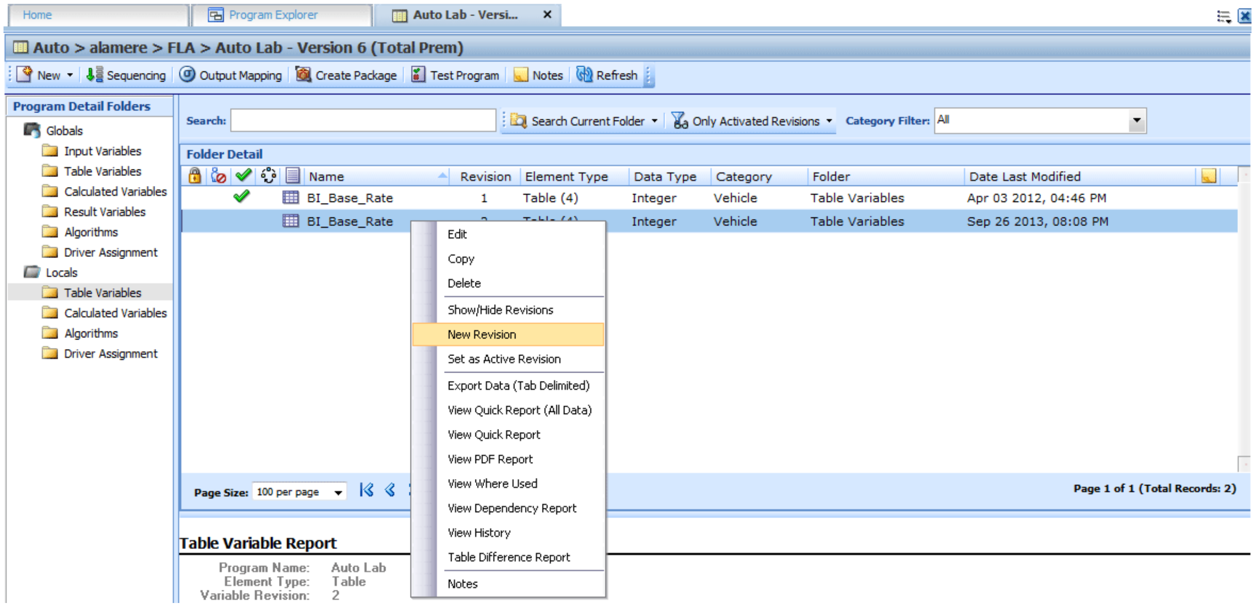
Folder Detail

Name	Revision	Element Type	Data Type	Category	Folder
BI	1	Table (4)	Integer	Vehicle	Table Variables
BI	1	Table (4)	Decimal	Vehicle	Table Variables
BI	1	Table (1)	Integer	Vehicle	Table Variables
BI	1	Table (1)	Integer	Vehicle	Table Variables
BI	1	Table (4)	Decimal	Vehicle	Table Variables
BI	1	Table (1)	Decimal	Vehicle	Table Variables
C	1	Table (4)	Integer	Vehicle	Table Variables
C	1	Table (4)	Decimal	Vehicle	Table Variables
C	1	Table (4)	Decimal	Vehicle	Table Variables
C	1	Table (4)	Integer	Vehicle	Table Variables
C	1	Table (4)	Decimal	Vehicle	Table Variables
C	1	Table (4)	Decimal	Vehicle	Table Variables

Page Size: 100 per

Figure 236 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 popup 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 will need to enter that information first.



Auto > alamere > FLA > Auto Lab - Version 6 (Total Prem)

Search: Category Filter: All

Folder Detail

✓	Name	Revision	Element Type	Data Type	Category	Folder	Date Last Modified
✓	BI_Base_Rate	1	Table (4)	Integer	Vehicle	Table Variables	Apr 03 2012, 04:46 PM
	BI_Base_Rate	2	Table (4)	Integer	Vehicle	Table Variables	Sep 26 2013, 08:08 PM

Page Size: 100 per page

Page 1 of 1 (Total Records: 2)

Table Variable Report

Program Name: Auto Lab
Element Type: Table
Variable Revision: 2

Figure 237 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.

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.

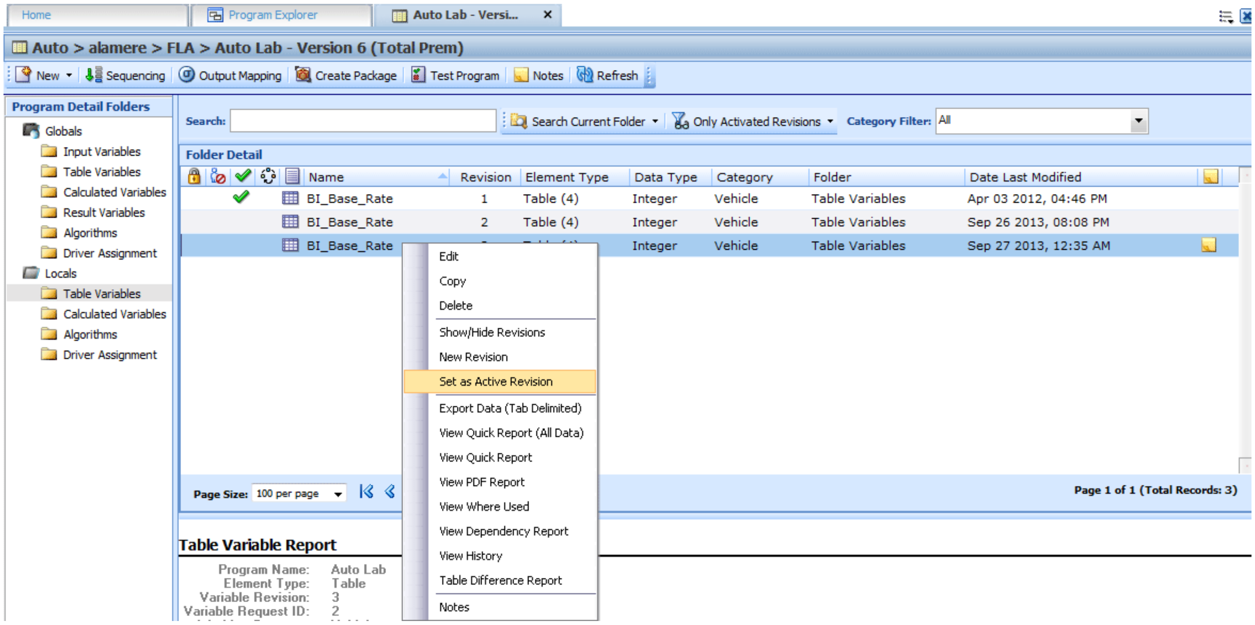
The active revision is indicated by the ✓ green checkmark.

The default active revision is revision 1. If there is another revision that you want to use, go into the program details tab and set the revision there.

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 it. Select **Show/Hide Revisions** from the popup menu.



Auto > alamere > FLA > Auto Lab - Version 6 (Total Prem)

Search:

Category Filter: All

	Name	Revision	Element Type	Data Type	Category	Folder	Date Last Modified
✓	BI_Base_Rate	1	Table (4)	Integer	Vehicle	Table Variables	Apr 03 2012, 04:46 PM
	BI_Base_Rate	2	Table (4)	Integer	Vehicle	Table Variables	Sep 26 2013, 08:08 PM
	BI_Base_Rate	[illegible]	[illegible]	Integer	Vehicle	Table Variables	Sep 27 2013, 12:35 AM

Page Size: 100 per page

Page 1 of 1 (Total Records: 3)

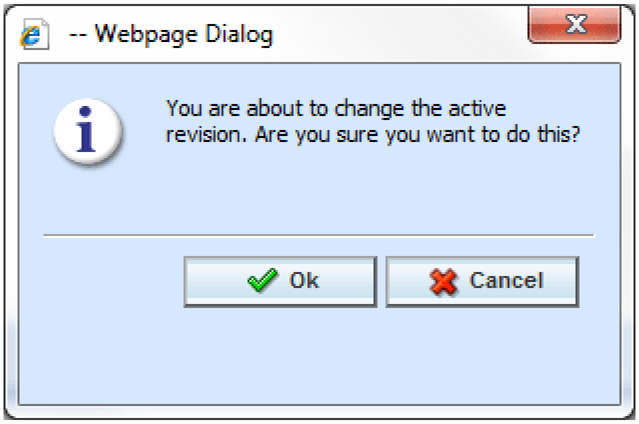
Table Variable Report

Program Name: Auto Lab
Element Type: Table
Variable Revision: 3
Variable Request ID: 2

Figure 238 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 will be asked to confirm changing the active revision.



You are about to change the active revision. Are you sure you want to do this?

Figure 239 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 will have the green checkmark ✓ in front.

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 will save time if the revision is needed and will not slow down rating.

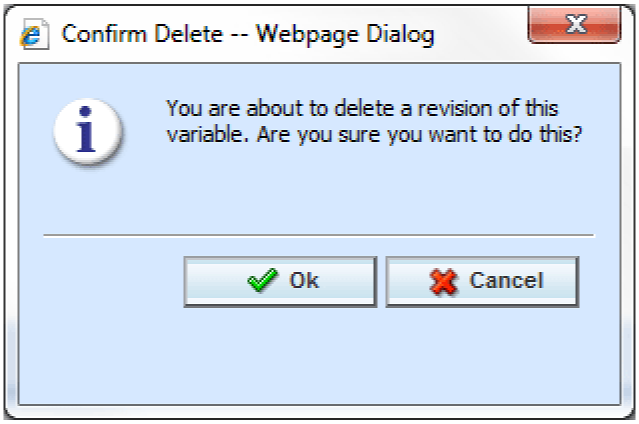
To Delete an Element Revision

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 it. Select **Show/Hide Revisions** from the popup menu.

3. Select the revision you would like to delete and right click it. Select **DELETE** from the popup menu.

4. You will be asked to confirm the deletion of the element revision.



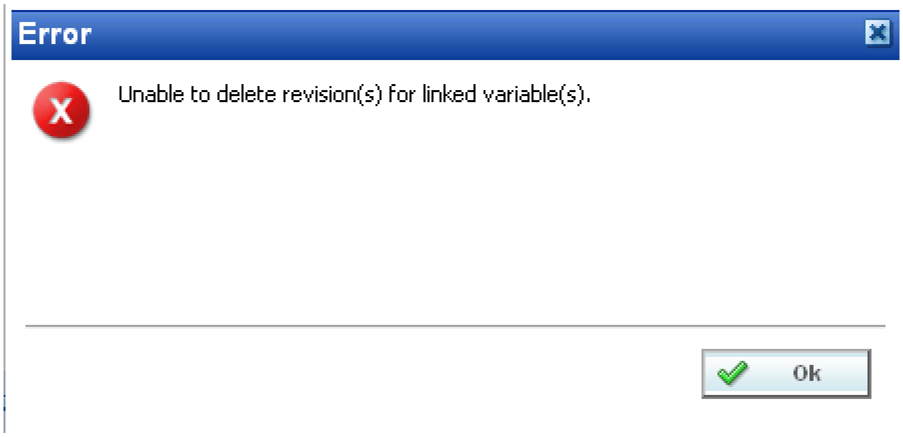
You are about to delete a revision of this variable. Are you sure you want to do this?

Figure 240 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.



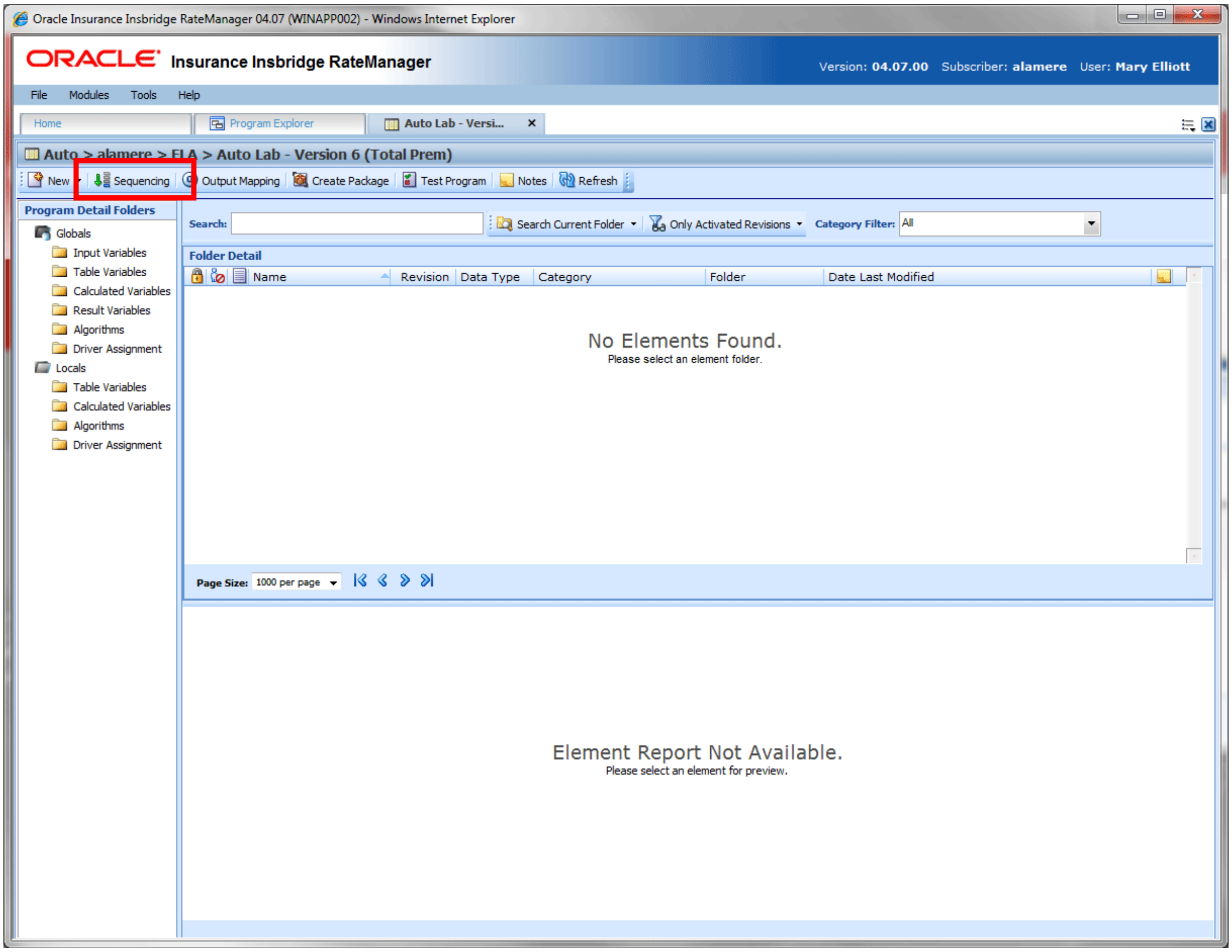
Unable to delete revision(s) for linked variable(s).

Figure 241 Unable to Delete Revision Due to Linked Variable

Chapter 20

INTRODUCTION TO SEQUENCING

Sequencing allows you to activate and order algorithms and underwriting rules. After creating your algorithms, you need to 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.



Version: 04.07.00 Subscriber: alamere User: Mary Elliott

Auto > alamere > FLA > Auto Lab - Version 6 (Total Prem)

Category Filter: All

No Elements Found.
Please select an element folder.

Page Size: 1000 per page

Element Report Not Available.
Please select an element for preview.

Figure 242 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 Subline→Folder→Program→Program Version where you want to view or manage sequences. This will open the Program Details tab where you can access the Sequencing tab.

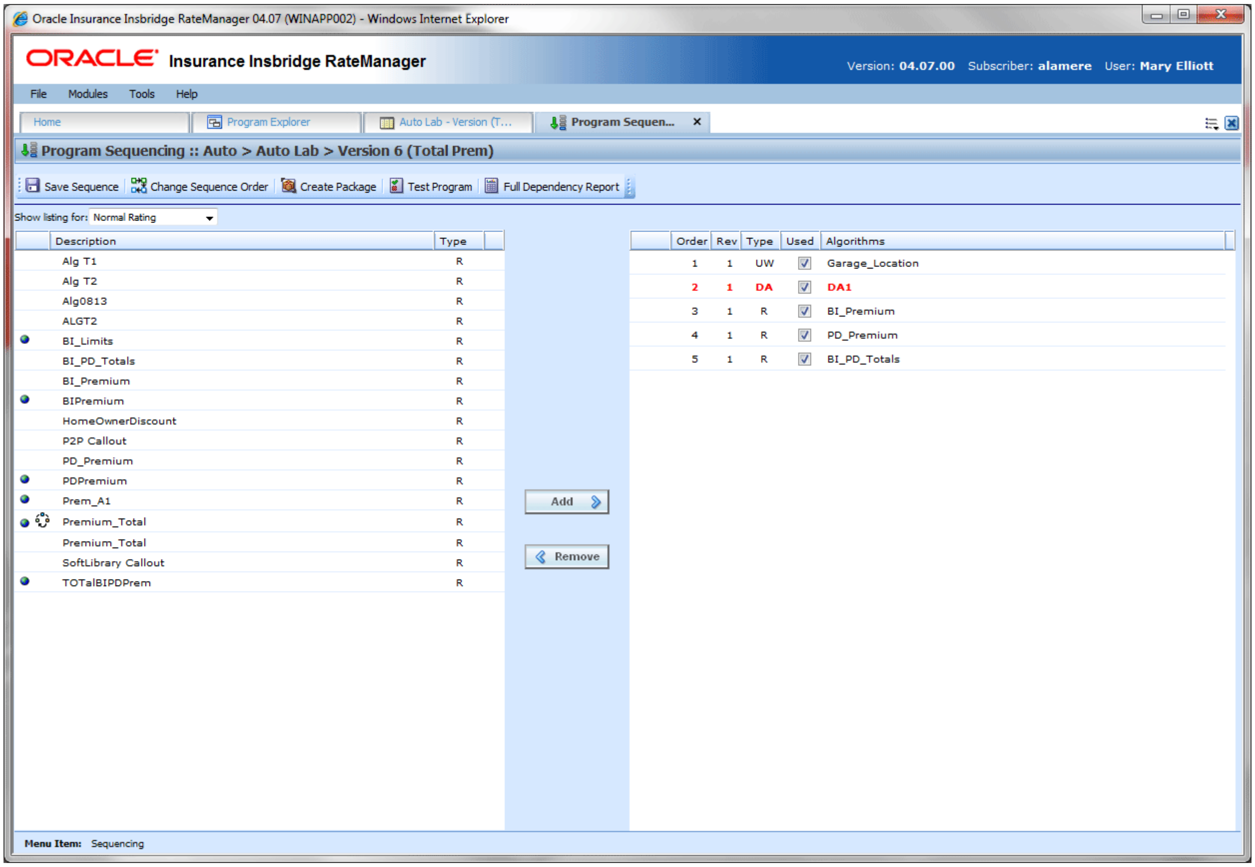
1. Modules→Program Explorer. This opens the Program Explorer tab.

2. Select the Line of Business→Subline→Program→Program Version. This will open 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. To create a sequence, begin on the left hand side of the tab. Select the element you want to run first and click the Add button in the center of the tab. The element will be placed on the right side of the tab. Continue until you have all the elements you want to run in the order you want them to run.


Oracle Insurance Insbridge RateManager 04.07 (WINAPP002) - Windows Internet Explorer
ORACLE Insurance Insbridge RateManager
Version: 04.07.00 Subscriber: alamere User: Mary Elliott
File
Modules
Tools
Help
Home
Program Explorer
Auto Lab - Version (T...
Program Sequen...
Save Sequence
Change Sequence Order
Create Package
Test Program
Full Dependency Report
Add
Remove

Program Sequencing :: Auto > Auto Lab > Version 6 (Total Prem)

Show listing for: Normal Rating

	Description	Type
	Alg T1	R
	Alg T2	R
	Alg0813	R
	ALGT2	R
●	BI_Limits	R
	BI_PD_Totals	R
	BI_Premium	R
●	BIPremium	R
	HomeOwnerDiscount	R
	P2P Callout	R
	PD_Premium	R
●	PDPremium	R
●	Prem_A1	R
● ⚙	Premium_Total	R
	Premium_Total	R
	SoftLibrary Callout	R
●	TOTalBIPDPrem	R

Order	Rev	Type	Used	Algorithms
1	1	UW	☑	Garage_Location
2	1	DA	☑	DA1
3	1	R	☑	BI_Premium
4	1	R	☑	PD_Premium
5	1	R	☑	BI_PD_Totals

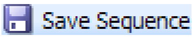
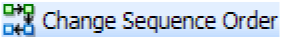
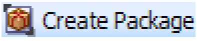
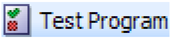
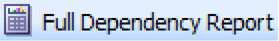
Menu Item: Sequencing

Figure 243 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.
- 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	**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	**Change Sequence Order:** Allows you to change the order of the sequences without using the drop and drag feature.
 Create Package	**Create Package:** Packages the program for rating and testing. See Introduction to Packaging for more information.
 Test Program	**Test Program:** Opens the Test Program, which allows you to test and debug a program within RateManager. See Introduction to the Test Program for more information.
 Full Dependency Report	**Full Dependency Report:** Opens the dependency
Show listing for: Normal Rating	**Show Listing For:** Allows you to filter the list of available elements to show one of the following: • All • Normal Rating • Underwriting Rule • DA Scenarios - on the Auto LOB only

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.*

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 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.

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.

- **Selected 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.

- **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 remove the last algorithm from a 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.

You can add as many algorithms and driver assignments as you need. The first algorithm you add to the selected column 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 green 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.

Once a sequence has been created, it cannot be deleted, only changed.

1. To add an algorithm or driver assignment to the sequencing, you can either highlight it and click **Add** or double-click the algorithm or driver assignment.

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.*

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 input 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 an input for the working category where that algorithm was created, then program will not package.

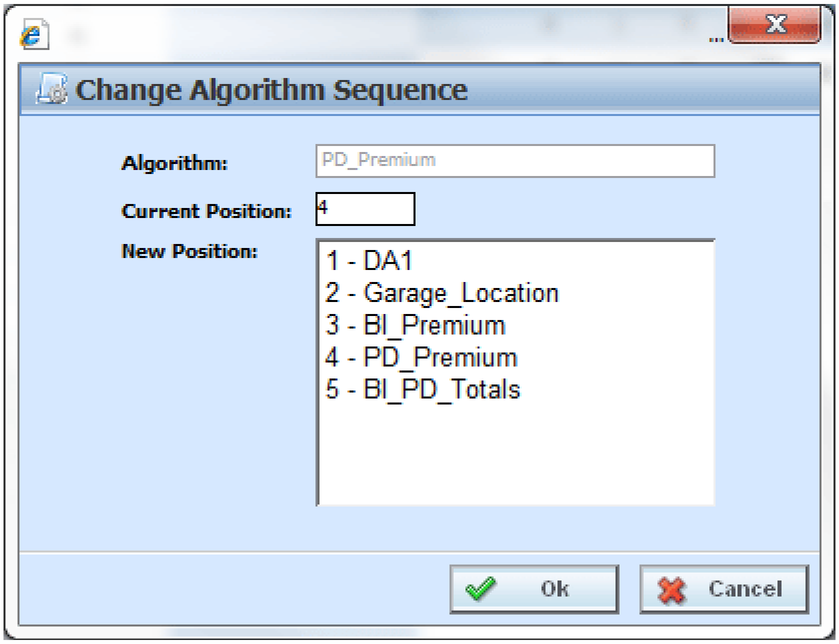
Changing Sequence Order

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.



Change Algorithm Sequence

Algorithm: PD_Premium

Current Position: 4

New Position:

1 - DA1
2 - Garage_Location
3 - BI_Premium
4 - PD_Premium
5 - BI_PD_Totals

Figure 244 Change Sequence

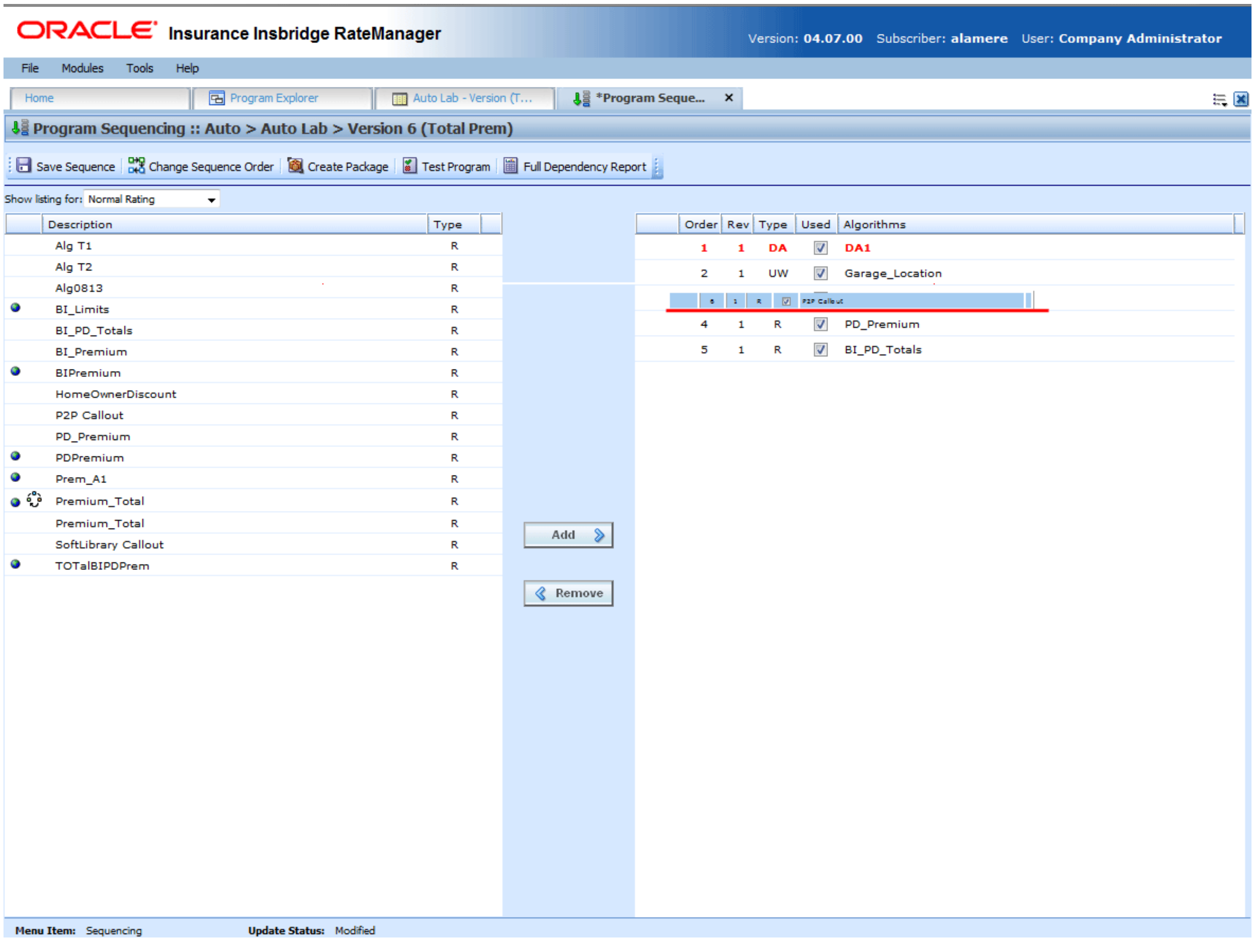
2. The current position will be displayed at the top of the screen. Select where you want the algorithm or driver assignment to be placed.

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:

To use the **drop and drag** method, highlight the algorithm or driver assignment you want to move, then drag it to the new location and drop it.

If you want to move an algorithm farther up than you can see or farther down than you can see, you can use the keyboard's arrow keys or you mouse wheel (if so equipped) to move the scroll bar up or down, while dragging the algorithm.



Program Sequencing :: Auto > Auto Lab > Version 6 (Total Prem)

Show listing for: Normal Rating

	Description	Type
	Alg T1	R
	Alg T2	R
	Alg0813	R
●	BI_Limits	R
	BI_PD_Totals	R
	BI_Premium	R
●	BIPremium	R
	HomeOwnerDiscount	R
	P2P Callout	R
	PD_Premium	R
●	PDPremium	R
●	Prem_A1	R
●	Premium_Total	R
	Premium_Total	R
	SoftLibrary Callout	R
●	TOTalBIPDPrem	R

Order	Rev	Type	Used	Algorithms
1	1	DA	☑	DA1
2	1	UW	☑	Garage_Location
6	1	R	☑	P2P Callout
4	1	R	☑	PD_Premium
5	1	R	☑	BI_PD_Totals

Menu Item: Sequencing Update Status: Modified

Figure 245 Changing Sequence Order Using Drop and Drag

When you are dragging the element you want to move, it will be encased in a blue box. A white line indicates where the element is currently positioned.

Removing an Element from a Sequence

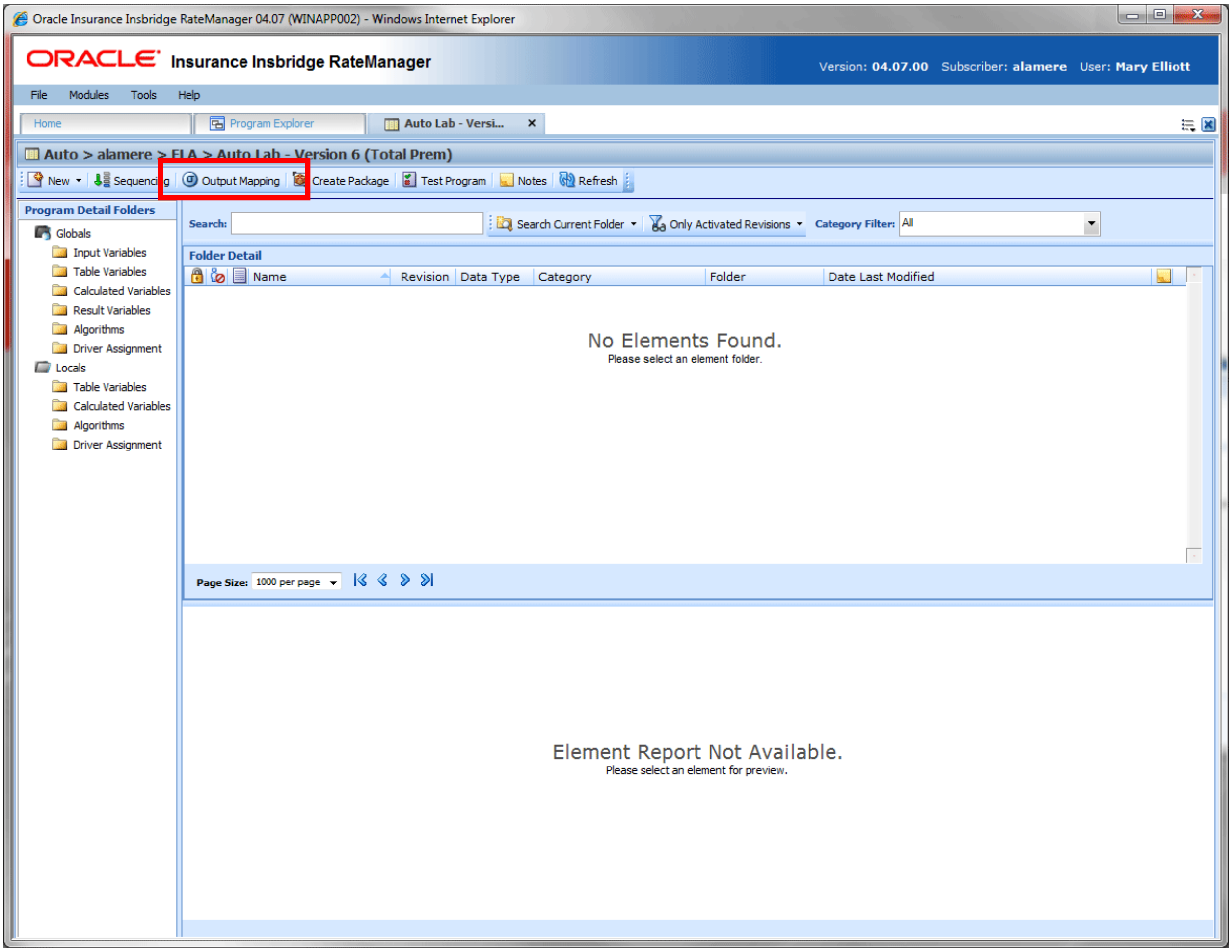
To remove an algorithm or driver assignment from the sequencing, highlight the element you want to remove and click **Remove**. The element will be removed from the selected list.

You cannot have a sequence without any algorithms or driver assignments. There will always be at least one algorithm or driver assignment. You will not be able to delete the last remaining 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 allows you to select the results that you want displayed in the output file. Whether it is a result, input, table or calculated variable, global or local, there is no limit to the type or number of results you can set. 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.



Version: 04.07.00 Subscriber: alamere User: Mary Elliott

Auto > alamere > FLA > Auto Lab - Version 6 (Total Prem)

Category Filter: All

No Elements Found.
Please select an element folder.

Page Size: 1000 per page

Element Report Not Available.
Please select an element for preview.

Figure 246 Navigating to Output Mapping Tab

Navigating to Output Mapping

Output mappings 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 Line of Business→Subline→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.

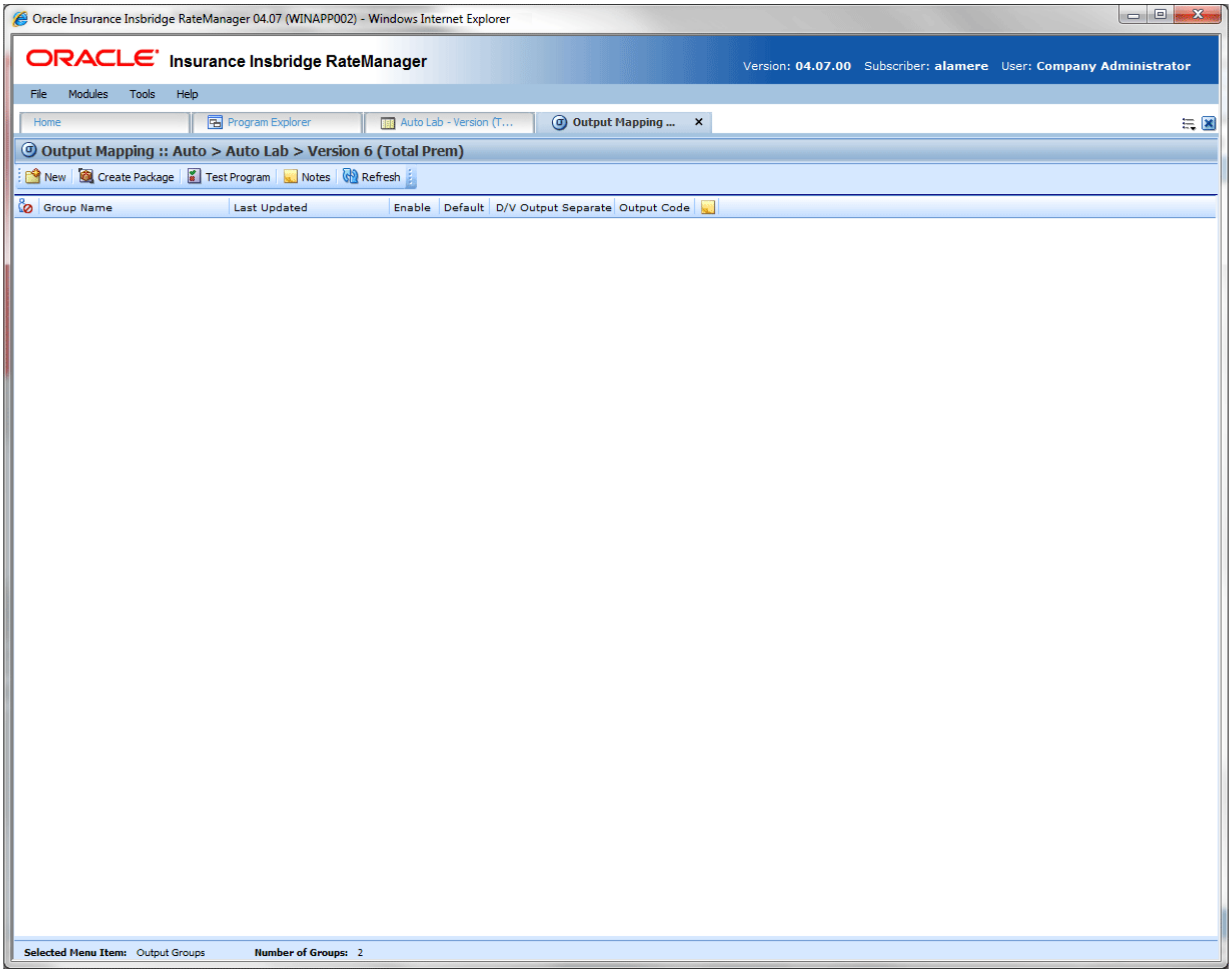
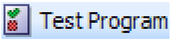
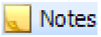

Oracle Insurance Insbridge RateManager 04.07 (WINAPP002) - Windows Internet Explorer
ORACLE Insurance Insbridge RateManager
Version: 04.07.00 Subscriber: alamere User: Company Administrator
File
Modules
Tools
Help
Home
Program Explorer
Auto Lab - Version (T...
Output Mapping ...
Output Mapping :: Auto > Auto Lab > Version 6 (Total Prem)
New
Create Package
Test Program
Notes
Refresh
Group Name
Last Updated
Enable
Default
D/V Output Separate
Output Code
Selected Menu Item: Output Groups
Number of Groups: 2

Figure 247 Output Mapping Tab

Output Mapping Functionality

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

New	**New:** Creates a new output group.
Create Package	**Create Package:** Packages the program for rating and testing. See Packaging for more information.

 Test Program	**Test Program:** Opens Test Case Editor where you can test and debug a program within RateManager. See Test Case Editor for more information.
 Notes	**Notes:** Enter, edit, and view notes to the output mapping.
Refresh	**Refresh:** Refreshes the screen with the current information.

OUTPUT MAPPING FIELDS

Shows a listing of the output groups 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.

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 input XML.

D/V Output Separate (Auto LOB 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 LOB 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.

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**: Allows you to change 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

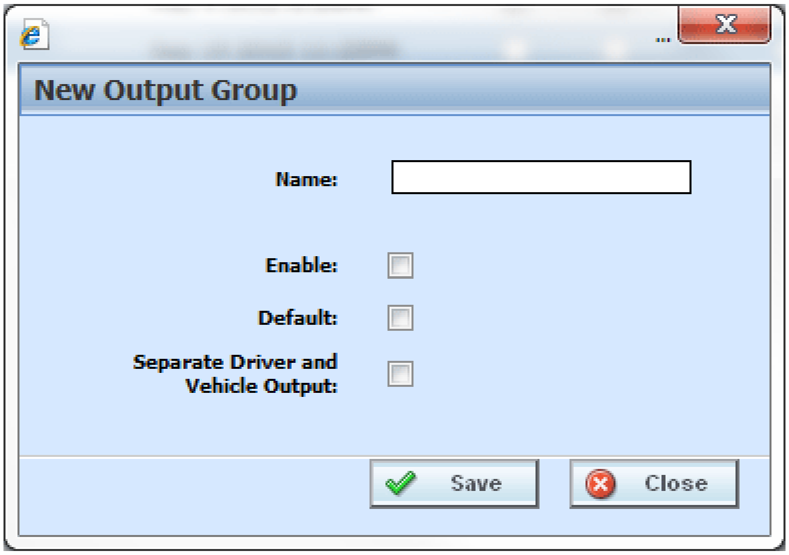
RateManager supports an unlimited number of output groups for any program version. Every program must have at least one output group. Each output group contains a specific set of elements that will be output when that group is used. The Output Mapping tab allows you to create, edit, copy and delete an output group.

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** from the menu bar. This opens the **New Output Group** window.



New Output Group

Name:

Enable: □

Default: □

Separate Driver and Vehicle Output: □

Figure 248 Entering a New Output Group

NOTE: *The Separate Driver and Vehicle Output option will only be available on Output Mappings in the Auto LOB.*

3. Enter a name for the new output group. 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.

4. Click **Save** when you have finished to save the new output group and update the output group listing.

Default Output Mapping

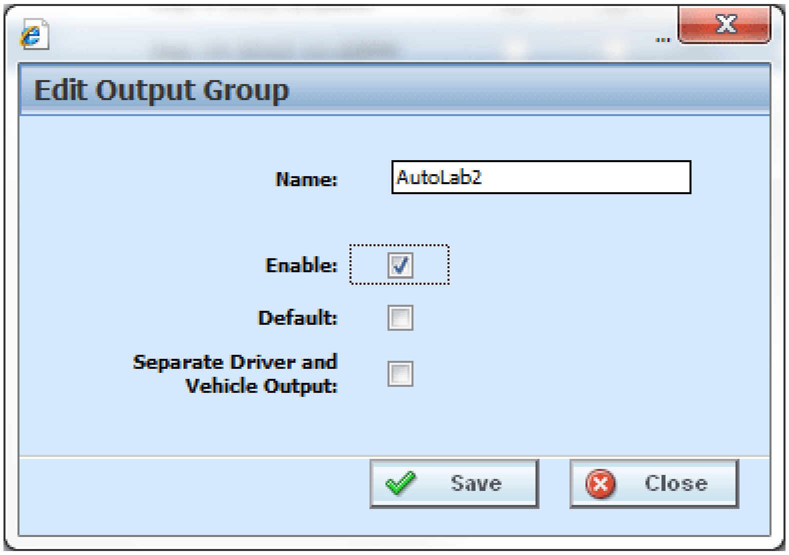
The default setting signifies that this output mapping group will be used if no output mapping group is specified for the program version.

EDITING GROUP OPTIONS

Editing an output group allows you to change the name of an output group, as well as, whether it is enabled and/or the default. For an auto line of business 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 **Edit Output Group** from the menu. This opens the Edit Output Group window.



Edit Output Group

Name: AutoLab2

Enable: ☑

Default: □

Separate Driver and Vehicle Output: □

Figure 249 Edit Output Group

3. Type in a new name for the output group. To enable the output group, select the **Enable** option. To set the group as the default, select the **Default** option. Click **Save** when you have finished to save the new output group and update the output group tab listing.
4. For an Auto program, you also will have the option to **Separate Driver and Vehicle Output**. For information on this option, see the driver assignment topic Separating Driver and Vehicle Output.

Edit Output Group Mapping Tab

You will 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 will need to add variables based on the categories used.

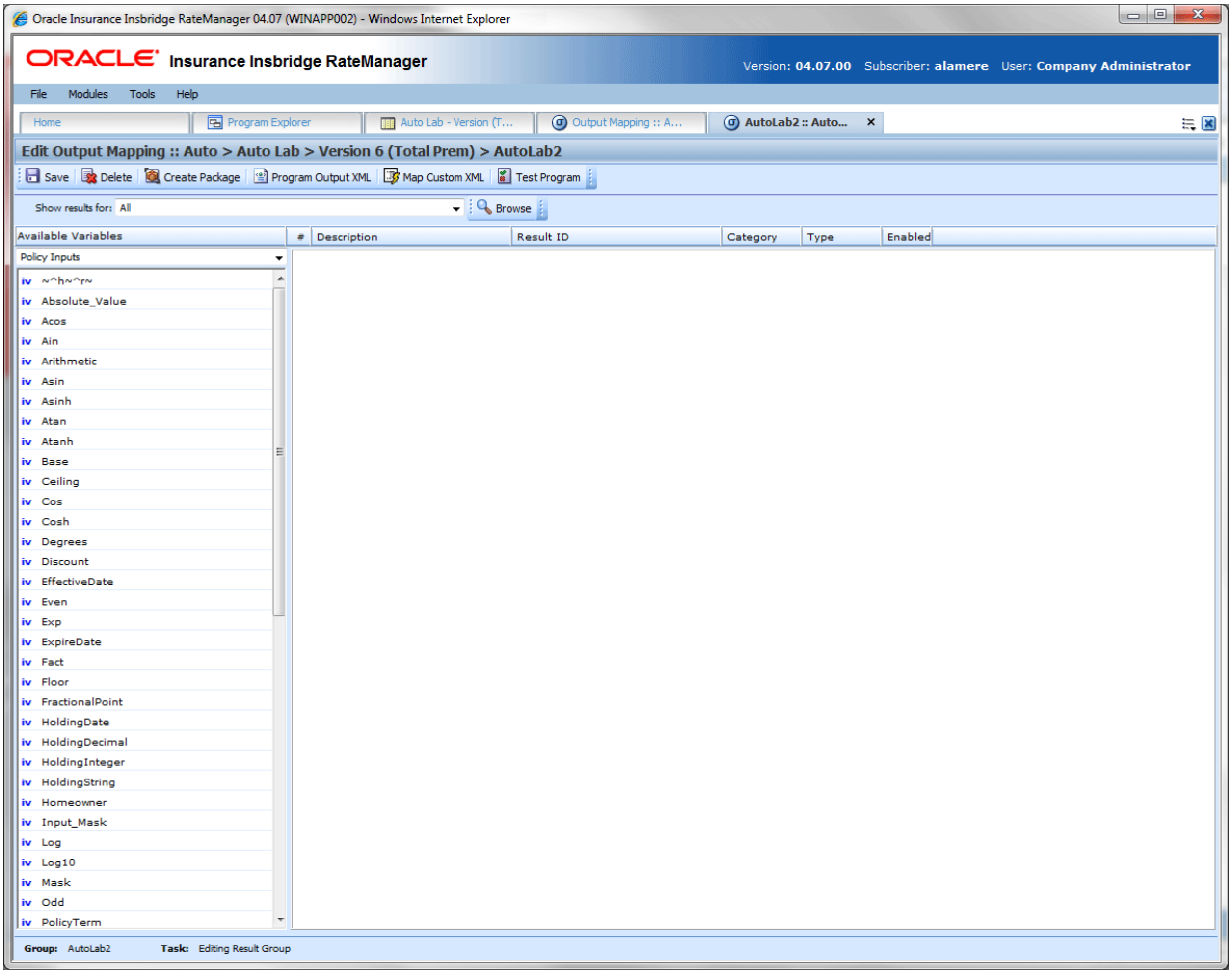
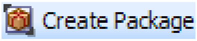
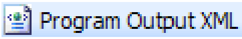
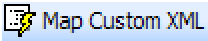
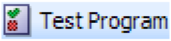


Figure 250 Editing an Output Mapping

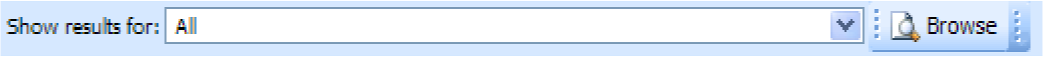
Edit Output Group Mapping Tab Functionality

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 a search option. The top bar menu has the options for creating packages, testing and working with custom XML.

Save	**Save:** Saves the output mapping.
Delete	**Delete:** Removes the selected result from the output mapping.

	**Create Package:** Packages the program for rating and testing. See Creating a Global Versioning Package for more information.
	**Program Outputs XML:** Opens a window that displays the Program Outputs XML.
	**Map Custom XML:** Opens a window that allows you to create a mapping that will transform the output XML. See Input and Output Mapping for more information.
	**Test Program:** Opens the Test Program, which allows the user to test and debug a program within RateManager. See Test Program for more information.

Search



Show results for: All

Show Results For: Allows you to filter the category shown. You also can browse for a category by clicking **BROWSE**. This will bring up the Browse Category popup. 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 show results for area.

All is the default category. This will display all selected variables and the corresponding Result ID.

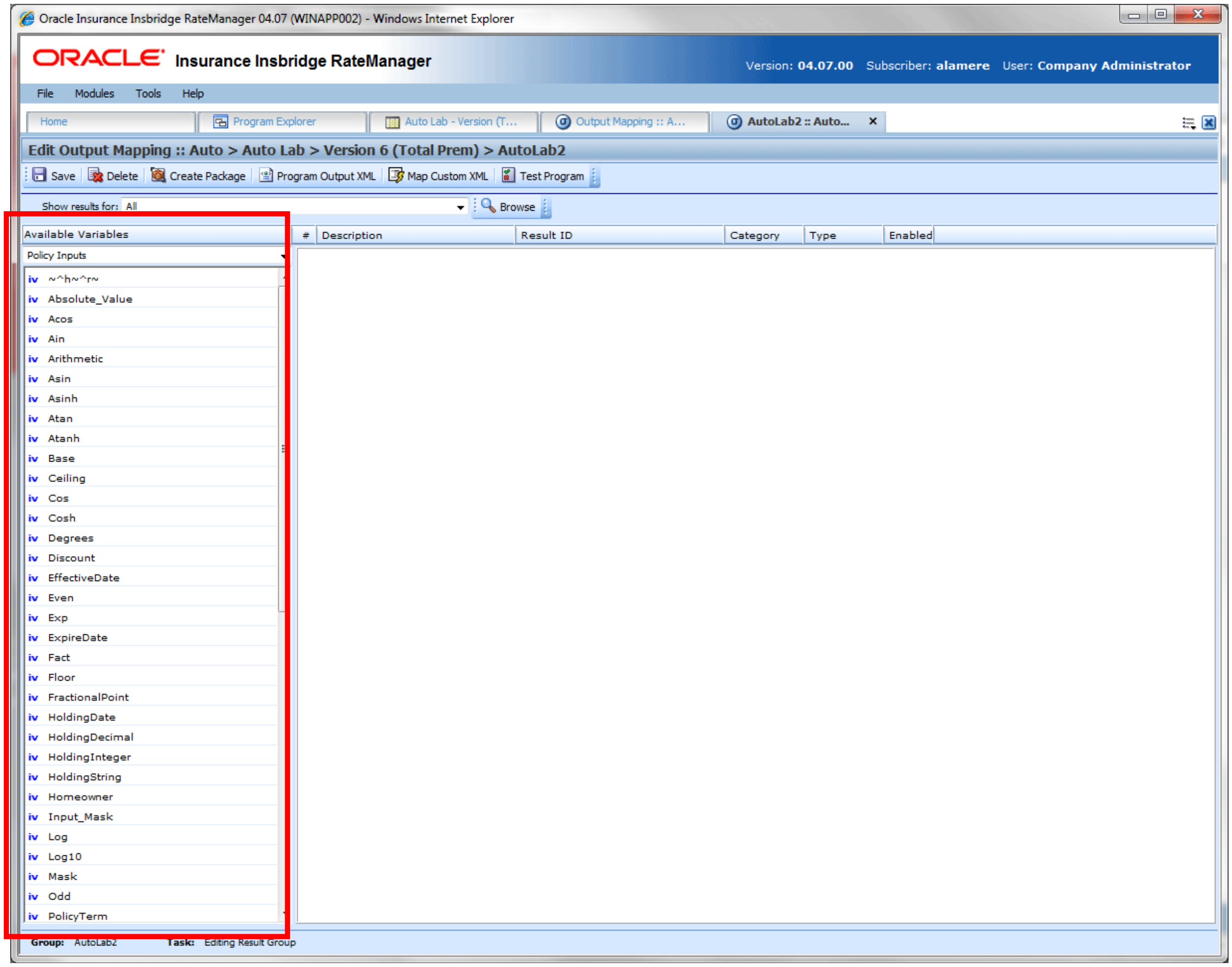
Edit Output Group Mapping Tab Fields

There are two areas on the Output Mapping Tab.

- **Available Variables:** The list of the available inputs, variables and results that can be used in the Selected Results area.
- **Selected Results:** A listing of inputs, 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

Drop down selection of available inputs, variables and results sorted by category. To add an item to the selected result mappings, find the element you want to add in the listing and double-click it. The element will be added to the selected results.


Oracle Insurance Insbridge RateManager 04.07 (WINAPP002) - Windows Internet Explorer
ORACLE Insurance Insbridge RateManager
Version: 04.07.00 Subscriber: alamere User: Company Administrator
File
Modules
Tools
Help
Home
Program Explorer
Auto Lab - Version (T…
Output Mapping :: A…
AutoLab2 :: Auto…
Edit Output Mapping :: Auto > Auto Lab > Version 6 (Total Prem) > AutoLab2
Save
Delete
Create Package
Program Output XML
Map Custom XML
Test Program
Browse
Available Variables
#
Description
Result ID
Category
Type
Enabled

Show results for: All

Policy Inputs

iv ~^h~^r~
iv Absolute_Value
iv Acos
iv Ain
iv Arithmetic
iv Asin
iv Asinh
iv Atan
iv Atanh
iv Base
iv Ceiling
iv Cos
iv Cosh
iv Degrees
iv Discount
iv EffectiveDate
iv Even
iv Exp
iv ExpireDate
iv Fact
iv Floor
iv FractionalPoint
iv HoldingDate
iv HoldingDecimal
iv HoldingInteger
iv HoldingString
iv Homeowner
iv Input_Mask
iv Log
iv Log10
iv Mask
iv Odd
iv PolicyTerm

Group: AutoLab2 Task: Editing Result Group

Figure 251 Available Variables for Output Mapping

Fields:

Menu: The menu is the top most field. Click the down arrow to select the category variables you want to view.

Selection: The available variables, for the selected category will be listed. Listing may also include other category inputs, variables and results.

Selection	Description
Policy Inputs	Inputs in the Policy category.
Policy Variables	Table and calculated variables in the Policy category.

Policy Results	Result variables in the Policy category.

Edit Output Group Mapping Tab Selected Results

A listing of inputs, variables and results that have been selected. The listing shows the results in the order they will be outputted. The results can be moved by selecting, dragging and dropping.

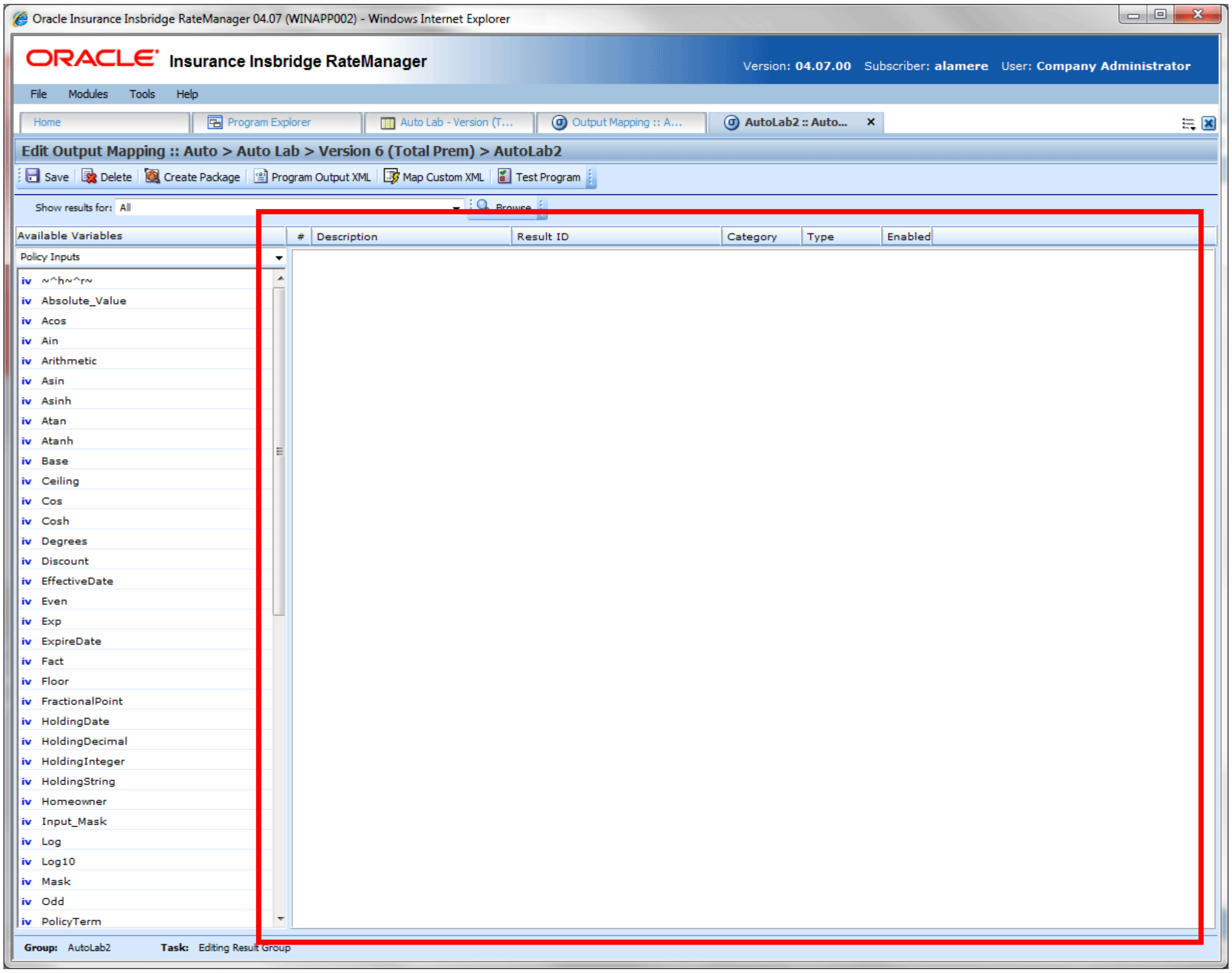


Figure 252 Available Variables for Output Mapping

Fields:

#: The order in which the result will be displayed.

Description: The name of the variable or input.

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 input.

Type: The type of variable or input. i.e. table, calculated, input or result.

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.

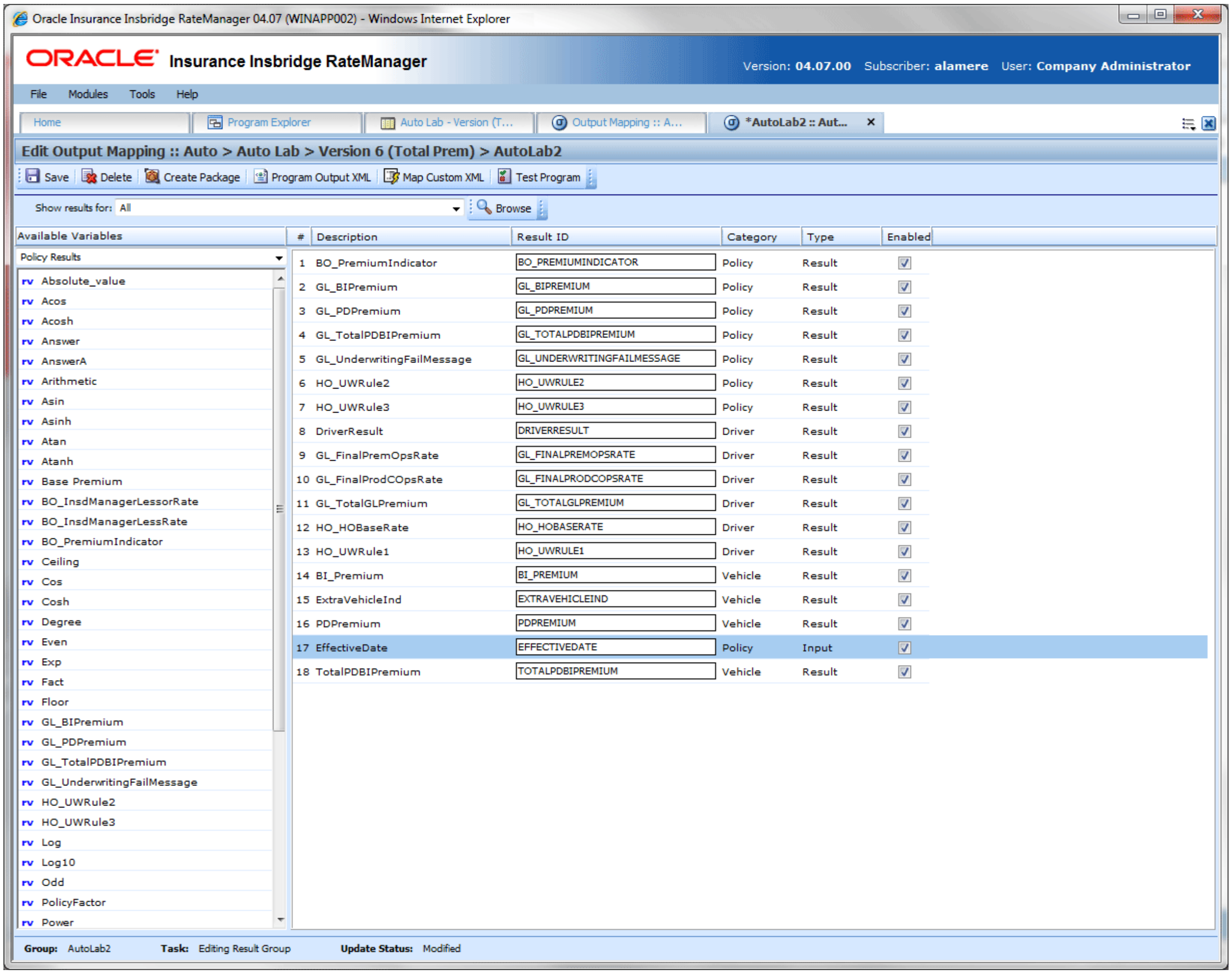
Editing an Output Mapping

On the **Edit Output Mapping** tab you can choose which inputs, 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 will open the **Edit Output Mapping** tab for the output group.

3. Select the category you want from the **Show result for** drop down menu.



Version: 04.07.00 Subscriber: alamere User: Company Administrator

Edit Output Mapping :: Auto > Auto Lab > Version 6 (Total Prem) > AutoLab2

Show results for: All

Available Variables

Policy Results

rv Absolute_value
rv Acos
rv Acosh
rv Answer
rv AnswerA
rv Arithmetic
rv Asin
rv Asinh
rv Atan
rv Atanh
rv Base Premium
rv BO_InsdManagerLessorRate
rv BO_InsdManagerLessRate
rv BO_PremiumIndicator
rv Ceiling
rv Cos
rv Cosh
rv Degree
rv Even
rv Exp
rv Fact
rv Floor
rv GL_BIPremium
rv GL_PDPremium
rv GL_TotalPDBIPremium
rv GL_UnderwritingFailMessage
rv HO_UWRule2
rv HO_UWRule3
rv Log
rv Log10
rv Odd
rv PolicyFactor
rv Power

#	Description	Result ID	Category	Type	Enabled
1	BO_PremiumIndicator	BO_PREMIUMINDICATOR	Policy	Result	☑
2	GL_BIPremium	GL_BIPREMIUM	Policy	Result	☑
3	GL_PDPremium	GL_PDPREMIUM	Policy	Result	☑
4	GL_TotalPDBIPremium	GL_TOTALPDBIPREMIUM	Policy	Result	☑
5	GL_UnderwritingFailMessage	GL_UNDERWRITINGFAILMESSAGE	Policy	Result	☑
6	HO_UWRule2	HO_UWRULE2	Policy	Result	☑
7	HO_UWRule3	HO_UWRULE3	Policy	Result	☑
8	DriverResult	DRIVERRESULT	Driver	Result	☑
9	GL_FinalPremOpsRate	GL_FINALPREMOPSRATE	Driver	Result	☑
10	GL_FinalProdCOpsRate	GL_FINALPRODCOPSRATE	Driver	Result	☑
11	GL_TotalGLPremium	GL_TOTALGLPREMIUM	Driver	Result	☑
12	HO_HOBaseRate	HO_HOBASERATE	Driver	Result	☑
13	HO_UWRule1	HO_UWRULE1	Driver	Result	☑
14	BI_Premium	BI_PREMIUM	Vehicle	Result	☑
15	ExtraVehicleInd	EXTRAVEHICLEIND	Vehicle	Result	☑
16	PDPremium	PDPREMIUM	Vehicle	Result	☑
17	EffectiveDate	EFFECTIVEDATE	Policy	Input	☑
18	TotalPDBIPremium	TOTALPDBIPREMIUM	Vehicle	Result	☑

Group: AutoLab2 Task: Editing Result Group Update Status: Modified

Figure 253 Editing an Output Group Mapping

4. Select the same category from the **Available Variables** drop down list on the left hand side of the screen. The categories must be the same. For example, if you select the Policy category up top, you will now be able to add the Policy inputs, variables and results.

5. Double-click the input, variable, or result you want to add. It will be added to the listing. You can add as many inputs, 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 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. **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.

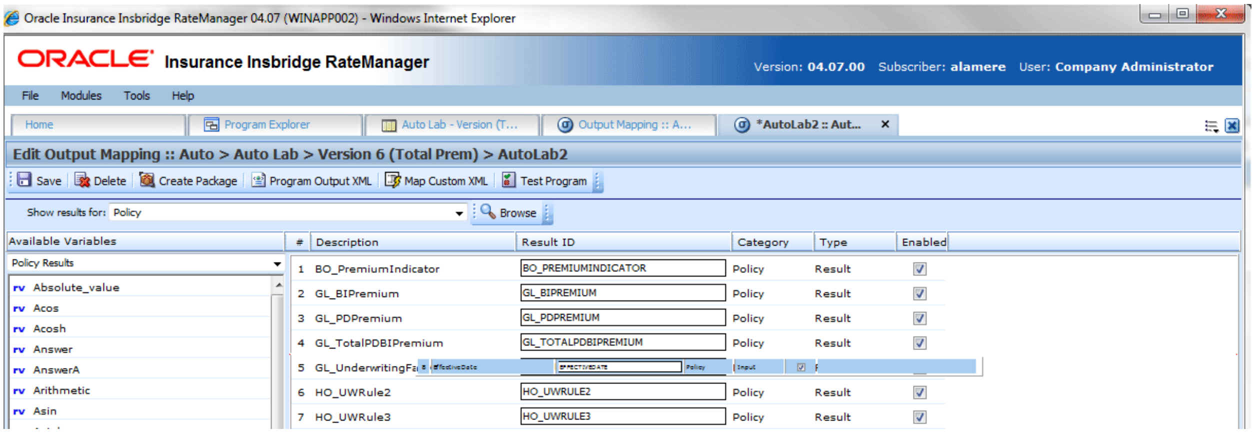
7. Make any edits that you may need. You can delete the elements you do not need and move the elements to the position you need as well. Make as many changes as you need

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. 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.*

Changing Output Order

To change the result order, use the **Show Results For** box to select a category. Highlight the item you want to move. Hold down the mouse button to drag and drop the results. You will know the result is ready to move when you see a blue shadow box containing the result information displayed.



Oracle Insurance Insbridge RateManager 04.07 (WINAPP002) - Windows Internet Explorer

ORACLE Insurance Insbridge RateManager

Version: **04.07.00** Subscriber: **alamere** User: **Company Administrator**

Edit Output Mapping :: Auto > Auto Lab > Version 6 (Total Prem) > AutoLab2

Show results for: Policy

Available Variables

Policy Results

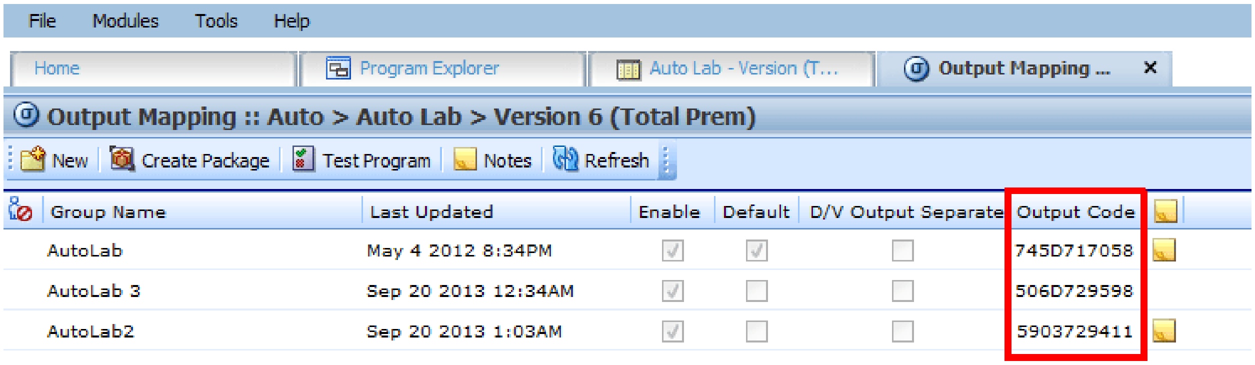
- rv Absolute_value
- rv Acos
- rv Acosh
- rv Answer
- rv AnswerA
- rv Arithmetic
- rv Asin

#	Description	Result ID	Category	Type	Enabled
1	BO_PremiumIndicator	BO_PREMIUMINDICATOR	Policy	Result	☑
2	GL_BIPremium	GL_BIPREMIUM	Policy	Result	☑
3	GL_PDPremium	GL_PDPREMIUM	Policy	Result	☑
4	GL_TotalPDBIPremium	GL_TOTALPDBIPREMIUM	Policy	Result	☑
5	GL_UnderwritingFa 8 EffectiveDate	EFFECTIVEDATE	Policy	Input	☑
6	HO_UWRule2	HO_UWRULE2	Policy	Result	☑
7	HO_UWRule3	HO_UWRULE3	Policy	Result	☑

Figure 254 Moving Output Mapping

Output Code Location

Output codes are required for custom XML mappings. This allows you to override 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.



Output Mapping :: Auto > Auto Lab > Version 6 (Total Prem)

Group Name	Last Updated	Enable	Default	D/V Output Separate	Output Code
AutoLab	May 4 2012 8:34PM	☑	☑	☐	745D717058
AutoLab 3	Sep 20 2013 12:34AM	☑	☐	☐	506D729598
AutoLab2	Sep 20 2013 1:03AM	☑	☐	☐	5903729411

Figure 255 Output Codes

If you are not using custom XML mappings, you will not need to enter the output code anywhere.

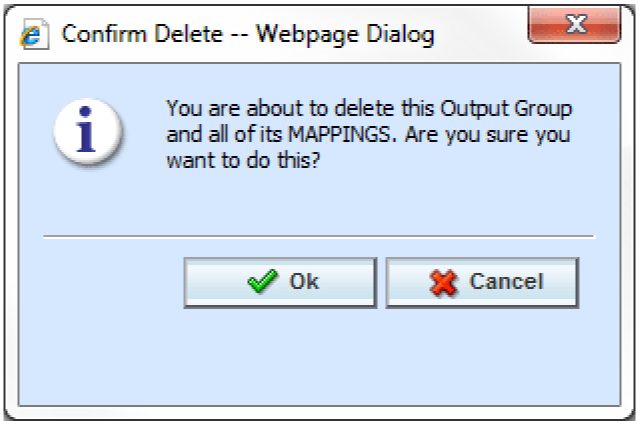
Deleting a Group

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 then right click and select **DELETE**. A warning message will be displayed.



You are about to delete this Output Group and all of its MAPPINGS. Are you sure you want to do this?

Figure 256 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.

Chapter 22

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 lob=“5”>
    <heading>
        <program parent_id=“123” program_id=“3” program_ver=“1”/>
    </heading>
    <c i=“0” desc=“Policy”>
        <m i=“2” n=“EffectiveDate” v=“04/11/2014”/>
        <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:

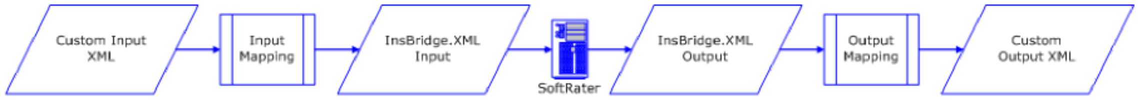


Figure 257 Using an existing XML Format

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 IBRU 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>03/05/2014</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 subline. This is useful if inputs are the same across all or most of the programs under that subline.
- If the inputs are mapped at a **program version level**, then the mapping is available for that program version only.

Custom inputs cannot be mapped at a **MultiLines level**.

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.

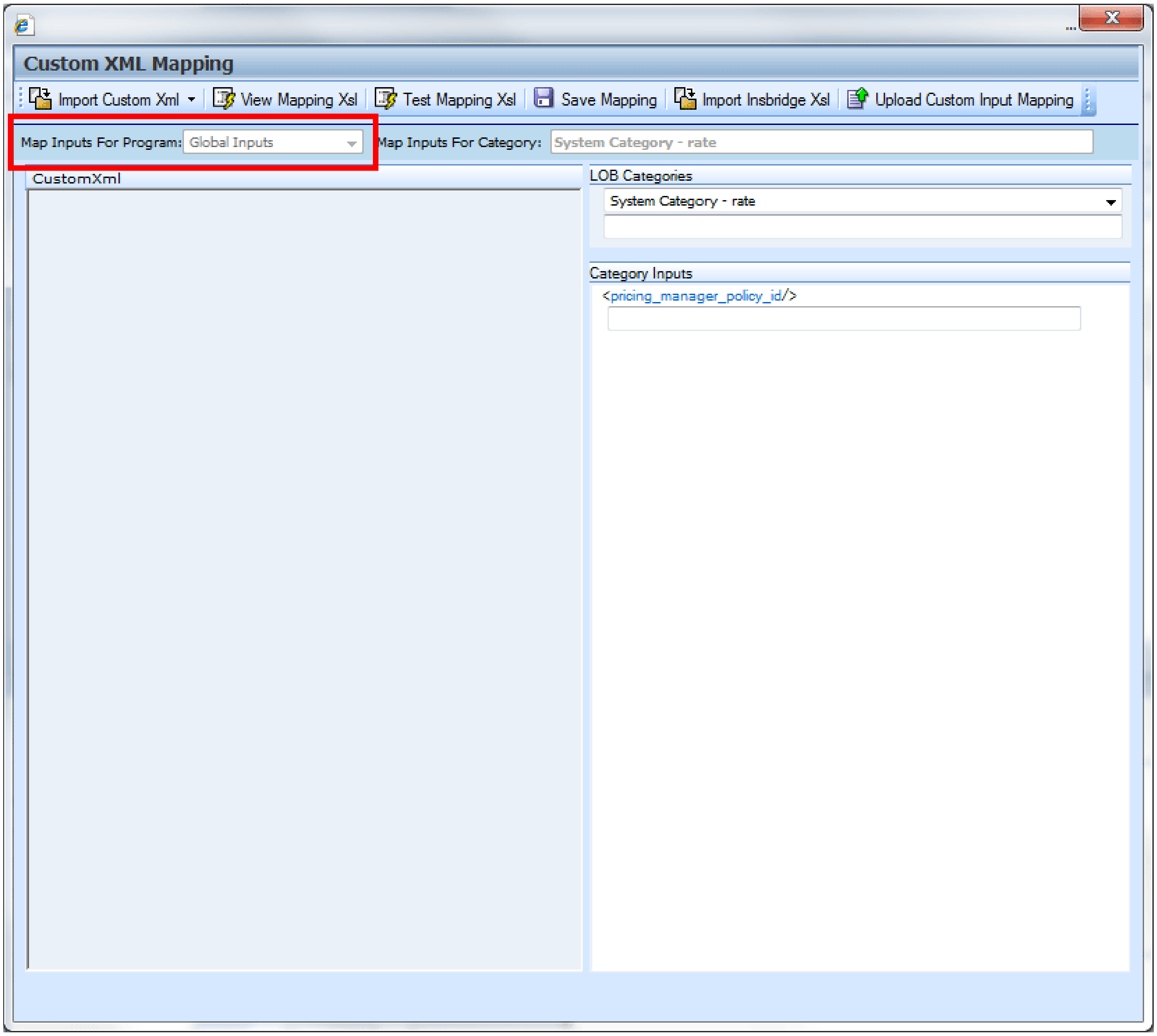
Step 3: Mapping Inputs

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→Sublines**, 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 subline.

1. From the Program Explorer tab, select the subline where you want to map inputs.
2. Right click and select **Map Custom XML**.
3. This opens the **Custom XML Mapping** window.



Map Inputs For Program: Global Inputs

Map Inputs For Category: System Category - rate

System Category - rate

<pricing_manager_policy_id/>

Figure 258 Custom XML Mapping at Global Level

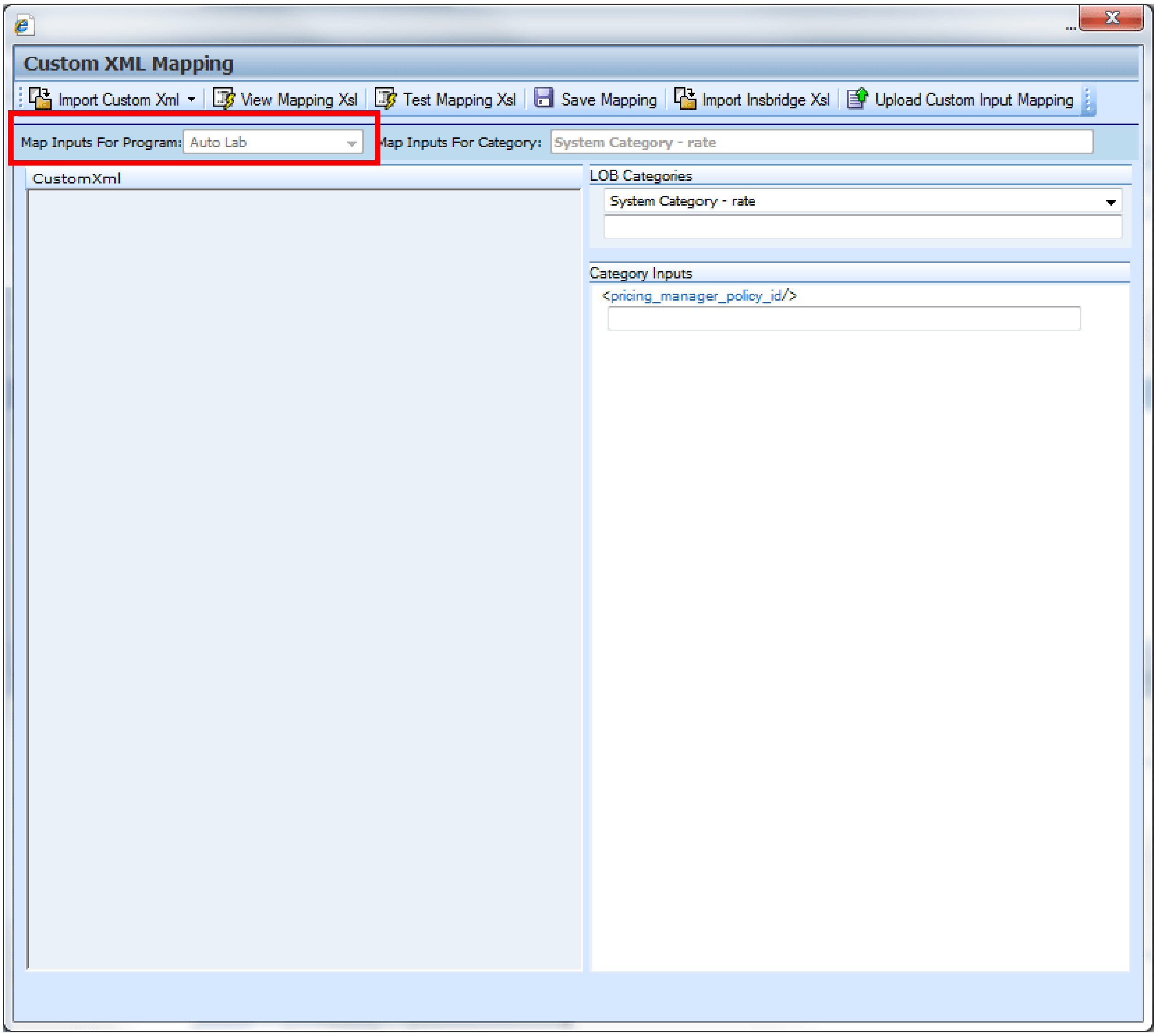
4. At this point, you can import your custom mappings. See **Importing Custom XML** for more information.

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.



Custom XML Mapping

Map Inputs For Program: Auto Lab

Map Inputs For Category: System Category - rate

System Category - rate

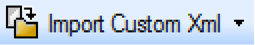
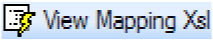
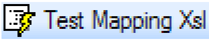
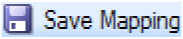
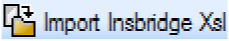
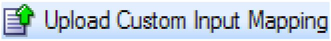
<pricing_manager_policy_id/>

Figure 259 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.

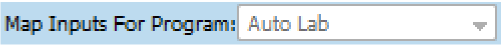
Custom XML Mapping Functionality

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:** Allows you to open a saved Insbridge Xsl file.
	**Upload Custom Input Mapping:** Allows you to upload your custom mapping. Uploaded mappings are available in the Testing module.

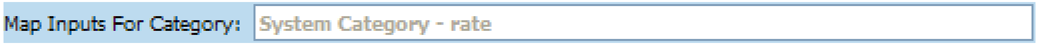
Custom XML Mapping Fields

An information bar is located near the top of the screen. There are two fields in the information bar:



Map Inputs For Program: Auto Lab

- The Map Inputs for Program displays the program or global information.



Map Inputs For Category: System Category - rate

- Map Inputs for Category displays the category that you have currently selected.

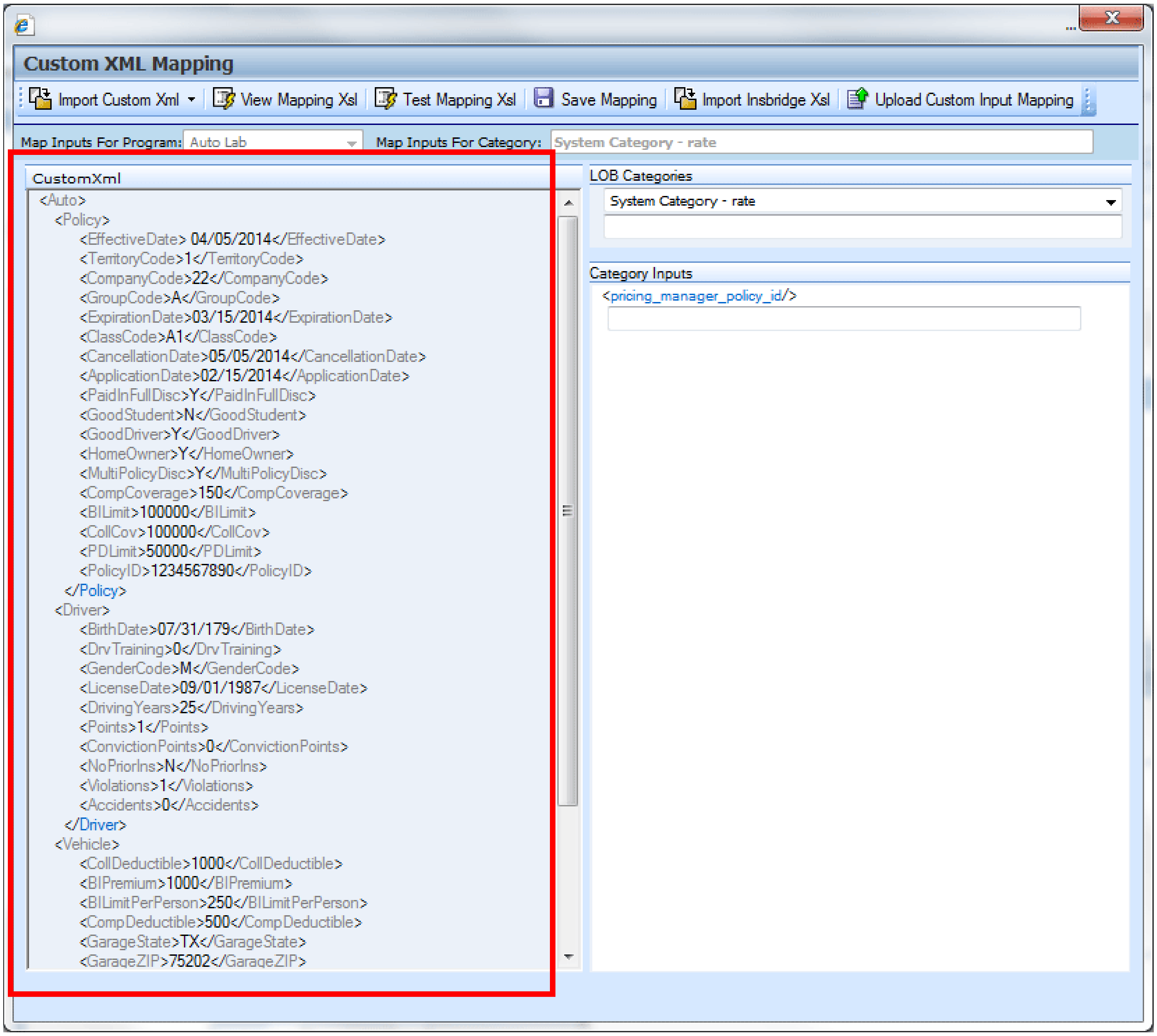
Custom XML Mapping Work Areas

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 need to make changes, please return to your XML editing program, make your changes and re-import.



Map Inputs For Program: Auto Lab

Map Inputs For Category: System Category - rate

System Category - rate

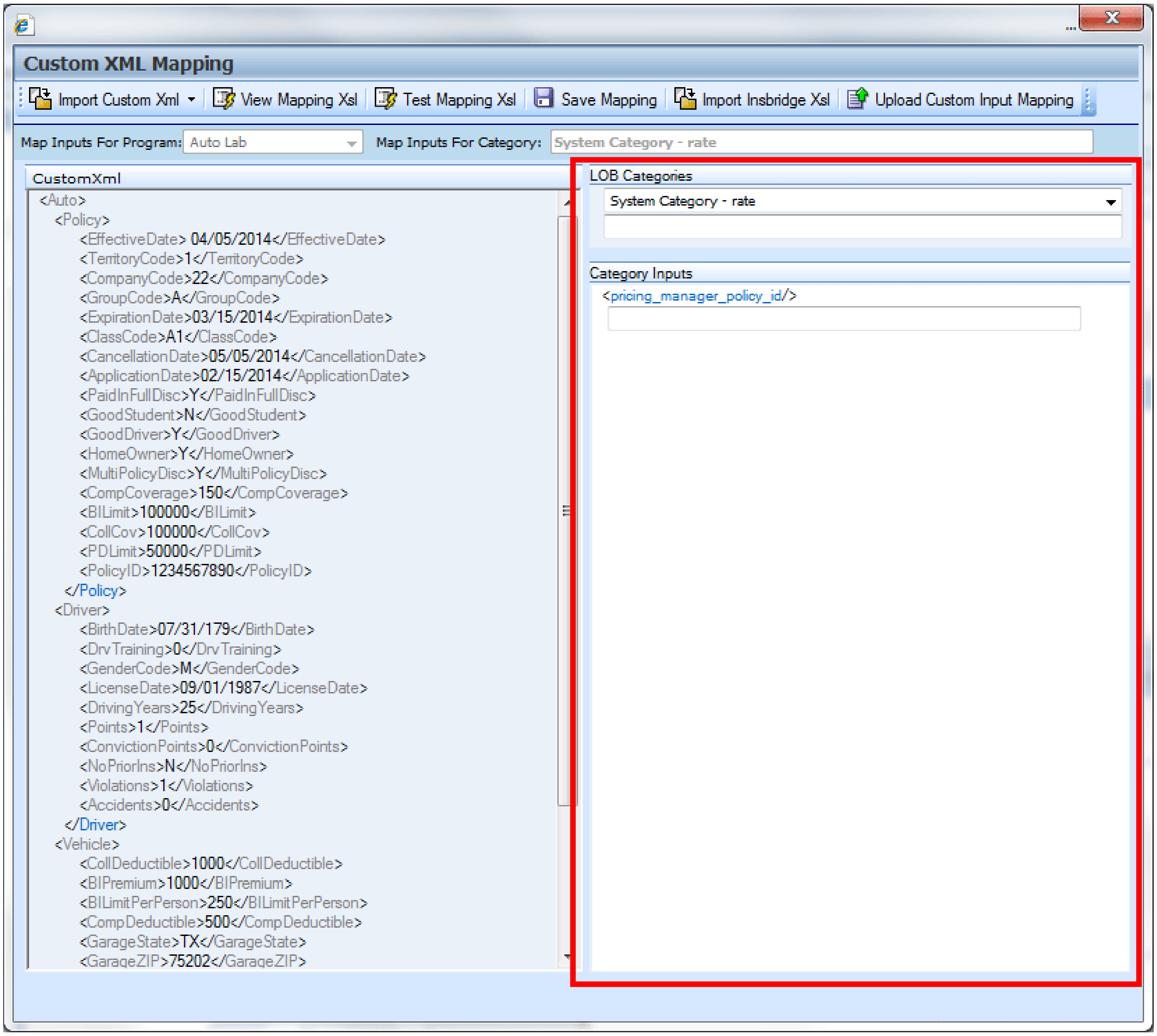
<pricing_manager_policy_id/>

```
<Auto>
  <Policy>
      <EffectiveDate> 04/05/2014</EffectiveDate>
      <TerritoryCode>1</TerritoryCode>
      <CompanyCode>22</CompanyCode>
      <GroupCode>A</GroupCode>
      <ExpirationDate>03/15/2014</ExpirationDate>
      <ClassCode>A1</ClassCode>
      <CancellationDate>05/05/2014</CancellationDate>
      <ApplicationDate>02/15/2014</ApplicationDate>
      <PaidInFullDisc>Y</PaidInFullDisc>
      <GoodStudent>N</GoodStudent>
      <GoodDriver>Y</GoodDriver>
      <HomeOwner>Y</HomeOwner>
      <MultiPolicyDisc>Y</MultiPolicyDisc>
      <CompCoverage>150</CompCoverage>
      <BILimit>100000</BILimit>
      <CollCov>100000</CollCov>
      <PDLimit>50000</PDLimit>
      <PolicyID>1234567890</PolicyID>
   </Policy>
  <Driver>
      <BirthDate>07/31/179</BirthDate>
      <DrvTraining>0</DrvTraining>
      <GenderCode>M</GenderCode>
      <LicenseDate>09/01/1987</LicenseDate>
      <DrivingYears>25</DrivingYears>
      <Points>1</Points>
      <ConvictionPoints>0</ConvictionPoints>
      <NoPriorIns>N</NoPriorIns>
      <Violations>1</Violations>
      <Accidents>0</Accidents>
   </Driver>
  <Vehicle>
      <CollDeductible>1000</CollDeductible>
      <BIPremium>1000</BIPremium>
      <BILimitPerPerson>250</BILimitPerPerson>
      <CompDeductible>500</CompDeductible>
      <GarageState>TX</GarageState>
      <GarageZIP>75202</GarageZIP>
```

Figure 260 Custom XML Work Area

Category Inputs

The Category Inputs area contains the categories and inputs used in your program or LOB globals. You will not be able to edit any of these fields. If you need to make changes, please return to your program or the globals 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 inputs.



Map Inputs For Program: Auto Lab

Map Inputs For Category: System Category - rate

CustomXml

```
<Auto>
  <Policy>
      <EffectiveDate> 04/05/2014</EffectiveDate>
      <TerritoryCode>1</TerritoryCode>
      <CompanyCode>22</CompanyCode>
      <GroupCode>A</GroupCode>
      <ExpirationDate>03/15/2014</ExpirationDate>
      <ClassCode>A1</ClassCode>
      <CancellationDate>05/05/2014</CancellationDate>
      <ApplicationDate>02/15/2014</ApplicationDate>
      <PaidInFullDisc>Y</PaidInFullDisc>
      <GoodStudent>N</GoodStudent>
      <GoodDriver>Y</GoodDriver>
      <HomeOwner>Y</HomeOwner>
      <MultiPolicyDisc>Y</MultiPolicyDisc>
      <CompCoverage>150</CompCoverage>
      <BILimit>100000</BILimit>
      <CollCov>100000</CollCov>
      <PDLimit>50000</PDLimit>
      <PolicyID>1234567890</PolicyID>
   </Policy>
  <Driver>
      <BirthDate>07/31/179</BirthDate>
      <DrvTraining>0</DrvTraining>
      <GenderCode>M</GenderCode>
      <LicenseDate>09/01/1987</LicenseDate>
      <DrivingYears>25</DrivingYears>
      <Points>1</Points>
      <ConvictionPoints>0</ConvictionPoints>
      <NoPriorIns>N</NoPriorIns>
      <Violations>1</Violations>
      <Accidents>0</Accidents>
   </Driver>
  <Vehicle>
      <CollDeductible>1000</CollDeductible>
      <BIPremium>1000</BIPremium>
      <BILimitPerPerson>250</BILimitPerPerson>
      <CompDeductible>500</CompDeductible>
      <GarageState>TX</GarageState>
      <GarageZIP>75202</GarageZIP>
```

LOB Categories

System Category - rate

Category Inputs

<pricing_manager_policy_id/>

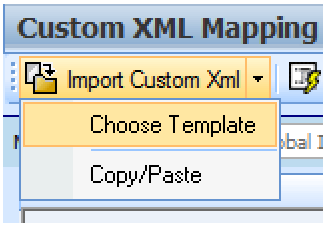
Figure 261 Category Inputs Work Area

The LOB Categories area has a category selection field at the top. Select your category and the inputs will be populated underneath. At the top level is a system category with one input. All other categories will contain the inputs used in the program or LOB globals.

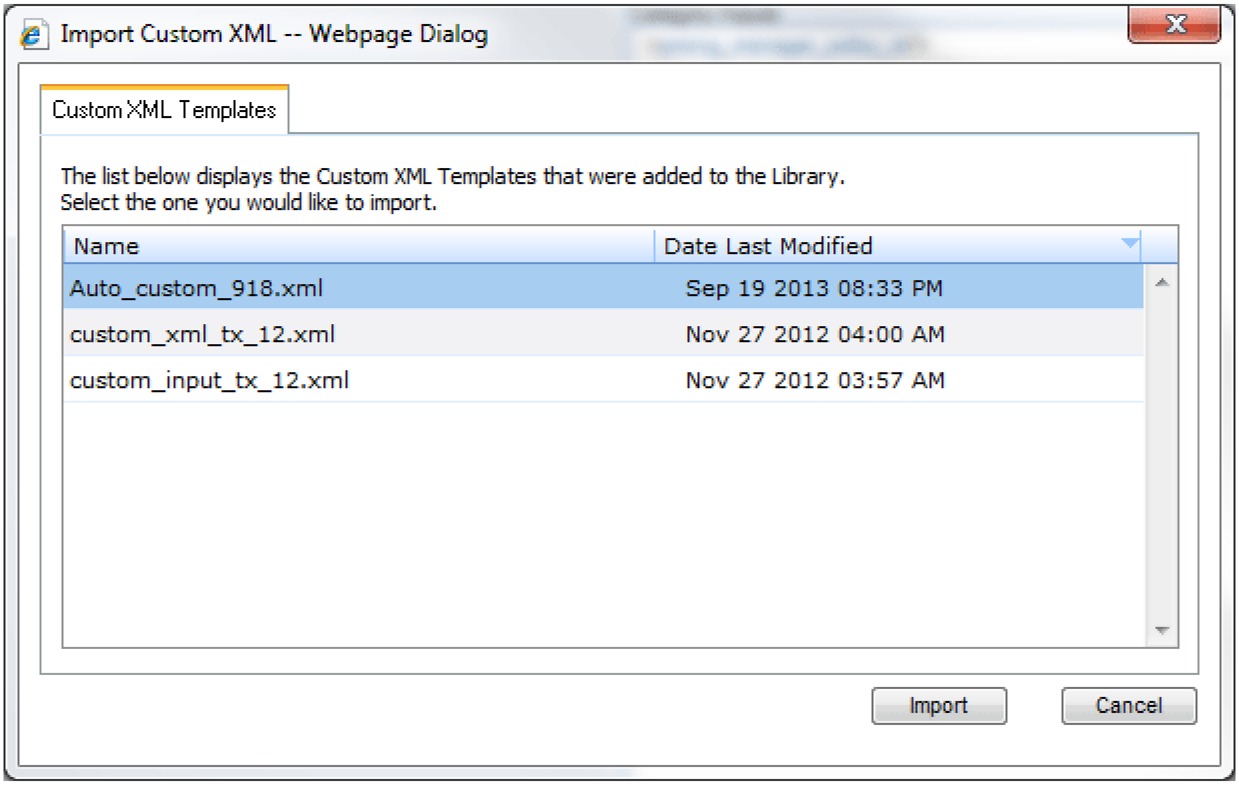
NOTE: *When using custom XML in Impact Analysis, you must map the System Category: rate and category input pricing_manager_policy_id. This input requires a unique policy identifier and is used in the batch and pricing manager process.*

Importing Custom XML Using Choose Template

1. From the Input Mapping screen, click **Import Custom XML** and select **Choose Template**.



2. A dialogue box allows you to 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.

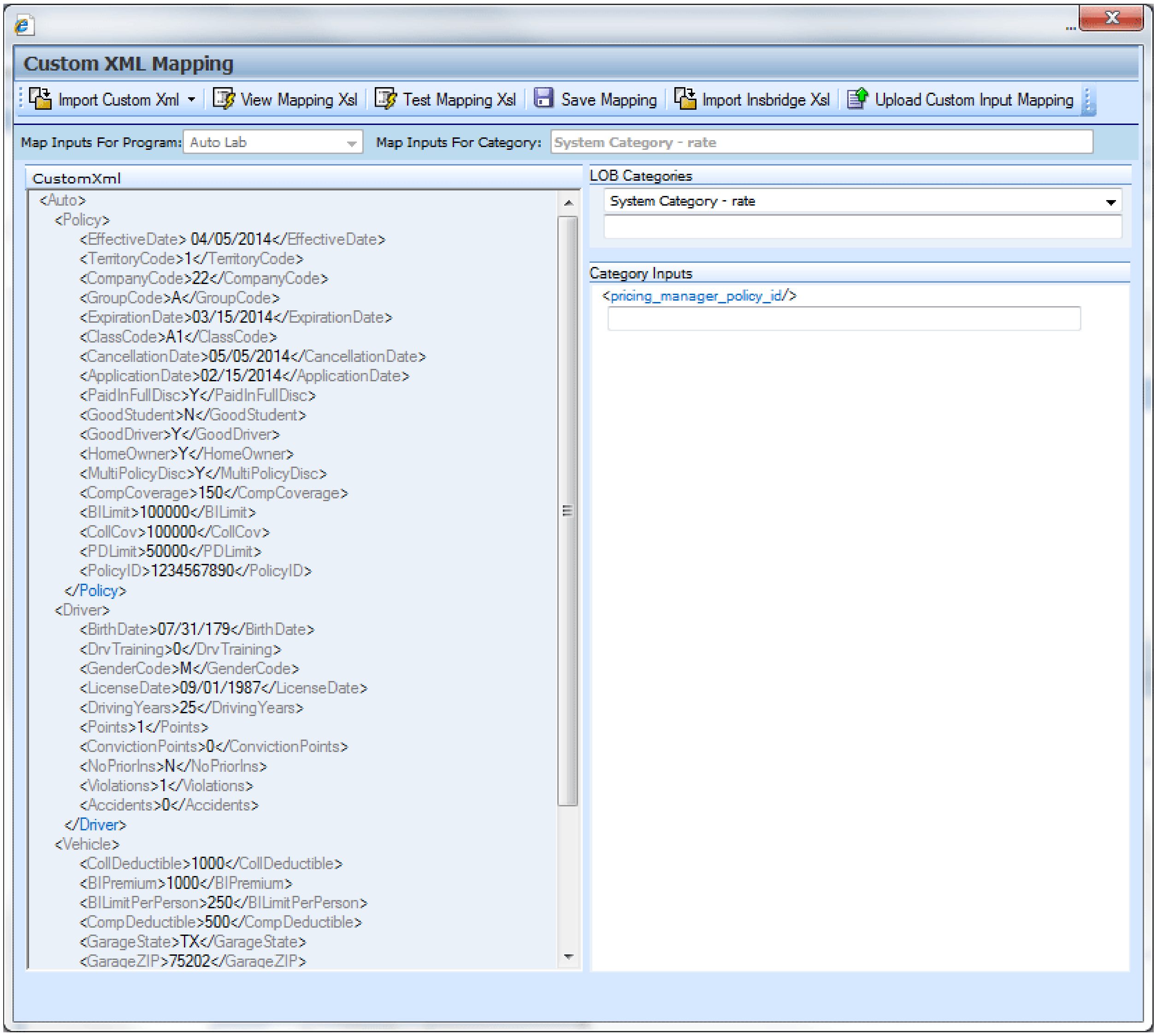


The list below displays the Custom XML Templates that were added to the Library.
Select the one you would like to import.

Name	Date Last Modified
Auto_custom_918.xml	Sep 19 2013 08:33 PM
custom_xml_tx_12.xml	Nov 27 2012 04:00 AM
custom_input_tx_12.xml	Nov 27 2012 03:57 AM

Figure 262 Uploading a Custom XML Template

3. Click Import.



Map Inputs For Program: Auto Lab Map Inputs For Category: System Category - rate

System Category - rate

<pricing_manager_policy_id/>

```
<Auto>
  <Policy>
      <EffectiveDate> 04/05/2014</EffectiveDate>
      <TerritoryCode>1</TerritoryCode>
      <CompanyCode>22</CompanyCode>
      <GroupCode>A</GroupCode>
      <ExpirationDate>03/15/2014</ExpirationDate>
      <ClassCode>A1</ClassCode>
      <CancellationDate>05/05/2014</CancellationDate>
      <ApplicationDate>02/15/2014</ApplicationDate>
      <PaidInFullDisc>Y</PaidInFullDisc>
      <GoodStudent>N</GoodStudent>
      <GoodDriver>Y</GoodDriver>
      <HomeOwner>Y</HomeOwner>
      <MultiPolicyDisc>Y</MultiPolicyDisc>
      <CompCoverage>150</CompCoverage>
      <BILimit>100000</BILimit>
      <CollCov>100000</CollCov>
      <PDLimit>50000</PDLimit>
      <PolicyID>1234567890</PolicyID>
   </Policy>
  <Driver>
      <BirthDate>07/31/179</BirthDate>
      <DrvTraining>0</DrvTraining>
      <GenderCode>M</GenderCode>
      <LicenseDate>09/01/1987</LicenseDate>
      <DrivingYears>25</DrivingYears>
      <Points>1</Points>
      <ConvictionPoints>0</ConvictionPoints>
      <NoPriorIns>N</NoPriorIns>
      <Violations>1</Violations>
      <Accidents>0</Accidents>
   </Driver>
  <Vehicle>
      <CollDeductible>1000</CollDeductible>
      <BIPremium>1000</BIPremium>
      <BILimitPerPerson>250</BILimitPerPerson>
      <CompDeductible>500</CompDeductible>
      <GarageState>TX</GarageState>
      <GarageZIP>75202</GarageZIP>
```

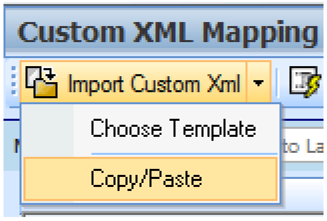
Figure 263 Imported Custom XML Template

4. The custom XML will be imported into RateManager and the **Custom XML Mapping** screen will refresh 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**.



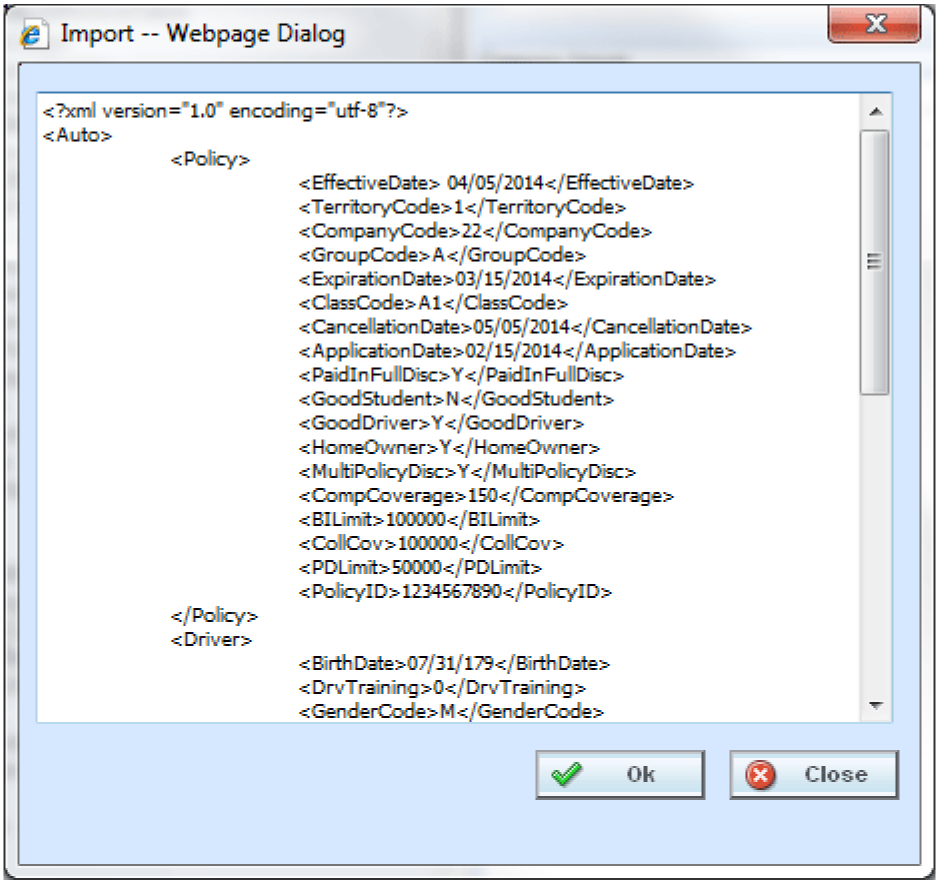
2. A text box allows you to paste your custom XML file.

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 popup menu or by pressing **Ctrl+V**.

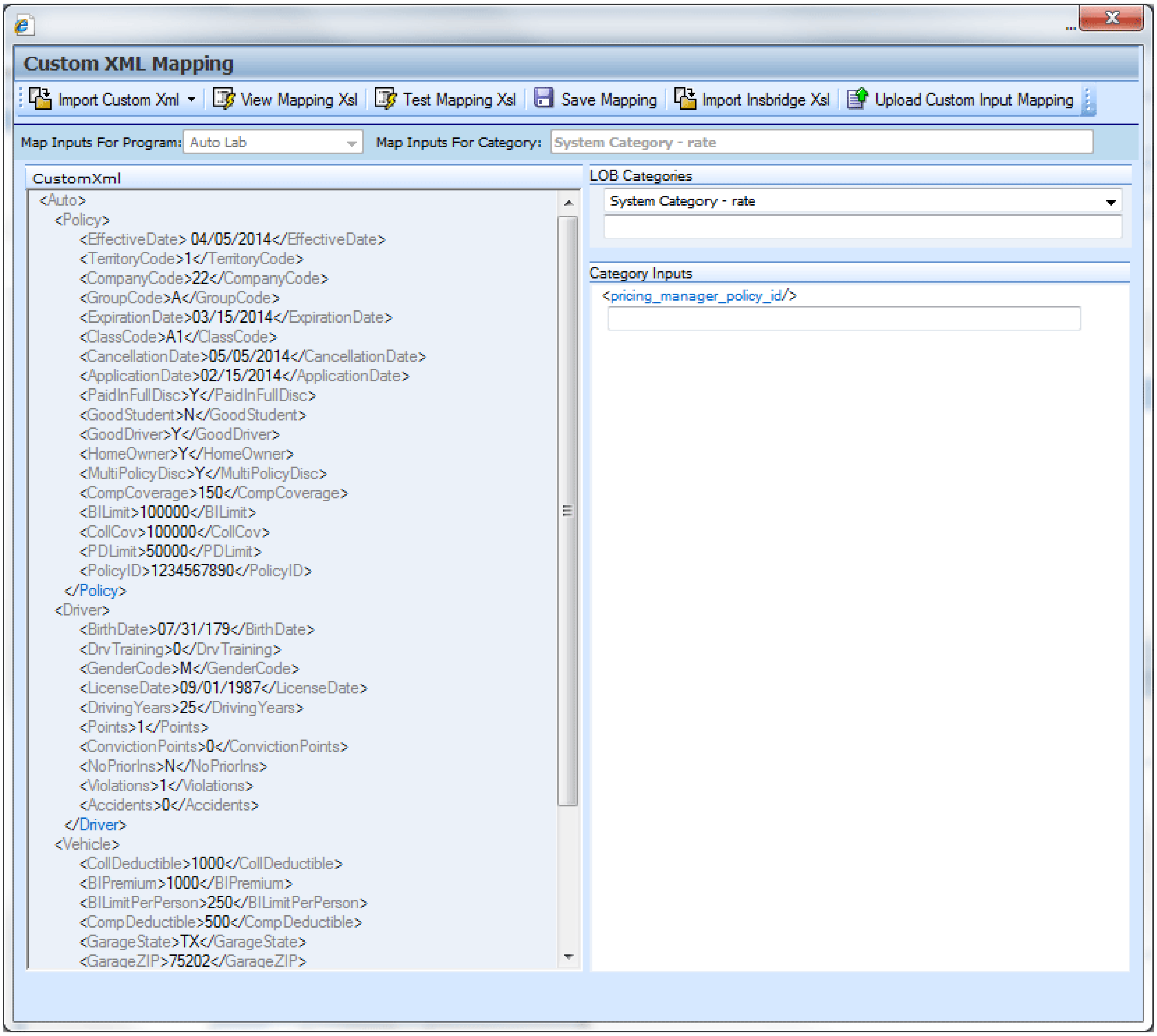


```
<?xml version="1.0" encoding="utf-8"?>
<Auto>
        <Policy>
                <EffectiveDate> 04/05/2014</EffectiveDate>
                <TerritoryCode>1</TerritoryCode>
                <CompanyCode>22</CompanyCode>
                <GroupCode>A</GroupCode>
                <ExpirationDate>03/15/2014</ExpirationDate>
                <ClassCode>A1</ClassCode>
                <CancellationDate>05/05/2014</CancellationDate>
                <ApplicationDate>02/15/2014</ApplicationDate>
                <PaidInFullDisc>Y</PaidInFullDisc>
                <GoodStudent>N</GoodStudent>
                <GoodDriver>Y</GoodDriver>
                <HomeOwner>Y</HomeOwner>
                <MultiPolicyDisc>Y</MultiPolicyDisc>
                <CompCoverage>150</CompCoverage>
                <BILimit>100000</BILimit>
                <CollCov>100000</CollCov>
                <PDLimit>50000</PDLimit>
                <PolicyID>1234567890</PolicyID>
        </Policy>
        <Driver>
                <BirthDate>07/31/179</BirthDate>
                <DrvTraining>0</DrvTraining>
                <GenderCode>M</GenderCode>
```

Figure 264 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.



Map Inputs For Program: Auto Lab

Map Inputs For Category: System Category - rate

CustomXml

```
<Auto>
  <Policy>
      <EffectiveDate> 04/05/2014</EffectiveDate>
      <TerritoryCode>1</TerritoryCode>
      <CompanyCode>22</CompanyCode>
      <GroupCode>A</GroupCode>
      <ExpirationDate>03/15/2014</ExpirationDate>
      <ClassCode>A1</ClassCode>
      <CancellationDate>05/05/2014</CancellationDate>
      <ApplicationDate>02/15/2014</ApplicationDate>
      <PaidInFullDisc>Y</PaidInFullDisc>
      <GoodStudent>N</GoodStudent>
      <GoodDriver>Y</GoodDriver>
      <HomeOwner>Y</HomeOwner>
      <MultiPolicyDisc>Y</MultiPolicyDisc>
      <CompCoverage>150</CompCoverage>
      <BILimit>100000</BILimit>
      <CollCov>100000</CollCov>
      <PDLimit>50000</PDLimit>
      <PolicyID>1234567890</PolicyID>
   </Policy>
  <Driver>
      <BirthDate>07/31/179</BirthDate>
      <DrvTraining>0</DrvTraining>
      <GenderCode>M</GenderCode>
      <LicenseDate>09/01/1987</LicenseDate>
      <DrivingYears>25</DrivingYears>
      <Points>1</Points>
      <ConvictionPoints>0</ConvictionPoints>
      <NoPriorIns>N</NoPriorIns>
      <Violations>1</Violations>
      <Accidents>0</Accidents>
   </Driver>
  <Vehicle>
      <CollDeductible>1000</CollDeductible>
      <BIPremium>1000</BIPremium>
      <BILimitPerPerson>250</BILimitPerPerson>
      <CompDeductible>500</CompDeductible>
      <GarageState>TX</GarageState>
      <GarageZIP>75202</GarageZIP>
```

LOB Categories

System Category - rate

Category Inputs

```
<pricing_manager_policy_id/>
```

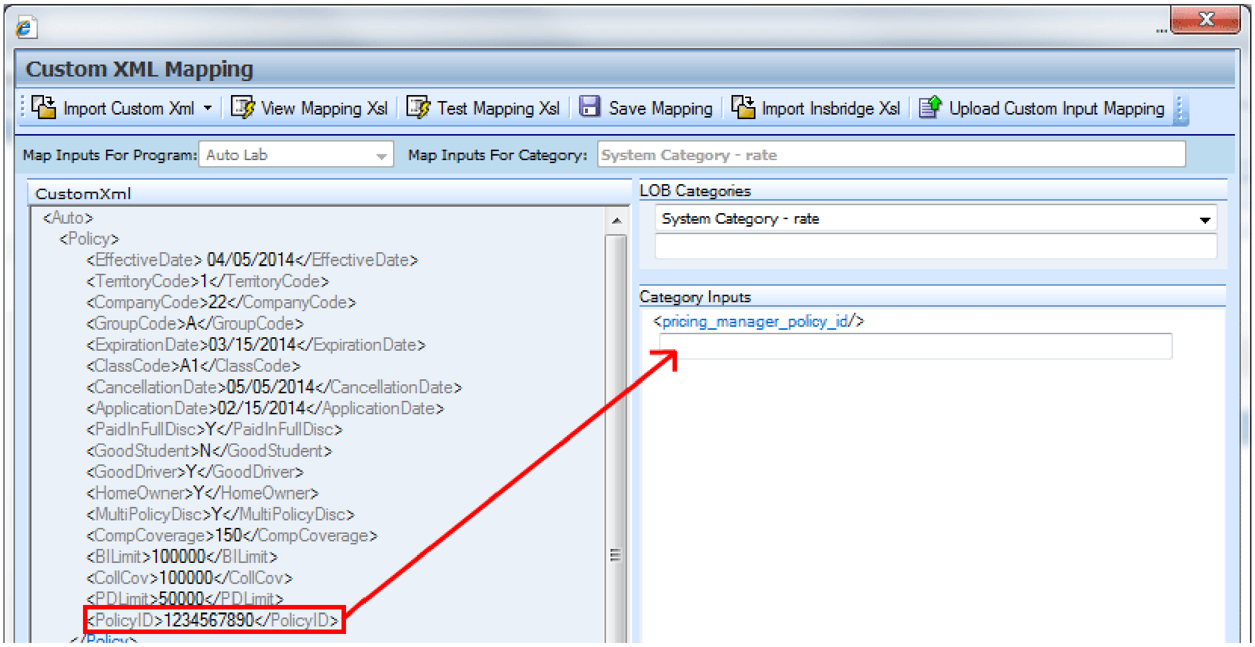
Figure 265 Imported Custom XML Using Copy and Paste

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 need 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 LOB Categories area on the right.



Map Inputs For Program: Auto Lab

Map Inputs For Category: System Category - rate

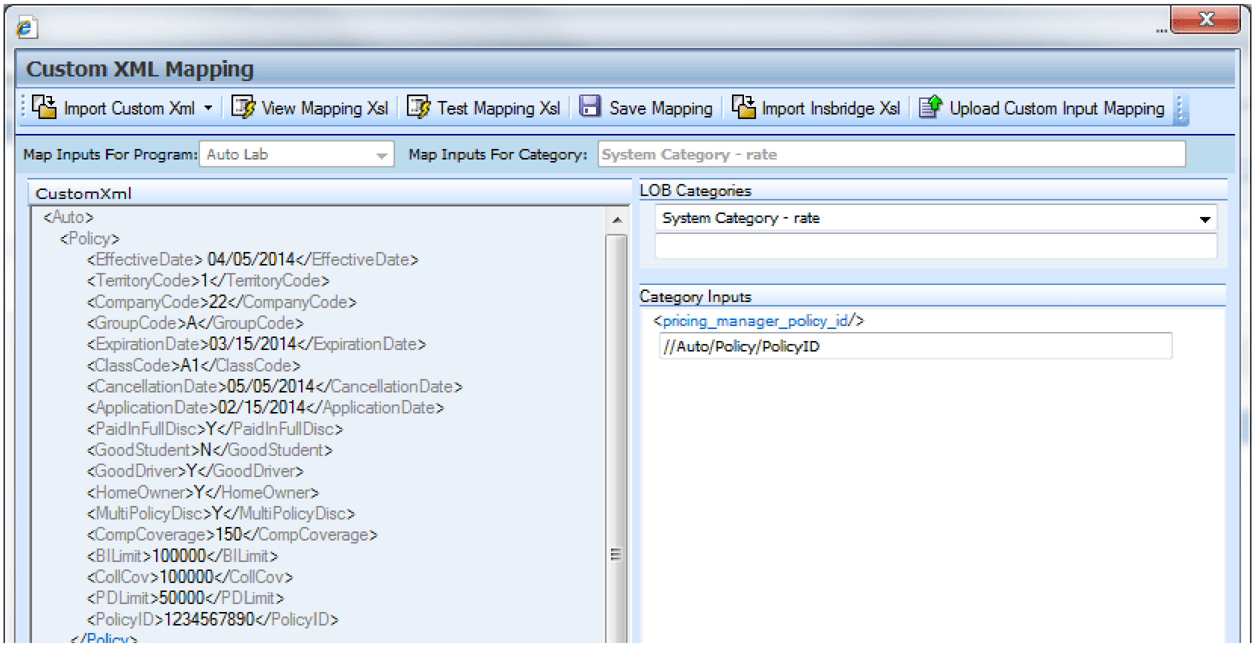
```
<Auto>
  <Policy>
      <EffectiveDate> 04/05/2014</EffectiveDate>
      <TerritoryCode>1</TerritoryCode>
      <CompanyCode>22</CompanyCode>
      <GroupCode>A</GroupCode>
      <ExpirationDate>03/15/2014</ExpirationDate>
      <ClassCode>A1</ClassCode>
      <CancellationDate>05/05/2014</CancellationDate>
      <ApplicationDate>02/15/2014</ApplicationDate>
      <PaidInFullDisc>Y</PaidInFullDisc>
      <GoodStudent>N</GoodStudent>
      <GoodDriver>Y</GoodDriver>
      <HomeOwner>Y</HomeOwner>
      <MultiPolicyDisc>Y</MultiPolicyDisc>
      <CompCoverage>150</CompCoverage>
      <BILimit>100000</BILimit>
      <CollCov>100000</CollCov>
      <PDLimit>50000</PDLimit>
      <PolicyID>1234567890</PolicyID>
    </Policy>
```

LOB Categories: System Category - rate

<pricing_manager_policy_id/>

Figure 266 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.



Map Inputs For Program: Auto Lab

Map Inputs For Category: System Category - rate

```
<Auto>
  <Policy>
      <EffectiveDate> 04/05/2014</EffectiveDate>
      <TerritoryCode>1</TerritoryCode>
      <CompanyCode>22</CompanyCode>
      <GroupCode>A</GroupCode>
      <ExpirationDate>03/15/2014</ExpirationDate>
      <ClassCode>A1</ClassCode>
      <CancellationDate>05/05/2014</CancellationDate>
      <ApplicationDate>02/15/2014</ApplicationDate>
      <PaidInFullDisc>Y</PaidInFullDisc>
      <GoodStudent>N</GoodStudent>
      <GoodDriver>Y</GoodDriver>
      <HomeOwner>Y</HomeOwner>
      <MultiPolicyDisc>Y</MultiPolicyDisc>
      <CompCoverage>150</CompCoverage>
      <BILimit>100000</BILimit>
      <CollCov>100000</CollCov>
      <PDLimit>50000</PDLimit>
      <PolicyID>1234567890</PolicyID>
    </Policy>
```

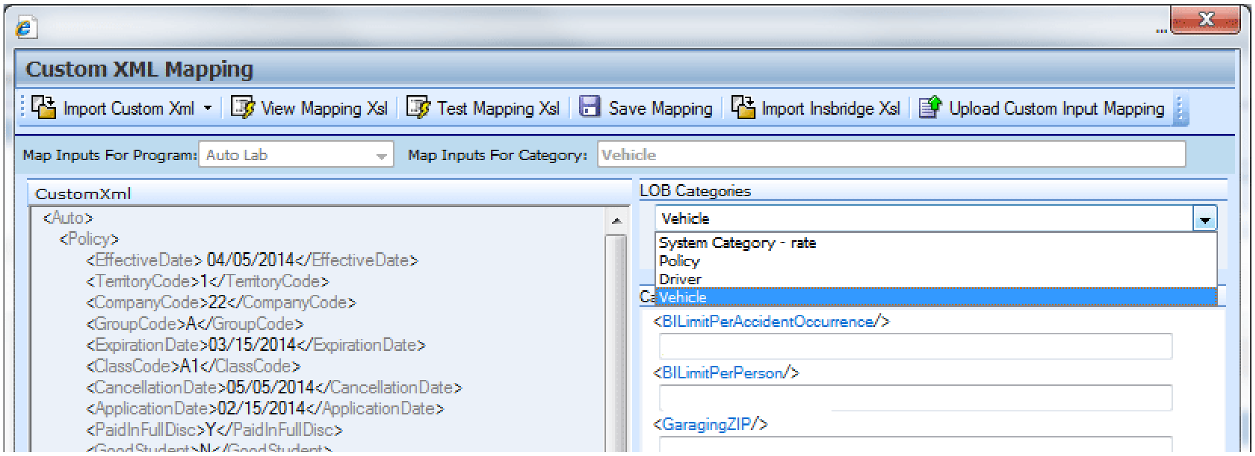
LOB Categories: System Category - rate

<pricing_manager_policy_id/>

//Auto/Policy/PolicyID

Figure 267 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 LOB 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 LOB 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.



Map Inputs For Program: Auto Lab
Map Inputs For Category: Vehicle

```
<Auto>
   <Policy>
      <EffectiveDate> 04/05/2014</EffectiveDate>
      <TerritoryCode>1</TerritoryCode>
      <CompanyCode>22</CompanyCode>
      <GroupCode>A</GroupCode>
      <ExpirationDate>03/15/2014</ExpirationDate>
      <ClassCode>A1</ClassCode>
      <CancellationDate>05/05/2014</CancellationDate>
      <ApplicationDate>02/15/2014</ApplicationDate>
      <PaidInFullDisc>Y</PaidInFullDisc>
```

LOB Categories: Vehicle

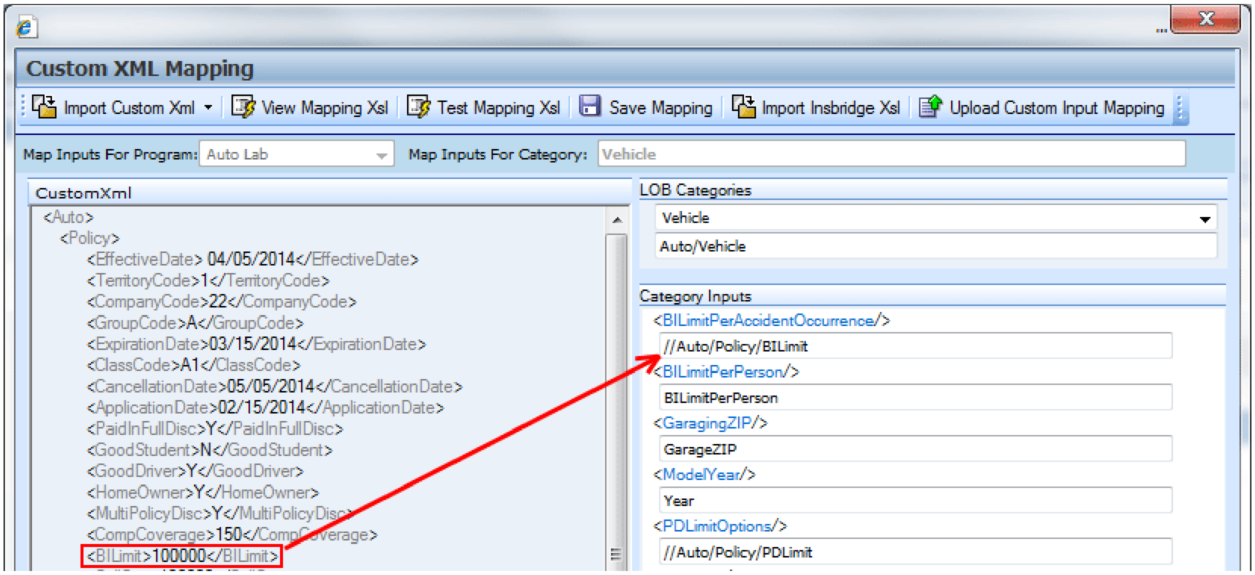
<BILimitPerAccidentOccurrence/>

<BILimitPerPerson/>

<GaragingZIP/>

Figure 268 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 subline. You will only need to map the inputs that are used in the programs that will use the mapping.



Map Inputs For Program: Auto Lab
Map Inputs For Category: Vehicle

```
<Auto>
   <Policy>
      <EffectiveDate> 04/05/2014</EffectiveDate>
      <TerritoryCode>1</TerritoryCode>
      <CompanyCode>22</CompanyCode>
      <GroupCode>A</GroupCode>
      <ExpirationDate>03/15/2014</ExpirationDate>
      <ClassCode>A1</ClassCode>
      <CancellationDate>05/05/2014</CancellationDate>
      <ApplicationDate>02/15/2014</ApplicationDate>
      <PaidInFullDisc>Y</PaidInFullDisc>
      <GoodStudent>N</GoodStudent>
      <GoodDriver>Y</GoodDriver>
      <HomeOwner>Y</HomeOwner>
      <MultiPolicyDisc>Y</MultiPolicyDisc>
      <CompCoverage>150</CompCoverage>
      <BILimit>100000</BILimit>
```

LOB Categories: Vehicle

Auto/Vehicle

<BILimitPerAccidentOccurrence/>
//Auto/Policy/BILimit

<BILimitPerPerson/>
BILimitPerPerson

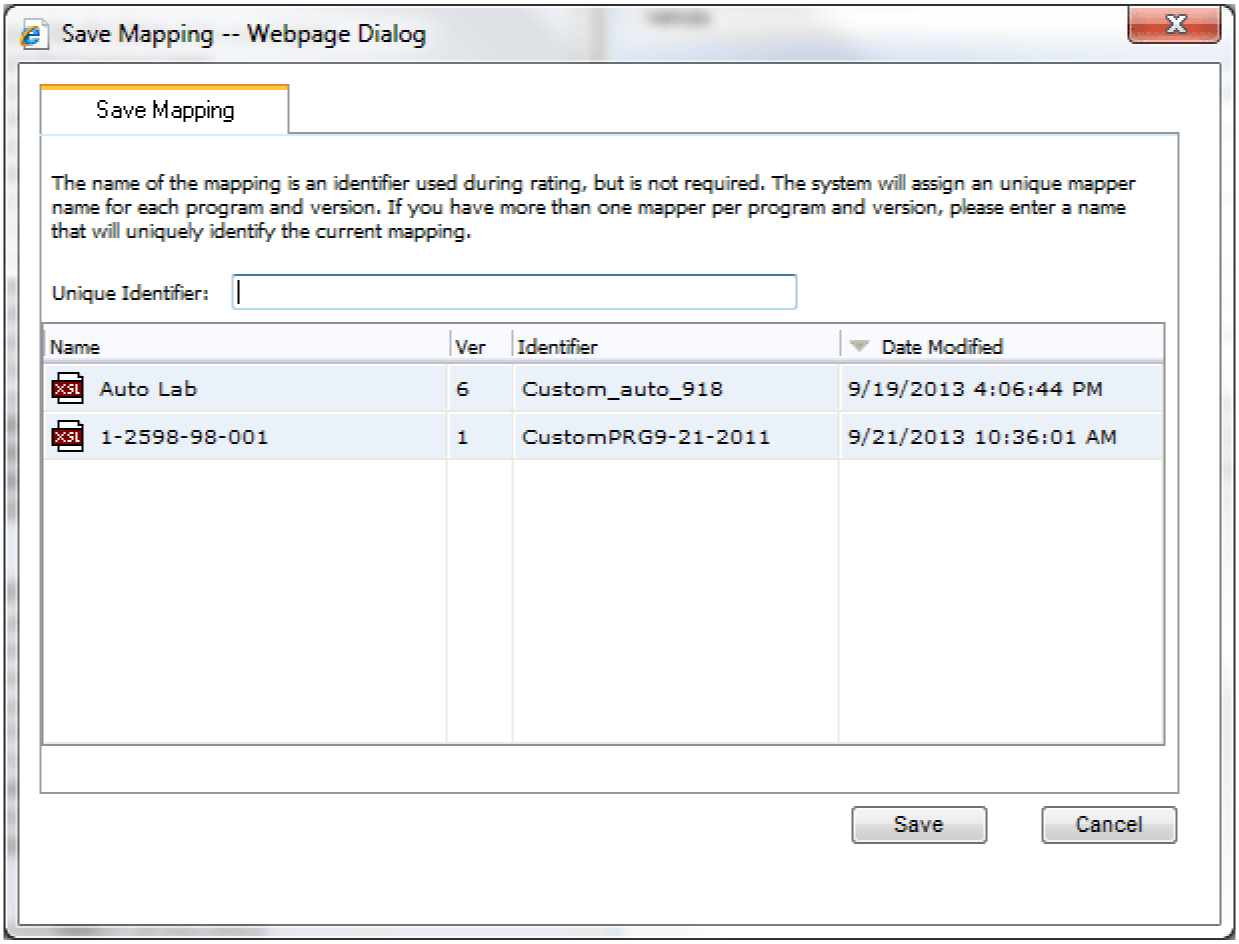
<GaragingZIP/>
GarageZIP

<ModelYear/>
Year

<PDLimitOptions/>
//Auto/Policy/PDLimit

Figure 269 Selecting Inputs

13. After you have finished mapping the categories and inputs, you need to save the mapping. To do this, click **Save Mapping**.



The name of the mapping is an identifier used during rating, but is not required. The system will assign an unique mapper name for each program and version. If you have more than one mapper per program and version, please enter a name that will uniquely identify the current mapping.

Unique Identifier:

Name	Ver	Identifier	Date Modified
Auto Lab	6	Custom_auto_918	9/19/2013 4:06:44 PM
1-2598-98-001	1	CustomPRG9-21-2011	9/21/2013 10:36:01 AM

Figure 270 Save Mapping

14. Click **Save** to save your mapping.

Save Mapping

The save mapping window allows you to 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 line of business number, the program ID and the program version. For example, 123_5_6_1.xml, which means subscriber 123, line of business 5 (fire), 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 line of business number. For example, 123_5.xml, which means subscriber 123, line of business 5 (fire).

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_5_6_1.InputMapping.xml at the program version level or 123_5.InputMapping.xml at the global level.

Although you do not need the filename when rating, you will need to know your subscriber ID, the line of business number, the program ID, the program version and the unique identifier for the mapping.

If you are utilizing sublines, adding a unique identifier to global mappings will keep the mapping within the subline that you intended. For example, if you do not enter a unique identifier, the global mapping will be saved as subscriber_line of business. If you save another mapping from a subline in the same line of business without a unique identifier, it will be saved in the same way, subscriber_line of business. This will overwrite your previously saved mapping, even though it is in another subline. Adding a unique identifier will help to keep the mappings separate.

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 line (all inputs for the line), 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 IBRU system is installed. To verify files, you must access to the IBRU server. Files are located:

C:\Program Files\Oracle\IGBU\Insbridge\Webs\RM4\Workfiles\Mappings\Input

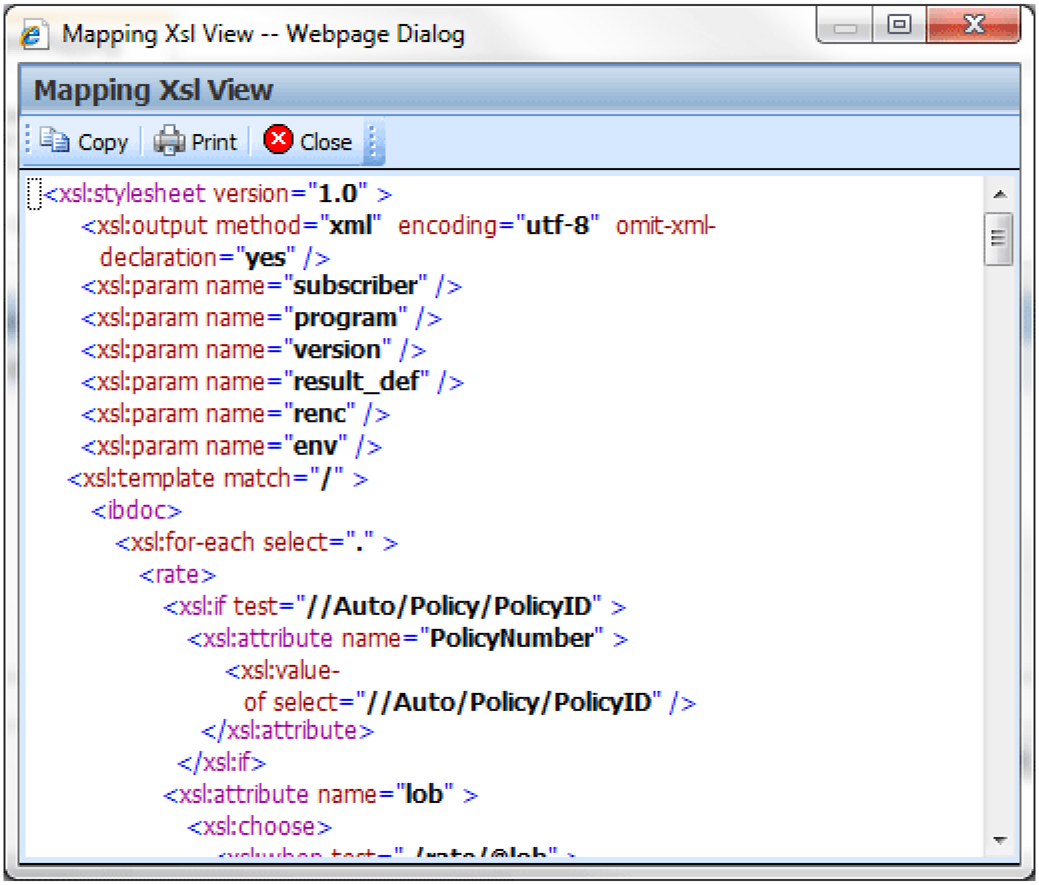
Save Mappings Options

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 allows you to 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.



```
<xsl:stylesheet version="1.0" >
    <xsl:output method="xml"  encoding="utf-8"  omit-xml-
      declaration="yes" />
    <xsl:param name="subscriber" />
    <xsl:param name="program" />
    <xsl:param name="version" />
    <xsl:param name="result_def" />
    <xsl:param name="renc" />
    <xsl:param name="env" />
  <xsl:template match="/" >
    <ibdoc>
      <xsl:for-each select="." >
        <rate>
          <xsl:if test="//Auto/Policy/PolicyID" >
            <xsl:attribute name="PolicyNumber" >
               <xsl:value-
                 of select="//Auto/Policy/PolicyID" />
              </xsl:attribute>
            </xsl:if>
          <xsl:attribute name="lob" >
            <xsl:choose>
              <xsl:when test="../rate/@lob" >
```

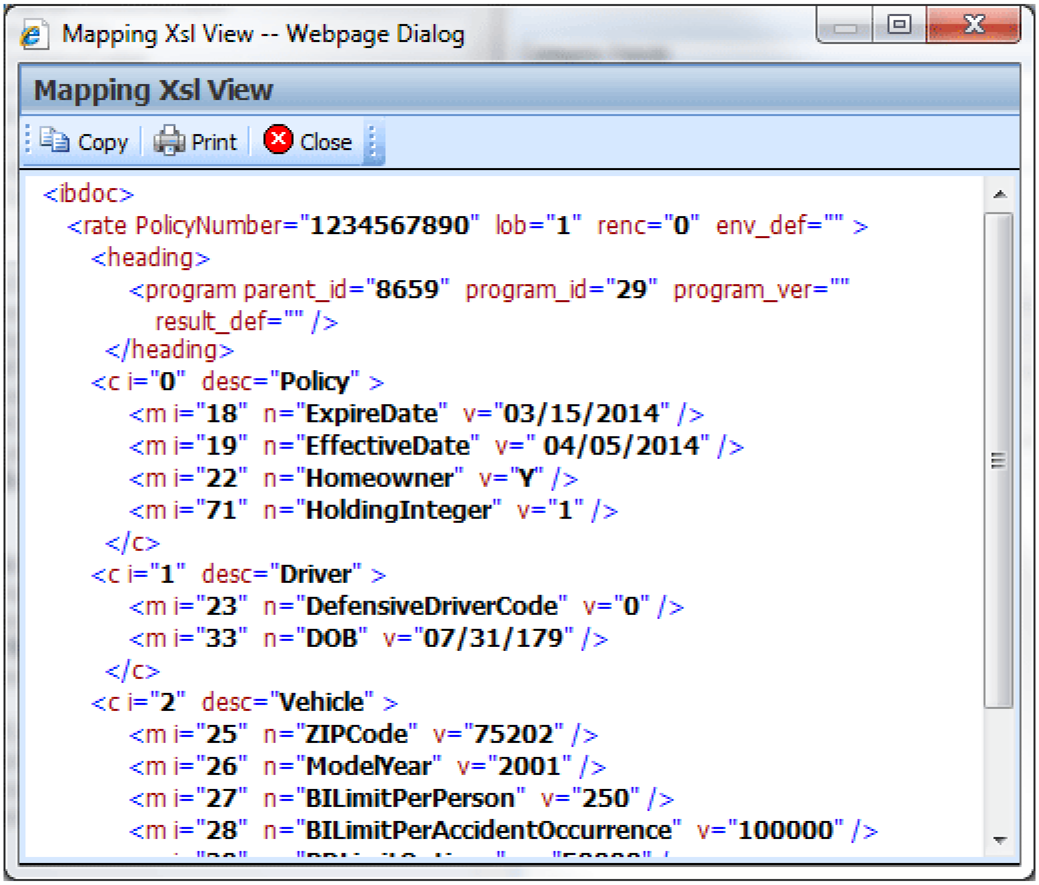
Figure 271 View Mappings Xsl

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.



```
<ibdoc>
  <rate PolicyNumber="1234567890"  lob="1"  renc="0"  env_def="" >
    <heading>
       <program parent_id="8659"  program_id="29"  program_ver=""
         result_def="" />
     </heading>
    <c i="0"  desc="Policy" >
       <m i="18"  n="ExpireDate"  v="03/15/2014" />
       <m i="19"  n="EffectiveDate"  v=" 04/05/2014" />
       <m i="22"  n="Homeowner"  v="Y" />
       <m i="71"  n="HoldingInteger"  v="1" />
     </c>
    <c i="1"  desc="Driver" >
       <m i="23"  n="DefensiveDriverCode"  v="0" />
       <m i="33"  n="DOB"  v="07/31/179" />
     </c>
    <c i="2"  desc="Vehicle" >
       <m i="25"  n="ZIPCode"  v="75202" />
       <m i="26"  n="ModelYear"  v="2001" />
       <m i="27"  n="BILimitPerPerson"  v="250" />
       <m i="28"  n="BILimitPerAccidentOccurrence"  v="100000" />
```

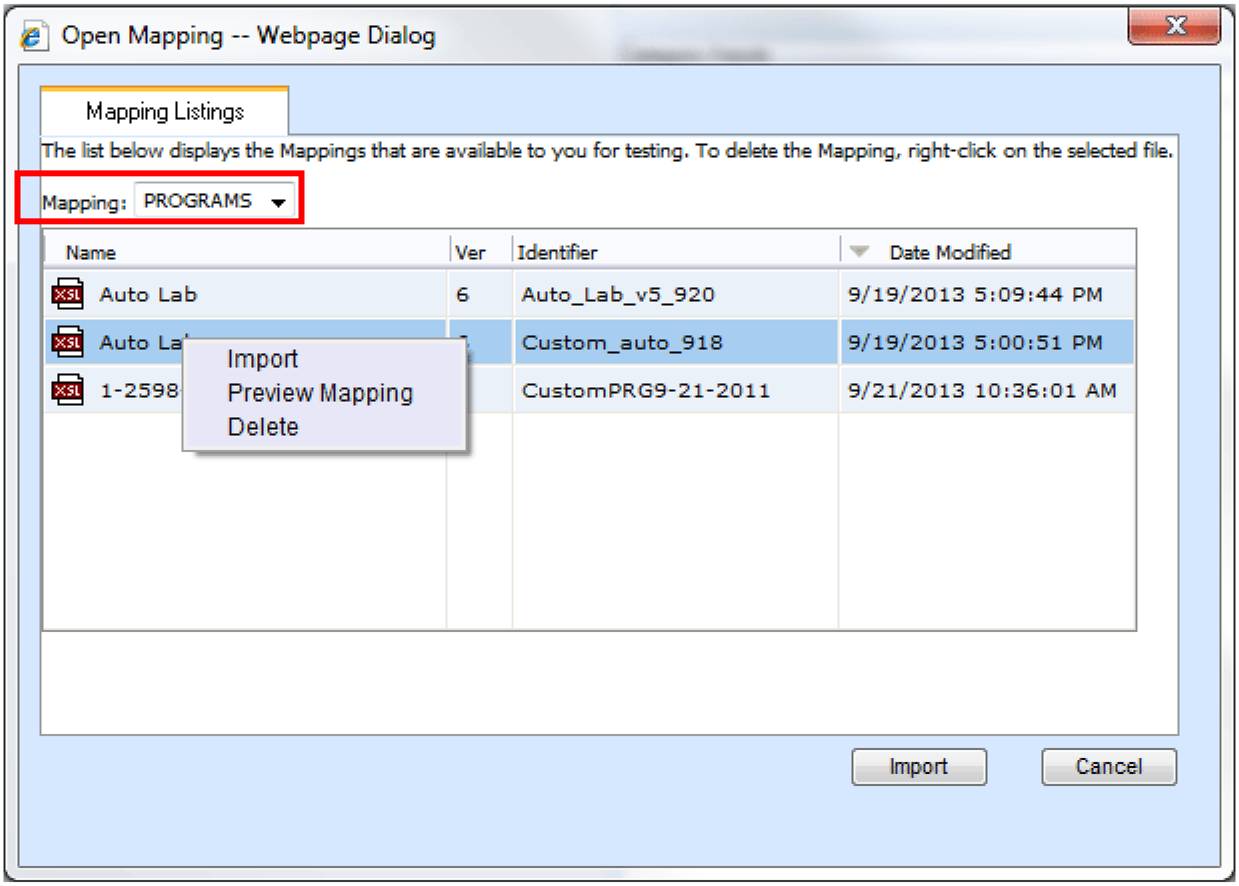
Figure 272 Test Mapping Xsl

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 IBRU server. This screen also will allow you to view or delete any custom mapping file that has been loaded to the IBRU server. Clicking on the Import Insbridge Xsl option opens a separate window where you can select the type of mapping file you want.



Open Mapping -- Webpage Dialog

Mapping Listings

The list below displays the Mappings that are available to you for testing. To delete the Mapping, right-click on the selected file.

Mapping: PROGRAMS

Name	Ver	Identifier	Date Modified
Auto Lab	6	Auto_Lab_v5_920	9/19/2013 5:09:44 PM
Auto La		Custom_auto_918	9/19/2013 5:00:51 PM
1-2598		CustomPRG9-21-2011	9/21/2013 10:36:01 AM

Figure 273 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 specific line of business. Custom global mappings are created on the **Subline**.
    - **CUSTOM:** A mapping that was created outside of RateManager and loaded into the IBRU 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.

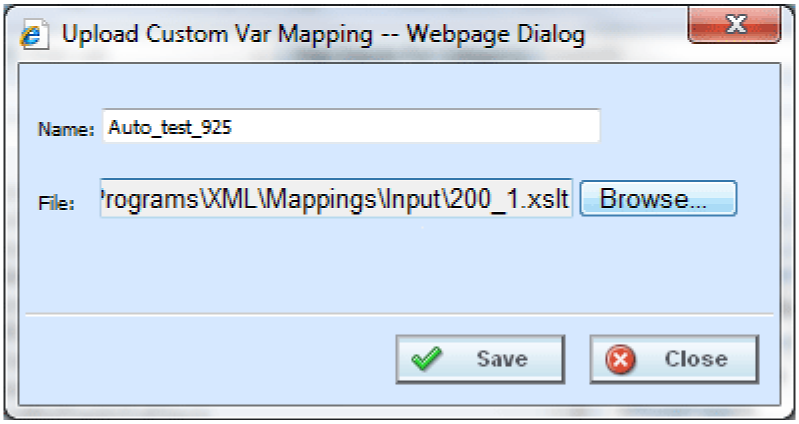
- **Import:** Imports the selected file into the Custom Xml area on the Mapping screen.
- **Preview Mapping**: Displays the mapping in the View Mapping XSL window. No changes will be allowed.
- **Delete**: Deletes the selected mapping. A warning message will be displayed before deleting.

Custom Mapping Files

Custom Mappings are any custom mapping files that have been loaded to the IBRU 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 allows you to upload your custom mappings. Uploaded mappings are available in the Testing module. Files must be xslt format.


Upload Custom Var Mapping -- Webpage Dialog

Name: Auto_test_925

File: 'rograms\XML\Mappings\Input\200_1.xslt

Figure 274 Upload Custom Input Mapping

Uploaded files are saved on the server where the IBRU system is installed. To verify files, you must access to the IBRU 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.

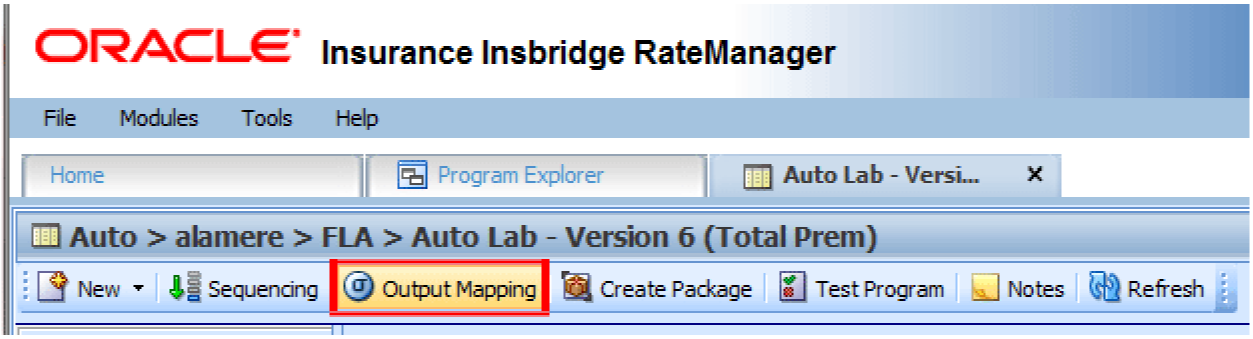
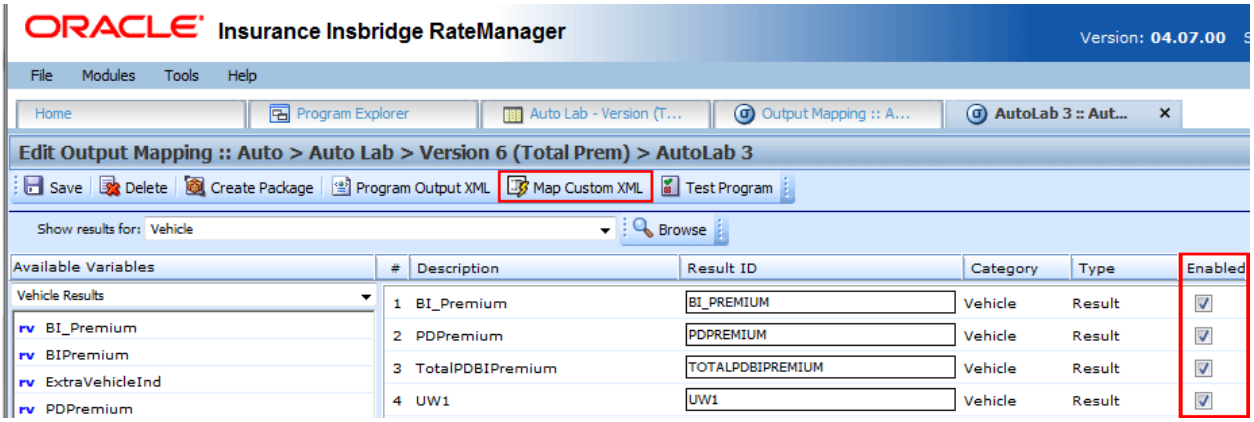


Figure 275 Navigating to Output Mapping

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.



Edit Output Mapping :: Auto > Auto Lab > Version 6 (Total Prem) > AutoLab 3

Show results for: Vehicle

Available Variables: Vehicle Results

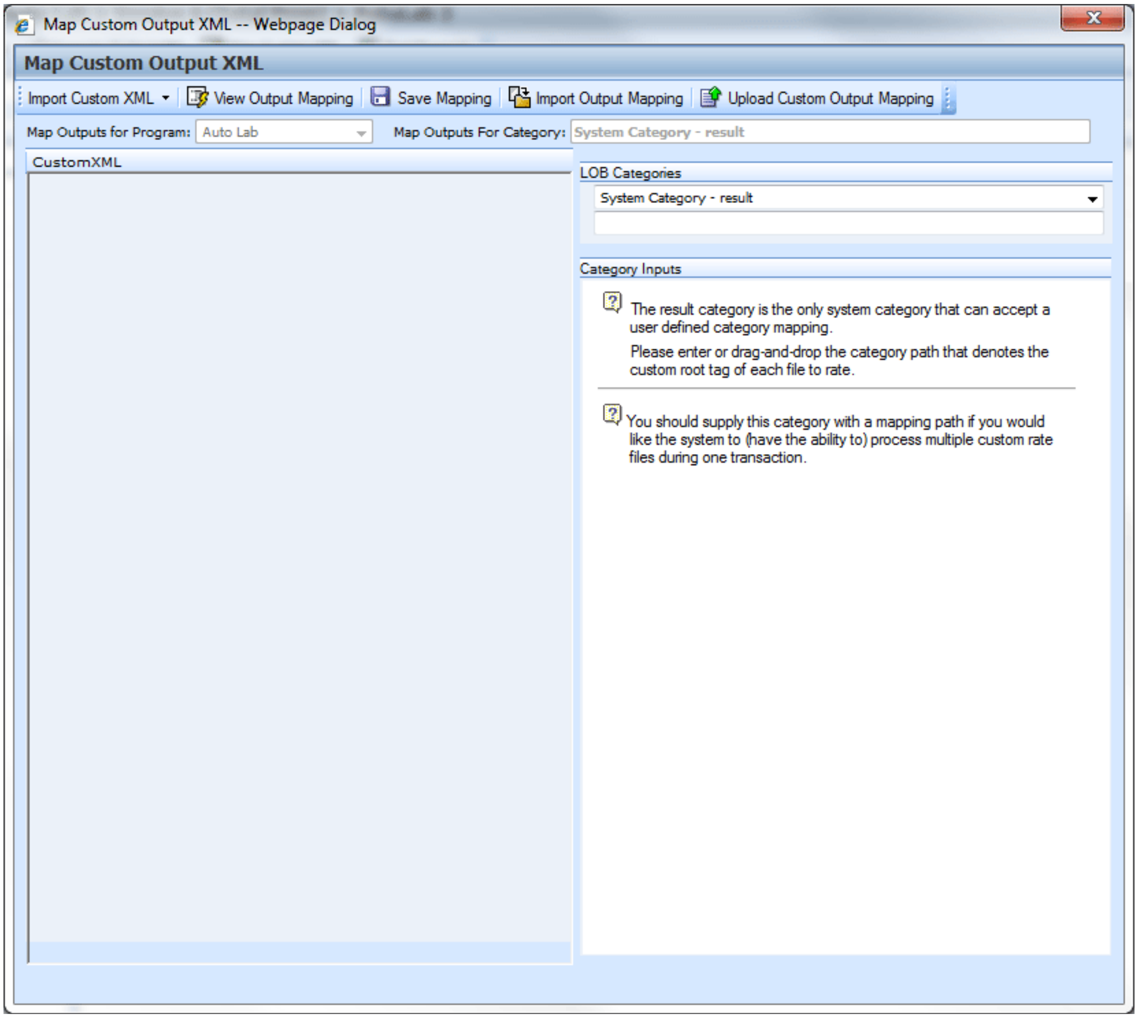
- rv BI_Premium
- rv BIPremium
- rv ExtraVehicleInd
- rv PDPremium

#	Description	Result ID	Category	Type	Enabled
1	BI_Premium	BI_PREMIUM	Vehicle	Result	☑
2	PDPremium	PDPREMIUM	Vehicle	Result	☑
3	TotalPDBIPremium	TOTALPDBIPREMIUM	Vehicle	Result	☑
4	UW1	UW1	Vehicle	Result	☑

Figure 276 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.



Map Outputs for Program: Auto Lab

Map Outputs For Category: System Category - result

LOB Categories: System Category - result

The result category is the only system category that can accept a user defined category mapping.

Please enter or drag-and-drop the category path that denotes the custom root tag of each file to rate.

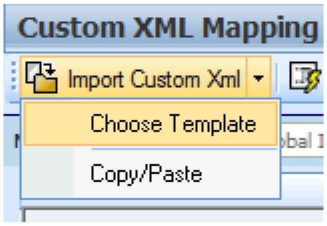
You should supply this category with a mapping path if you would like the system to (have the ability to) process multiple custom rate files during one transaction.

Figure 277 Mapping Custom Output XML

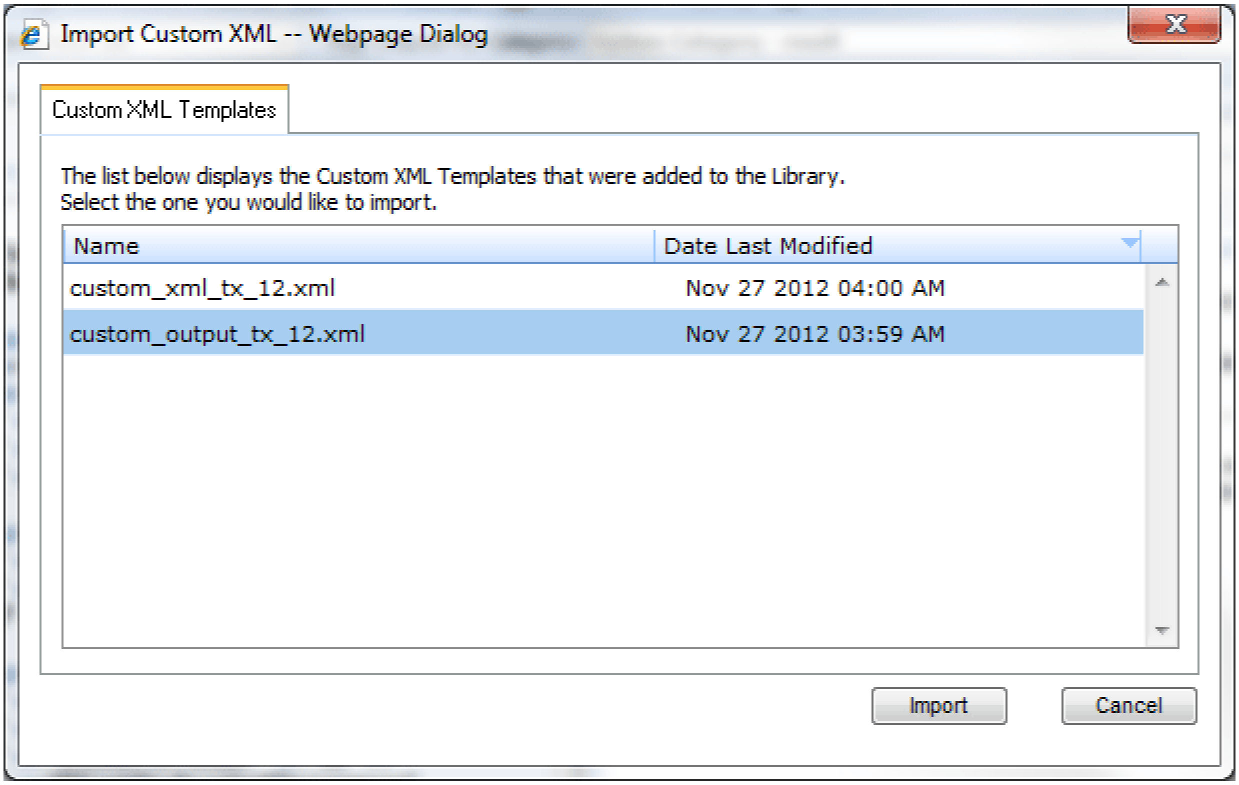
Importing Custom XML Using Choose Template

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. A dialogue box allows you to 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.



The list below displays the Custom XML Templates that were added to the Library.
Select the one you would like to import.

Name	Date Last Modified
custom_xml_tx_12.xml	Nov 27 2012 04:00 AM
custom_output_tx_12.xml	Nov 27 2012 03:59 AM

Figure 278 Uploading a Custom Output XML Template

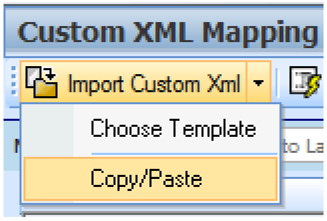
3. Click Import.

4. The custom XML will be imported into RateManager and the Map Custom Output XML screen will refresh 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 allows you to 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. 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 popup menu or by pressing **Ctrl+V**.
7. The XML will be pasted into the text box.
8. Click **OK**.
9. The custom XML will be imported into RateManager and the **Map Custom Output XML** screen will refresh.

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 need 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 LOB 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, you need to save the mapping. To do this, click **Save Mapping**.

Save Output Mapping

The save mapping screen allows you to 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 allows you 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 line of business number, the program ID and the program version. For example, 123_5_6_1.xslt, which means subscriber 123, line of business 5 (fire), 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_5_6_1.OutputMapping.xslt.

   Although you will not need the filename when rating, you will need to know your subscriber ID, the line of business number, the program ID, the program version and the unique identifier for the mapping.

6. Finally, you will need 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.

Testing in Testing

1. Navigate to Testing.

2. Click . The will open the **Rate New Custom File** window.

3. Browse for the XML file you want to rate.

4. Enter your subscriber ID, if it is not already listed.

5. Enter the number for the line of business you are rating. For example, if you are rating a file for the fire line of business, then enter **5**.

6. Enter the program ID you are rating.

7. Enter the program version you are rating.

8. Select the type of mapping you are using, **GLOBAL, LOCAL** or **CUSTOM**.

   GLOBAL: A mapping that can be used by all programs under a specific line of business. Global mappings are created from the **Global Inputs** screen.

   LOCAL: A mapping that is specific to a particular program version. Local mappings are created from the **Program Details** tab.

   CUSTOM: A mapping that was created outside of RateManager and then imported.

9. Select the unique Identifier from the drop down menu.

   When you select the file you want to test, the Identifier will automatically be updated with the available files for you to select.

10. The example below is for subscriber **1002,** line of business **5** (fire), program **6,** version **1**. It uses a **GLOBAL** mapping that has a unique identifier of **InputMapping**.

11. When you have finished entering the information, click **RATE XML**.

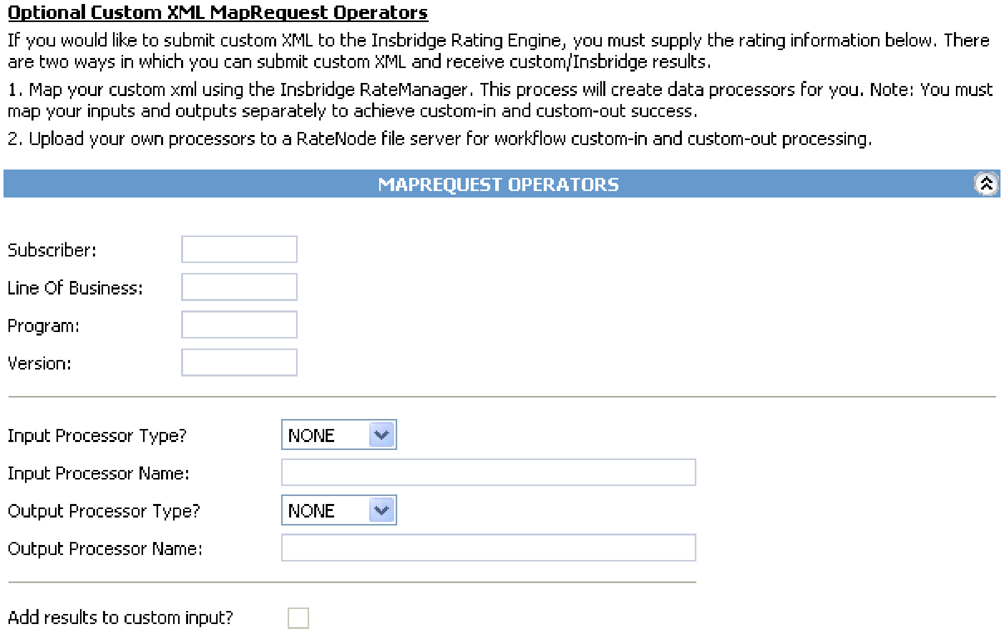
12. The file will be rated and displayed in the list of input files and the results will be in the list of result files. You can open the result file by double-clicking it, just like a standard rating report.

13. If you want to rate a custom file with debug, be sure to select the debug checkbox in the top bar menu before rating the file.

14. If you want to re-rate the file, you can either double-click it or select it and click **RATE XML FILE**. If you double-click the file, it will be re-rated with the same mapping. If you click **RATE XML FILE**, you will have the option to change the mapping used.

TESTING IN IBFA

1. Navigate to the **SoftRater Test Interface** located on the **SoftRater** screen.
2. Browse for the file you want to rate or enter the XML in the text box.
3. Expand the **Maprequest Operators** section by clicking ⌄.



Optional Custom XML MapRequest Operators

If you would like to submit custom XML to the Insbridge Rating Engine, you must supply the rating information below. There are two ways in which you can submit custom XML and receive custom/Insbridge results.

1. Map your custom xml using the Insbridge RateManager. This process will create data processors for you. Note: You must map your inputs and outputs separately to achieve custom-in and custom-out success.

2. Upload your own processors to a RateNode file server for workflow custom-in and custom-out processing.

Subscriber:

Line Of Business:

Program:

Version:

Input Processor Type? NONE

Input Processor Name:

Output Processor Type? NONE

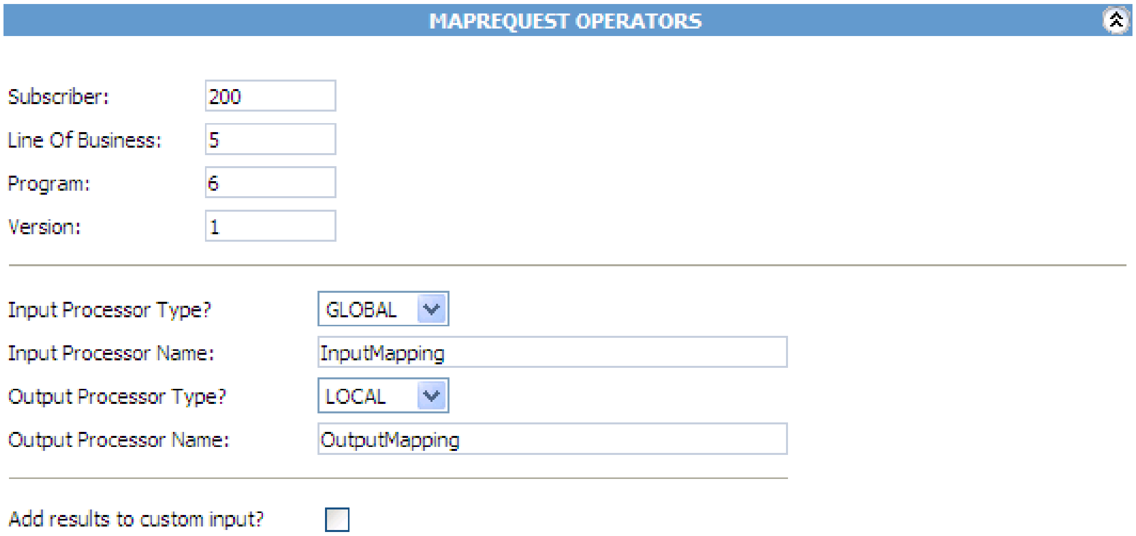
Output Processor Name:

Add results to custom input? □

Figure 279 Testing a Mapping in IBFA

4. Enter your subscriber ID.
5. Enter the number for the line of business you are rating. For example, if you are rating a file for the fire line of business, then enter **5**.
6. Enter the program ID you are rating.
7. Enter the program version you are rating.
8. Select the type of input mapping you are using, **GLOBAL**, **LOCAL** or **CUSTOM**.
9. If you saved the input mapping with a unique identifier, enter it in the **Input Processor Name** box.
10. Select the type of output mapping, if any, you are using, **GLOBAL, LOCAL** or **CUSTOM**.
11. If you saved the output mapping with a unique identifier, enter it in the **Output Processor Name** box.
12. The example below is for subscriber 200, line of business 5 (fire), program 6, version 1. It uses a

GLOBAL input mapping that has a unique identifier of InputMapping. It uses a LOCAL output mapping that has a unique identifier of OutputMapping.



Subscriber: 200

Line Of Business: 5

Program: 6

Version: 1

Input Processor Type? GLOBAL

Input Processor Name: InputMapping

Output Processor Type? LOCAL

Output Processor Name: OutputMapping

Add results to custom input? □

Figure 280 Completed IBFA Testing

13. If you are using both a custom input and custom output mapping and you want the results to be added to the original input file, select **Add results to custom input**.

14. Select any Rate Operators you want to use in conjunction with the Maprequest Operators. For more information, see the SoftRater Test Interface topic Optional Rate Operators.

15. Click **Execute**.

16. The file will be rated and the results will be returned in the **Result Information** text box.

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 for the subline where you want to map inputs.
2. On the **Global Inputs Listing** screen, click **MAP CUSTOM XML**.
3. This will open the **Input Mapping** screen.
4. Click **UPLOAD CUSTOM INPUT MAPPING**. This will open 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 23

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.*

To Create a Package

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:

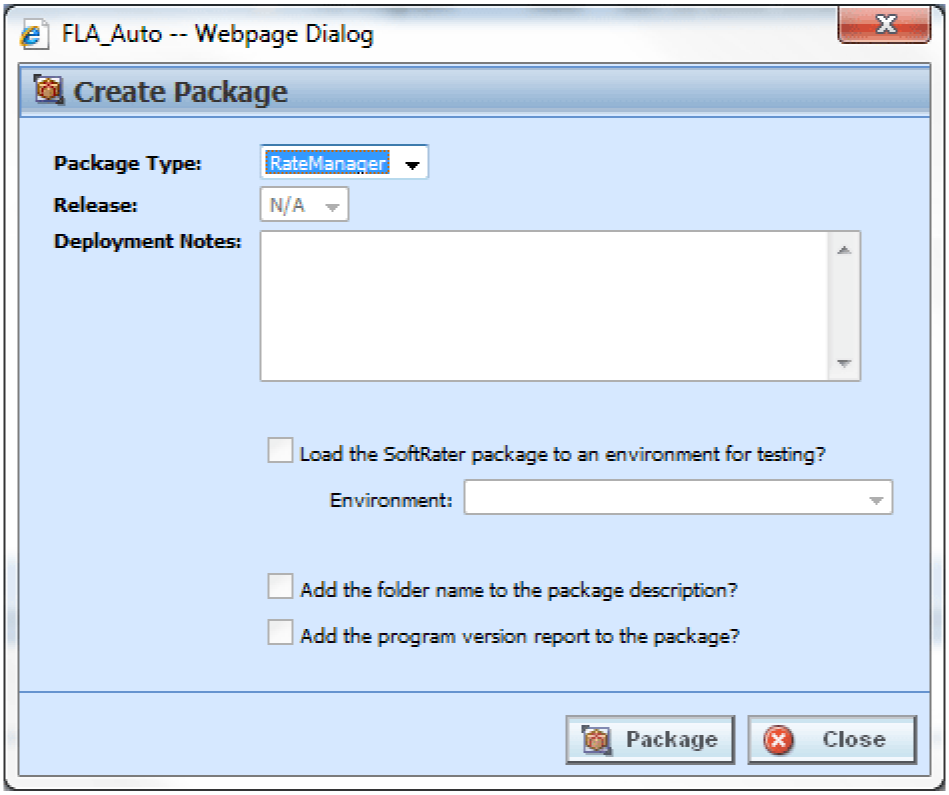
- Program Details
- Program Explorer
- Sequencing
- Output Mapping
- Edit Output Mapping
- Edit Releases
- MultiLines

On these tabs, click **Create Package** on the navigation bar.

1. Navigate to the **Program Details** tab for the subline 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 will open allowing you to set options for the package.

   If "Enforce Release Packaging" is set to True in Preferences, the Create Package screen will have additional options that can be set.



Package Type: RateManager

Release: N/A

Deployment Notes:

□ Load the SoftRater package to an environment for testing?

Environment:

□ Add the folder name to the package description?

□ Add the program version report to the package?

Figure 281 Creating a Package

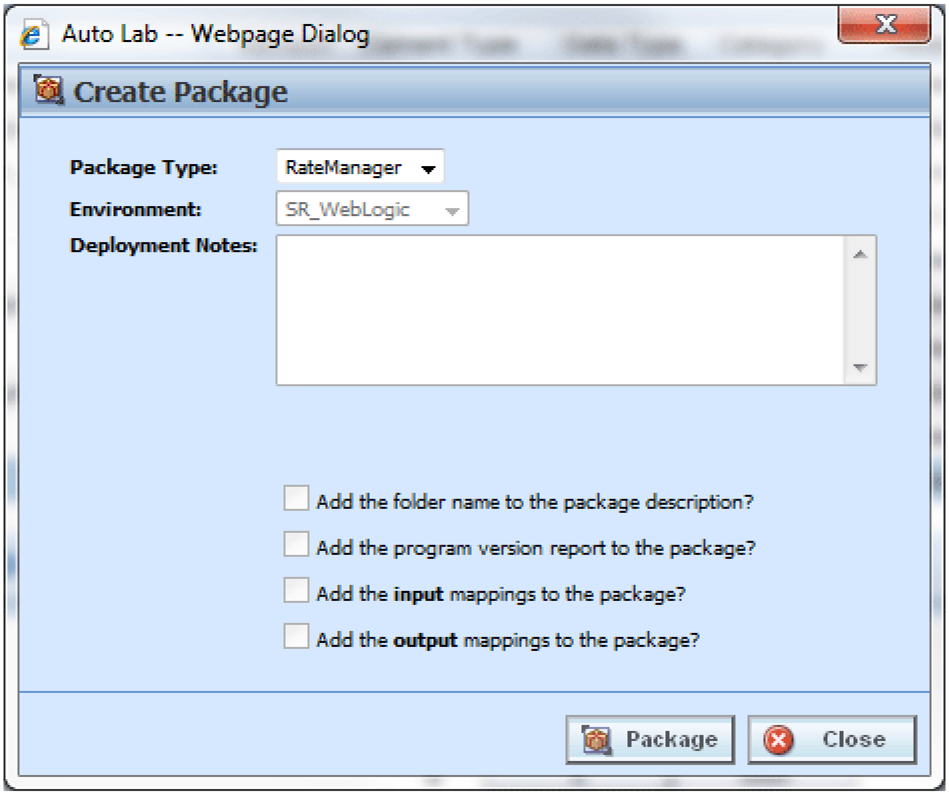
3. When you have finished setting the package options, click **Package**. The packaging process will start and the progress will be shown. A summary will be shown when packaging is finished.

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.*

Fields: 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 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.



Auto Lab -- Webpage Dialog

Create Package

Package Type: RateManager

Environment: SR_WebLogic

Deployment Notes:

□ Add the folder name to the package description?

□ Add the program version report to the package?

□ Add the **input** mappings to the package?

□ Add the **output** mappings to the package?

Figure 282 Creating a Package Release Packaging Set to True

Package Type: Allows you to select the type of package to be created:

- RateManager
- SoftRater

Release: If you are creating a SoftRater package, this allows you to set the environment the package will initially be placed on. 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.

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, this allows you to enter comments about the package. The comments are viewable from the SoftRater Explorer.

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.

Fields: Enforce Release Packaging Set to False:

Package Type: Allows you to select the type of package to be created:

- **RateManager**
- **SoftRater**

Environment: If you are creating a SoftRater package, this allows you to set the environment the package will initially be placed in. 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.

NOTE: *If the Enforce Release Packaging is set to False, users will have the option to select the Environment and not the Release when they go to package.*

Deployment Notes: If you a creating a SoftRater package, this allows you to enter comments about the package. The comments are viewable from the SoftRater Explorer.

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.

GLOBAL VERSIONING PACKAGE

In order for SoftRater to automatically choose the version of a program to rate, a global versioning package (or versioning package) must be created. A versioning package is global to all programs under a particular subline. Only one versioning package needs to be created for each subline. Create a new versioning package anytime you make changes to the way program versions are selected for a program.

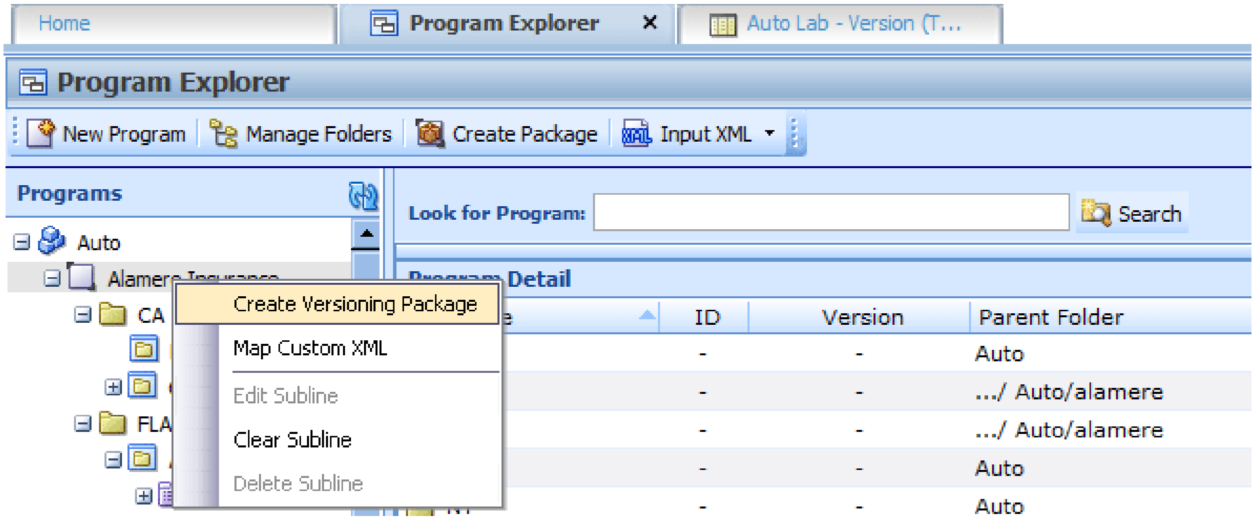
Versioning packages are created the same way program packages are created, except they are created from the subline Programs Folders screen. For more information on versioning, see Versioning.
Global Versioning logic uses the program versioning criteria to determine which program version should be used by the program being called.

Creating a Global Versioning Package: Allowing for Override

System administrators set override options. If you want to select a different environment for loading a SoftRater package to, the override option must be set to true.

There are two ways to create a versioning package.

The **first** way is to right click any subline and select Create Versioning Package.



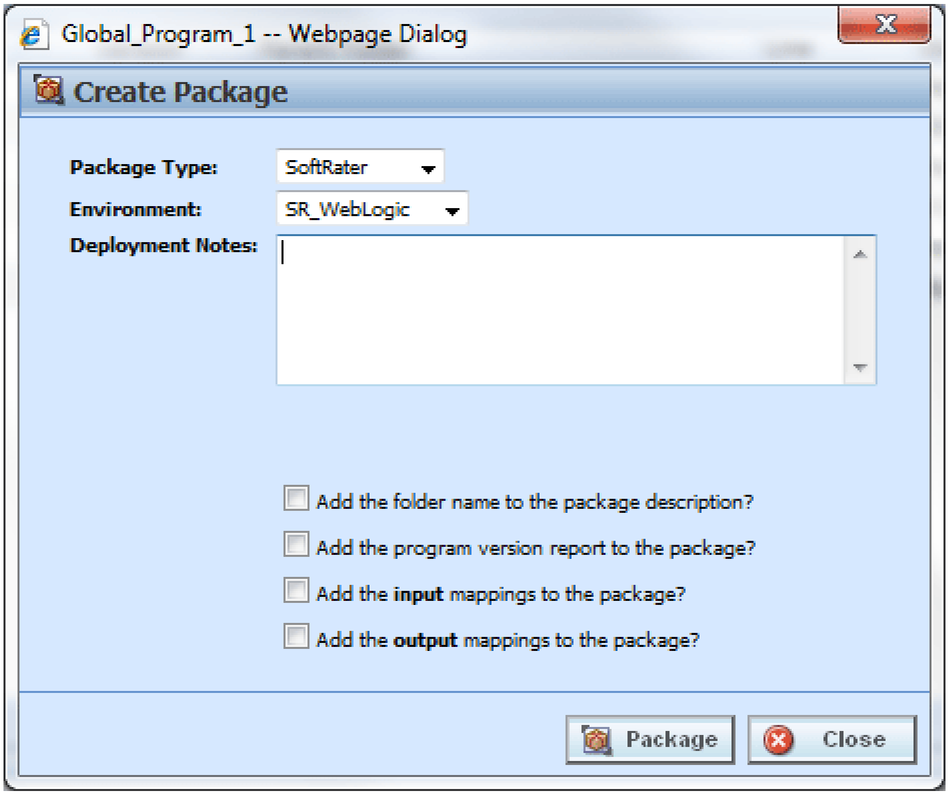
ID	Version	Parent Folder
-	-	Auto
-	-	.../ Auto/alamere
-	-	.../ Auto/alamere
-	-	Auto
-	-	Auto

Figure 283 Cresting Global Versioning Package

This pulls up the global versioning package screen and allow you to create a versioning package for that subline. This feature can be used at any time and does not require you to leave the screen you are currently working in.

The **second** way is to:

1. Navigate to the **Program Explorer**.
2. Click **Manage Folders**. The Manage Program Folders screen will open.
3. Click **Create Versioning Package** in the top bar menu. The Global Packaging screen is displayed.



Create Package

Package Type: SoftRater

Environment: SR_WebLogic

Deployment Notes:

□ Add the folder name to the package description?

□ Add the program version report to the package?

□ Add the **input** mappings to the package?

□ Add the **output** mappings to the package?

Figure 284 Creating a Global Versioning Package

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: Allows you to select the type of package to be created:

   - **RateManager**
   - **SoftRater**

   Release: If you are creating a SoftRater package, this allows you to set the environment the package will initially be placed on. 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.

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, this allows you to enter comments about the package. The comments are viewable from the SoftRater Explorer.

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.

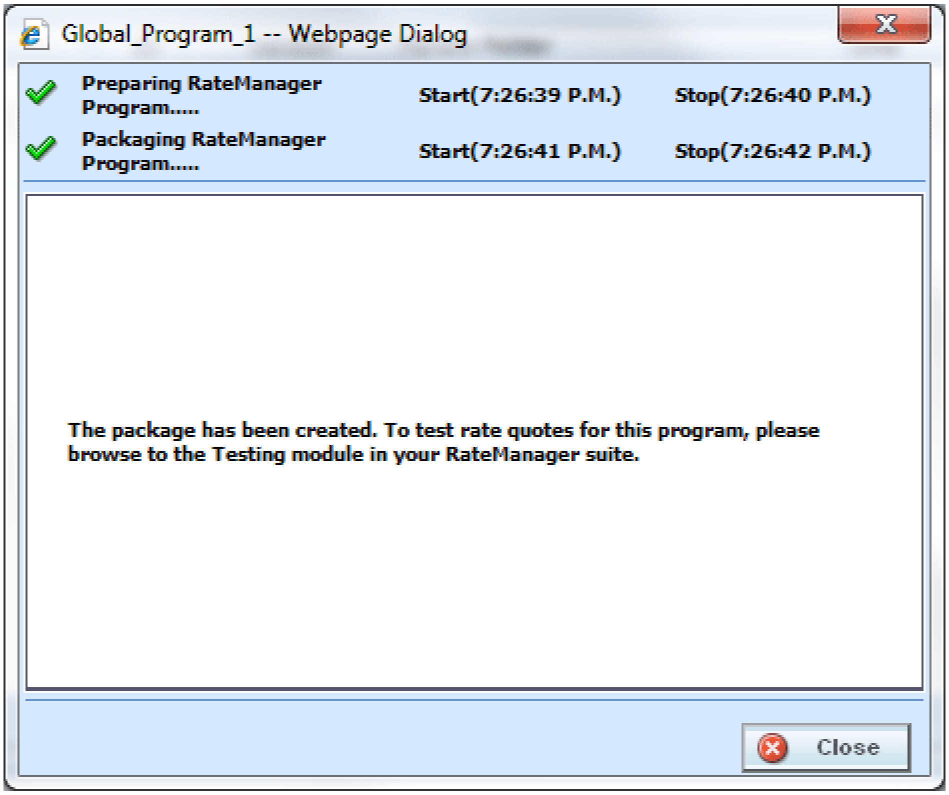
Environment: If you checked Yes for loading SoftRater package to an environment for testing, you must select the environment that the testing package will be loaded to. This option will be available if the system has been setup to allow release override.

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.



Global_Program_1 -- Webpage Dialog

✓	Preparing RateManager Program.....	Start(7:26:39 P.M.)	Stop(7:26:40 P.M.)
✓	Packaging RateManager Program.....	Start(7:26:41 P.M.)	Stop(7:26:42 P.M.)

The package has been created. To test rate quotes for this program, please browse to the Testing module in your RateManager suite.

Figure 285 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 Global Versioning Package: Not Allowing for Override

System administrators set override options. If the system does not allow for overrides, the create global 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: Allows you to select the type of package to be created:

- **RateManager**
- **SoftRater**

Environment: If you are creating a SoftRater package, this allows you to set the environment the package will initially be placed in. 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, this allows you to enter comments about the package. The comments are viewable from the SoftRater Explorer.

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**

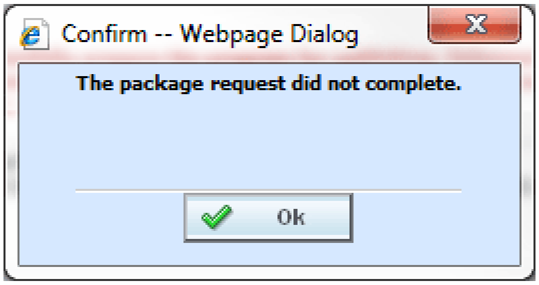
category. SoftRater uses inputs to determine which categories need to be processed. If you receive the error below, it means that you have an algorithm in the sequence whose category does not have any inputs used.

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.



The package request did not complete.

Figure 286 Package Failure

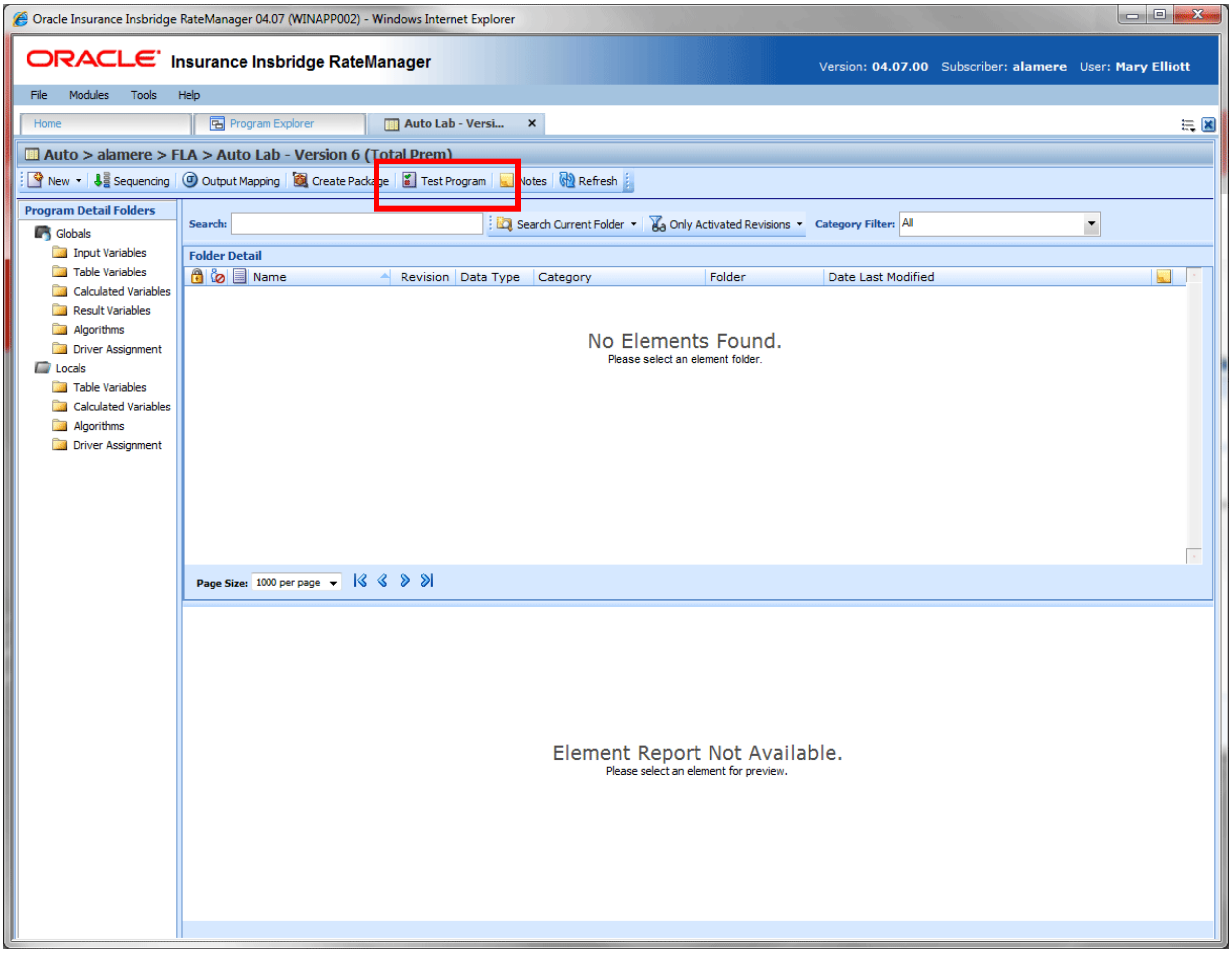
Chapter 24

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 box fields 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.



Version: 04.07.00 Subscriber: alamere User: Mary Elliott

Auto > alamere > FLA > Auto Lab - Version 6 (Total Prem)

Category Filter: All

No Elements Found.
Please select an element folder.

Page Size: 1000 per page

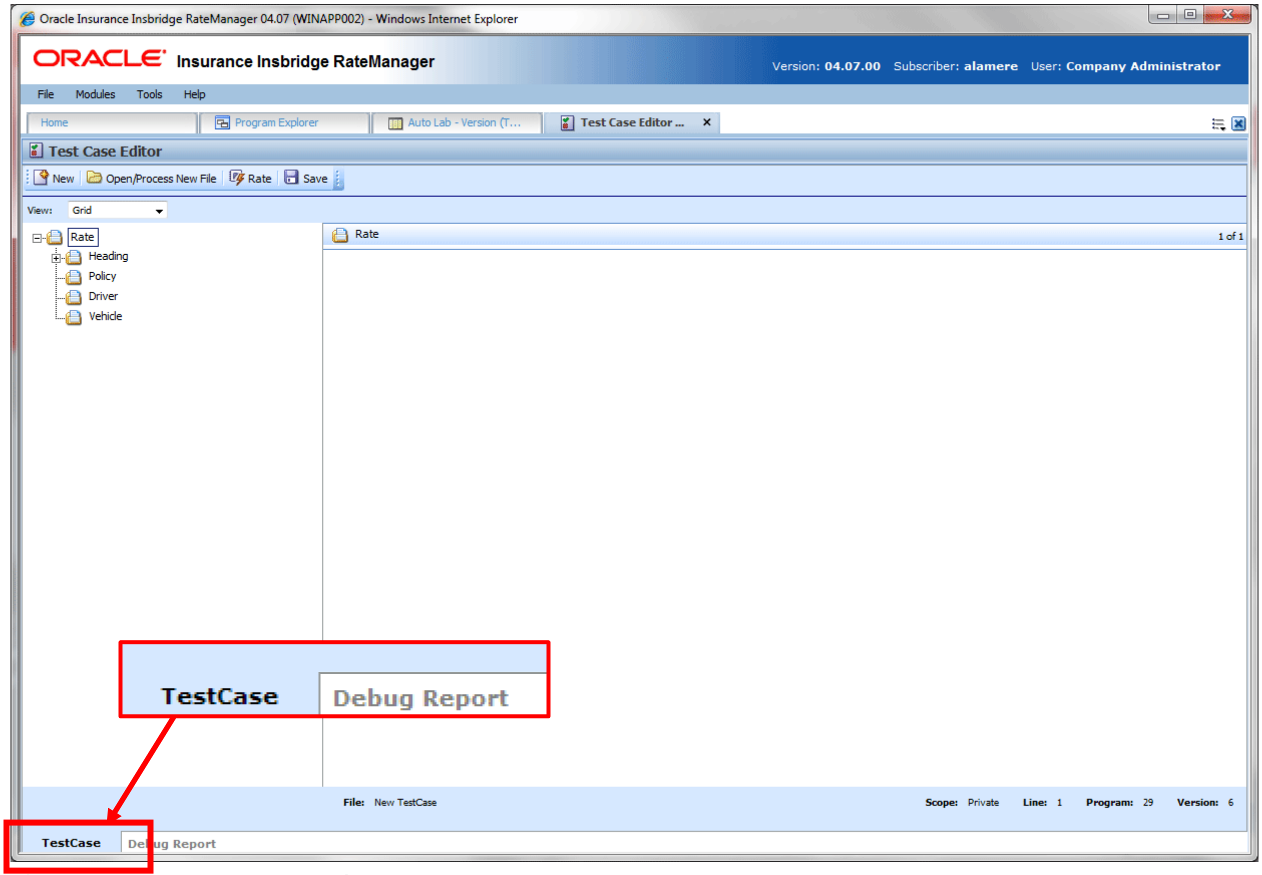
Element Report Not Available.
Please select an element for preview.

Figure 287 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 Line of Business→Subline→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.



Version: 04.07.00 Subscriber: alamere User: Company Administrator

View: Grid

File: New TestCase Scope: Private Line: 1 Program: 29 Version: 6

Figure 288 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 allows you to 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 will open 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	**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/Process New File	**Open:** Allows you to browse for existing test cases.
Rate	**Rate:** Rates the current test case.
ValidateXML	**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	**Save:** Saves the current test case. This option is only for test cases within the program version.
View: Grid	**View:** Allows you to switch between XML view and Grid view. When opening a new test case, Grid view is always the default.

Fields:

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 LOB 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 hand window of the Test Program will show the categories and category structure created for the program. When a specific category is clicked, the right hand 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 hand side has a right click menu that allows you to add, copy, or delete. The Heading and the Policy category do not have a right click menu.

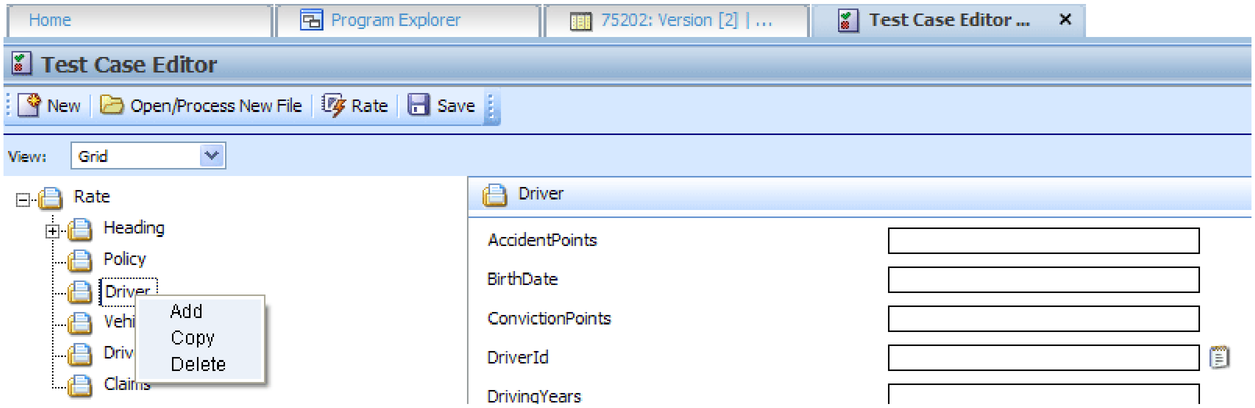


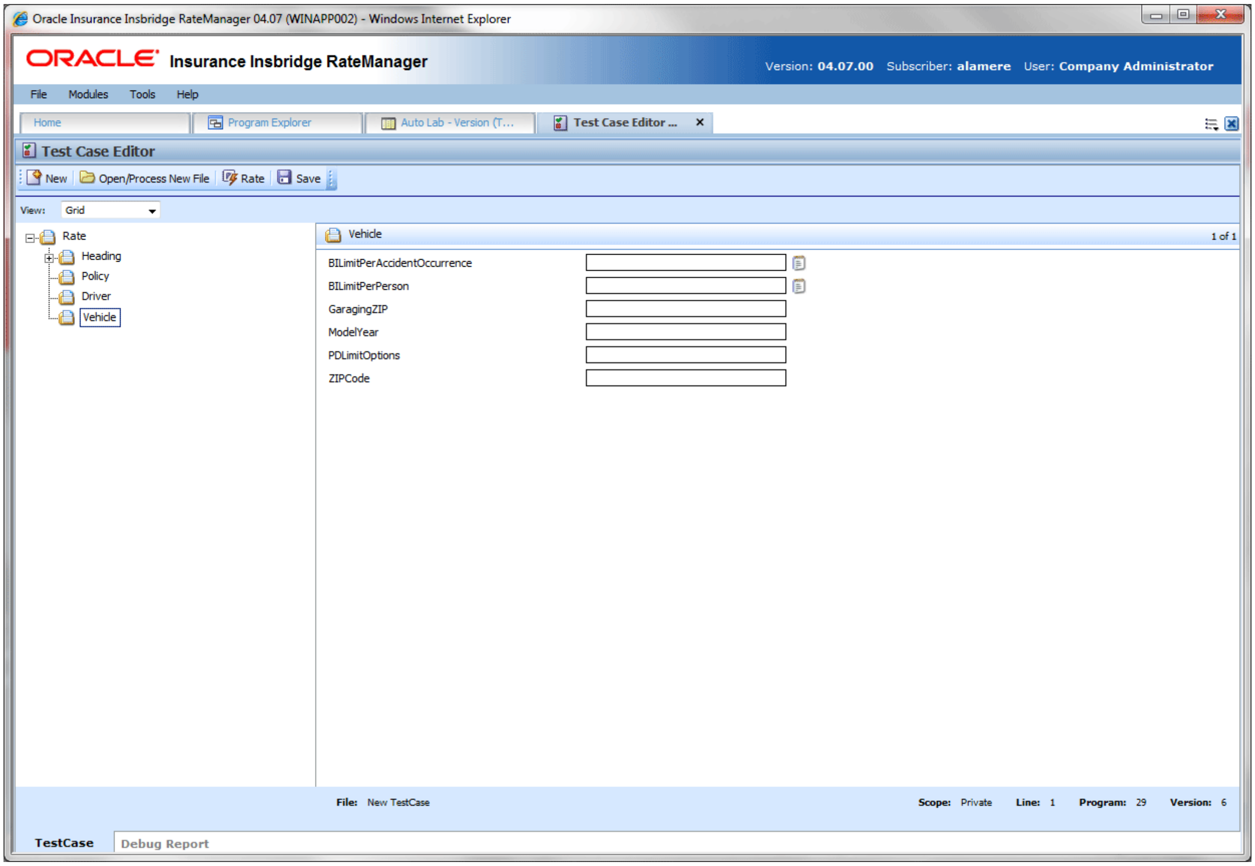
Figure 289 Right Click Menu for Test Case

- **Add**: Adds another category. Use this when you need 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.



Version: 04.07.00 Subscriber: alamere User: Company Administrator

View: Grid

BILimitPerAccidentOccurrence	
BILimitPerPerson	
GaragingZIP	
ModelYear	
PDLimitOptions	
ZIPCode	

File: New TestCase Scope: Private Line: 1 Program: 29 Version: 6

Figure 290 Inputs for Selected Category

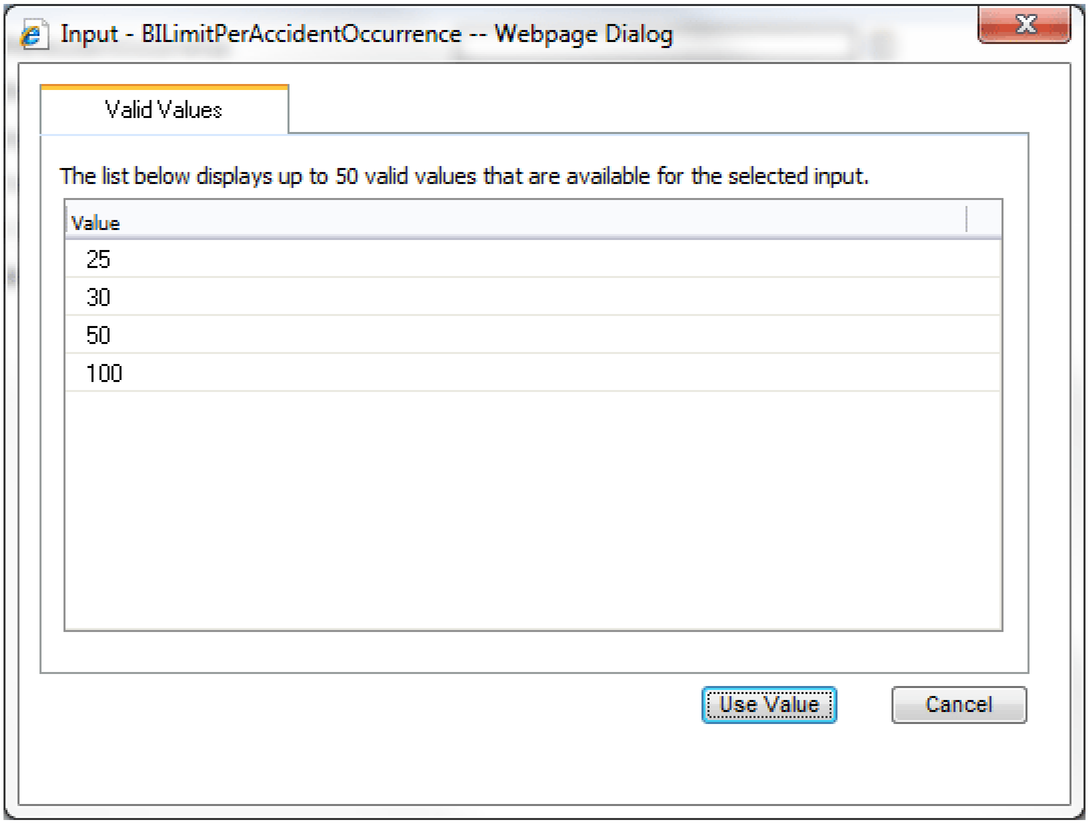
2. If not already displayed, choose Grid from the view menu.

3. In the left hand window, click on the category name where you want to enter data. The right hand window will display the applicable inputs for the chosen category.

4. In the right hand 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 popup menu.



The list below displays up to 50 valid values that are available for the selected input.

Value
25
30
50
100

Figure 291 Entering Inputs in Test Program

c. Select the option you want. Click **USE VALUE**. This will close the popup and return 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 will switch 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

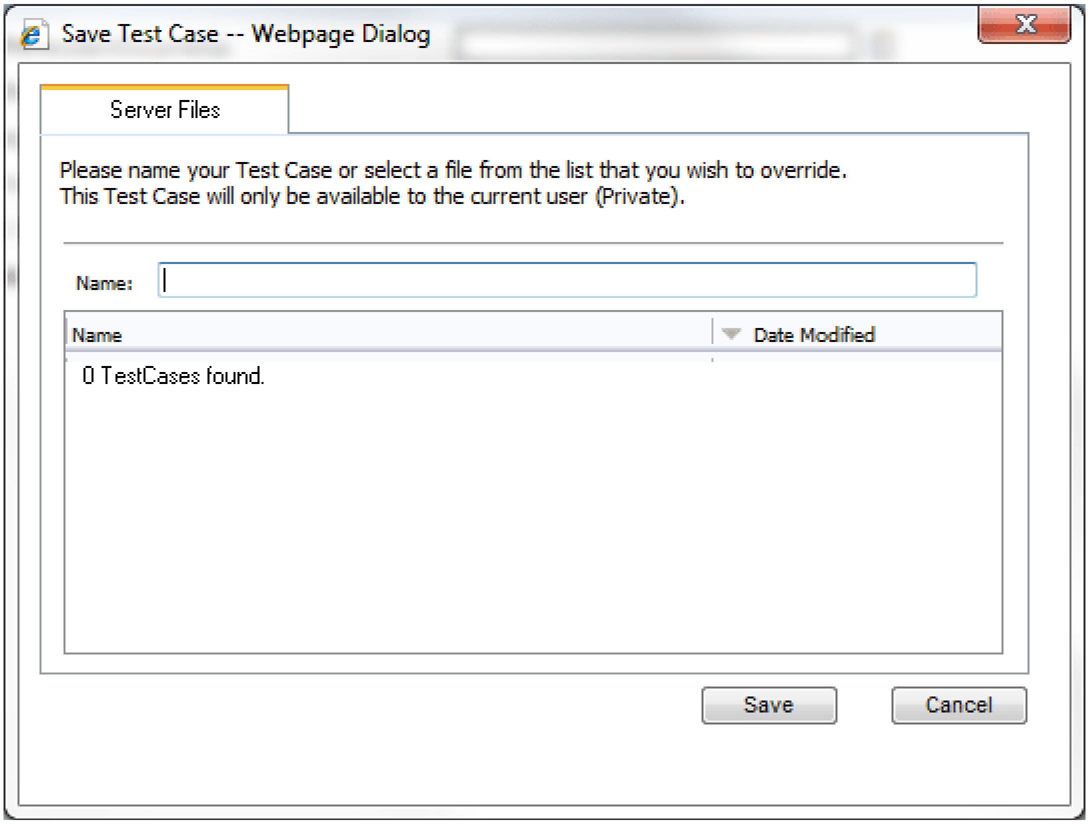
You can clear the fields and begin your test case again by clicking **New** in the top menu bar. This will clear the fields and allow you to enter new values for testing. Any new or deleted categories will also be cleared or added back to the test case.

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 allows you to 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.



Please name your Test Case or select a file from the list that you wish to override.
This Test Case will only be available to the current user (Private).

Name:

Name	Date Modified
0 TestCases found.	

Figure 292 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.

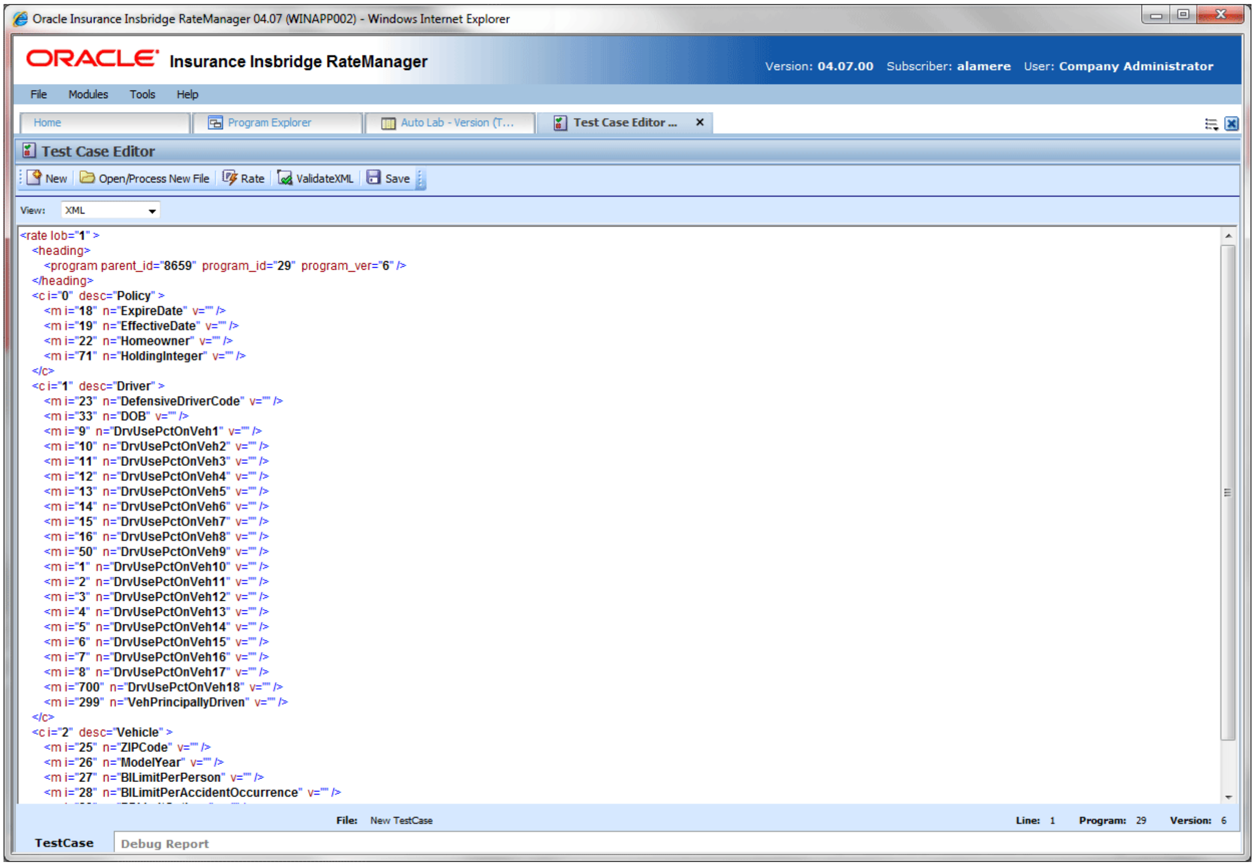
Creating a Test Case in XML View

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.

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.



Version: 04.07.00 Subscriber: alamere User: Company Administrator

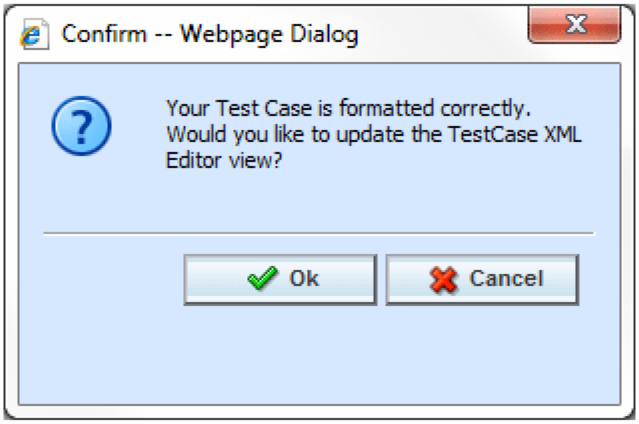
View: XML

```
<rate lob="1" >
  <heading>
    <program parent_id="8659"  program_id="29"  program_ver="6" />
  </heading>
  <c i="0"  desc="Policy" >
    <m i="18"  n="ExpireDate"  v="" />
    <m i="19"  n="EffectiveDate"  v="" />
    <m i="22"  n="Homeowner"  v="" />
    <m i="71"  n="HoldingInteger"  v="" />
  </c>
  <c i="1"  desc="Driver" >
    <m i="23"  n="DefensiveDriverCode"  v="" />
    <m i="33"  n="DOB"  v="" />
    <m i="9"  n="DrvUsePctOnVeh1"  v="" />
    <m i="10"  n="DrvUsePctOnVeh2"  v="" />
    <m i="11"  n="DrvUsePctOnVeh3"  v="" />
    <m i="12"  n="DrvUsePctOnVeh4"  v="" />
    <m i="13"  n="DrvUsePctOnVeh5"  v="" />
    <m i="14"  n="DrvUsePctOnVeh6"  v="" />
    <m i="15"  n="DrvUsePctOnVeh7"  v="" />
    <m i="16"  n="DrvUsePctOnVeh8"  v="" />
    <m i="50"  n="DrvUsePctOnVeh9"  v="" />
    <m i="1"  n="DrvUsePctOnVeh10"  v="" />
    <m i="2"  n="DrvUsePctOnVeh11"  v="" />
    <m i="3"  n="DrvUsePctOnVeh12"  v="" />
    <m i="4"  n="DrvUsePctOnVeh13"  v="" />
    <m i="5"  n="DrvUsePctOnVeh14"  v="" />
    <m i="6"  n="DrvUsePctOnVeh15"  v="" />
    <m i="7"  n="DrvUsePctOnVeh16"  v="" />
    <m i="8"  n="DrvUsePctOnVeh17"  v="" />
    <m i="700"  n="DrvUsePctOnVeh18"  v="" />
    <m i="299"  n="VehPrincipallyDriven"  v="" />
  </c>
  <c i="2"  desc="Vehicle" >
    <m i="25"  n="ZIPCode"  v="" />
    <m i="26"  n="ModelYear"  v="" />
    <m i="27"  n="BILimitPerPerson"  v="" />
    <m i="28"  n="BILimitPerAccidentOccurrence"  v="" />
```

File: New TestCase   Line: 1   Program: 29   Version: 6

Figure 293 Test Case XML View

4. Click **VALIDATE XML** to validate the XML before rating the test case.


Confirm -- Webpage Dialog

Your Test Case is formatted correctly.
Would you like to update the TestCase XML Editor view?

Figure 294 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 will switch to the Debug Report sub-tab once the test case has finished rating. See Using the Debug Tab for more information.

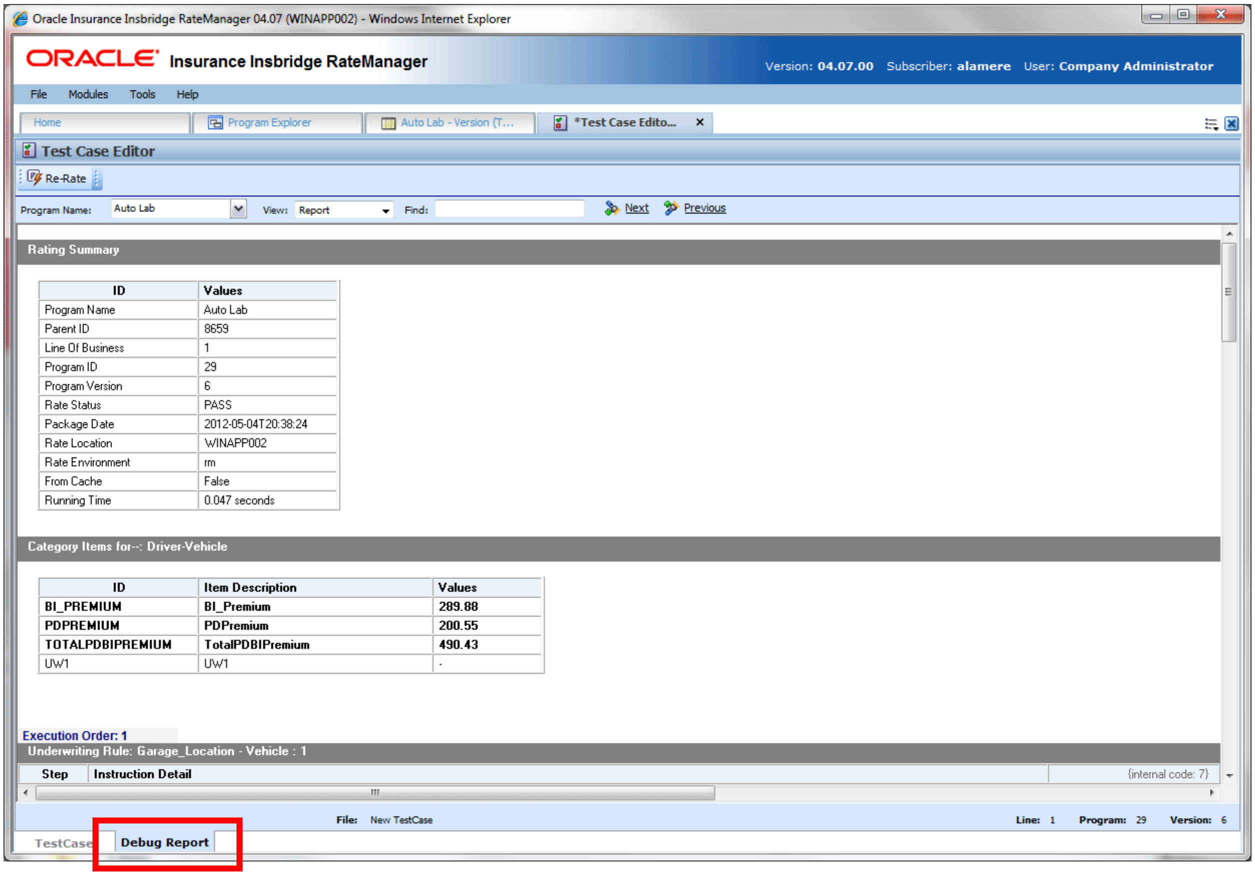
8. If you would like to save your test case, return to the Test Case sub-tab

9. Click **Save** to save the test case.

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 will switch you to the Debug Report sub-tab.

Rating Summary information will be at the top, followed by detailed Category Information. The execution order is listed next.


Oracle Insurance Insbridge RateManager 04.07 (WINAPP002) - Windows Internet Explorer
ORACLE Insurance Insbridge RateManager
Version: 04.07.00 Subscriber: alamere User: Company Administrator
File
Modules
Tools
Help
Home
Program Explorer
Auto Lab - Version (T…
*Test Case Edito…
Test Case Editor
Re-Rate
Next
Previous
TestCase
Debug Report

Program Name: Auto Lab View: Report Find:

Rating Summary

ID	Values
Program Name	Auto Lab
Parent ID	8659
Line Of Business	1
Program ID	29
Program Version	6
Rate Status	PASS
Package Date	2012-05-04T20:38:24
Rate Location	WINAPP002
Rate Environment	rm
From Cache	False
Running Time	0.047 seconds

Category Items for--: Driver-Vehicle

ID	Item Description	Values
BI_PREMIUM	BI_Premium	289.88
PDPREMIUM	PDPremium	200.55
TOTALPDBIPREMIUM	TotalPDBIPremium	490.43
UW1	UW1	-

Execution Order: 1

Underwriting Rule: Garage_Location - Vehicle : 1

Step	Instruction Detail	
		(internal code: 7)

File: New TestCase Line: 1 Program: 29 Version: 6

Figure 295 View Debug Reports

Functions:

Re-Rate	**Re-rate:** Re-rate will rate the selected test case and replace any pre-existing debug report with the newly rated one.
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 using MultiLines. See MultiLines Programs for more information.
View: XML	**View:** The debug tab will automatically show results in a report view. You may change this view to show the raw XML output.
Find:	**Find:** Allows you to 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.

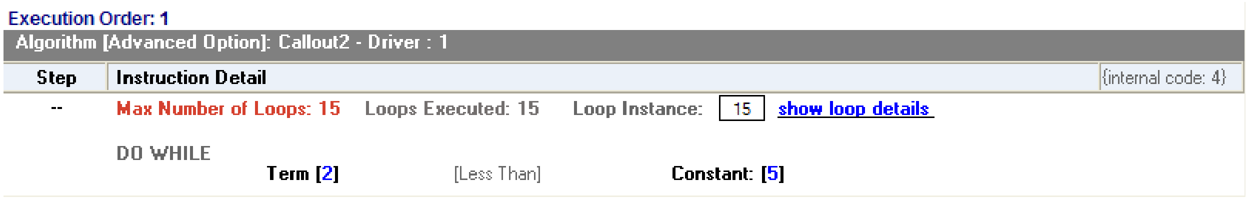
Fields:

Rating Summary: A summary of program information.

Category Items: Elements in the rating listed by category.

Advanced Options

Advanced Algorithm Looping will be displayed on the Debug Report.



Execution Order: 1

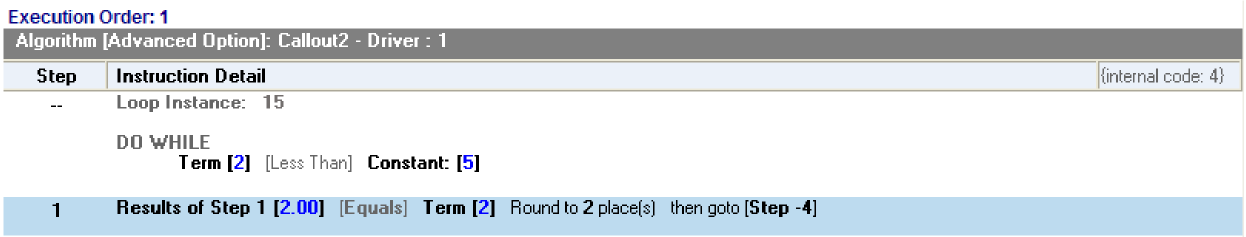
Algorithm [Advanced Option]: Callout2 - Driver : 1

Step	Instruction Detail	{internal code: 4}
--	Max Number of Loops: 15 Loops Executed: 15 Loop Instance: 15 show loop details DO WHILE Term [2] [Less Than] Constant: [5]	

Figure 296 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.



Execution Order: 1

Algorithm [Advanced Option]: Callout2 - Driver : 1

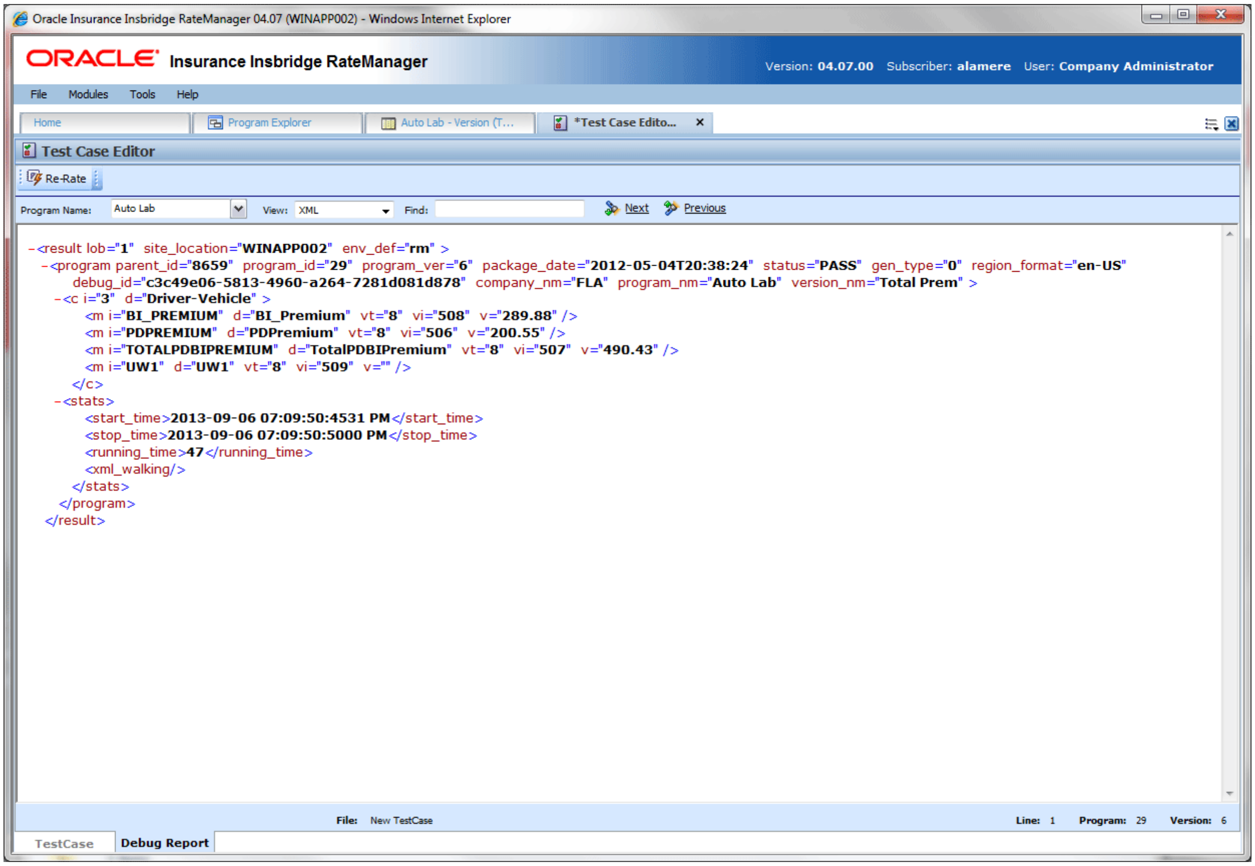
Step	Instruction Detail	{internal code: 4}
--	Loop Instance: 15 DO WHILE Term [2] [Less Than] Constant: [5]	
1	Results of Step 1 [2.00] [Equals] Term [2] Round to 2 place(s) then goto [Step -4]	

Figure 297 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.

Debug Report XML

Debug reports also can be displayed as XML. You will be able to expand and reduce nodes as well as enter values and re-rate.



```
-<result lob="1"  site_location="WINAPP002"  env_def="rm" >
  -<program parent_id="8659"  program_id="29"  program_ver="6"  package_date="2012-05-04T20:38:24"  status="PASS"  gen_type="0"  region_format="en-US"
      debug_id="c3c49e06-5813-4960-a264-7281d081d878"  company_nm="FLA"  program_nm="Auto Lab"  version_nm="Total Prem" >
    -<c i="3"  d="Driver-Vehicle" >
        <m i="BI_PREMIUM"  d="BI_Premium"  vt="8"  vi="508"  v="289.88" />
        <m i="PDPREMIUM"  d="PDPremium"  vt="8"  vi="506"  v="200.55" />
        <m i="TOTALPDBIPREMIUM"  d="TotalPDBIPremium"  vt="8"  vi="507"  v="490.43" />
        <m i="UW1"  d="UW1"  vt="8"  vi="509"  v="" />
      </c>
    -<stats>
        <start_time>2013-09-06 07:09:50:4531 PM</start_time>
        <stop_time>2013-09-06 07:09:50:5000 PM</stop_time>
        <running_time>47</running_time>
        <xml_walking/>
      </stats>
    </program>
  </result>
```

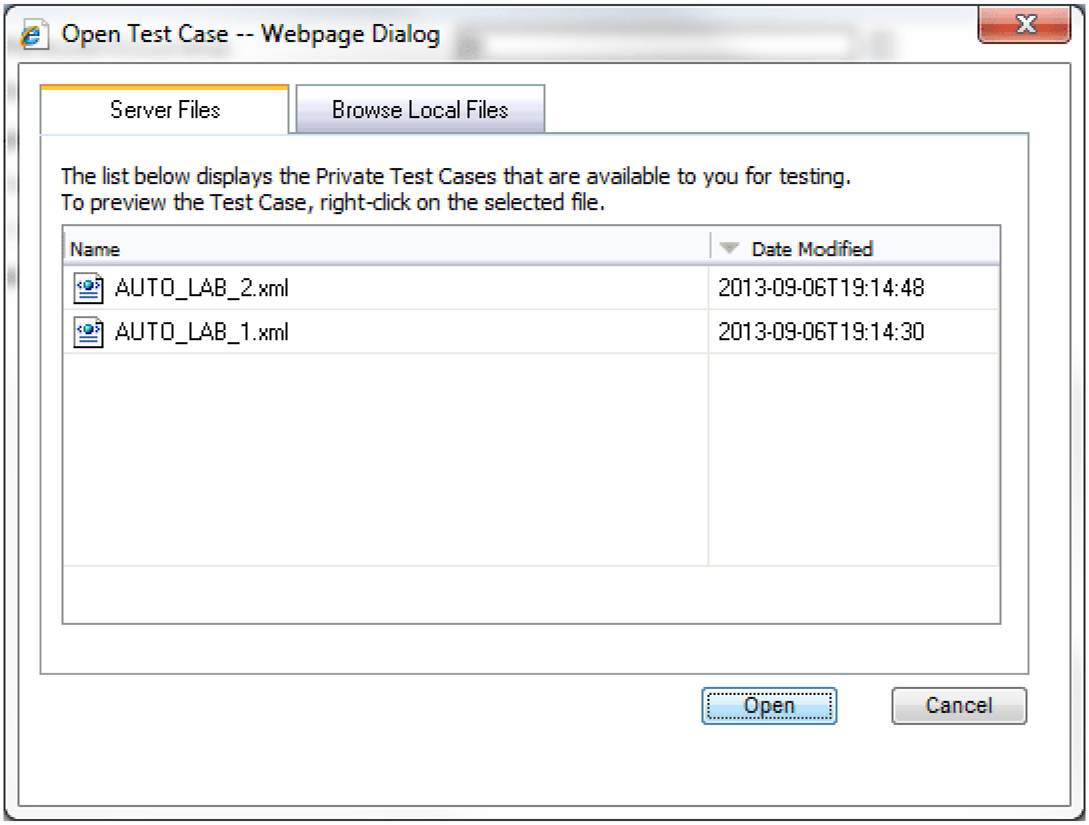
Figure 298 View Debug Reports XML

See Debug a Rating Report for more information on using the debug report.

NOTE: *The date format used by Test Case Editor will be in ISO format YYYY-MM-DD. This may not reflect the date format used by the program.*

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.



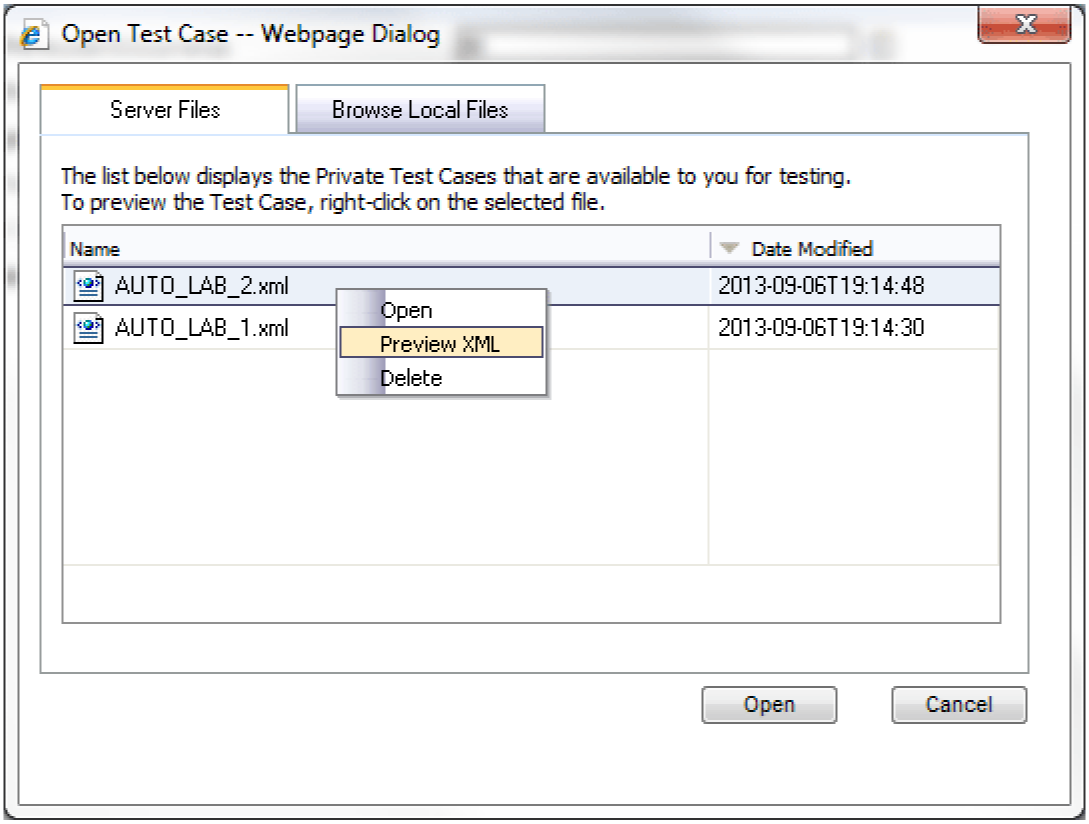
The list below displays the Private Test Cases that are available to you for testing.
To preview the Test Case, right-click on the selected file.

Name	Date Modified
AUTO_LAB_2.xml	2013-09-06T19:14:48
AUTO_LAB_1.xml	2013-09-06T19:14:30

Figure 299 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.



The list below displays the Private Test Cases that are available to you for testing.
To preview the Test Case, right-click on the selected file.

Name	Date Modified
AUTO_LAB_2.xml	2013-09-06T19:14:48
AUTO_LAB_1.xml	2013-09-06T19:14:30

Figure 300 Right Click Menu for Opening Test Cases

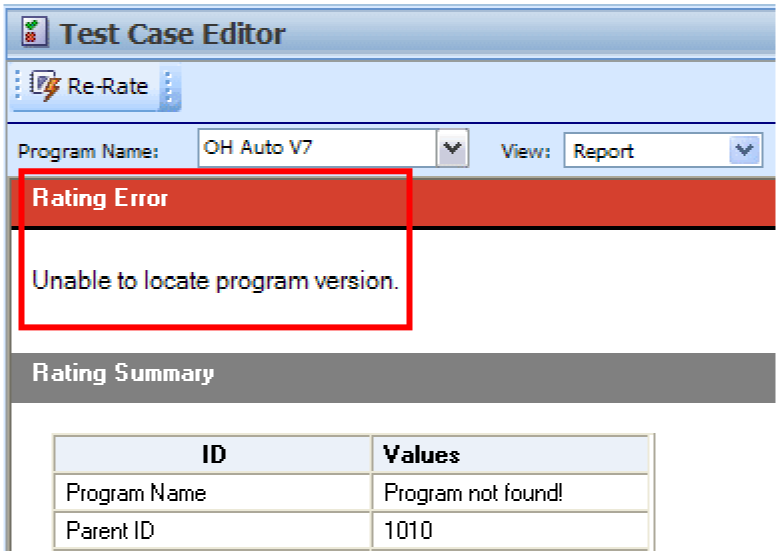
Server files also have a right click menu. There are three options:

- **Open**: Opens the file in Test Program.
- **Preview XML**: Displays the XML in a separate screen. Preview allows you to copy the XML file.
- **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.



Program Name: OH Auto V7 View: Report

Rating Error

Unable to locate program version.

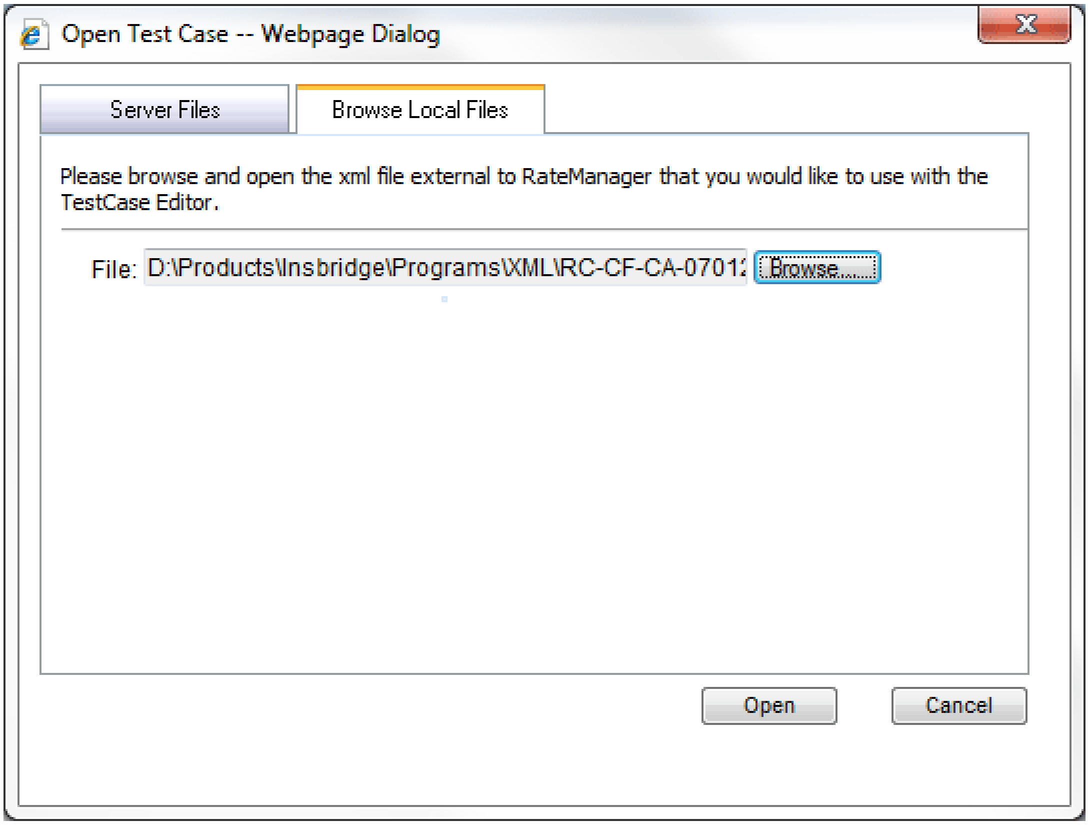
Rating Summary

ID	Values
Program Name	Program not found!
Parent ID	1010

Figure 301 Test Case Error

Browse Local Files

You can browse to a test case XML file saved to your local machine or network.



Please browse and open the xml file external to RateManager that you would like to use with the TestCase Editor.

File: D:\Products\Insbridge\Programs\XML\RC-CF-CA-0701

Figure 302 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 25

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.

Line Input XML and Program Input 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

- **Line Input XML Report** in XML format presents the XML schema consisting of all categories and inputs from all defined programs under the selected line of business.
  - Tool bar option when positioned on a selected program or program version

- **Program Version Input XML Report** in XML format presents the XML schema consisting of all categories and inputs used by the selected programs version.
  - Tool bar option 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

- **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** allows you to see if/where an element is used and in what program version, under a given line of business.
  - 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 must be created in RateManager 3.12 or greater. Lower versions of RateManager do not have this feature.
- 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.
- If the option to select a package is grayed out, you have not created the package in RateManager 3.12 or greater. If you want to compare a package, please exit the report and create either a RateManager or SoftRater package.

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.

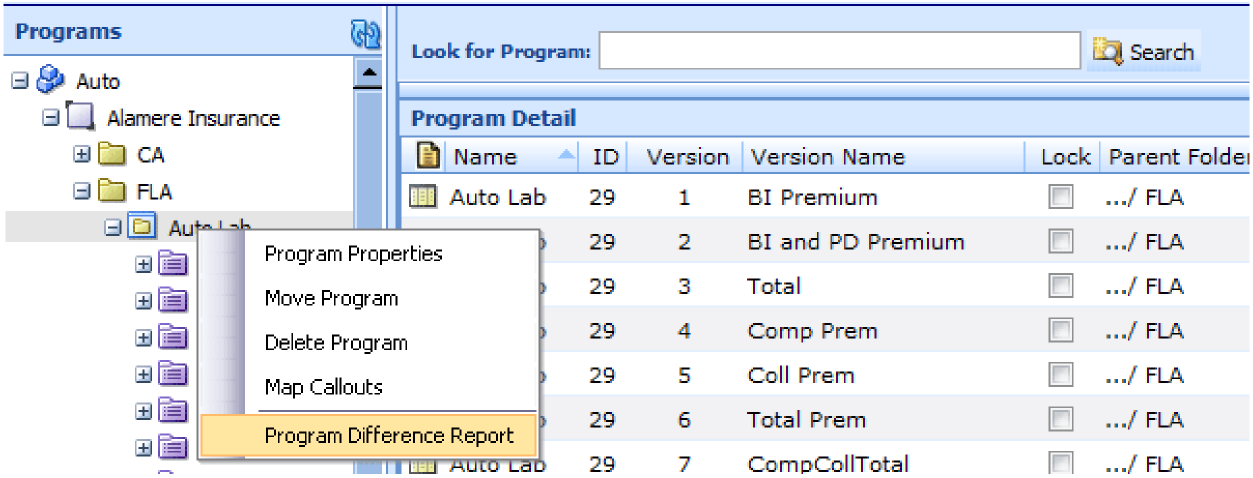
This option is available on all programs in all sublines. To create a Program Difference Report, you must have either two versions, or one version and one package. If you have one version and no package, you will not be able to create a Program Difference Report.

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

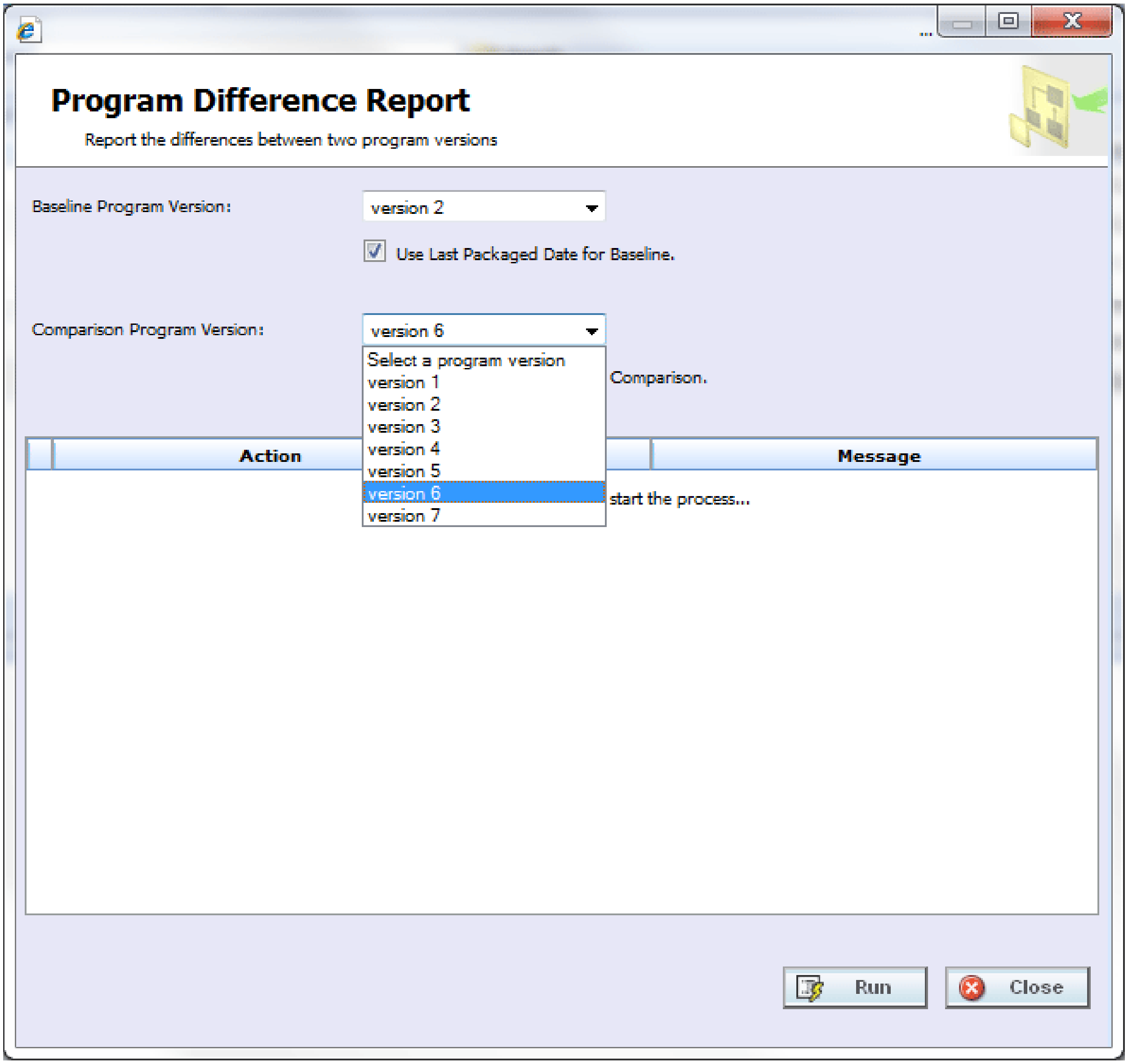
1. From the Program Explorer tab, right click on the program where you want to run a Program Difference Report.



Name	ID	Version	Version Name	Lock	Parent Folde
Auto Lab	29	1	BI Premium	□	.../ FLA
	29	2	BI and PD Premium	□	.../ FLA
	29	3	Total	□	.../ FLA
	29	4	Comp Prem	□	.../ FLA
	29	5	Coll Prem	□	.../ FLA
	29	6	Total Prem	□	.../ FLA
Auto Lab	29	7	CompCollTotal	□	.../ FLA

Figure 303 Program Difference Report Right Click Menu

2. Select **Program Difference Report** from the right click menu. A wizard is displayed.



Program Difference Report

Report the differences between two program versions

Baseline Program Version: version 2

☑ Use Last Packaged Date for Baseline.

Comparison Program Version: version 6

Comparison.

Action	Message
	start the process...

Figure 304 Selecting Programs Versions to Compare

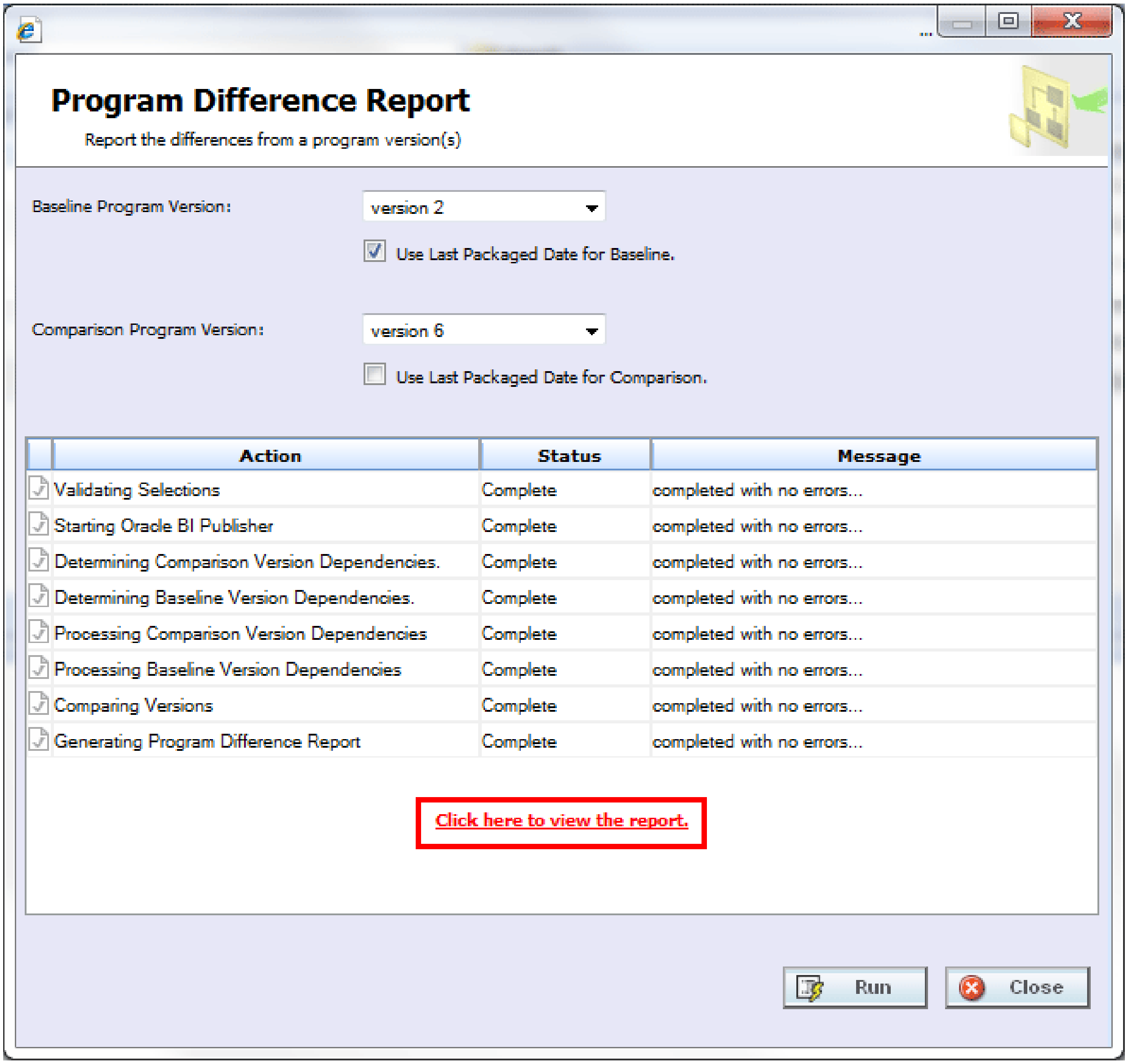
3. Select the **Baseline Program Version**. All versions are listed. This version is the standard that the other version is compared against.

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.



Program Difference Report

Report the differences from a program version(s)

Baseline Program Version: version 2

☑ Use Last Packaged Date for Baseline.

Comparison Program Version: version 6

□ Use Last Packaged Date for Comparison.

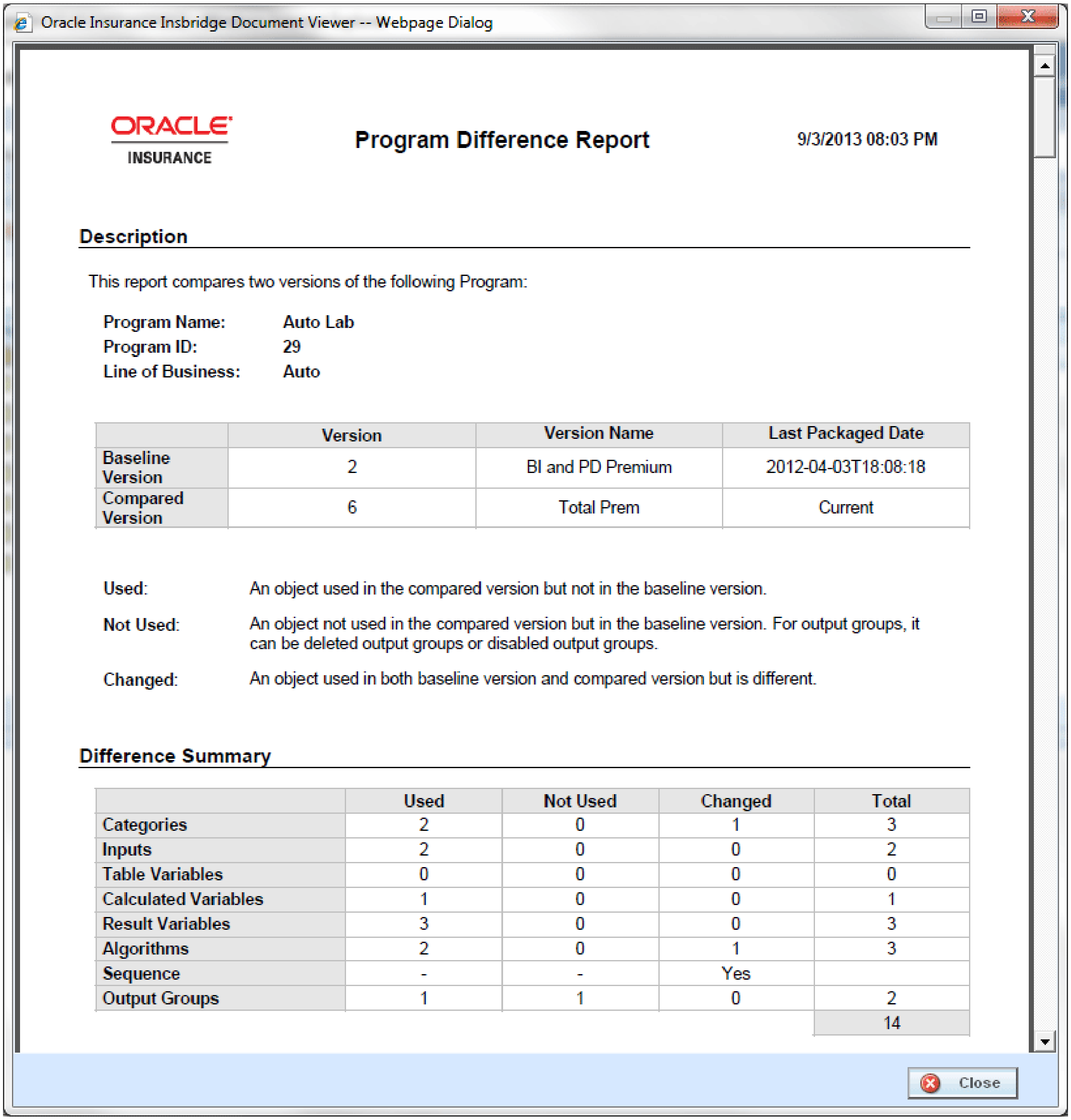
Action	Status	Message
Validating Selections	Complete	completed with no errors...
Starting Oracle BI Publisher	Complete	completed with no errors...
Determining Comparison Version Dependencies.	Complete	completed with no errors...
Determining Baseline Version Dependencies.	Complete	completed with no errors...
Processing Comparison Version Dependencies	Complete	completed with no errors...
Processing Baseline Version Dependencies	Complete	completed with no errors...
Comparing Versions	Complete	completed with no errors...
Generating Program Difference Report	Complete	completed with no errors...

Click here to view the report.

Figure 305 Completed Program Difference Report

8. Click on 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.


Oracle Insurance Insbridge Document Viewer -- Webpage Dialog
ORACLE INSURANCE
Close

Program Difference Report

9/3/2013 08:03 PM

Description

This report compares two versions of the following Program:

Program Name:	Auto Lab
Program ID:	29
Line of Business:	Auto

	Version	Version Name	Last Packaged Date
Baseline Version	2	BI and PD Premium	2012-04-03T18:08:18
Compared Version	6	Total Prem	Current

Used: An object used in the compared version but not in the baseline version.

Not Used: An object not used in the compared version but in the baseline version. For output groups, it can be deleted output groups or disabled output groups.

Changed: An object used in both baseline version and compared version but is different.

Difference Summary

	Used	Not Used	Changed	Total
Categories	2	0	1	3
Inputs	2	0	0	2
Table Variables	0	0	0	0
Calculated Variables	1	0	0	1
Result Variables	3	0	0	3
Algorithms	2	0	1	3
Sequence	-	-	Yes	
Output Groups	1	1	0	2
				14

Figure 306 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 re-name the file.*

Report Details

The report has a description at the top, identifying the program, program ID and LOB. 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:

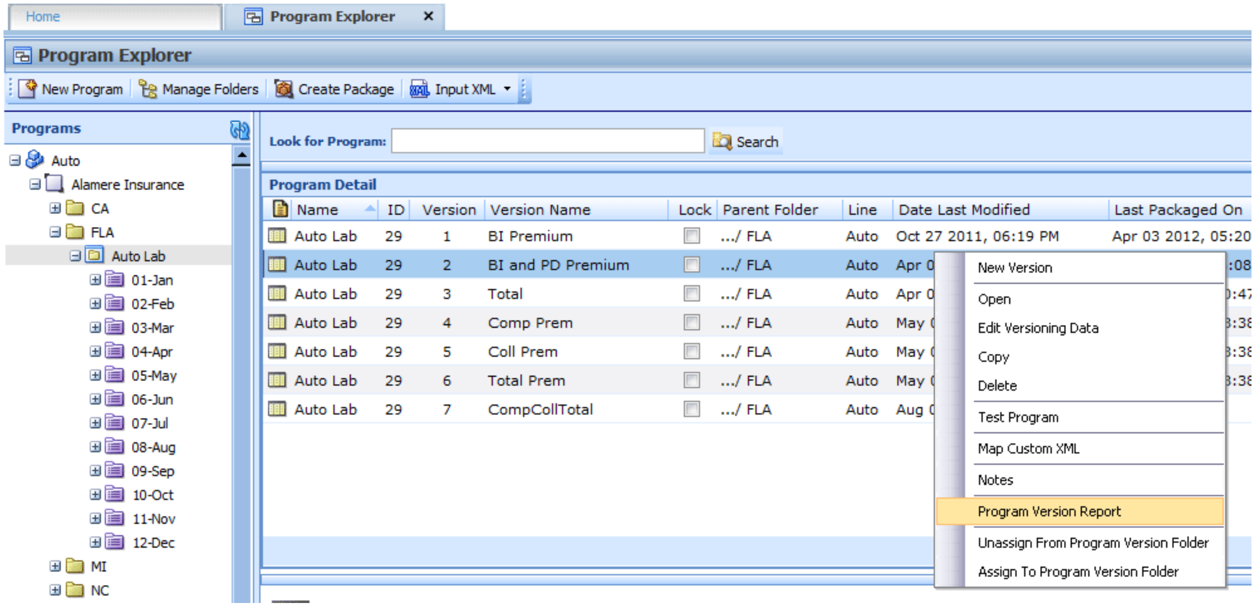
- **Categories:** Parent Category changes.
- **Inputs:** Data Type or Category changes.
- **Table Variables:** Data Type, Category, Default Value, Qualifiers or Data changes. Data changes list the dates of the last time the data was modified. The first date is for the baseline and the second is for the 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 on the program version where you want to run a Program Version Report.

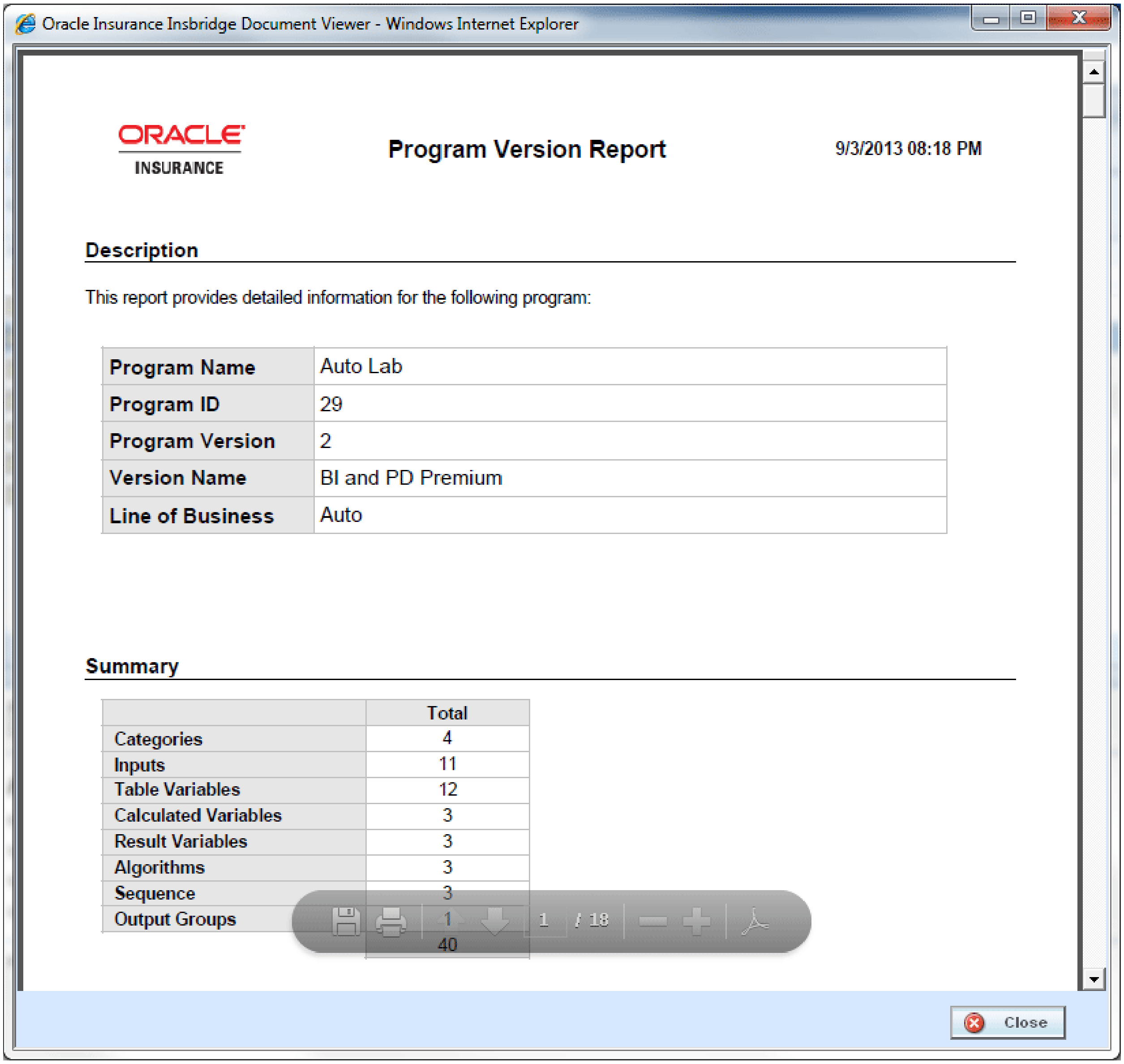


Name	ID	Version	Version Name	Lock	Parent Folder	Line	Date Last Modified	Last Packaged On
Auto Lab	29	1	BI Premium	□	.../ FLA	Auto	Oct 27 2011, 06:19 PM	Apr 03 2012, 05:20
Auto Lab	29	2	BI and PD Premium	□	.../ FLA	Auto	Apr 0	:08
Auto Lab	29	3	Total	□	.../ FLA	Auto	Apr 0	):47
Auto Lab	29	4	Comp Prem	□	.../ FLA	Auto	May (	3:38
Auto Lab	29	5	Coll Prem	□	.../ FLA	Auto	May (	3:38
Auto Lab	29	6	Total Prem	□	.../ FLA	Auto	May (	3:38
Auto Lab	29	7	CompCollTotal	□	.../ FLA	Auto	Aug (	

Figure 307 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.



Oracle Insurance Insbridge Document Viewer - Windows Internet Explorer
ORACLE INSURANCE
1 / 18
Close

Program Version Report

9/3/2013 08:18 PM

Description

This report provides detailed information for the following program:

Program Name	Auto Lab
Program ID	29
Program Version	2
Version Name	BI and PD Premium
Line of Business	Auto

Summary

	Total
Categories	4
Inputs	11
Table Variables	12
Calculated Variables	3
Result Variables	3
Algorithms	3
Sequence	3
Output Groups	1
	40

Figure 308 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 re-name the file.*

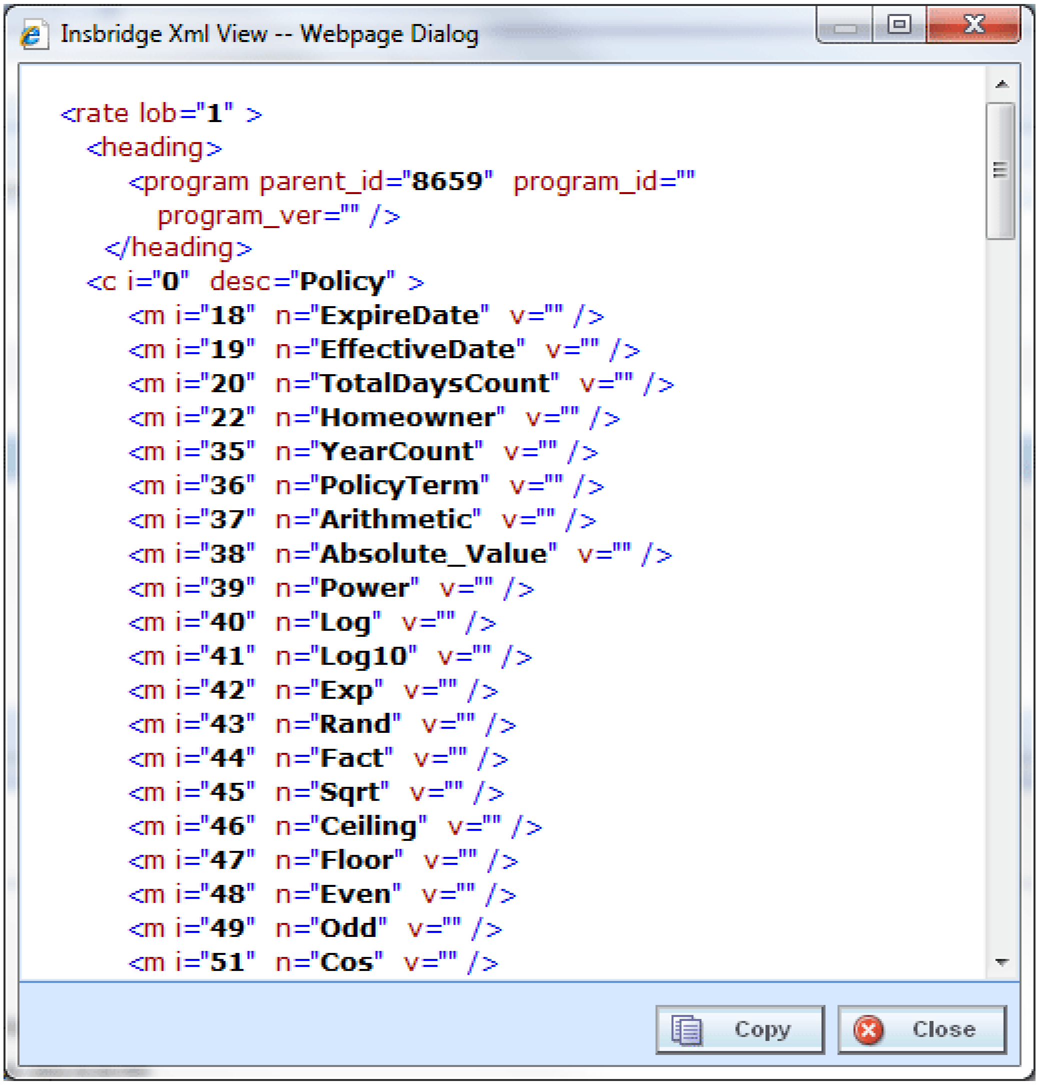
NOTE: *Report controls are displayed when the cursor is positioned in the lower portion of the report.*

Report Details

The report has a description at the top, identifying the program, program ID, program version and LOB. 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

LINE INPUTS XML

Selecting the **Inputs XML→Line** menu option displays the line inputs XML for this subline. You can view or copy the line inputs XML. This feature can be accessed from the Program Explorer tab.



```
<rate lob="1" >
  <heading>
     <program parent_id="8659"  program_id=""
       program_ver="" />
   </heading>
  <c i="0"  desc="Policy" >
     <m i="18"  n="ExpireDate"  v="" />
     <m i="19"  n="EffectiveDate"  v="" />
     <m i="20"  n="TotalDaysCount"  v="" />
     <m i="22"  n="Homeowner"  v="" />
     <m i="35"  n="YearCount"  v="" />
     <m i="36"  n="PolicyTerm"  v="" />
     <m i="37"  n="Arithmetic"  v="" />
     <m i="38"  n="Absolute_Value"  v="" />
     <m i="39"  n="Power"  v="" />
     <m i="40"  n="Log"  v="" />
     <m i="41"  n="Log10"  v="" />
     <m i="42"  n="Exp"  v="" />
     <m i="43"  n="Rand"  v="" />
     <m i="44"  n="Fact"  v="" />
     <m i="45"  n="Sqrt"  v="" />
     <m i="46"  n="Ceiling"  v="" />
     <m i="47"  n="Floor"  v="" />
     <m i="48"  n="Even"  v="" />
     <m i="49"  n="Odd"  v="" />
     <m i="51"  n="Cos"  v="" />
```

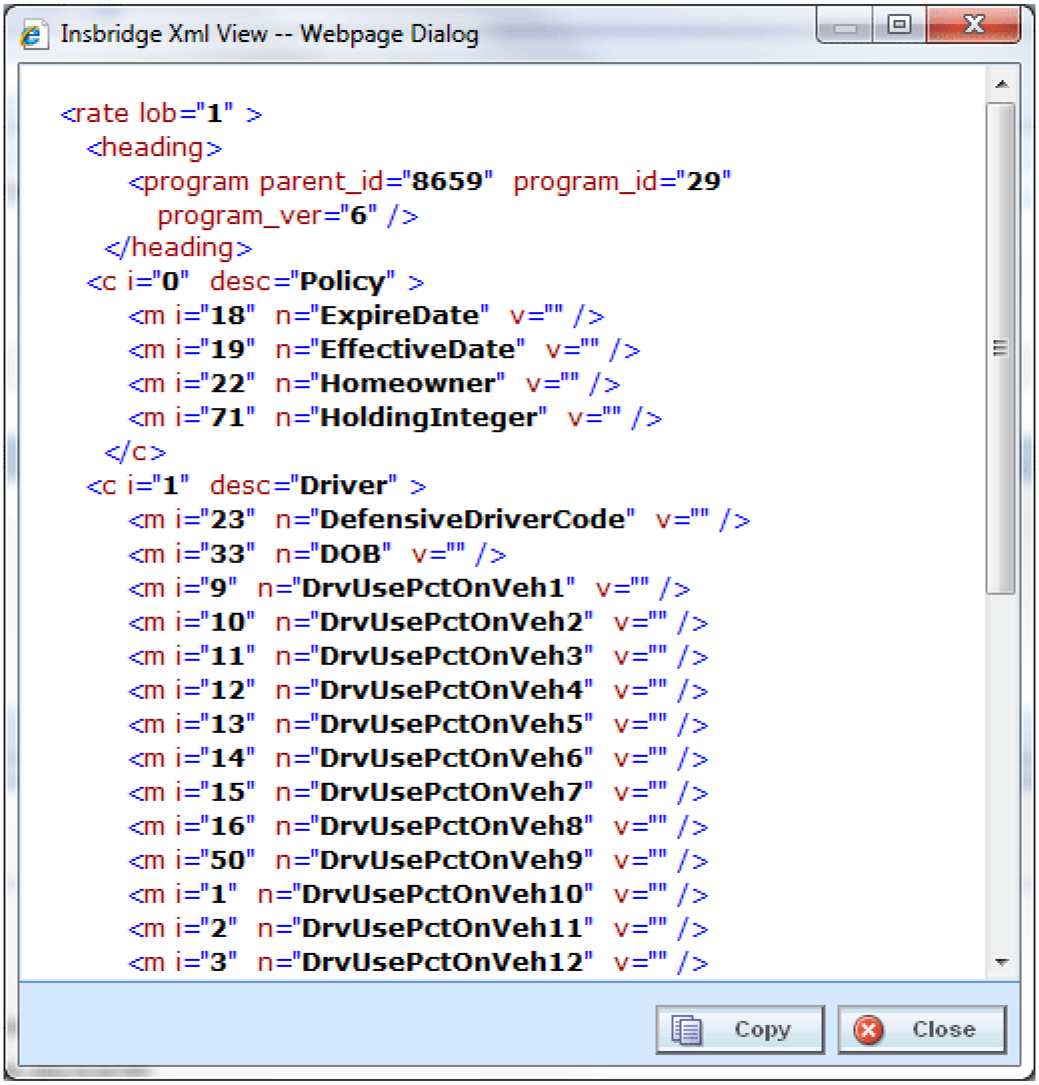
Figure 309 Line Inputs XML

To View Line Inputs XML

1. From the Program Explorer tab, select a subline.
2. Select **Inputs XML→Line** from the top bar menu.
3. A separate window opens where you can view or copy XML for this subline.
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.



```
<rate lob="1" >
  <heading>
     <program parent_id="8659"  program_id="29"
       program_ver="6" />
   </heading>
  <c i="0"  desc="Policy" >
     <m i="18"  n="ExpireDate"  v="" />
     <m i="19"  n="EffectiveDate"  v="" />
     <m i="22"  n="Homeowner"  v="" />
     <m i="71"  n="HoldingInteger"  v="" />
   </c>
  <c i="1"  desc="Driver" >
     <m i="23"  n="DefensiveDriverCode"  v="" />
     <m i="33"  n="DOB"  v="" />
     <m i="9"  n="DrvUsePctOnVeh1"  v="" />
     <m i="10"  n="DrvUsePctOnVeh2"  v="" />
     <m i="11"  n="DrvUsePctOnVeh3"  v="" />
     <m i="12"  n="DrvUsePctOnVeh4"  v="" />
     <m i="13"  n="DrvUsePctOnVeh5"  v="" />
     <m i="14"  n="DrvUsePctOnVeh6"  v="" />
     <m i="15"  n="DrvUsePctOnVeh7"  v="" />
     <m i="16"  n="DrvUsePctOnVeh8"  v="" />
     <m i="50"  n="DrvUsePctOnVeh9"  v="" />
     <m i="1"  n="DrvUsePctOnVeh10"  v="" />
     <m i="2"  n="DrvUsePctOnVeh11"  v="" />
     <m i="3"  n="DrvUsePctOnVeh12"  v="" />
```

Figure 310 Program Inputs XML

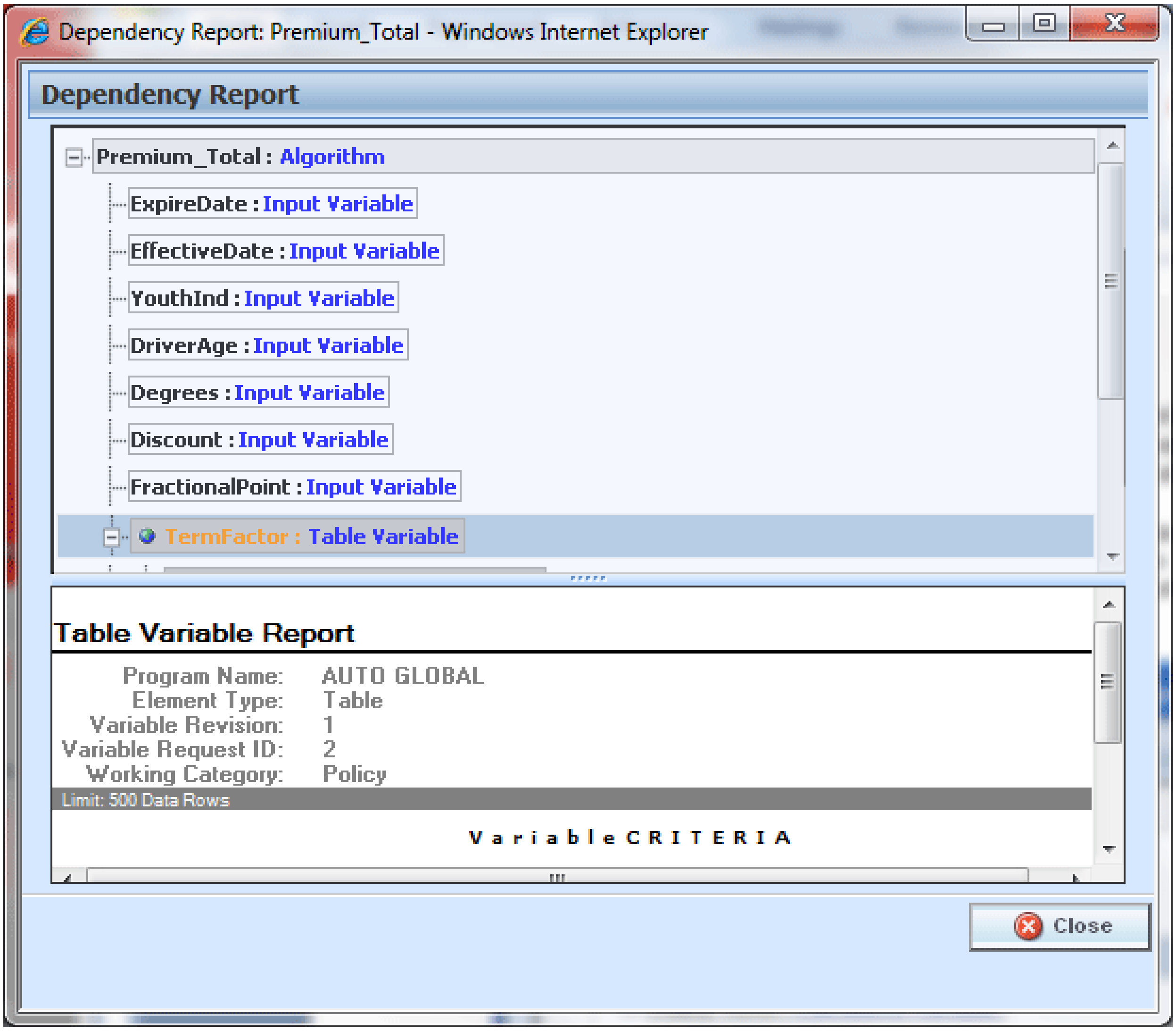
Viewing Program Inputs XML

1. From the **Program Explorer** tab, select the Line of Business→Subline→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.


Dependency Report: Premium_Total - Windows Internet Explorer
Dependency Report
Close

Premium_Total : Algorithm
ExpireDate : Input Variable
EffectiveDate : Input Variable
YouthInd : Input Variable
DriverAge : Input Variable
Degrees : Input Variable
Discount : Input Variable
FractionalPoint : Input Variable
TermFactor : Table Variable

Table Variable Report

Program Name: AUTO GLOBAL
Element Type: Table
Variable Revision: 1
Variable Request ID: 2
Working Category: Policy

Limit: 500 Data Rows

VariableCRITERIA

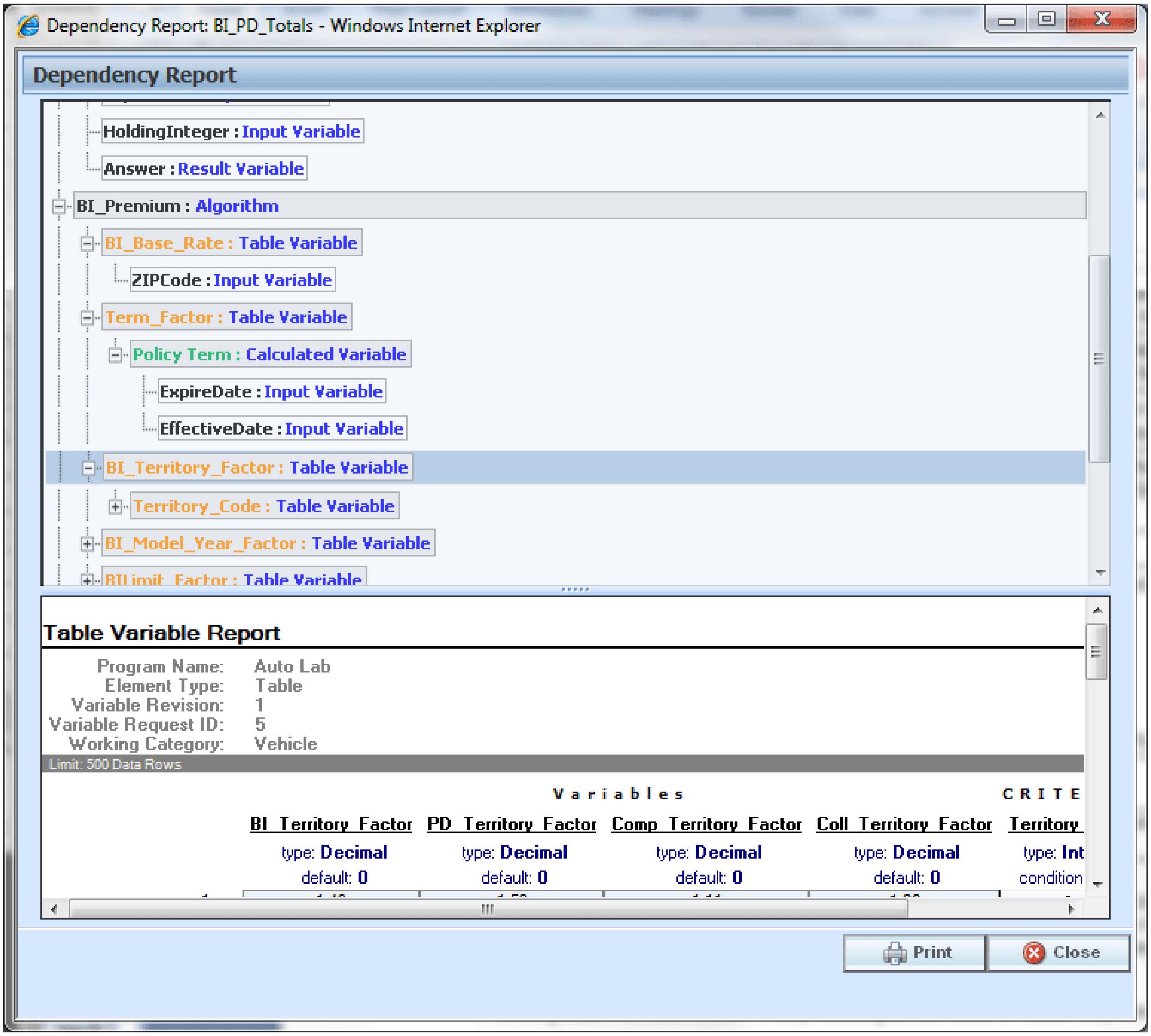
Figure 311 Dependency Report

- Yellow text indicates a algorithm.
- Green text indicates a result or input 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 only. Click the Full Dependency button in the top bar menu.



Dependency Report

HoldingInteger : Input Variable
Answer : Result Variable
BI_Premium : Algorithm
BI_Base_Rate : Table Variable
ZIPCode : Input Variable
Term_Factor : Table Variable
Policy Term : Calculated Variable
ExpireDate : Input Variable
EffectiveDate : Input Variable
BI_Territory_Factor : Table Variable
Territory_Code : Table Variable
BI_Model_Year_Factor : Table Variable
BILimit_Factor : Table Variable

Table Variable Report

Program Name:	Auto Lab
Element Type:	Table
Variable Revision:	1
Variable Request ID:	5
Working Category:	Vehicle

Limit: 500 Data Rows

Variables				CRITE
BI Territory Factor	PD Territory Factor	Comp Territory Factor	Coll Territory Factor	Territory
type: Decimal	type: Decimal	type: Decimal	type: Decimal	type: Int
default: 0	default: 0	default: 0	default: 0	condition

Figure 312 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.

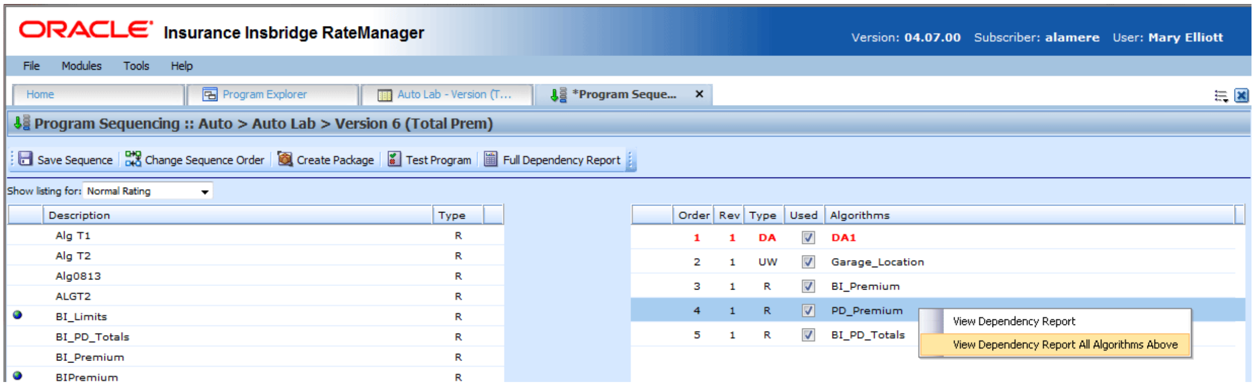
Variables are listed in sequence order. When you select to view an algorithm, all the inputs, results, table, and calculated variables used in the algorithm will be listed. Inputs are listed first, with tables and calculated variables next, and result variables last. Inputs are listed in XML ID order.

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.



Version: 04.07.00 Subscriber: alamere User: Mary Elliott

Program Sequencing :: Auto > Auto Lab > Version 6 (Total Prem)

Show listing for: Normal Rating

	Description	Type
	Alg T1	R
	Alg T2	R
	Alg0813	R
	ALGT2	R
●	BI_Limits	R
	BI_PD_Totals	R
	BI_Premium	R
●	BIPremium	R

Order	Rev	Type	Used	Algorithms
1	1	DA	☑	DA1
2	1	UW	☑	Garage_Location
3	1	R	☑	BI_Premium
4	1	R	☑	PD_Premium
5	1	R	☑	BI_PD_Totals

Figure 313 View Dependency Report All Algorithm Above

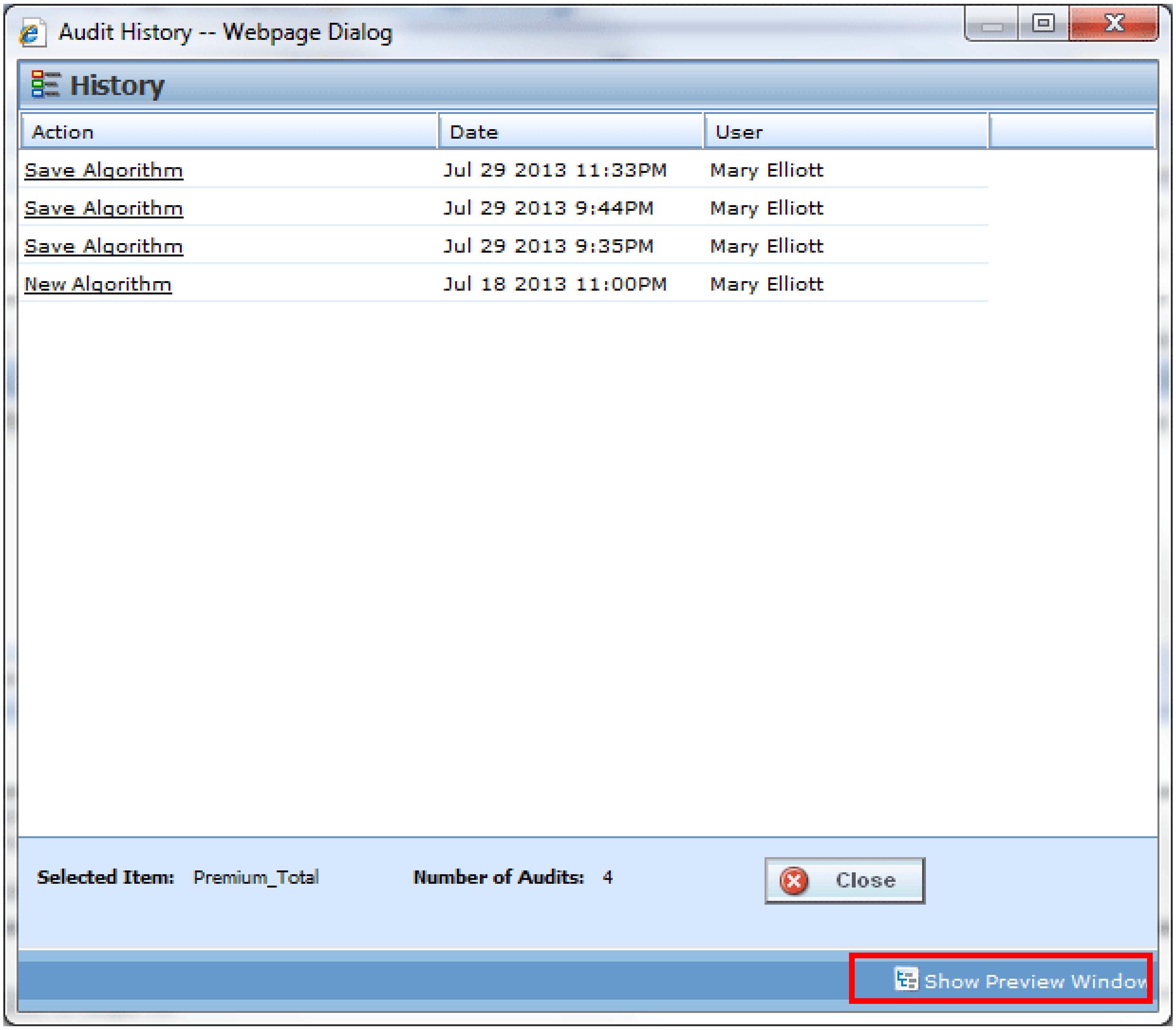
Variables are listed in sequence order. When you select to view an algorithm, all the inputs, results, table, and calculated variables used in the algorithm will be listed. Inputs are listed first, with tables and calculated variables next, and result variables last. Inputs are listed in XML ID order.

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 hand corner of the screen and then click on an action you want to view. The details will be displayed in the lower portion of the screen.



Action	Date	User
Save Algorithm	Jul 29 2013 11:33PM	Mary Elliott
Save Algorithm	Jul 29 2013 9:44PM	Mary Elliott
Save Algorithm	Jul 29 2013 9:35PM	Mary Elliott
New Algorithm	Jul 18 2013 11:00PM	Mary Elliott

Selected Item: Premium_Total

Number of Audits: 4

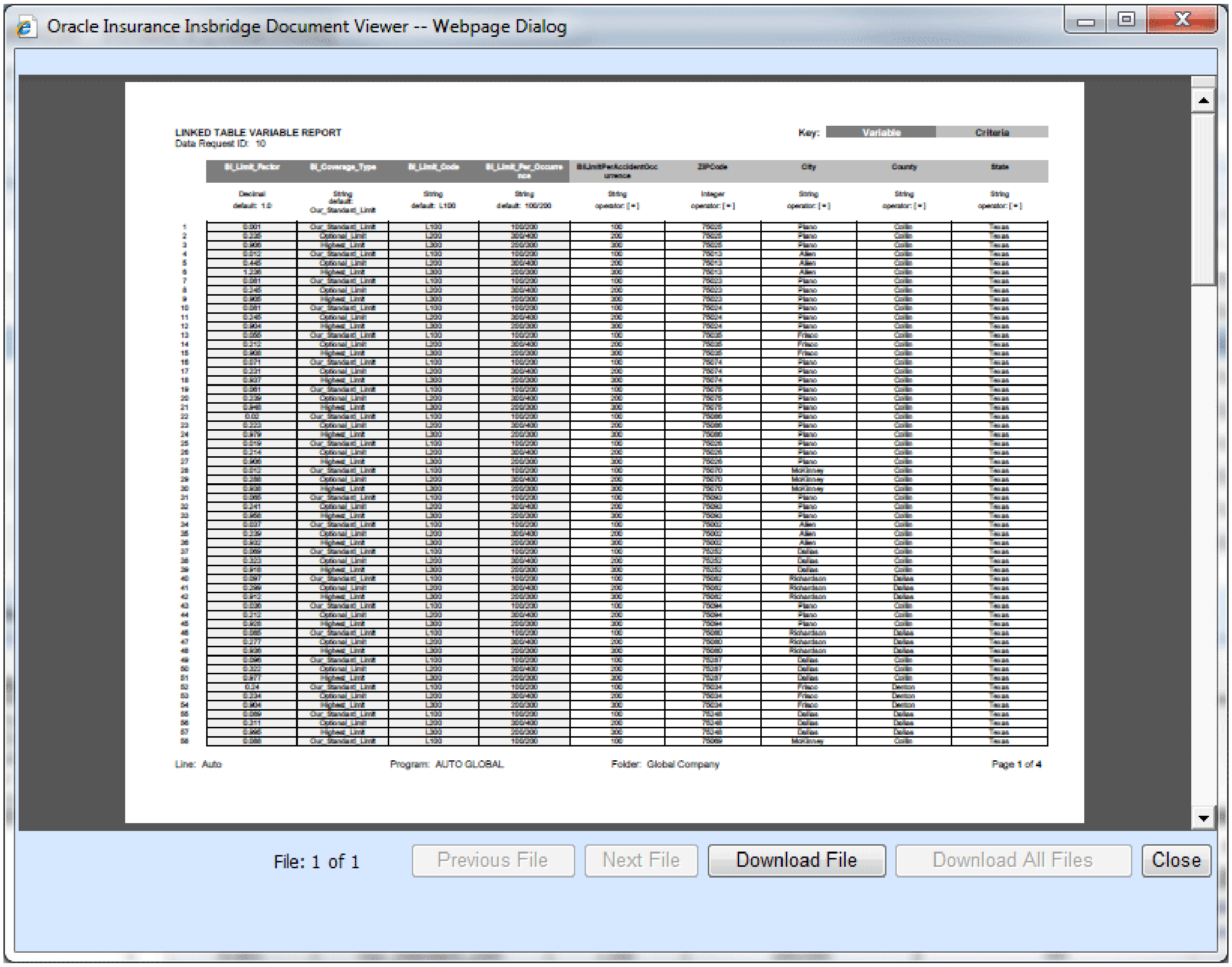
Figure 314 View History

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.


Oracle Insurance Insbridge Document Viewer -- Webpage Dialog

LINKED TABLE VARIABLE REPORT
Data Request ID: 10

Key: Variable Criteria

	BI_Limit_Factor	BI_Coverage_Type	BI_Limit_Code	BI_Limit_Per_Occurrence	BILimitPerAccidentOccurrence	ZIPCode	City	County	State
	Decimal default: 1.0	String default: Our_Standard_Limit	String default: L100	String default: 100/200	String operator: [=]	Integer operator: [=]	String operator: [=]	String operator: [=]	String operator: [=]
1	0.001	Our_Standard_Limit	L100	100/200	100	75025	Plano	Collin	Texas
2	0.235	Optional_Limit	L200	300/400	200	75025	Plano	Collin	Texas
3	0.906	Highest_Limit	L300	200/300	300	75025	Plano	Collin	Texas
4	0.012	Our_Standard_Limit	L100	100/200	100	75013	Allen	Collin	Texas
5	0.445	Optional_Limit	L200	300/400	200	75013	Allen	Collin	Texas
6	1.236	Highest_Limit	L300	200/300	300	75013	Allen	Collin	Texas
7	0.081	Our_Standard_Limit	L100	100/200	100	75023	Plano	Collin	Texas
8	0.345	Optional_Limit	L200	300/400	200	75023	Plano	Collin	Texas
9	0.905	Highest_Limit	L300	200/300	300	75023	Plano	Collin	Texas
10	0.081	Our_Standard_Limit	L100	100/200	100	75024	Plano	Collin	Texas
11	0.345	Optional_Limit	L200	300/400	200	75024	Plano	Collin	Texas
12	0.904	Highest_Limit	L300	200/300	300	75024	Plano	Collin	Texas
13	0.065	Our_Standard_Limit	L100	100/200	100	75035	Frisco	Collin	Texas
14	0.212	Optional_Limit	L200	300/400	200	75035	Frisco	Collin	Texas
15	0.908	Highest_Limit	L300	200/300	300	75035	Frisco	Collin	Texas
16	0.071	Our_Standard_Limit	L100	100/200	100	75074	Plano	Collin	Texas
17	0.221	Optional_Limit	L200	300/400	200	75074	Plano	Collin	Texas
18	0.927	Highest_Limit	L300	200/300	300	75074	Plano	Collin	Texas
19	0.061	Our_Standard_Limit	L100	100/200	100	75075	Plano	Collin	Texas
20	0.239	Optional_Limit	L200	300/400	200	75075	Plano	Collin	Texas
21	0.948	Highest_Limit	L300	200/300	300	75075	Plano	Collin	Texas
22	0.02	Our_Standard_Limit	L100	100/200	100	75086	Plano	Collin	Texas
23	0.223	Optional_Limit	L200	300/400	200	75086	Plano	Collin	Texas
24	0.979	Highest_Limit	L300	200/300	300	75086	Plano	Collin	Texas
25	0.019	Our_Standard_Limit	L100	100/200	100	75026	Plano	Collin	Texas
26	0.214	Optional_Limit	L200	300/400	200	75026	Plano	Collin	Texas
27	0.906	Highest_Limit	L300	200/300	300	75026	Plano	Collin	Texas
28	0.012	Our_Standard_Limit	L100	100/200	100	75070	McKinney	Collin	Texas
29	0.288	Optional_Limit	L200	300/400	200	75070	McKinney	Collin	Texas
30	0.938	Highest_Limit	L300	200/300	300	75070	McKinney	Collin	Texas
31	0.065	Our_Standard_Limit	L100	100/200	100	75093	Plano	Collin	Texas
32	0.241	Optional_Limit	L200	300/400	200	75093	Plano	Collin	Texas
33	0.958	Highest_Limit	L300	200/300	300	75093	Plano	Collin	Texas
34	0.037	Our_Standard_Limit	L100	100/200	100	75002	Allen	Collin	Texas
35	0.239	Optional_Limit	L200	300/400	200	75002	Allen	Collin	Texas
36	0.932	Highest_Limit	L300	200/300	300	75002	Allen	Collin	Texas
37	0.069	Our_Standard_Limit	L100	100/200	100	75252	Dallas	Collin	Texas
38	0.323	Optional_Limit	L200	300/400	200	75252	Dallas	Collin	Texas
39	0.918	Highest_Limit	L300	200/300	300	75252	Dallas	Collin	Texas
40	0.097	Our_Standard_Limit	L100	100/200	100	75082	Richardson	Dallas	Texas
41	0.299	Optional_Limit	L200	300/400	200	75082	Richardson	Dallas	Texas
42	0.912	Highest_Limit	L300	200/300	300	75082	Richardson	Dallas	Texas
43	0.036	Our_Standard_Limit	L100	100/200	100	75094	Plano	Collin	Texas
44	0.212	Optional_Limit	L200	300/400	200	75094	Plano	Collin	Texas
45	0.928	Highest_Limit	L300	200/300	300	75094	Plano	Collin	Texas
46	0.085	Our_Standard_Limit	L100	100/200	100	75080	Richardson	Dallas	Texas
47	0.277	Optional_Limit	L200	300/400	200	75080	Richardson	Dallas	Texas
48	0.936	Highest_Limit	L300	200/300	300	75080	Richardson	Dallas	Texas
49	0.096	Our_Standard_Limit	L100	100/200	100	75287	Dallas	Collin	Texas
50	0.322	Optional_Limit	L200	300/400	200	75287	Dallas	Collin	Texas
51	0.977	Highest_Limit	L300	200/300	300	75287	Dallas	Collin	Texas
52	0.24	Our_Standard_Limit	L100	100/200	100	75034	Frisco	Denton	Texas
53	0.234	Optional_Limit	L200	300/400	200	75034	Frisco	Denton	Texas
54	0.904	Highest_Limit	L300	200/300	300	75034	Frisco	Denton	Texas
55	0.069	Our_Standard_Limit	L100	100/200	100	75248	Dallas	Dallas	Texas
56	0.211	Optional_Limit	L200	300/400	200	75248	Dallas	Dallas	Texas
57	0.966	Highest_Limit	L300	200/300	300	75248	Dallas	Dallas	Texas
58	0.088	Our_Standard_Limit	L100	100/200	100	75069	McKinney	Collin	Texas

Line: Auto Program: AUTO GLOBAL Folder: Global Company Page 1 of 4

File: 1 of 1

Figure 315 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.

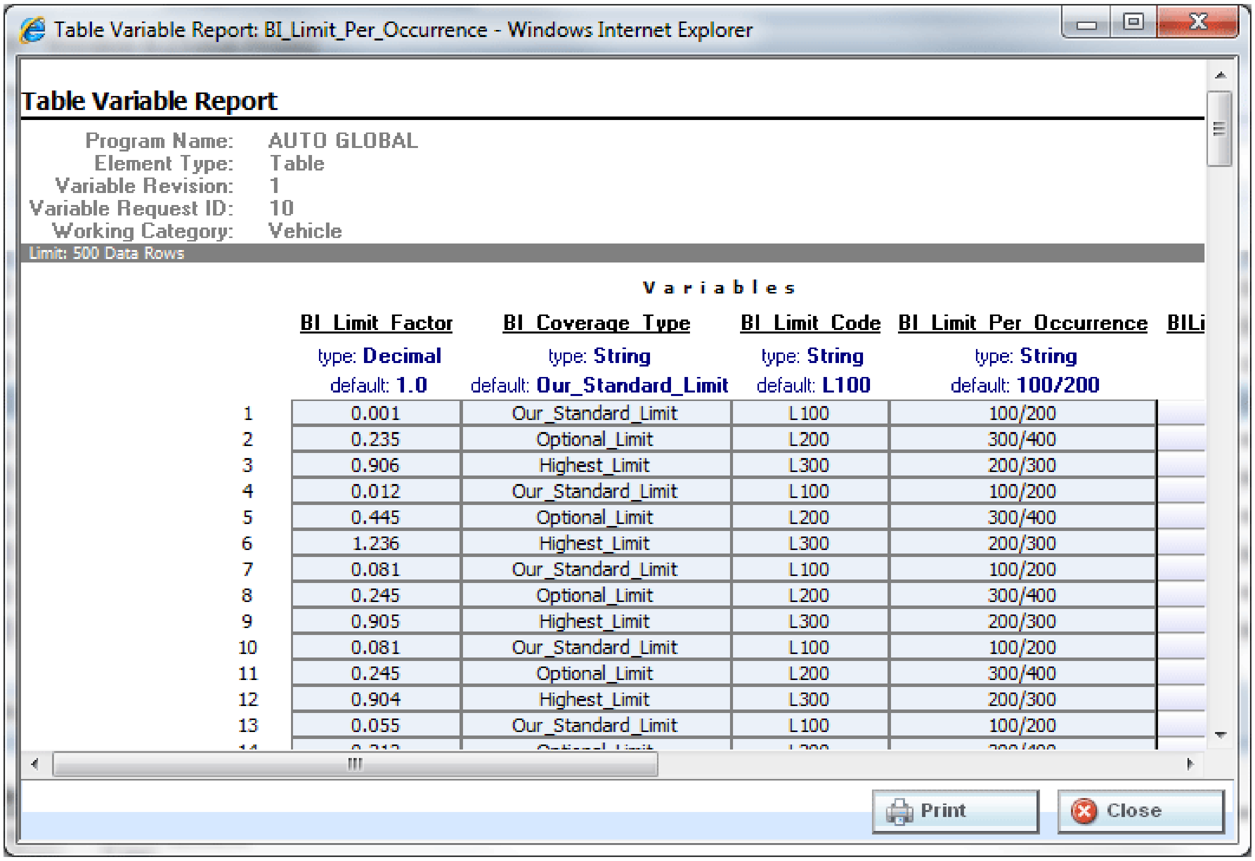


Table Variable Report: BI_Limit_Per_Occurrence - Windows Internet Explorer
Print
Close

Table Variable Report

Program Name: AUTO GLOBAL
Element Type: Table
Variable Revision: 1
Variable Request ID: 10
Working Category: Vehicle

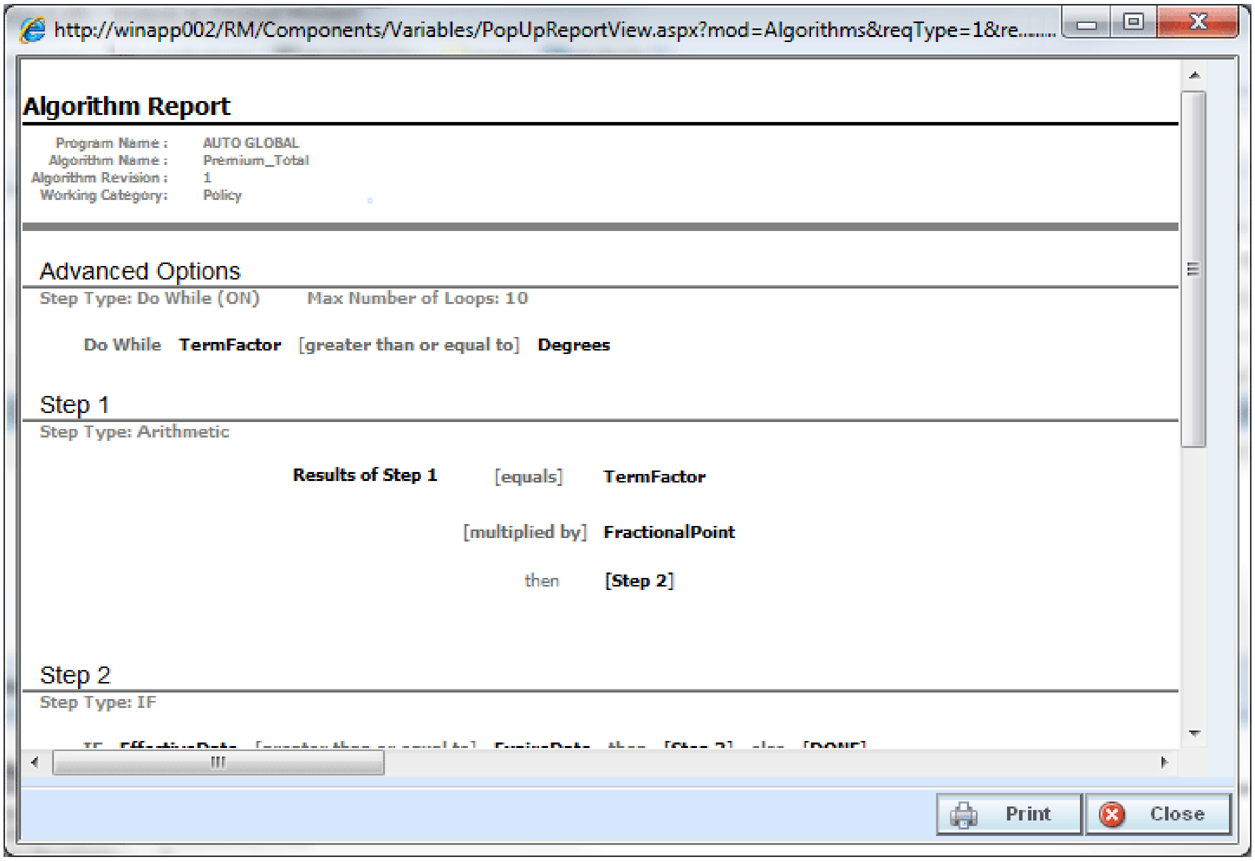
Limit: 500 Data Rows

Variables

	BI_Limit_Factor	BI_Coverage_Type	BI_Limit_Code	BI_Limit_Per_Occurrence	BILi
	type: Decimal	type: String	type: String	type: String	
	default: 1.0	default: Our_Standard_Limit	default: L100	default: 100/200	
1	0.001	Our_Standard_Limit	L100	100/200	
2	0.235	Optional_Limit	L200	300/400	
3	0.906	Highest_Limit	L300	200/300	
4	0.012	Our_Standard_Limit	L100	100/200	
5	0.445	Optional_Limit	L200	300/400	
6	1.236	Highest_Limit	L300	200/300	
7	0.081	Our_Standard_Limit	L100	100/200	
8	0.245	Optional_Limit	L200	300/400	
9	0.905	Highest_Limit	L300	200/300	
10	0.081	Our_Standard_Limit	L100	100/200	
11	0.245	Optional_Limit	L200	300/400	
12	0.904	Highest_Limit	L300	200/300	
13	0.055	Our_Standard_Limit	L100	100/200	

Figure 316 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.



http://winapp002/RM/Components/Variables/PopUpReportView.aspx?mod=Algorithms&reqType=1&re.......

Algorithm Report

Program Name : AUTO GLOBAL
Algorithm Name : Premium_Total
Algorithm Revision : 1
Working Category: Policy

Advanced Options

Step Type: Do While (ON)   Max Number of Loops: 10

Do While **TermFactor** [greater than or equal to] **Degrees**

Step 1

Step Type: Arithmetic

Results of Step 1 [equals] **TermFactor**
[multiplied by] **FractionalPoint**
then **[Step 2]**

Step 2

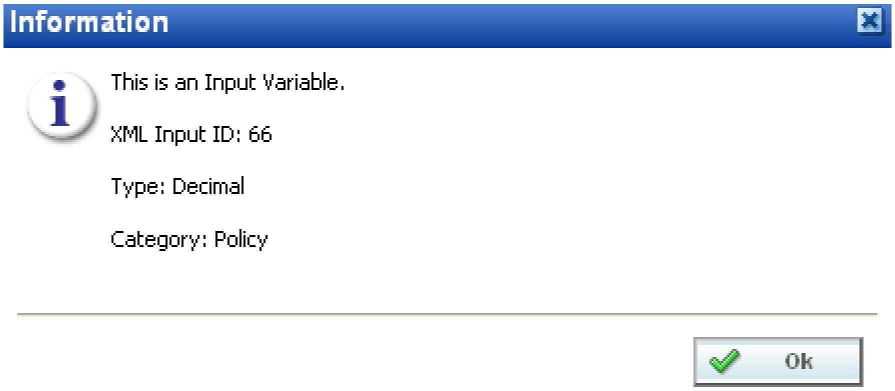
Step Type: IF

IF **EffectiveDate** [greater than or equal to] **ExpireDate** then **[Step 3]** else **[DONE]**

Figure 317 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 popup will be displayed with the information you need, including Type and Category, allowing you to stay in Edit mode.



This is an Input Variable.

XML Input ID: 66

Type: Decimal

Category: Policy

Figure 318 View Report

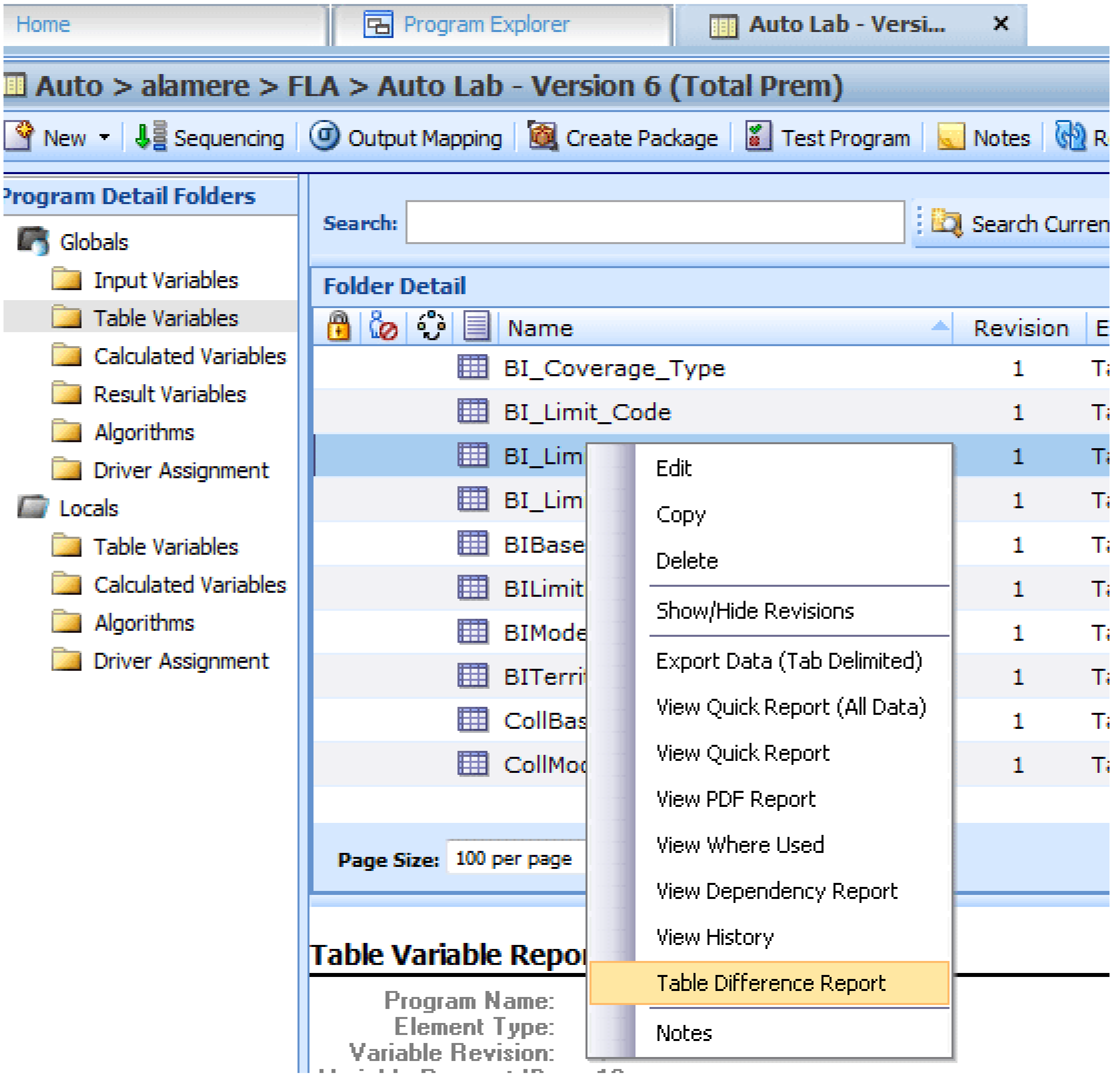
TABLE DIFFERENCE REPORT

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.



Name	Revision	E
BI_Coverage_Type	1	T
BI_Limit_Code	1	T
BI_Lim	1	T
BI_Lim	1	T
BIBase	1	T
BILimit	1	T
BIMode	1	T
BITerri	1	T
CollBas	1	T
CollMo	1	T

Figure 319 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.

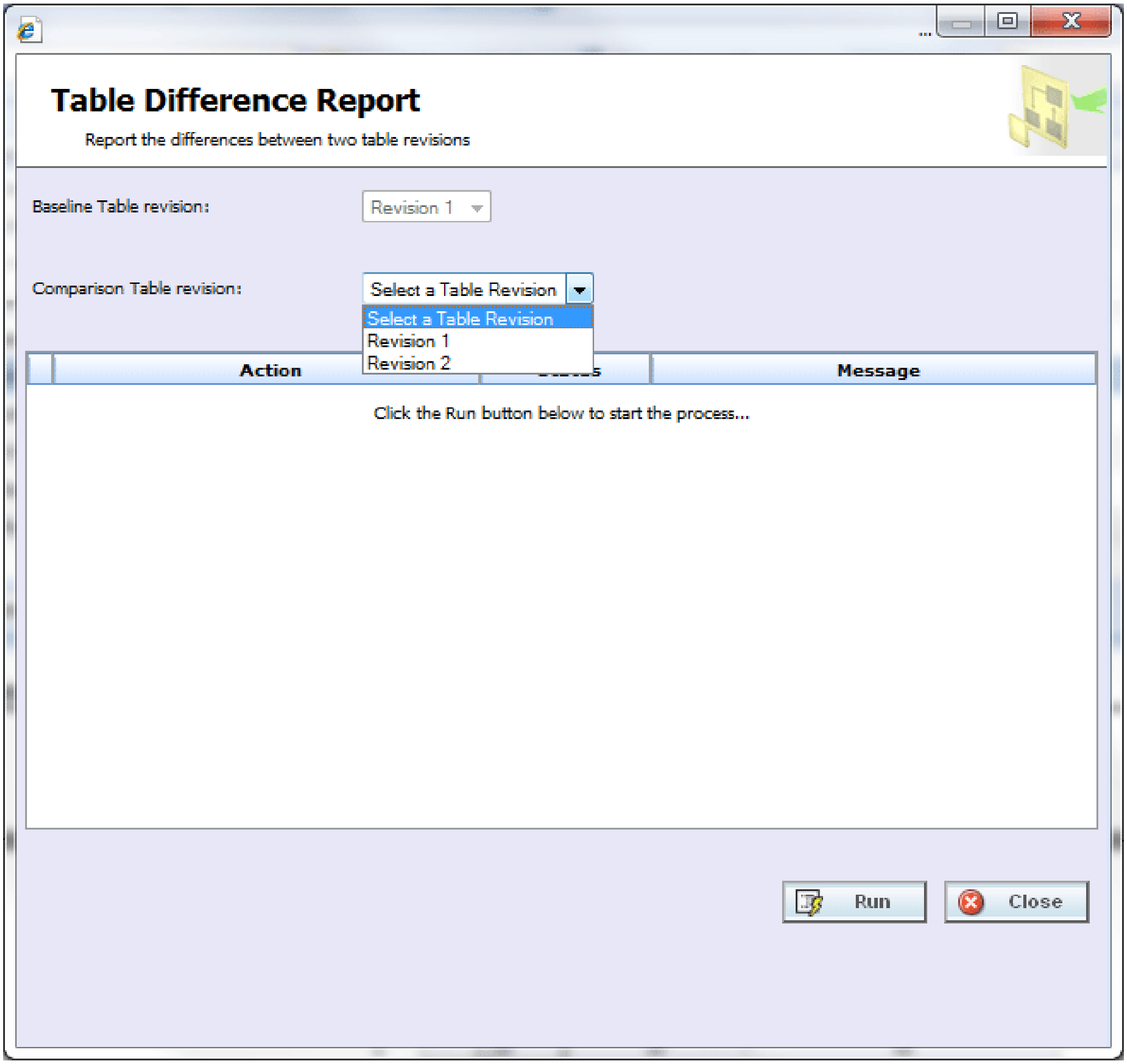


Table Difference Report

Report the differences between two table revisions

Baseline Table revision: Revision 1

Comparison Table revision: Select a Table Revision

Click the Run button below to start the process...

Figure 320 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

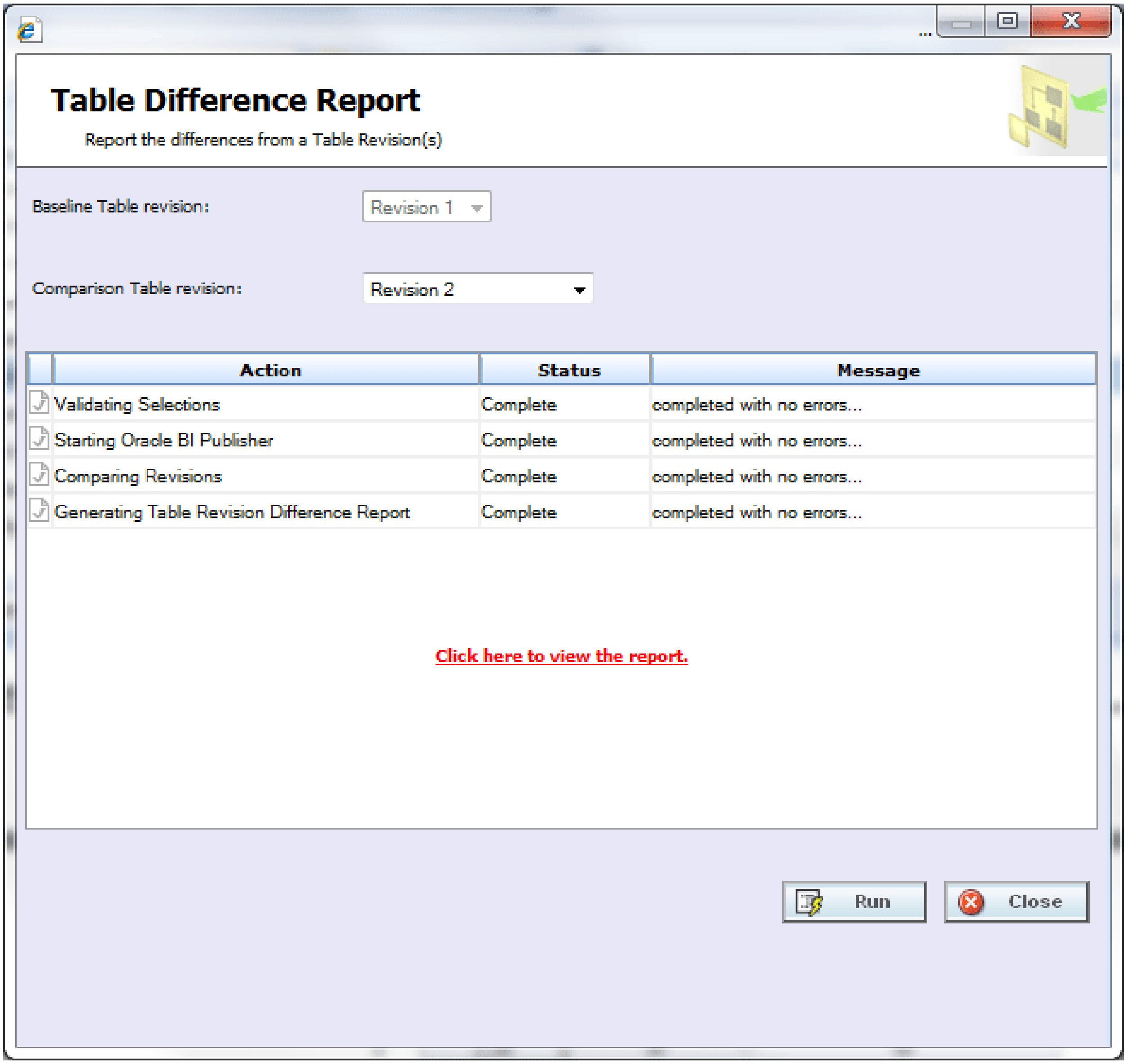


Table Difference Report

Report the differences from a Table Revision(s)

Baseline Table revision: Revision 1

Comparison Table revision: Revision 2

Action	Status	Message
Validating Selections	Complete	completed with no errors...
Starting Oracle BI Publisher	Complete	completed with no errors...
Comparing Revisions	Complete	completed with no errors...
Generating Table Revision Difference Report	Complete	completed with no errors...

Click here to view the report.

Figure 321 Successful Run of a Table Comparison Report

6. When the wizard finishes, click on 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.

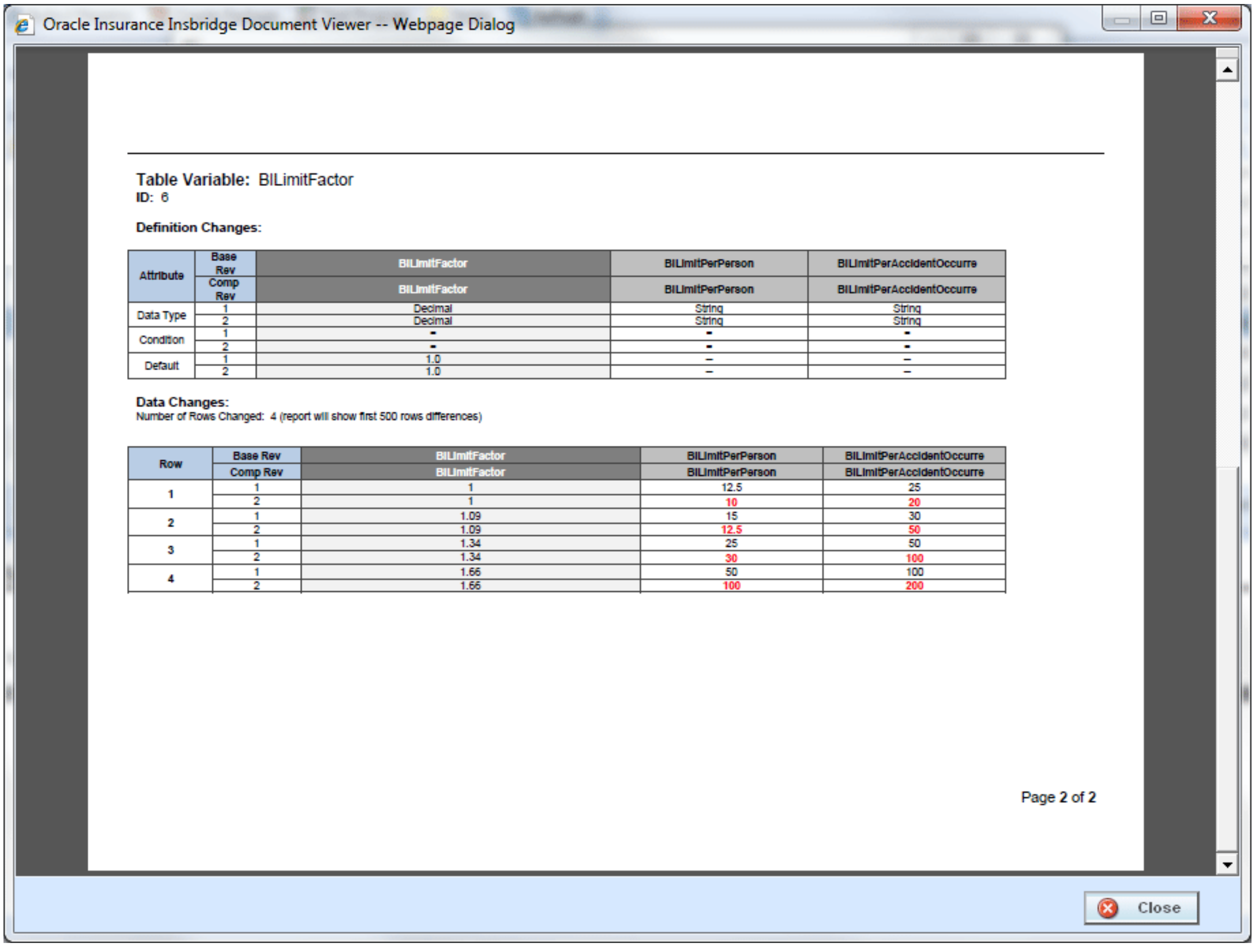

Oracle Insurance Insbridge Document Viewer -- Webpage Dialog
Close

Table Variable: BILimitFactor
ID: 6

Definition Changes:

Attribute	Base Rev	BILimitFactor	BILimitPerPerson	BILimitPerAccidentOccurre
	Comp Rev	BILimitFactor	BILimitPerPerson	BILimitPerAccidentOccurre
Data Type	1	Decimal	String	String
	2	Decimal	String	String
Condition	1	-	-	-
	2	-	-	-
Default	1	1.0	–	–
	2	1.0	–	–

Data Changes:
Number of Rows Changed: 4 (report will show first 500 rows differences)

Row	Base Rev	BILimitFactor	BILimitPerPerson	BILimitPerAccidentOccurre
	Comp Rev	BILimitFactor	BILimitPerPerson	BILimitPerAccidentOccurre
1	1	1	12.5	25
	2	1	10	20
2	1	1.09	15	30
	2	1.09	12.5	50
3	1	1.34	25	50
	2	1.34	30	100
4	1	1.66	50	100
	2	1.66	100	200

Page 2 of 2

Figure 322 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.

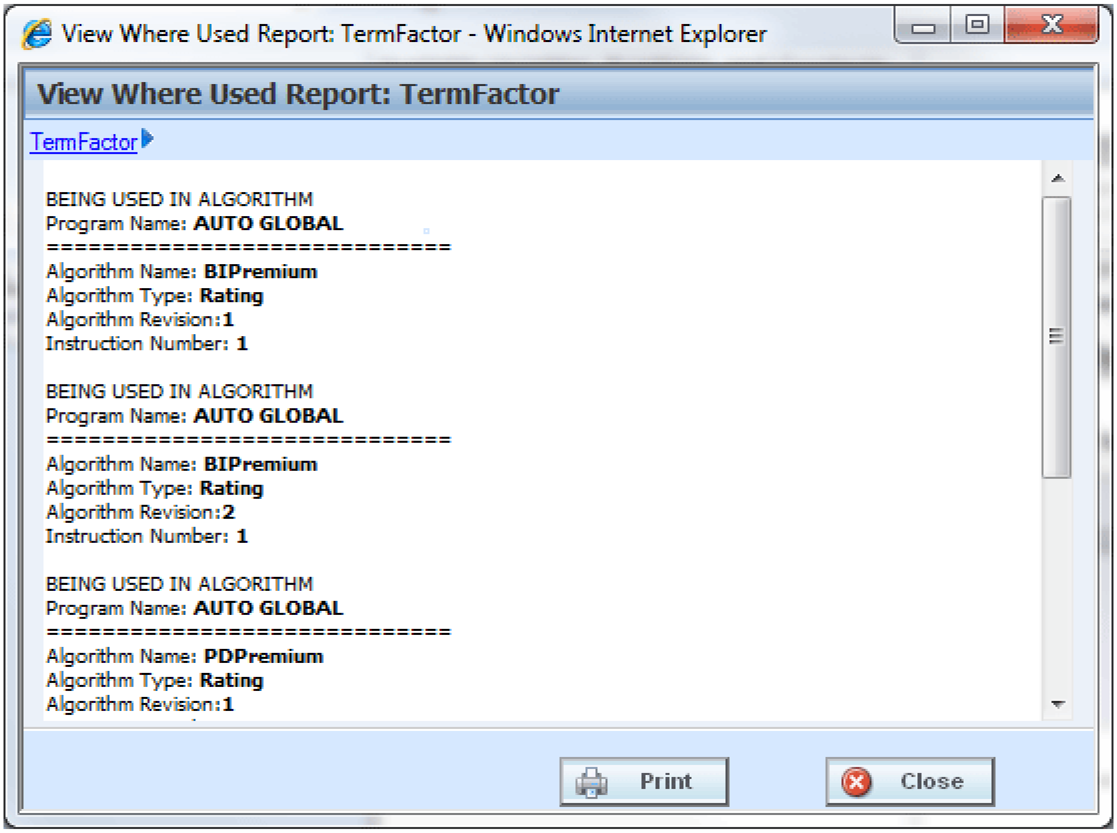
Table Difference Reports are limited to the first 500 rows of data.

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 re-name the file.*

VIEW WHERE USED

View Where Used allows you to see if and where a variable or input is used in a program. The View Where Used Report will show all occurrences of the variable or input in other variables, algorithms, or result mapping, in all programs across the subline. You also can click through to further see where the variable or input is used in calculated variables or algorithms. The report will load in a popup window and may be viewed or printed.



View Where Used Report: TermFactor - Windows Internet Explorer
View Where Used Report: TermFactor
TermFactor
Print
Close

```
BEING USED IN ALGORITHM
Program Name: AUTO GLOBAL
=============================
Algorithm Name: BIPremium
Algorithm Type: Rating
Algorithm Revision:1
Instruction Number: 1

BEING USED IN ALGORITHM
Program Name: AUTO GLOBAL
=============================
Algorithm Name: BIPremium
Algorithm Type: Rating
Algorithm Revision:2
Instruction Number: 1

BEING USED IN ALGORITHM
Program Name: AUTO GLOBAL
=============================
Algorithm Name: PDPremium
Algorithm Type: Rating
Algorithm Revision:1
```

Figure 323 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 constants.

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 on 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 26

COMMON FUNCTIONALITY

There are some functions that can be performed in many different areas of the IBRU system such as:

- Locking
- Notes
- Change Control
- Constants

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 copying, 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.

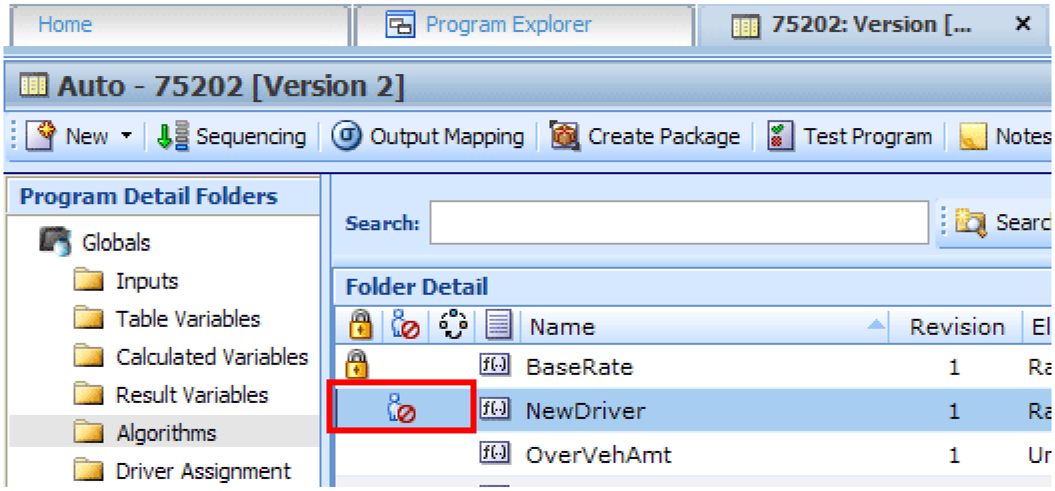
IN-USE LOCKS

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 will be locked automatically when you enter the edit tab. The element will be unlocked when you exit the edit tab, not when you save.

An element will display 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 will be displayed by the variable name and the revision you are editing. If you edit the non-active revision, a lock icon will be displayed next to the revision you are editing and not next to the variable name.

Single elements such as categories, input variables, result variables, program sequencing, and result mapping will be closed to editing when a lock is in place. Revisioned elements, such as table variables, calculated variables and algorithms will 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.



Name	Revision	El
BaseRate	1	Ra
NewDriver	1	Ra
OverVehAmt	1	Ur

Figure 324 In-Use Locking on an Element

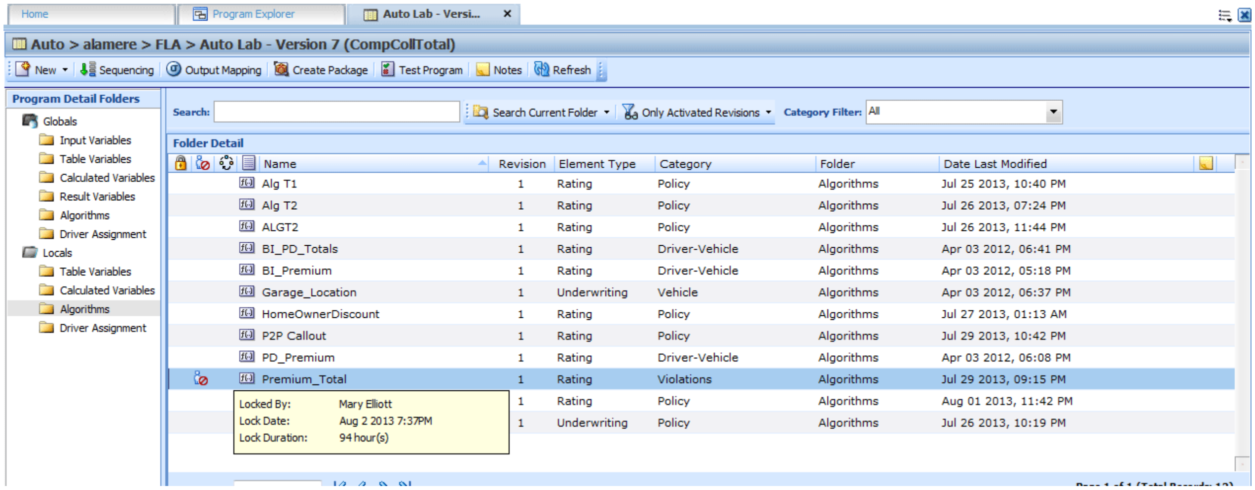
Any open element can have an element lock placed on it when a user goes to edit it. An element lock will be placed on the element you are currently editing. If you are editing an algorithm and open up a new tab to look at inputs, the algorithm you are editing will display an In-Use lock until you close the editing tab.

Elements that are locked by a program lock or a template lock will not be 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.



Auto > alamere > FLA > Auto Lab - Version 7 (CompCollTotal)

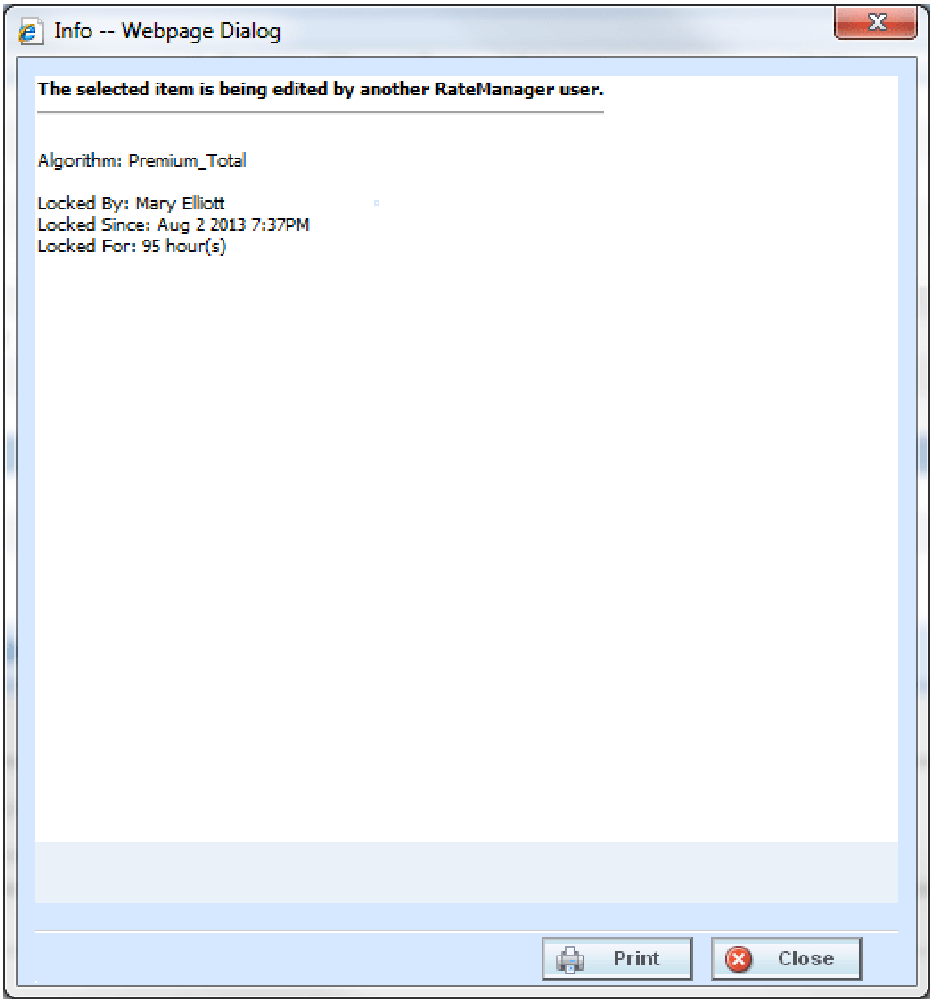
Category Filter: All

Name	Revision	Element Type	Category	Folder	Date Last Modified
Alg T1	1	Rating	Policy	Algorithms	Jul 25 2013, 10:40 PM
Alg T2	1	Rating	Policy	Algorithms	Jul 26 2013, 07:24 PM
ALGT2	1	Rating	Policy	Algorithms	Jul 26 2013, 11:44 PM
BI_PD_Totals	1	Rating	Driver-Vehicle	Algorithms	Apr 03 2012, 06:41 PM
BI_Premium	1	Rating	Driver-Vehicle	Algorithms	Apr 03 2012, 05:18 PM
Garage_Location	1	Underwriting	Vehicle	Algorithms	Apr 03 2012, 06:37 PM
HomeOwnerDiscount	1	Rating	Policy	Algorithms	Jul 27 2013, 01:13 AM
P2P Callout	1	Rating	Policy	Algorithms	Jul 29 2013, 10:42 PM
PD_Premium	1	Rating	Driver-Vehicle	Algorithms	Apr 03 2012, 06:08 PM
Premium_Total	1	Rating	Violations	Algorithms	Jul 29 2013, 09:15 PM
	1	Rating	Policy	Algorithms	Aug 01 2013, 11:42 PM
	1	Underwriting	Policy	Algorithms	Jul 26 2013, 10:19 PM

Locked By: Mary Elliott
Lock Date: Aug 2 2013 7:37PM
Lock Duration: 94 hour(s)

Figure 325 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.



The selected item is being edited by another RateManager user.

Algorithm: Premium_Total

Locked By: Mary Elliott
Locked Since: Aug 2 2013 7:37PM
Locked For: 95 hour(s)

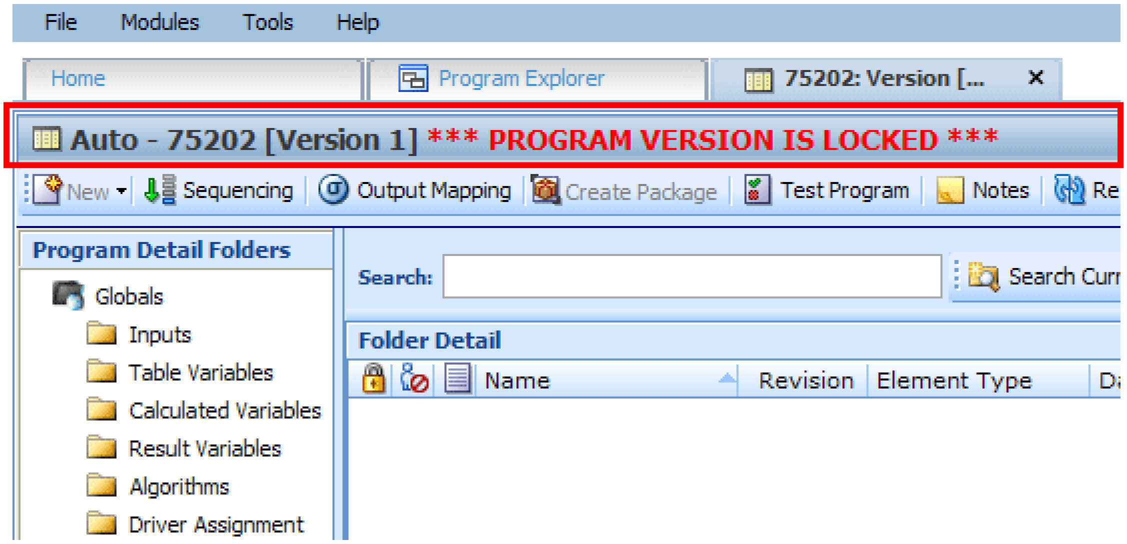
Figure 326 In-Use Lock Information

Program Version Locks

An administrator can lock a program version to prevent it from being changed or deleted. This will lock 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 will be locked throughout the subline based on the last packaged date. Changes to these element versions in this subline will not be allowed, regardless of what folder or program version you are working in. Locking a version will not affect other sublines.

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 will be locked in all three program versions. You will not be able to edit or delete any of these element versions in any program in the subline. Elements in program versions outside of the version that has the lock applied will be open for copying. Elements that are locked in version 2 will not be open for copying. The same elements in version 3 will be open for copying.

On The Program Details tab there will be a program lock icon indicating which items are locked at the program version level. Locking a program version will also lock the associated Global Categories, Inputs, Variables, Algorithms, and Driver Assignments throughout the subline. Lock icons will not be displayed on the Program Explorer tab.



Auto - 75202 [Version 1] *** PROGRAM VERSION IS LOCKED ***

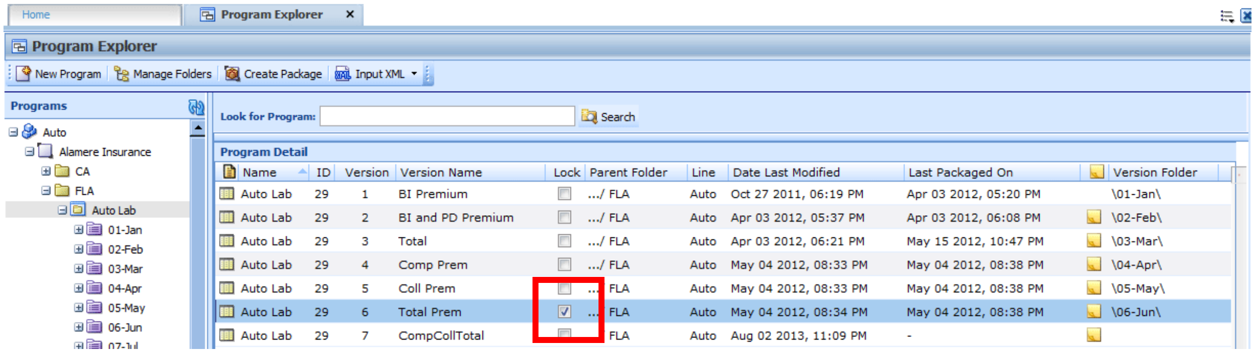
Figure 327 Locked Program Version Banner

When you enter a program version that is locked, a **VERSION LOCKED** banner is at the top of the tab. Other versions in that subline will not have the banner but will have the lock icons indicating the locked items.

Locking a Program Version

1. Navigate to the **Program Explorer** tab. Select the subline, folder, and program that contains the program version you want to lock.

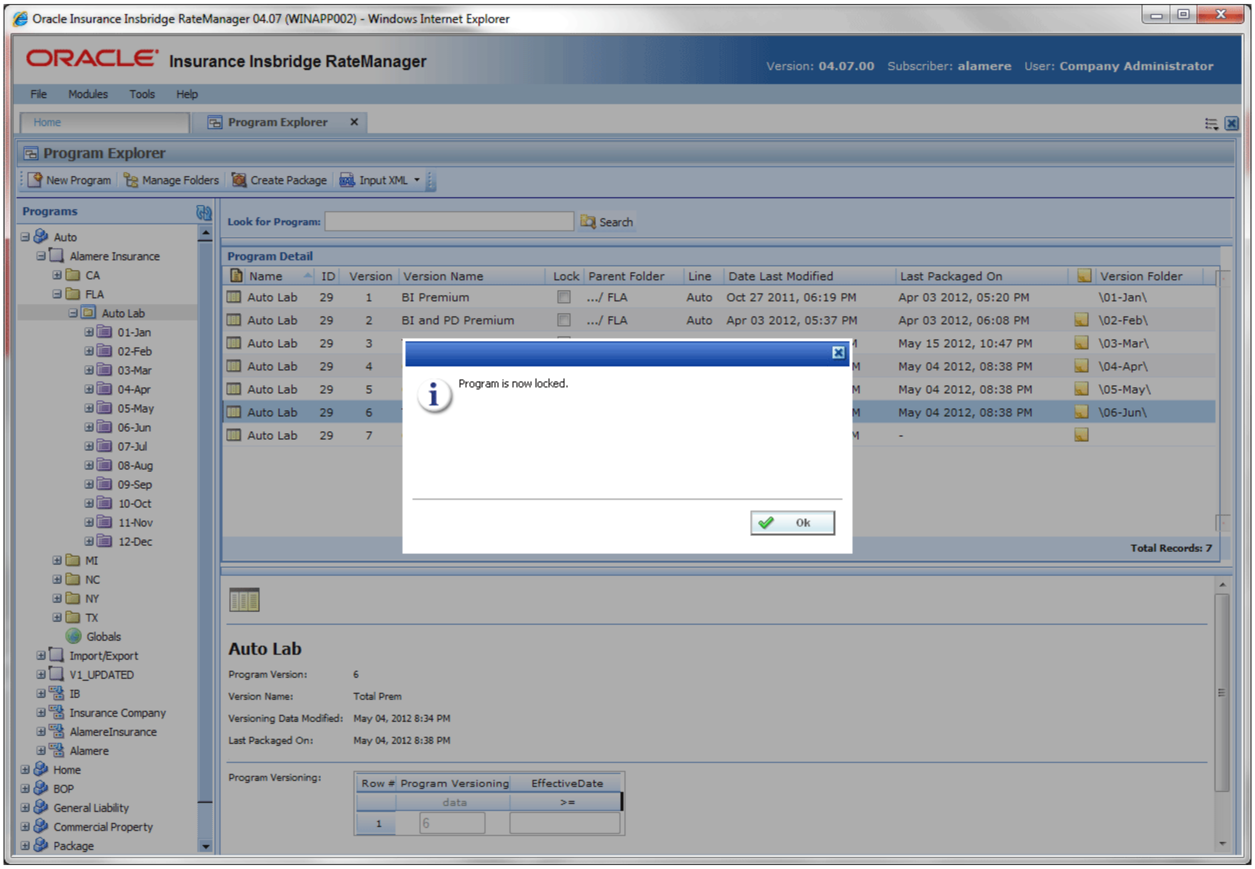
2. Place a check in the **LOCK** checkbox for the version you want to lock.



Name	ID	Version	Version Name	Lock	Parent Folder	Line	Date Last Modified	Last Packaged On	Version Folder
Auto Lab	29	1	BI Premium	□	.../ FLA	Auto	Oct 27 2011, 06:19 PM	Apr 03 2012, 05:20 PM	\01-Jan\
Auto Lab	29	2	BI and PD Premium	□	.../ FLA	Auto	Apr 03 2012, 05:37 PM	Apr 03 2012, 06:08 PM	\02-Feb\
Auto Lab	29	3	Total	□	.../ FLA	Auto	Apr 03 2012, 06:21 PM	May 15 2012, 10:47 PM	\03-Mar\
Auto Lab	29	4	Comp Prem	□	.../ FLA	Auto	May 04 2012, 08:33 PM	May 04 2012, 08:38 PM	\04-Apr\
Auto Lab	29	5	Coll Prem	□	.../ FLA	Auto	May 04 2012, 08:33 PM	May 04 2012, 08:38 PM	\05-May\
Auto Lab	29	6	Total Prem	☑	.../ FLA	Auto	May 04 2012, 08:34 PM	May 04 2012, 08:38 PM	\06-Jun\
Auto Lab	29	7	CompCollTotal	□	.../ FLA	Auto	Aug 02 2013, 11:09 PM	-	

Figure 328 Selecting a Program Version to Lock

3. A message is displayed. Click **OK**. This version is now locked.



Program is now locked.

Auto Lab

Program Version: 6

Version Name: Total Prem

Versioning Data Modified: May 04, 2012 8:34 PM

Last Packaged On: May 04, 2012 8:38 PM

Program Versioning:

Row #	Program Versioning	EffectiveDate
	data	>=
1	6	

Figure 329 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.

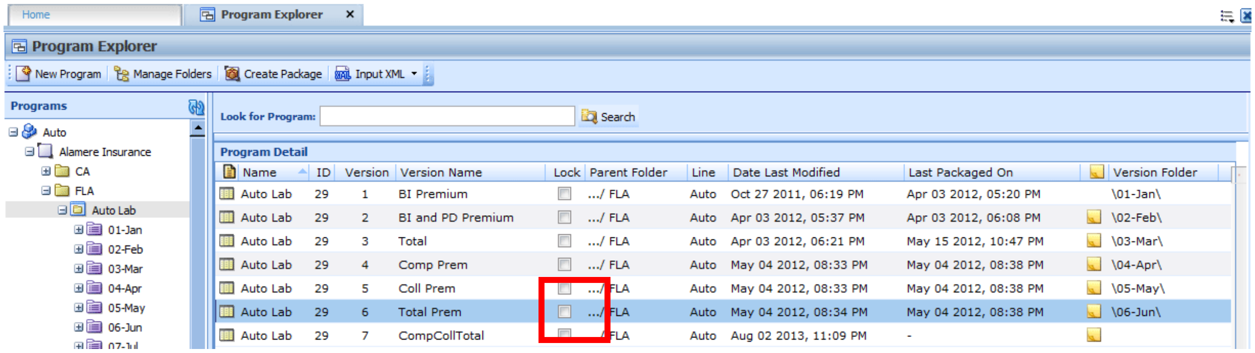
- The New, Delete, and Create Package options along with the Copy, Show/Hide Revisions and Delete options will be removed from the right click menu of the program elements.
- Result groups will have the New Group and Copy Group options removed. Edit Groups will have the Save option removed.
- Program 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 will free 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 subline, folder, and program that contains the program version you want to unlock.

2. Remove the check in the **LOCK** checkbox for the version you want to unlock.



Name	ID	Version	Version Name	Lock	Parent Folder	Line	Date Last Modified	Last Packaged On	Version Folder
Auto Lab	29	1	BI Premium	□	.../ FLA	Auto	Oct 27 2011, 06:19 PM	Apr 03 2012, 05:20 PM	\01-Jan\
Auto Lab	29	2	BI and PD Premium	□	.../ FLA	Auto	Apr 03 2012, 05:37 PM	Apr 03 2012, 06:08 PM	\02-Feb\
Auto Lab	29	3	Total	□	.../ FLA	Auto	Apr 03 2012, 06:21 PM	May 15 2012, 10:47 PM	\03-Mar\
Auto Lab	29	4	Comp Prem	□	.../ FLA	Auto	May 04 2012, 08:33 PM	May 04 2012, 08:38 PM	\04-Apr\
Auto Lab	29	5	Coll Prem	□	.../ FLA	Auto	May 04 2012, 08:33 PM	May 04 2012, 08:38 PM	\05-May\
Auto Lab	29	6	Total Prem	□	.../ FLA	Auto	May 04 2012, 08:34 PM	May 04 2012, 08:38 PM	\06-Jun\
Auto Lab	29	7	CompCollTotal	□	.../ FLA	Auto	Aug 02 2013, 11:09 PM	-	

Figure 330 Selecting a Program Version to Unlock

3. A message will be displayed. Click **OK**. This version is now unlocked.

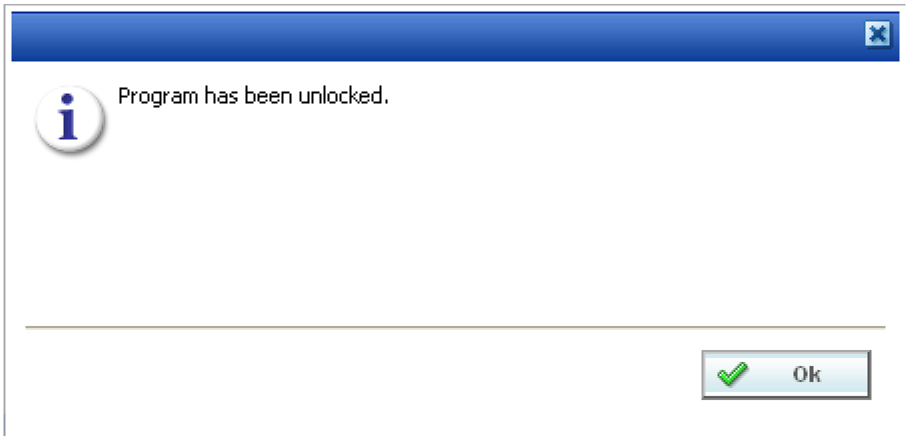

Program has been unlocked.

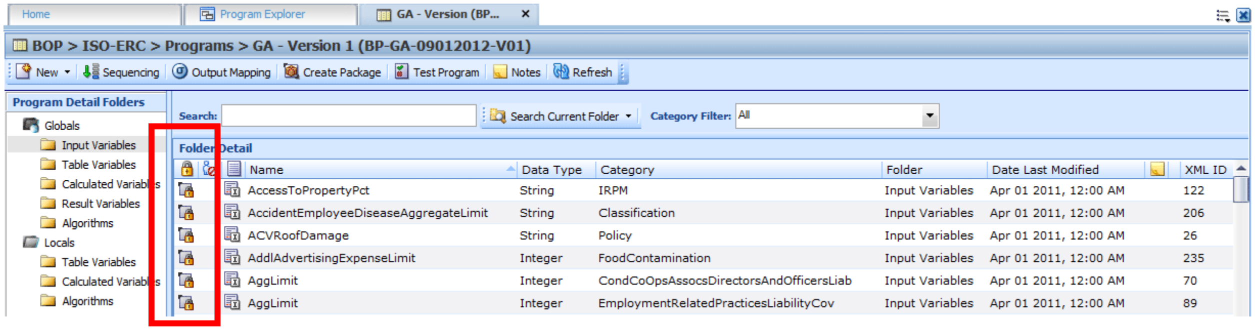
Figure 331 Unlocking a Program Version

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, input variables, result variables, program sequencing, and result mapping will be closed to editing or deletion by a template lock. Revisioned elements, such as table variables, calculated variables and algorithms will have the selected revision closed to editing or deletion by a template lock. There will be 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 will not have a template lock. You can edit or delete as needed.



Category Filter: All

🔒	Name	Data Type	Category	Folder	Date Last Modified	XML ID
🔒	AccessToPropertyPct	String	IRPM	Input Variables	Apr 01 2011, 12:00 AM	122
🔒	AccidentEmployeeDiseaseAggregateLimit	String	Classification	Input Variables	Apr 01 2011, 12:00 AM	206
🔒	ACVRoofDamage	String	Policy	Input Variables	Apr 01 2011, 12:00 AM	26
🔒	AddlAdvertisingExpenseLimit	Integer	FoodContamination	Input Variables	Apr 01 2011, 12:00 AM	235
🔒	AggLimit	Integer	CondCoOpsAssocsDirectorsAndOfficersLiab	Input Variables	Apr 01 2011, 12:00 AM	70
🔒	AggLimit	Integer	EmploymentRelatedPracticesLiabilityCov	Input Variables	Apr 01 2011, 12:00 AM	89

Figure 332 Template Locks

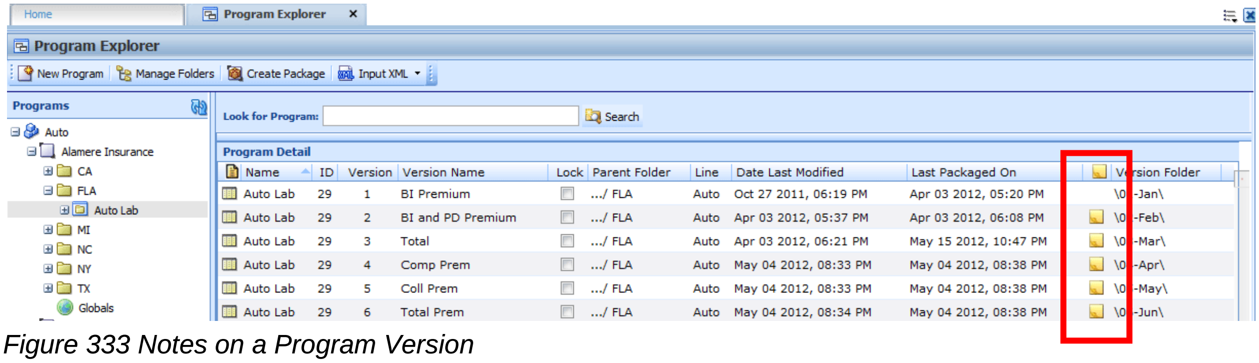
Notes can be added to a template locked element at any time.

NOTES

Notes allow for explanations or comments to be added to an element or a program version. These are optional entries that can be added to program versions, variable revisions, algorithm revisions, sequencing, result mapping, categories, and inputs. 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.



Name	ID	Version	Version Name	Lock	Parent Folder	Line	Date Last Modified	Last Packaged On	🗒	Version Folder
Auto Lab	29	1	BI Premium	□	.../ FLA	Auto	Oct 27 2011, 06:19 PM	Apr 03 2012, 05:20 PM		\0-Jan\
Auto Lab	29	2	BI and PD Premium	□	.../ FLA	Auto	Apr 03 2012, 05:37 PM	Apr 03 2012, 06:08 PM	🗒	\0-Feb\
Auto Lab	29	3	Total	□	.../ FLA	Auto	Apr 03 2012, 06:21 PM	May 15 2012, 10:47 PM	🗒	\0-Mar\
Auto Lab	29	4	Comp Prem	□	.../ FLA	Auto	May 04 2012, 08:33 PM	May 04 2012, 08:38 PM	🗒	\0-Apr\
Auto Lab	29	5	Coll Prem	□	.../ FLA	Auto	May 04 2012, 08:33 PM	May 04 2012, 08:38 PM	🗒	\0-May\
Auto Lab	29	6	Total Prem	□	.../ FLA	Auto	May 04 2012, 08:34 PM	May 04 2012, 08:38 PM	🗒	\0-Jun\

Figure 333 Notes on a Program Version

To view the note, highlight the element and click NOTES. Notes are displayed in a separate screen.

- Notes placed on global elements and linked variables are carried through to all instances of the element revision or linked variable revision. This means that if you enter a note on revision 1 of a variable that is linked to 10 others, the note will be displayed on revision 1 of each of the linked variables. Likewise a note entered on a global variable will be displayed on that variable throughout the subline.

- Program versions must be selected for a note. You will not be able to enter a note at the program level. You must select the version where you want the note placed.

- Elements do not have to have a specific revision selected. Elements will default to the active revision. If you select a variable, but not a specific revision, the note will be 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 will not be 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 will automatically create 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

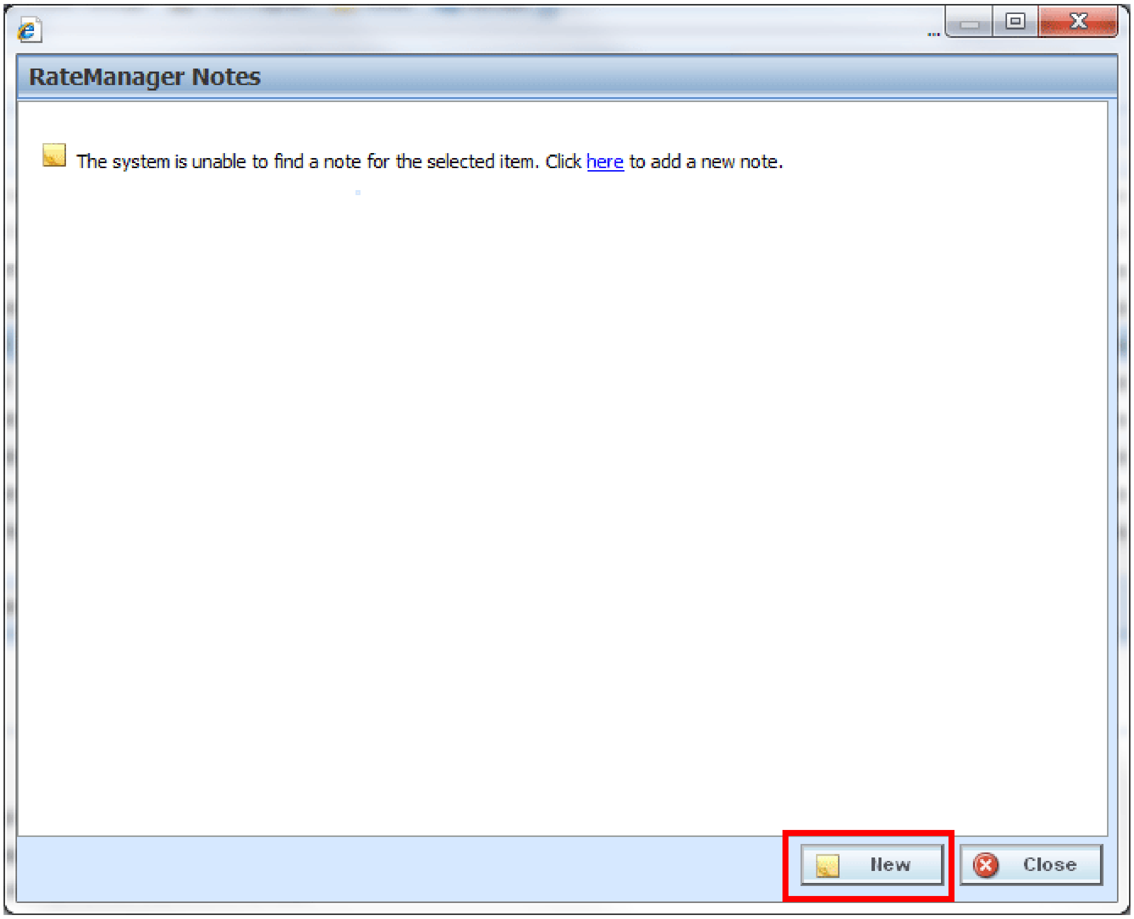
Notes can be added at any time. A note can contain up to 200 characters including spaces. Certain special characters will be restricted:

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

All other characters will be allowed.

When you enter your note, you will be able to 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 will only save the first two hundred. Any extra characters will be cut off. You cannot save a blank note.

1. Click NOTES and the main notes screen is displayed.



RateManager Notes

The system is unable to find a note for the selected item. Click here to add a new note.

Figure 334 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 will be displayed.

3. The *Notes Detail Screen* contains the details of the note and/or any attachments. Enter your **Note** and/or attachment.

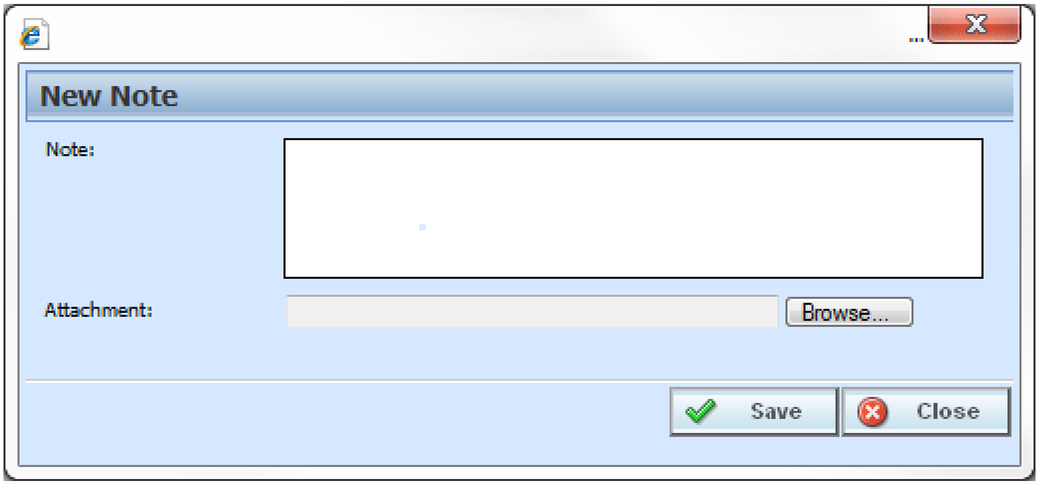


Figure 335 Completing the Notes Detail Screen

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.

FIELDS

A blank Notes screen has two buttons, New and Close. A populated Notes screen will also allow for editing and deleting. The Notes Detail screen has Save and Close.

New	**New:** Opens a Notes Detail screen where you can enter and save your note.
Close	**Close:** Closes the screen. If you have not saved your note entry, this will close the screen without saving any additions or changes.
Save	**Save:** Saves your entry.

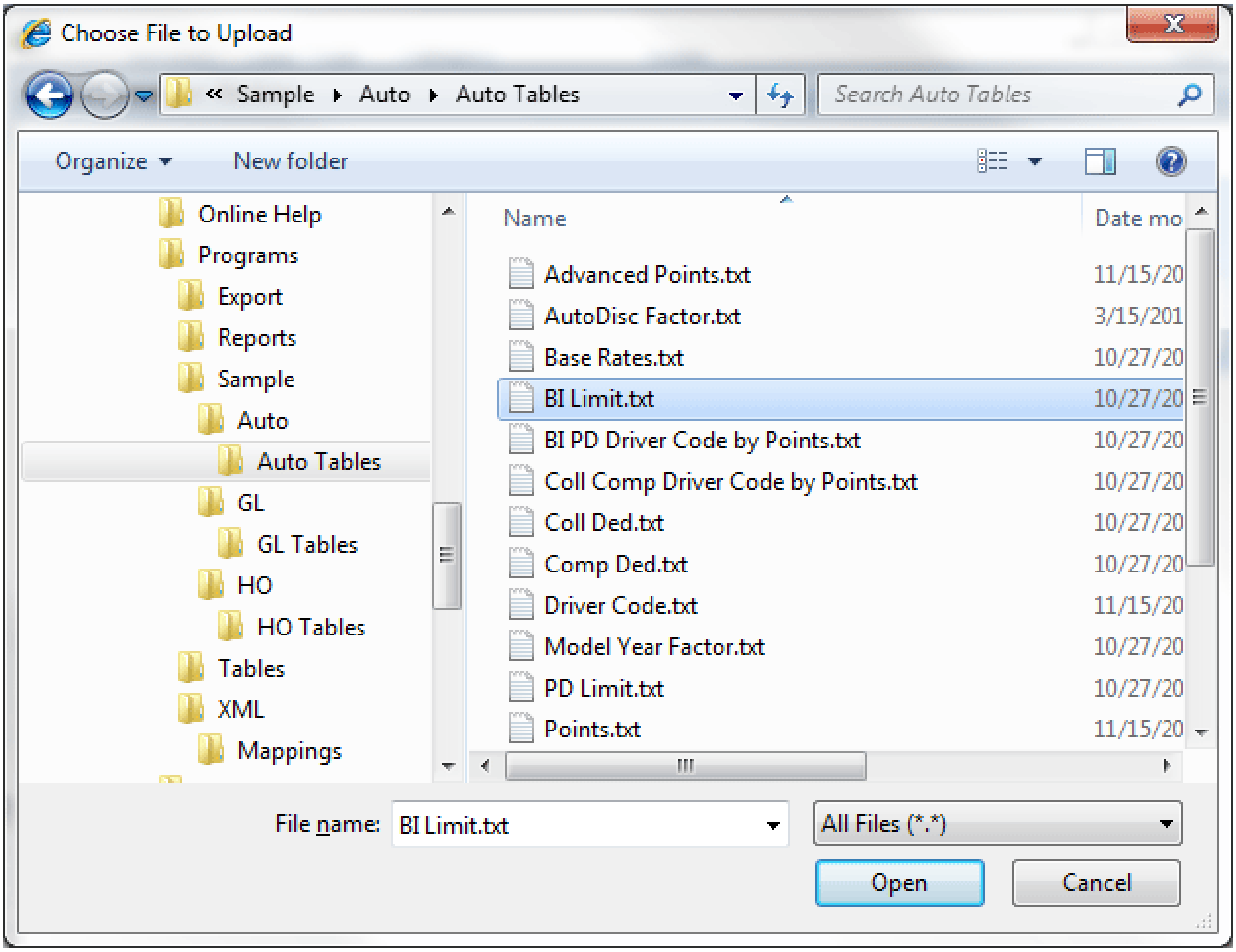
✎	**Edit:** Opens the selected entry for editing. The notes will be saved under the new date and time. Change Controls cannot be edited.
✖	**Delete:** Removes the selected entry.

Attaching Files

You can place attachments on a note. Attachments can be 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 will be cut off.

NOTE: *You can attach one file per note. If you need to attach more than one document, you will need to create a new note for each document you need to attach.*

1. To attach a file, on the notes detail screen, click **BROWSE…**, this pulls up your computers dialog box.



Name	Date mo
Advanced Points.txt	11/15/20
AutoDisc Factor.txt	3/15/201
Base Rates.txt	10/27/20
BI Limit.txt	10/27/20
BI PD Driver Code by Points.txt	10/27/20
Coll Comp Driver Code by Points.txt	10/27/20
Coll Ded.txt	10/27/20
Comp Ded.txt	10/27/20
Driver Code.txt	11/15/20
Model Year Factor.txt	10/27/20
PD Limit.txt	10/27/20
Points.txt	11/15/20

File name: BI Limit.txt

All Files (*.*)

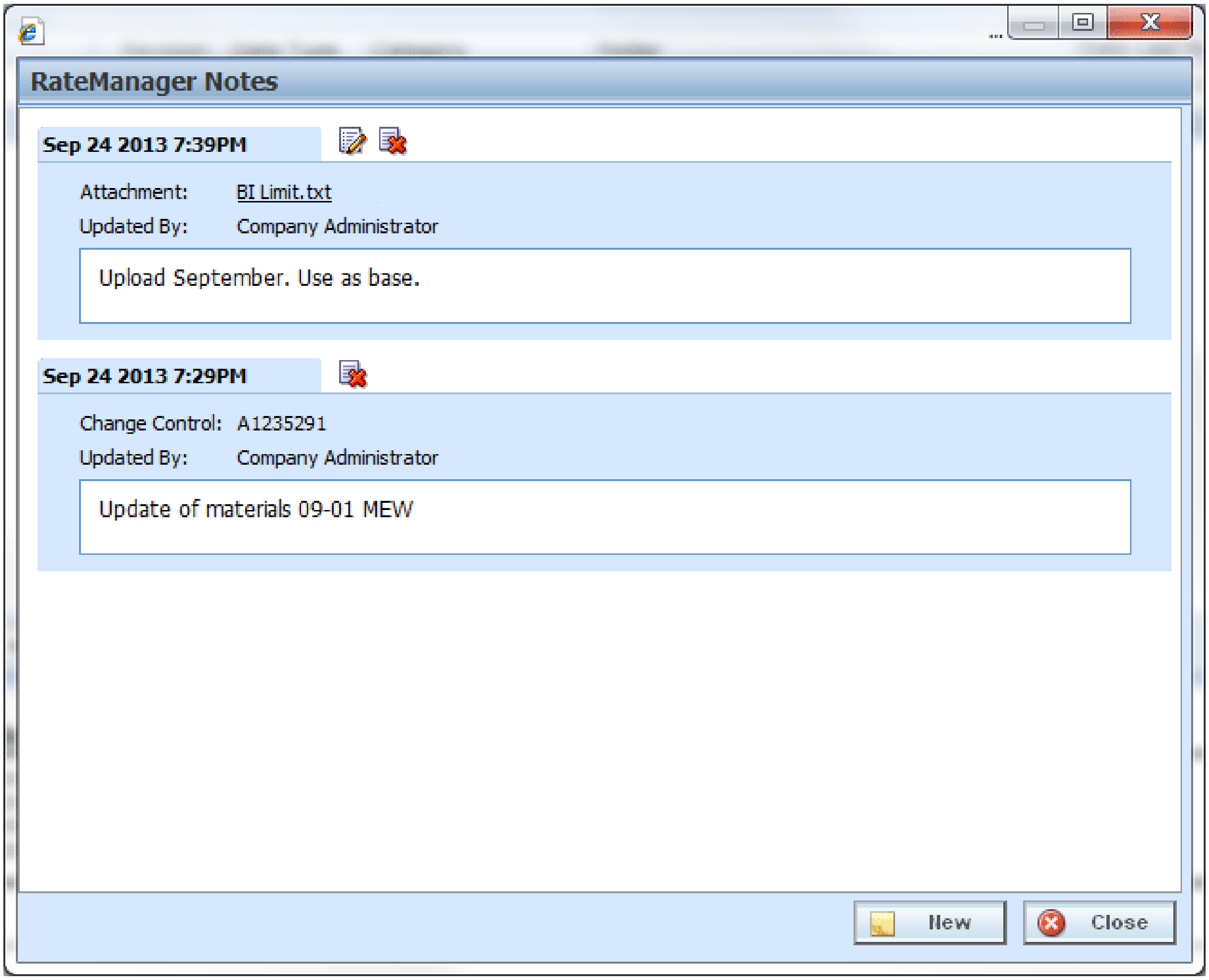
Figure 336 Dialogue Box

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 will be listed.



RateManager Notes

Sep 24 2013 7:39PM

Attachment: BI Limit.txt
Updated By: Company Administrator

Upload September. Use as base.

Sep 24 2013 7:29PM

Change Control: A1235291
Updated By: Company Administrator

Update of materials 09-01 MEW

Figure 337 Completed Notes Screen

6. Enter any additional notes, if needed.

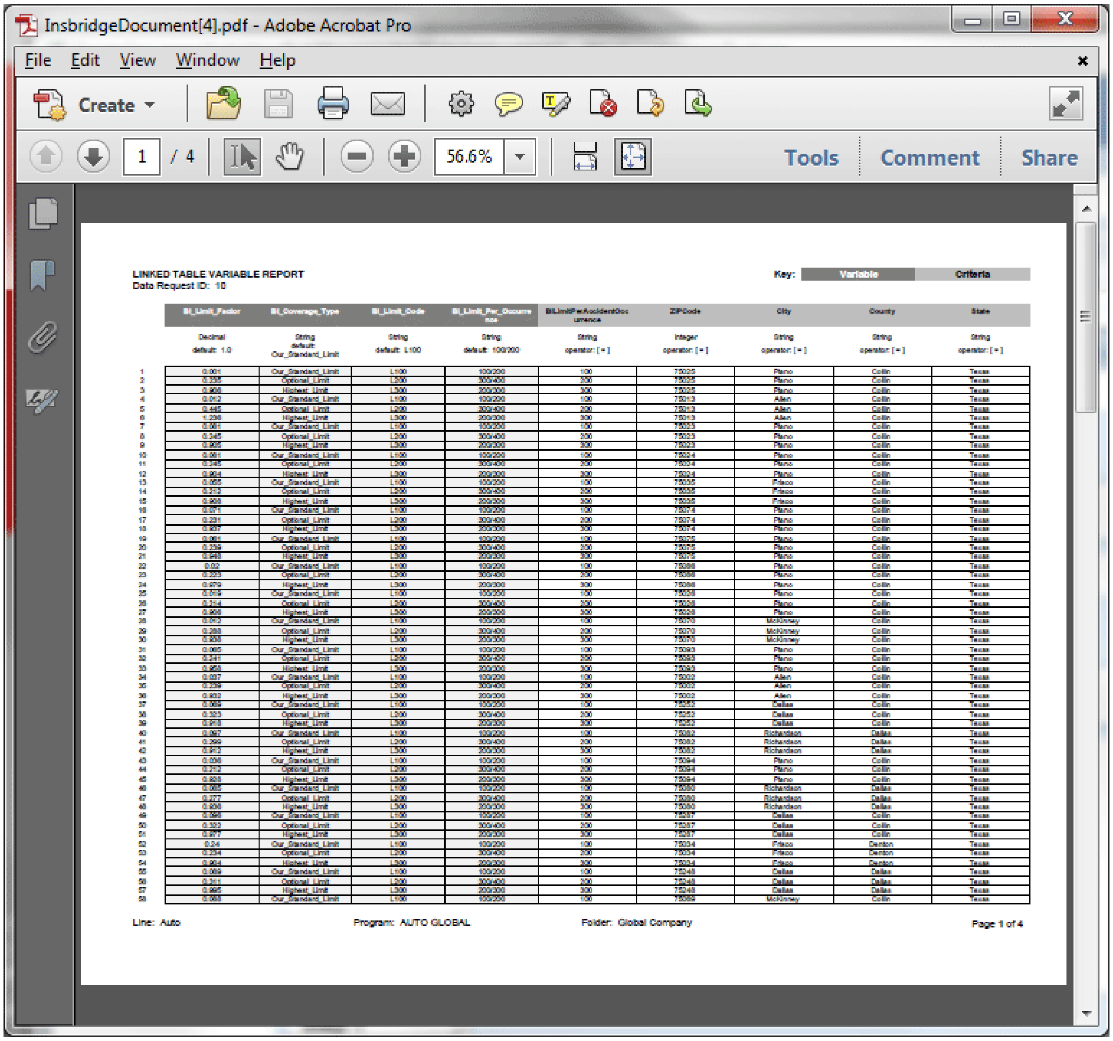
7. When you are finished, click **Close** to close the screen.

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 will be able to view the attachment. Click the attachment file you want to view. Your attachment will be displayed in a separate window, depending upon file type. Word documents will be displayed in MS Word, excel documents will be displayed in MS Excel, and PDF documents will be 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 will not be made to the attachment on the note.

If the attachment needs to be edited, you will need to go to the original source and edit there. For example, if you have uploaded a Word document, you will 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.


InsbridgeDocument[4].pdf - Adobe Acrobat Pro
File Edit View Window Help
Create
1 / 4
56.6%
Tools
Comment
Share

LINKED TABLE VARIABLE REPORT
Data Request ID: 10

Key: Variable Criteria

	BI_Limit_Factor	BI_Coverage_Type	BI_Limit_Code	BI_Limit_Per_Occurrence	BILimitPerAccidentOccurrence	ZIPCode	City	County	State
	Decimal default: 1.0	String default: Our_Standard_Limit	String default: L100	String default: 100/200	String operator: [=]	Integer operator: [=]	String operator: [=]	String operator: [=]	String operator: [=]
1	0.001	Our_Standard_Limit	L100	100/200	100	75025	Plano	Collin	Texas
2	0.235	Optional_Limit	L200	300/400	200	75025	Plano	Collin	Texas
3	0.908	Highest_Limit	L300	200/300	300	75025	Plano	Collin	Texas
4	0.012	Our_Standard_Limit	L100	100/200	100	75013	Allen	Collin	Texas
5	0.445	Optional_Limit	L200	300/400	200	75013	Allen	Collin	Texas
6	1.236	Highest_Limit	L300	200/300	300	75013	Allen	Collin	Texas
7	0.001	Our_Standard_Limit	L100	100/200	100	75023	Plano	Collin	Texas
8	0.345	Optional_Limit	L200	300/400	200	75023	Plano	Collin	Texas
9	0.905	Highest_Limit	L300	200/300	300	75023	Plano	Collin	Texas
10	0.001	Our_Standard_Limit	L100	100/200	100	75024	Plano	Collin	Texas
11	0.345	Optional_Limit	L200	300/400	200	75024	Plano	Collin	Texas
12	0.904	Highest_Limit	L300	200/300	300	75024	Plano	Collin	Texas
13	0.055	Our_Standard_Limit	L100	100/200	100	75035	Frisco	Collin	Texas
14	0.212	Optional_Limit	L200	300/400	200	75035	Frisco	Collin	Texas
15	0.908	Highest_Limit	L300	200/300	300	75035	Frisco	Collin	Texas
16	0.071	Our_Standard_Limit	L100	100/200	100	75074	Plano	Collin	Texas
17	0.231	Optional_Limit	L200	300/400	200	75074	Plano	Collin	Texas
18	0.907	Highest_Limit	L300	200/300	300	75074	Plano	Collin	Texas
19	0.001	Our_Standard_Limit	L100	100/200	100	75075	Plano	Collin	Texas
20	0.239	Optional_Limit	L200	300/400	200	75075	Plano	Collin	Texas
21	0.948	Highest_Limit	L300	200/300	300	75075	Plano	Collin	Texas
22	0.02	Our_Standard_Limit	L100	100/200	100	75086	Plano	Collin	Texas
23	0.223	Optional_Limit	L200	300/400	200	75086	Plano	Collin	Texas
24	0.979	Highest_Limit	L300	200/300	300	75086	Plano	Collin	Texas
25	0.019	Our_Standard_Limit	L100	100/200	100	75026	Plano	Collin	Texas
26	0.214	Optional_Limit	L200	300/400	200	75026	Plano	Collin	Texas
27	0.906	Highest_Limit	L300	200/300	300	75026	Plano	Collin	Texas
28	0.012	Our_Standard_Limit	L100	100/200	100	75070	McKinney	Collin	Texas
29	0.288	Optional_Limit	L200	300/400	200	75070	McKinney	Collin	Texas
30	0.938	Highest_Limit	L300	200/300	300	75070	McKinney	Collin	Texas
31	0.085	Our_Standard_Limit	L100	100/200	100	75093	Plano	Collin	Texas
32	0.241	Optional_Limit	L200	300/400	200	75093	Plano	Collin	Texas
33	0.958	Highest_Limit	L300	200/300	300	75093	Plano	Collin	Texas
34	0.037	Our_Standard_Limit	L100	100/200	100	75002	Allen	Collin	Texas
35	0.239	Optional_Limit	L200	300/400	200	75002	Allen	Collin	Texas
36	0.932	Highest_Limit	L300	200/300	300	75002	Allen	Collin	Texas
37	0.089	Our_Standard_Limit	L100	100/200	100	75252	Dallas	Collin	Texas
38	0.323	Optional_Limit	L200	300/400	200	75252	Dallas	Collin	Texas
39	0.918	Highest_Limit	L300	200/300	300	75252	Dallas	Collin	Texas
40	0.097	Our_Standard_Limit	L100	100/200	100	75082	Richardson	Dallas	Texas
41	0.299	Optional_Limit	L200	300/400	200	75082	Richardson	Dallas	Texas
42	0.912	Highest_Limit	L300	200/300	300	75082	Richardson	Dallas	Texas
43	0.036	Our_Standard_Limit	L100	100/200	100	75094	Plano	Collin	Texas
44	0.212	Optional_Limit	L200	300/400	200	75094	Plano	Collin	Texas
45	0.928	Highest_Limit	L300	200/300	300	75094	Plano	Collin	Texas
46	0.085	Our_Standard_Limit	L100	100/200	100	75080	Richardson	Dallas	Texas
47	0.277	Optional_Limit	L200	300/400	200	75080	Richardson	Dallas	Texas
48	0.936	Highest_Limit	L300	200/300	300	75080	Richardson	Dallas	Texas
49	0.096	Our_Standard_Limit	L100	100/200	100	75287	Dallas	Collin	Texas
50	0.322	Optional_Limit	L200	300/400	200	75287	Dallas	Collin	Texas
51	0.977	Highest_Limit	L300	200/300	300	75287	Dallas	Collin	Texas
52	0.24	Our_Standard_Limit	L100	100/200	100	75034	Frisco	Denton	Texas
53	0.234	Optional_Limit	L200	300/400	200	75034	Frisco	Denton	Texas
54	0.904	Highest_Limit	L300	200/300	300	75034	Frisco	Denton	Texas
55	0.089	Our_Standard_Limit	L100	100/200	100	75248	Dallas	Dallas	Texas
56	0.311	Optional_Limit	L200	300/400	200	75248	Dallas	Dallas	Texas
57	0.995	Highest_Limit	L300	200/300	300	75248	Dallas	Dallas	Texas
58	0.088	Our_Standard_Limit	L100	100/200	100	75069	McKinney	Collin	Texas

Line: Auto Program: AUTO GLOBAL Folder: Global Company Page 1 of 4

Figure 338 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 will be named “InsbridgeDocument”. It is recommended that you re-name the file.*

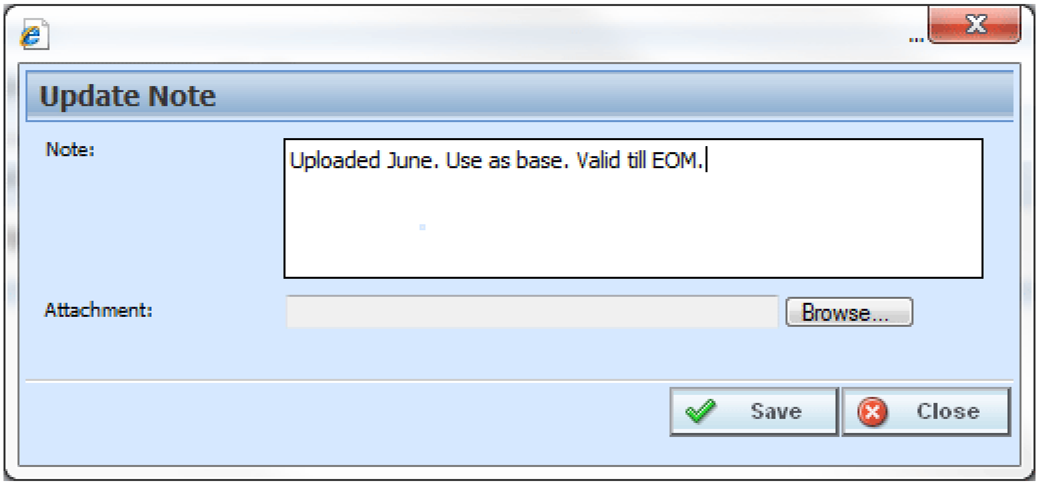
Remove Attached Files

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 will replace the first attachment.

Editing Notes

Notes can be edited at any time. You will not be able to replace or edit attachments. If you need to update or replace an attachment, you will 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.


Update Note

Note: Uploaded June. Use as base. Valid till EOM.

Attachment:

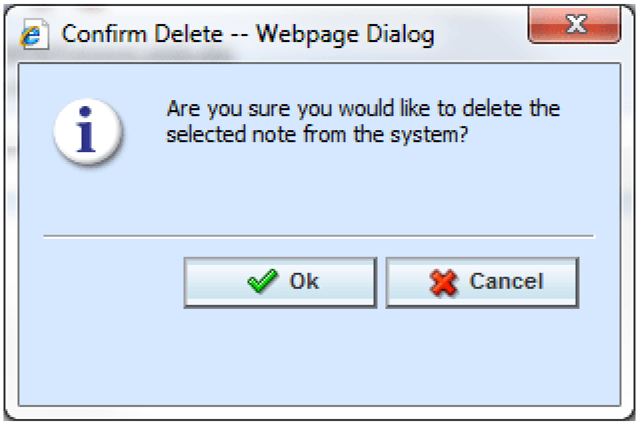
Figure 339 Editing a Note

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 will not be displayed on the program version or element. Deleting a Note will delete 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.



Confirm Delete -- Webpage Dialog

Are you sure you would like to delete the selected note from the system?

Figure 340 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 will be mandatory to make an entry to any element or program when it is revisioned. If change control is not activated, the option will not be presented. Any program, table, linked or calculated variable, and algorithm 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 or program revision has a change control, the notes icon will be displayed towards the end of the program or element revision. The notes icon will not be placed at the program level, only at the version level. The notes icon will be 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 will not be displayed at the element level.
- Change controls entered on global elements and linked variables will be carried through to all instances of the element revision or linked variable revision. This means that if you enter a change control on revision 3 of a variable that is linked to 10 others, the note will be displayed on revision 3 of each of the linked variables. Likewise a change control entered on a global variable will be displayed on that variable throughout the subline.
- To view the change control, highlight the element and click NOTES. Your note will be displayed in a separate screen.

NOTE: *Change controls are not carried through to packages, copies or new revisions.*

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:

The Change Control fields restrict:

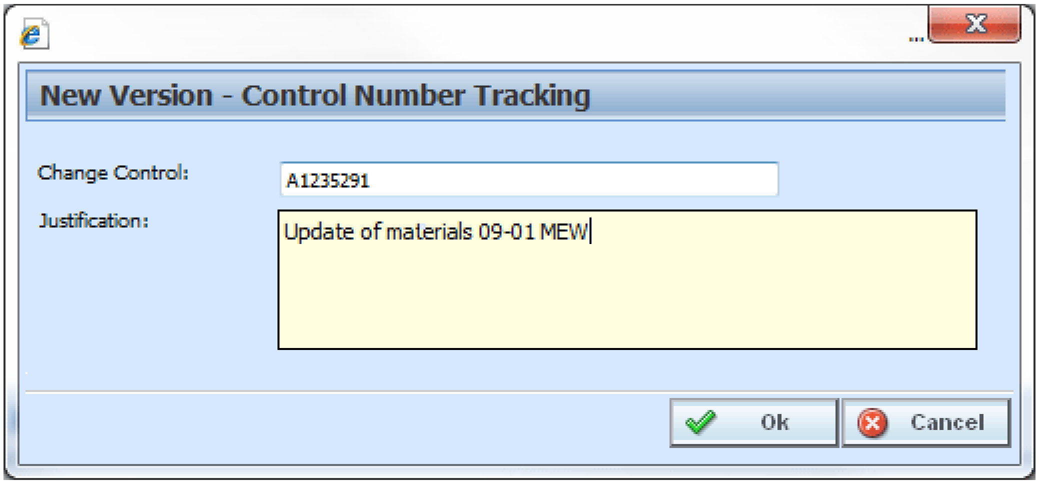
Name	Character
Ampersand	&
Asterisk	*
Backslash	\
Greater-Than Sign	>
Less-Than Sign	<
Question Mark	?
Quotation Mark	"

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.



New Version - Control Number Tracking

Change Control: A1235291

Justification: Update of materials 09-01 MEW

Figure 341 Change Control

3. Enter the change **control tracking 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 will be placed on the revision details screen for the element or program version.

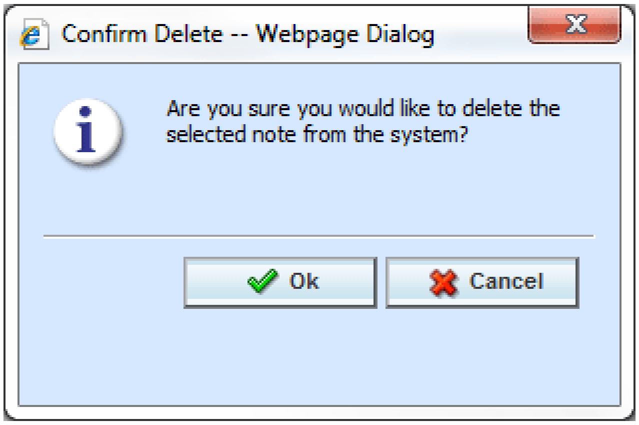
5. When you complete your revision, your change control is listed on the notes screen for that revision. Notes may be added after the revision has been created. Notes are listed in date order of newest to oldest with the newest note first. The change order will be the oldest item on the notes screen.

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 will be displayed.
3. Click **Delete** on the change control you want to delete.
4. A warning message will be displayed. Click **OK** if you want to delete the change control. Cancel will return you to the notes detail screen without any changes being made.



Are you sure you would like to delete the selected note from the system?

Figure 342 Deleting a Change Control

5. Click **Close** to close the screen.

NOTE: *Please take care when deleting change controls. This action cannot be undone.*

CONSTANTS

Constants can be used in table variables, calculated variables, algorithms and drivers assignments.

There are three types of constants in RateManager:

- **Fixed:** These constants have a fixed value. 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 100, 1000, 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 or create an input with that value.

- **System:** A system constant is a value that comes from the system. There are two system constants.

  - **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**: This is the current server time and date at the time of execution.

- **Custom Value:** 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 will be for that step only. It will not be displayed with other constant values. If you want to use this value again, you will have to enter it in again. If this is a value that you will use frequently, you may want to create a variable using that value.

Adding a New Custom Value

A new custom value can be created while entering or editing a calculated variable, table variable, algorithm, or drivers assignment without leaving the edit tab.

There are two ways to enter in a custom value:

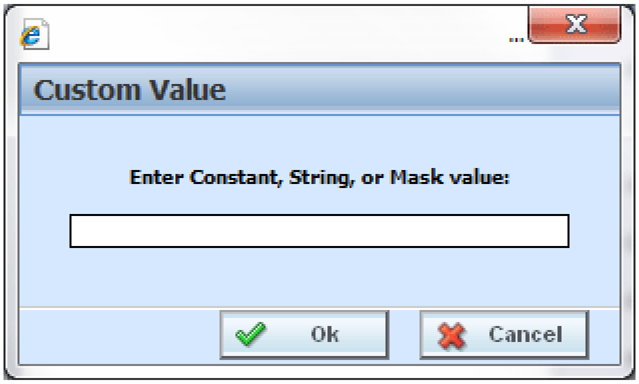
- Right click menu on an algorithm step input.

- Select **Custom Value** from the Constants menu on any table variable, calculated variable, algorithm or drivers assignment.

To Add a Custom Value

1. To create a custom value:

   a. Scroll down to the bottom of the constants menu options. Select **Custom Value**. The Custom Value popup is shown.
   b. Right click on an algorithm step input. Select **Custom Value**. The Custom Value popup is shown.



Custom Value

Enter Constant, String, or Mask value:

Figure 343 Custom Value Popup

2. Enter in a **Value**.
3. Click **Ok** to save your entry. The popup window closes and your new custom value will be populated in the input.

CONTACTING SUPPORT

If you need assistance with an Oracle Insurance Insbridge Rating and Underwriting System product, please log a Service Request using My Oracle Support at https://support.oracle.com/.

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

TTY Access to Oracle Support Services

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, seven days a week. For TTY support, call 800.446.2398.

Deaf/Hard of Hearing Access to Oracle Support Services

To reach Oracle Support Services, use a telecommunications relay service (TRS) to call Oracle Support at 1.800.223.1711. An Oracle Support Services engineer will handle technical issues and provide customer support according to the Oracle service request process. Information about TRS is available at http://www.fcc.gov/cgb/consumerfacts/trs.html, and a list of phone numbers is available at http://www.fcc.gov/cgb/dro/trsphonebk.html.

Appendix A

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 RateManager's 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 In the Testing module, you will only be able to rate files that are located on a network share, 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. This is because RateManager uses an ActiveX control to access your file system. The **Medium** security level disables some ActiveX functionality. When rating a file, the Testing module will prompt you 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 In the Testing module, you will be able to rate files located on your machine, a mapped network drive or a network share.

Warning

In all cases Low security presents no prompt to the user, and all macros are allowed to run. When macro security is set to Low, certificates of trust attached to macros are not examined by the system and are not presented to the user for acceptance. Since the user is never prompted to accept or reject these certificates, they are not posted to the trusted Trust Publishers store for Office applications.

If you use the **Low** setting, you increase your vulnerability even if you clear the **Trust all installed add-ins and templates** checkbox.

Note: If the security setting is set to **Low**, Office applications will not warn you before running a macro, therefore, all macros are run without user intervention. Because of the potential security risk, Microsoft does not recommend using the **Low** setting.

With no warnings, controls can be initialized or scripted regardless of data source or scripts.

IE 8 and IE 9

Internet Explorer 8 and Internet Explorer 9 (IE) use 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.

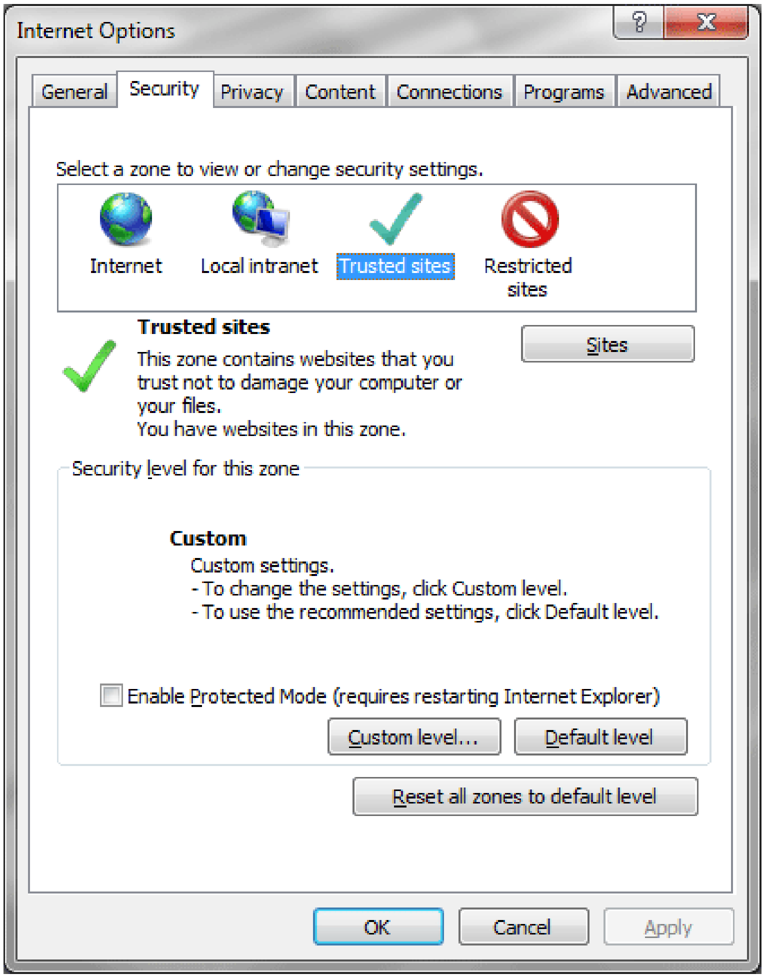
In IE, the real path to the file will be shown only if the page that contains the control is added to the trusted sites collection of the browser.

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**.



Select a zone to view or change security settings.

Trusted sites

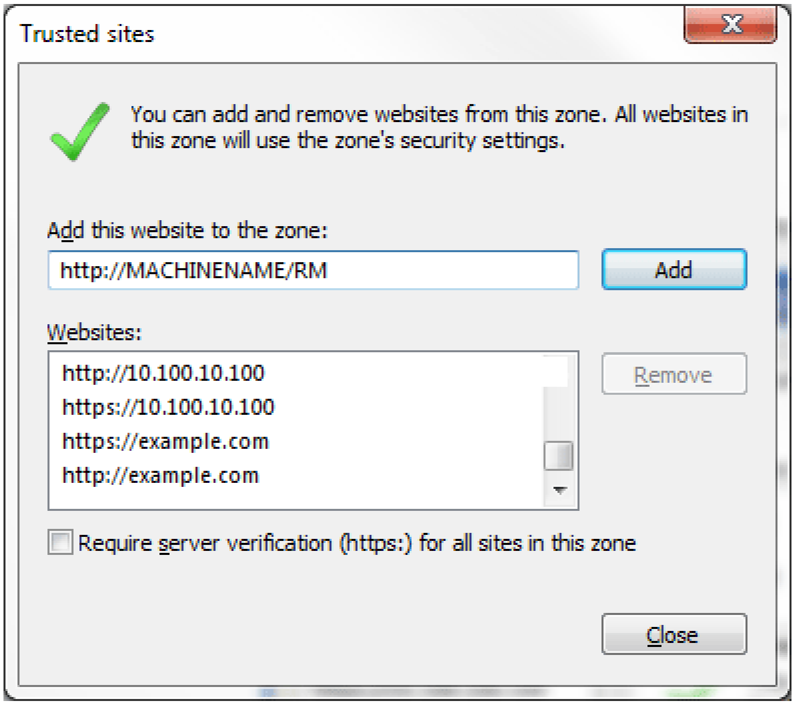
This zone contains websites that you trust not to damage your computer or your files.
You have websites in this zone.

Security level for this zone

Custom
Custom settings.
- To change the settings, click Custom level.
- To use the recommended settings, click Default level.

☐ Enable Protected Mode (requires restarting Internet Explorer)

3. Click **Sites**. A separate window is displayed.



Trusted sites

You can add and remove websites from this zone. All websites in this zone will use the zone's security settings.

Add this website to the zone:
http://MACHINENAME/RM

Websites:
http://10.100.10.100
https://10.100.10.100
https://example.com
http://example.com

□ Require server verification (https:) for all sites in this zone

4. Add your IBRU 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 IBRU 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 IBRU 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 on 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 popup blockers. Popup blocker programs will severely limit any type of auxiliary screen from being displayed. IBRU 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 8

If you have installed Service Pack 2, then you have a popup blocker on your Internet Explorer.

To turn it off:

1. Open an Internet Explorer window.
2. Click on Tools in the menu.
3. Click on Internet Options.
4. Click on the Privacy tab.
5. Uncheck Pop-up blocker.
6. 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.

Active X Controls

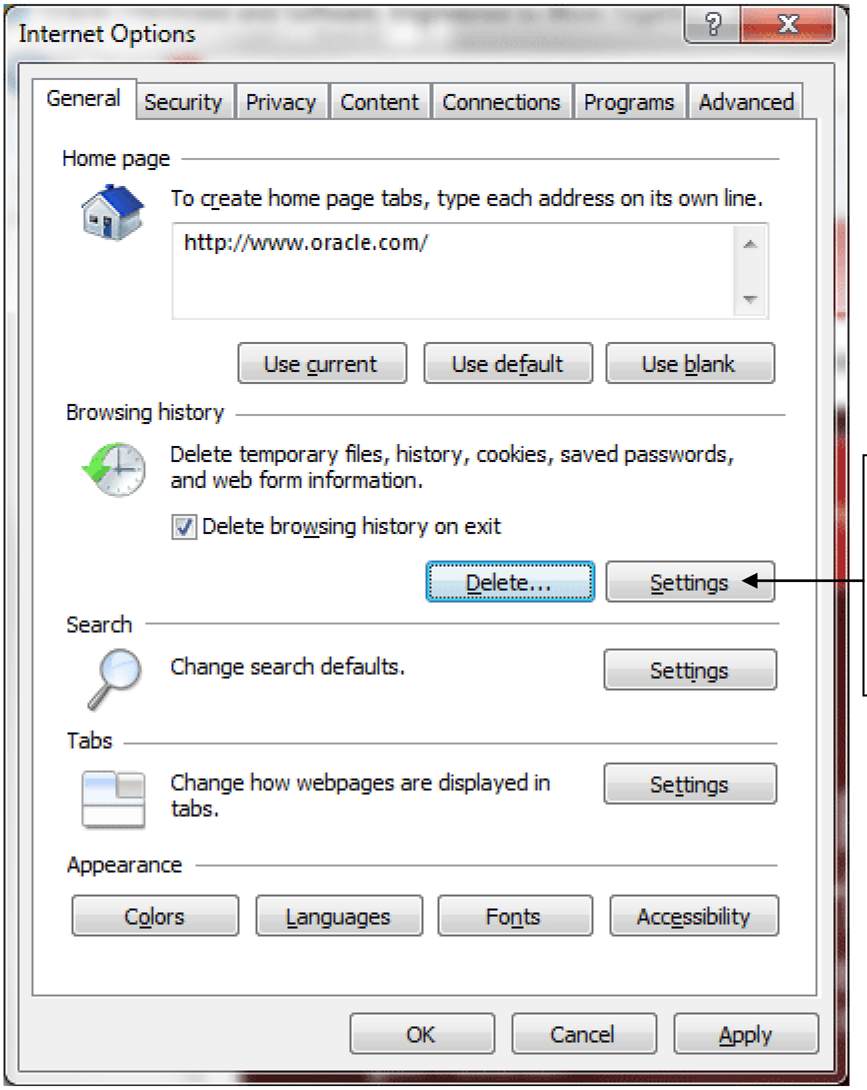
RateManager requires an Active X controller. Please see the RateManager End User Setup Guide for instructions on installing Active X available on the Oracle Technical Network (OTN) http://www.oracle.com/technetwork/documentation/insurance-097481.html

Clearing Temp Files

Because IBRU 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 create home page tabs, type each address on its own line.

http://www.oracle.com/

Delete temporary files, history, cookies, saved passwords, and web form information.

☑ Delete browsing history on exit

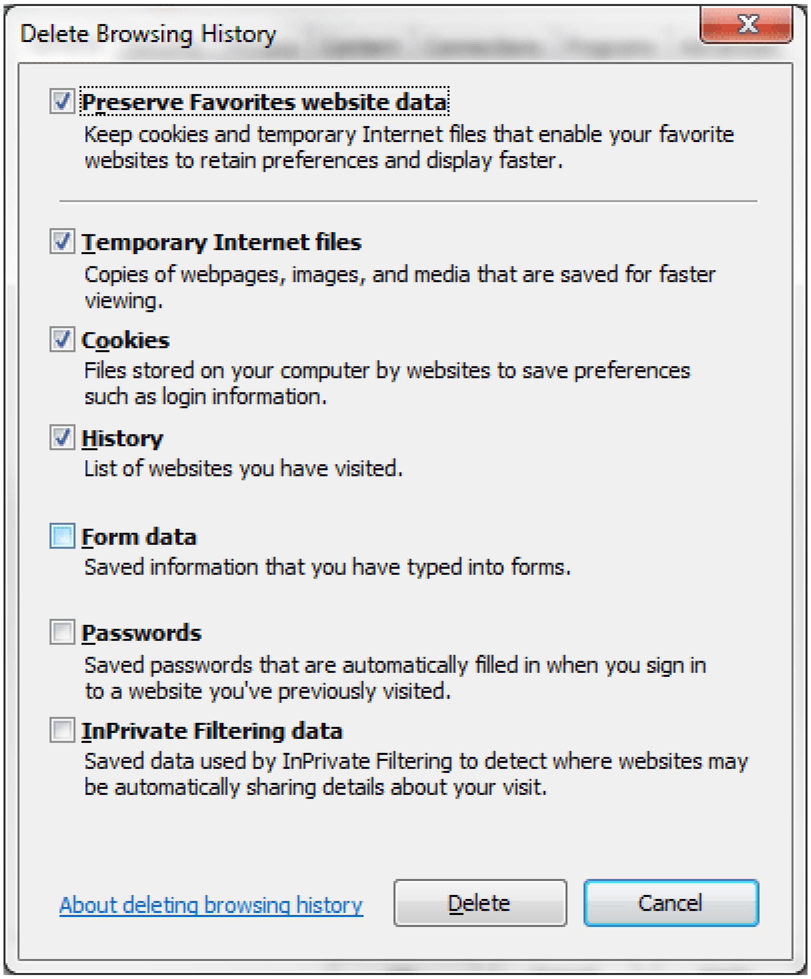
Change search defaults.

Change how webpages are displayed in tabs.

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 344 Delete Files

The Delete Browsing History is displayed.



Delete Browsing History

☑ **Preserve Favorites website data**
Keep cookies and temporary Internet files that enable your favorite websites to retain preferences and display faster.

☑ **Temporary Internet files**
Copies of webpages, images, and media that are saved for faster viewing.

☑ **Cookies**
Files stored on your computer by websites to save preferences such as login information.

☑ **History**
List of websites you have visited.

□ **Form data**
Saved information that you have typed into forms.

□ **Passwords**
Saved passwords that are automatically filled in when you sign in to a website you've previously visited.

□ **InPrivate Filtering data**
Saved data used by InPrivate Filtering to detect where websites may be automatically sharing details about your visit.

Figure 345 Temp File Warning Message

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

NAME	ABBR
CAMBODIA	KHM
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

NAME	ABBR
FRENCH POLYNESIA	PYF
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

NAME	ABBR
LAO PEOPLE'S DEMOCRATIC REPUBLIC	LAO
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

NAME	ABBR
NIGERIA	NGA
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

NAME	ABBR
SRI LANKA	LKA
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

NAME	ABBR
YUGOSLAVIA	YUG
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 inputs, variables, algorithms and the overall structure of the program.
Criteria:	An input or variable used in a table variable to determine which value to return. Any input or variable can be used as criteria.
D	
Data Type:	The type of data associated with a parameter. RateManager supports three (decimal, integer and string) data types for variables and inputs and a fourth (date) for inputs and result variables only.
Date:	A data type supported for inputs 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 inputs. Examples of decimal values are 3.1415, 18 and 0.995.
Default Value:	Used by a table 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.
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 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.
G	
Global:	An input or variable that is available to all programs under a specific subline.
Group:	A set of users that have the same access rights.
I	
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 IBRU 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:	Insbridge Rating and Underwriting System. This is the entire system.
If/Then:	A step type available for use in calculated variables and algorithms. For more information, see If in the contents section.
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:	A 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 is where templates are stored and managed.
Linked Variables:	Two or more table 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 in a table should be interpreted.
N	
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.
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.
Program 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 subline.
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	A built-in function used in the main driver assignment that defines the scope and sorting order of the ranking operation. Most require an

Function:	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.
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, inputs and variables displayed in the output file.
Revision:	A variable specific type of versioning.
RM:	RateManager. RateManager is a component within IBRU 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 subline.
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".
Subline:	Sublines are classifications that fall in between lines of business and program folders. Sublines allow for the separation of programs by source.
T	
Tab-delimited:	A type of text file in which columns are separated by tabs. This is the required format for importing tables into RateManager.
Table Variable:	A variable that uses other variables and inputs as criteria in determining the appropriate value.
Template:	Templates are exact copies of existing programs within a line of business that can be from within your own user group, any other user group within the company or even from an outside company.

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.
V	
Variable:	A name used to represent a value that can change.
Version:	One of a sequence of copies of a program, each incorporating new modifications. See Versioning in the Contents section for more information.
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 table variables that tells RateManager that one or more rows ignore the value passed in for the criteria.
MultiLines:	A MultiLines program is a type of program that allows you to call multiple programs from different lines of business together under one universal program.
Working Category:	A classification used to define how elements should run.
X	
XML ID:	A number automatically assigned by RateManager to identify inputs and categories.

Index

INDEX