ORACLE
INSURANCE

Oracle Insurance

Insbridge Enterprise Rating

Public XML Format Guide

Release 4.8

August 2014

Oracle Insurance Insbridge Enterprise Rating Public XML Format Guide

Release 4.8.x

Part # E54870-01

Library # E54888-01

August 2014

Primary Author: Mary Elizabeth Wiger, Laura Childers

CONTENTS

INTRODUCTION

Public XML is an option on the Library tab within RateManager. Public XML allows users to upload a single program created in an outside source using standard XML format, in to RateManager.

Public XML files can be created outside of RateManager in any XML editor. Once the XML file is zipped, it can be uploaded. The XML file can be a complete single program version, from categories to output mapping or a partial single program version for example, only categories and table variables. A program with more than one program version must be separted into multiple XML files.

The minimum amount of data that can be contained in the XML file is program details and categories. Categories are the only required element. If categories are not present, the applied XML file will fail. Public XML will allow users to upload an XML file with program details, categories and one or no other elements or a complete program including global and local elements plus sequencing and result groups.

All elements and programs will be a single version or revisions. Multiple element revisions and programs with multiple versions cannot be created using Public XML.

The components of a Public XML file are:

Program

- Effective Date
- Version ID
- Version
- Line of Business

Global Elements

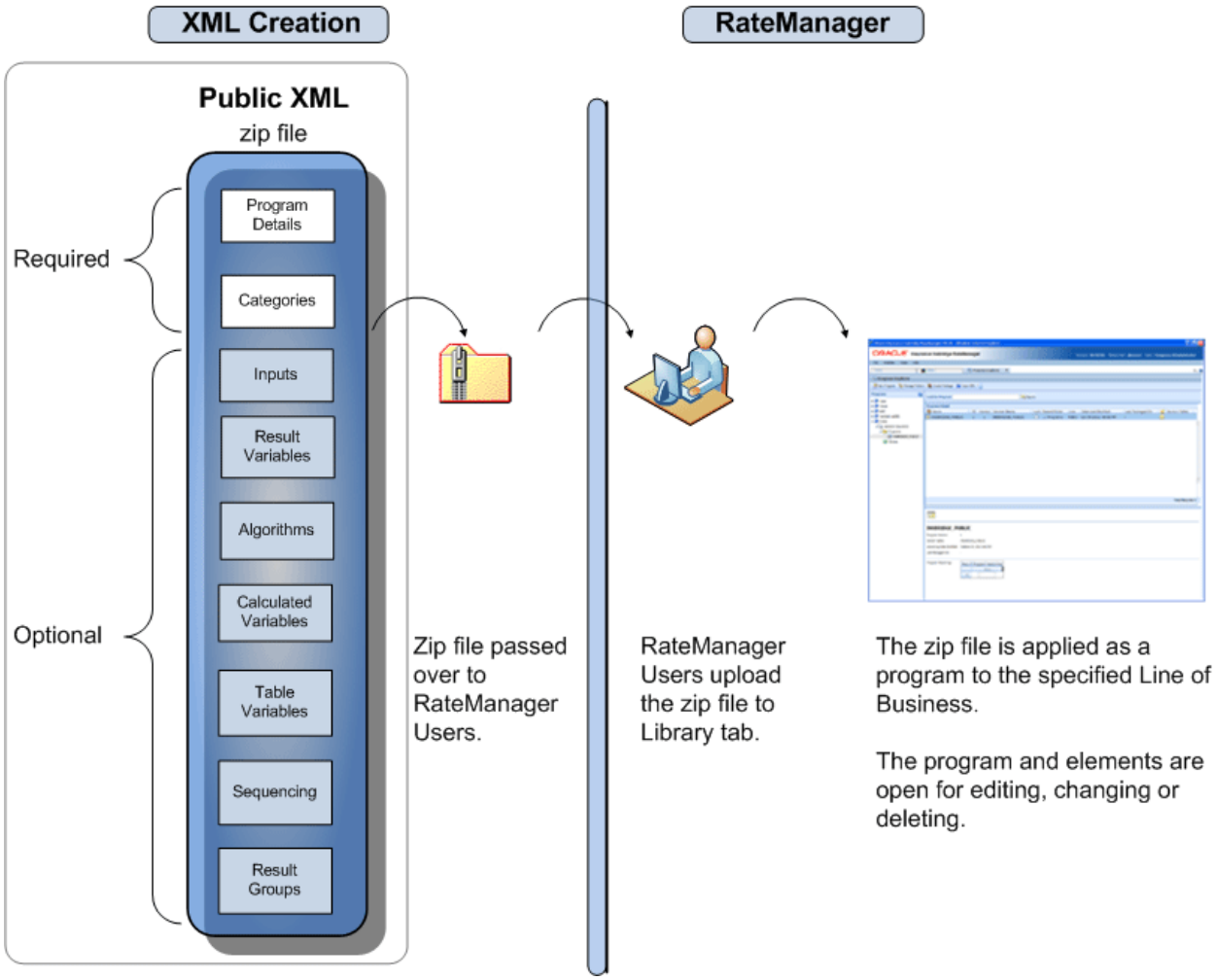
- Categories
- Inputs
- Table Variables
- Calculated Variables
- Result Variables
- Algorithms

Local / Program Elements

- Table Variables
- Table Data
- Calculated Variables
- Algorithms
- Sequencing
- Result Groups

When a variable is created under a specific program, it is referred to as a Local or program variable. When a variable is created that applies to all programs under an Insbridge subline, it is referred to as a Global variable. Both global and local elements can be used in a Public XML file.

Once the Public XML file has been created and zipped, it can be stored on a local or network drive. The drive must be accessible to the RateManager user. The XML file can be uploaded in to the Library tab in RateManager and applied. An applied Public XML file can be view, edited, updated or deleted. There are no locks on any elements or restrictions on actions to the program.



Public XML has two types of users:

- **XML Creators** – The XML creators will create the Public XML files. Required nodes and attributes are defined in this document.
- **RateManager** – The RateManager users will upload and apply the XML file within RateManager. RateManager users should use the RateManager User Guide Volume 2 for information on uploading and applying Public XML files.

Public XML Requirements and Recommendations

- **XML Editing Tool** – Public XML can be created outside of RateManager using any XML editing tool.
- **GUID Generator** – A GUID Generator is recommended. Each element must have a unique ID number. The number must be unique across the LOB.
- **Zip Tool** – Public XML files can be zipped by any file zipping tool. The archive format needs to be zip.

- **Naming Conventions** – Attribute/Node naming conventions may be able to accept some special characters. For a complete listing of special characters that are not allowed, please refer to the Special Characters Not Allowed section.
- **The file must be named: INSBRIDGE_PUBLIC.XML** – The file located inside the zip file must be named: INSBRIDGE_PUBLIC.XML. The name of the zip file can be any name you choose.
- **The zip file name must be unique** – The zip file that users import into RateManager must have a unique name across all lines of business and sublines. For example, if a zip file for the AUTO LOB is named 'oct_2012' then a zip file for the HOME LOB cannot also be named 'oct_2012'. Make zip file names unique. The AUTO LOB could be named 'auto_oct_2012_v1.2' and the HOME LOB could be named "home_oct_2012_v1.6'.

Updated Files

Files can be re-imported in to RateManager only if the original file is deleted from the Public XML tab or the updated file has a new name. An updated file cannot be applied to an existing program, only to a new program. This means that even though the file is updated, a new program will be created. No new program versions will be created. Users will have the option to create the new program in the same subline or a new subline.

If the file is applied to the same subline as the earlier file, updates to global elements may not be recognized. If the element ID number is found in the database, the element is not updated or over-written. The element will not be updated, it will be ignored. No new element revisions or program versions will be created.

If the file is applied to a new subline, then the program and elements will be created as expected.

When a Public XML file is updated, it is recommended that global elements not be updated/changed or instruct the RateManager users to apply the program to a new subline. A new subline will create the program and elements as expected.

Notice

- Linked variables are not supported. Any linked variables will have to be manually entered.
- Driver Assignment is not supported. Any driver assignment will have to be manually entered.
- Ranking step types are not supported. Any ranking steps will have to be manually entered.
- Advanced Option –Algorithm Looping is not supported. Any advanced option-algorithm looping will have to be manually entered.

High Level View of the XML Format

Insbridge uses a standard XML format to import rating program elements.

The top level program node holds the program definitions. The second level node is the line of business node that defines where the program is located.

The next node level is the global node where global elements are located. Categories, inputs and results are located under the global node with inputs and results located under the category node. Other elements, such as table variables, calculated variables, algorithms and driver assignments can be placed at the global level or at the program level.

The program node is on the same level as the global node and holds elements along with table data. This node also holds the program level elements sequence and result groups. All table data, even for global level table variables, is entered under the program node.

Calculated variables, algorithms and driver assignments can be detailed out with step nodes and term nodes. Table variables can be detailed out with criteria nodes.

HIGH LEVEL EXAMPLE

```
<OIRUProgramImport effectiveDate="" versionID="" versionDescription="">
     <LineOfBusiness name="">
         <Global>
            <Categories>
                 <Inputs />
                 <Results />
            </Categories>
            <TableVariables>
                 <Criteria />
            </TableVariables>
            <CalculatedVariables>
                 <Steps />
            </CalculatedVariables>
            <Algorithms>
                 <Steps />
            </Algorithms>
         </Global>
         <Program>
            <TableVariables>
                 <Criteria />
            </TableVariables>
            <TableData />
            <CalculatedVariables>
                 <Steps />
            </CalculatedVariables>
            <Algorithms>
                 <Steps />
            </Algorithms>
            <Sequence />
            <ResultGroups />
```

```
        </Program>
    </LineOfBusiness>
</OIRUProgramImport>
```

OIRUPROGRAMIMPORT NODE

The OIRUProgramImport node is required to identify the XML as a Public XML format file. All attributes are required.

Prior to saving, please name the file INSBRIDGE_PUBLIC.XML. The file locatedmust be named: INSBRIDGE_PUBLIC.XML.

OIRUProgramImport Attributes

The information required to define an Insbridge program is:

- **effectiveDate** [required] – the effective date for this program version.
- **versionDescription** [required] – a description of the program version.
- **version ID** [required] – a numeric ID of the program version. This is an information field and is visible in the XML file only. When the file is applied, the version number will be defaulted to 1.

TYPE	NAME	OCCURANCE	REQUIRED	RESTRICTIONS/FORMATS
Node	OIRUProgramImport	1	Yes	-
Attribute	effectiveDate	-	Yes	Two types of Date Formats are allowed (i.e., YYYY-MM-DD & MM/DD/YYYY)
Attribute	versionDescription	-	Yes	String type, spaces allowed
Attribute	versionID	-	Yes	String type, numeric only

XML EXAMPLE

```
<OIRUProgramImport effectiveDate="01-01-2013" versionDescription="Jan V01"
versionID="1">

   …

</OIRUProgramImport>
```

Where: … indicates XML contents.

LINEOFBUSINESS NODE

The LineOfBusiness node is required to identify the line of business where the program needs to be located. The name attribute is required and must exactly match the line of business name in RateManager. The line of business (LOB) ID number in RateManager is not used. Prior to creating the XML, verify the exact spelling of the line of business in RateManager. The entry is not case sensitive.

LineOfBusiness Attributes

The information required to define the line of business used for the Insbridge program is:

- **name** [required] – The line of business name in RateManager.

TYPE	NAME	OCCURANCE	REQUIRED	RESTRICTIONS/FORMATS
Node	LineOfBusiness	1	Yes	-
Attribute	name	-	Yes	String type. The name must exactly match a Line of Business name within RateManager. The entry is NOT case sensitive.

XML EXAMPLE

```
<OIRUProgramImport effectiveDate="01-01-2013" versionDescription="Jan V01"
versionID="1">
   <LineOfBusiness name="Home">

      …

   </LineOfBusiness>
</OIRUProgramImport>
```

Where: … indicates XML contents.

GLOBAL NODE

The Global node is required to identify the global elements used in the program. There are no attributes for this node. This node is required for categories and other global level elements.

Global elements can be used in more than one program. If the global elements in one Public XML file matches the global elements in another Public XML file, no action will be done. If there is a change to the global element, a new element revision may be created.

TYPE	NAME	OCCURANCE	REQUIRED	RESTRICTIONS/FORMATS
Node	Global	1	Yes	-

XML EXAMPLE

```
<OIRUProgramImport effectiveDate="01-01-2013" versionDescription="Jan V01"
versionID="1">
   <LineOfBusiness name="Home">
      <Global>

      …

      </Global>
   </LineOfBusiness>
</OIRUProgramImport>
```

Where: … indicates XML contents.

CATEGORIES NODE

The Categories node is required to identify the types of items on which information is input and output of the system (to/from rules and/or rating) and their relationships (parent, child, or sibling). Insbridge defines a category as a group of information that is specific to an item. For example, an auto policy would have categories such as Policy, Driver and Vehicle. A home policy would have Policy and Dwelling categories.

Insbridge categories are defined within a line of business at a 'global' level. The 'root' of the Insbridge category hierarchy structure is 'Policy'. Policy is a system level category and must be entered as the top level category in the Public XML file. The policy level category will be matched up in RateManager. If Policy is not found, the file may fail to be applied in RateManager.

System created categories for the Auto LOB must also be entered as well; driver, vehicle, and driver-vehicle. Categories are defined in a nested fashion. If the category is at the policy level, then create the entry at the same level as Policy. If the category is a child of another category, for example, accidents may be defined as a child category of vehicle, then you would create the accidents category under the vehicle category.

Categories are required for all programs. Make note of the category id and name. These attributes may be required in other elements.

NOTE: *Categories are always under the Global node. Categories cannot be created under the program node.*

Category Attributes

The information required to define an Insbridge category is:

- **name** [required] – a category name that should clearly describe the item and its intent (e.g. Vehicles, Locations, Buildings, Dwelling, Coverages, Forms, etc.). Policy must be entered as the top level category.
- **id** [required] – a consistent, universally unique identifier.
- **dateModified** – the date the category definition was last modified. The date must be in yyyy-mm-dd format. If the dateModifed attribute is not included, the default is the date the file was imported in to RateManager.
- **currentStatus** – the current status of the category in this version of the XML file. If a currentStatus is not included, the apply process looks for the id in the database. If the id exists, the apply process compares the current item defined with the existing id and determines whether the status is NoChange or Updated. If the id does not exist in the database, the import process sets the status to New.
    - New
    - NoChange (i.e., no changes for this release, program version, etc.)
    - Updated
    - Deleted

TYPE	NAME	OCCURANCE	REQUIRED	RESTRICTIONS
Node	Categories	1	Yes	-
Node	Category	Unbounded	Yes	-
Attribute	name	-	Yes	String type. Does not allow some special characters.
Attribute	id	-	Yes	Allows String type. This id will be used by other elements in the XML file.
Attribute	dateModified	-	No	Two types of Date Formats are allowed (i.e., YYYY-MM-DD & DD/MM/YYYY)
Attribute	currentStatus	-	No	New/NoChange/Updated/Deleted

XML EXAMPLE

```
…

   <Categories>
      <Category name="Policy" id="A43375C0-8899-4B45-B0E1-697D60661BD1"
dateModified="9-1-2012" currentStatus="New"/>
       <Category name="Dwelling" id="0C70E248-EC76-4799-B7F8-
24EEA4D4F065" dateModified="8-1-2012" currentStatus="New">
       </Category>
      </Category>
   </Categories>

…
```

Where: … indicates XML contents.

System Defined Auto LOB Categories

The Auto line of business has system-defined categories and inputs that are related to driver assignment functionality. Driver assignment is not supported for Public XML in release 4.7. Driver assignments can be created manually inside RateManager. If driver assignment is being used, these categories must be entered in the categories section of the XML file. If driver assignment is not being used, please be aware that these category names should **NOT** be used for any category created in Public XML.

	Category Name	XML ID	XML Path
0	Policy	0	Policy
1	Driver	1	Driver
2	Vehicle	2	Vehicle
3	Driver-Vehicle	3	Driver-Vehicle

INPUT NODE

The Input node is not required. However if inputs are being used by other elements in the XML, the file may fail to be applied. Inputs are the specific data elements needed to initiate the rating process. Inputs are those values that cannot be derived by other means. Examples of inputs could include Effective Date, Policy Term, Coverage Limits, Year Built, Construction Type, Date of Birth, Postal Code, and so on.

Examples of values that could be derived from other means are:

- Expiration Date if the effective date and policy term are known.
- Policy Term if the effective date and expiration date are known.
- Age if the date of birth and either effective date or today's date are known.
- Territory Code if a postal code is known.

Inputs and results are located under the categories where they are to be created. For example, if there is an expiration date as an input at the policy level, then the policy category would have the input - expiration date - located underneath it.

Input Attribute

The information required to define an Insbridge input is:

- **name** [required] – the input name, which should clearly describe the item and its intent. For example, EffectiveDt, Limit, YrBuilt, ConstructionCd, BirthDt, etc.

- **id** [required] – a consistent, universally unique identifier.

- **dateModified** – the date the category definition was last modified. The date must be in yyyy-mm-dd format. If the dateModifed attribute is not included, the default is the date the file was imported in to RateManager.

- **dataType** [required] – the type of data expected in this input.

    - String
    - Decimal
    - Integer
    - Date

- **currentStatus** – the current status of the input in this version of the XML file. If a currentStatus is not included, the apply process looks for the id in the database. If the id exists, the apply process compares the current item defined with the existing id and determines whether the status is NoChange or Updated. If the id does not exist in the database, the import process sets the status to New.

    - New
    - NoChange (i.e., no changes for this release, rules, circular)
    - Updated
    - Deleted

TYPE	NAME	OCCURANCE	REQUIRED	RESTRICTIONS
Node	Input	Unbounded	Yes	-
Attribute	Name	-	Yes	String type. Does not allow some special characters.
Attribute	Id	-	Yes	Allows String type
Attribute	dateModified	-	No	Two types of Date Formats are allowed (i.e., YYYY-MM-DD & DD/MM/YYYY)
Attribute	dataType	-	Yes	String/Decimal/Integer/Date
Attribute	currentStatus	-	No	New/NoChange/Updated/Deleted

XML EXAMPLE

```
…

            <Input name="BusnPrsnlPropLimit" id="1C977EA3-8CD1-4E59-B7C9-
64465428228E" dateModified="9-1-2012" dataType="Integer"
currentStatus="New"/>

…

Where: … indicates XML contents.
```

System Defined Auto LOB Inputs

The Auto line of business has system-defined categories and inputs that are related to driver assignment functionality. Driver assignment is not supported for Public XML in release 4.7. Driver assignments can be created manually inside RateManager. If driver assignment will be used, the following inputs should be defined in the defined category section of the XML file. If driver assignment is not being used, please be aware that these input names should **NOT** be used for any input created in Public XML.

	Name	Data Type	Category	XML ID
1	AssignedDriver	String	Vehicle	17
2	AssignmentOverride	String	Vehicle	400
3	DriverId	String	Driver	367
4	DrvUsePctOnVeh1	String	Driver	9
5	DrvUsePctOnVeh2	String	Driver	10
6	DrvUsePctOnVeh3	String	Driver	11
7	DrvUsePctOnVeh4	String	Driver	12
8	DrvUsePctOnVeh5	String	Driver	13
9	DrvUsePctOnVeh6	String	Driver	14

Name	Data Type	Category	XML ID
DrvUsePctOnVeh7	String	Driver	15
DrvUsePctOnVeh8	String	Driver	16
DrvUsePctOnVeh9	String	Driver	50
PrincipalOperatorInd	String	Driver	253
VehicleID	Integer	Vehicle	368
VehPrincipallyDriven	Integer	Driver	299
VehUsePctOnDrv1	String	Vehicle	1
VehUsePctOnDrv2	String	Vehicle	2
VehUsePctOnDrv3	String	Vehicle	3
VehUsePctOnDrv4	String	Vehicle	4
VehUsePctOnDrv5	String	Vehicle	5
VehUsePctOnDrv6	String	Vehicle	6
VehUsePctOnDrv7	String	Vehicle	7
VehUsePctOnDrv8	String	Vehicle	8
VehUsePctOnDrv9	String	Vehicle	700

RESULTVARIABLES NODE

The ResultVariable node is not required. However if results are being used by other elements in the XML, the file may fail to be applied. Result Variables generally represent the final result of a calculation, usually in an algorithm but can also be used as placeholders in calculation steps, including arithmetic, and string values.

Results are located under the categories where they are to be created. For example, if you have a 'total premium' as a result at the policy level, then you would create the policy category with the result total premium located underneath it.

Result Variables are included in the category structure, allowing for the category to be derived from the XML hierarchy.

ResultVariable Attributes

The information required to define an Insbridge result variable is:

- **name** [required] – the variable name that clearly describes the item and its intent. For example, TotalTerrorismPremium, GeneralLiabilityClassificationPollutionCoveragePremium, etc.
- **id** [required] – a consistent, universally unique identifier.
- **dateModified** – the date the result definition was last modified. The date must be in yyyy-mm-dd format. If the dateModifed attribute is not included, the default is the date the file was imported in to RateManager.
- **currentStatus** – the current status of the result in this version of the XML file. If a currentStatus is not included, the apply process looks for the id in the database. If the id exists, the apply process compares the current item defined with the existing id and determines whether the status is NoChange or Updated. If the id does not exist in the database, the import process sets the status to New.
    - New
    - NoChange (i.e., no changes for this release, rules, circular)
    - Updated
    - Deleted
- **dataType** [required] – the type of data expected in this result variable.
    - String
    - Decimal
    - Integer
    - Date

TYPE	NAME	OCCURANCE	REQUIRED	RESTRICTIONS
Node	ResultVariable	Unbounded	No	-
Attribute	Name	-	Yes	String type. Does not allow some

				special characters.
Attribute	Id	-	Yes	Allows String type
Attribute	dateModified	-	No	Two types of Date Formats are allowed (i.e., YYYY-MM-DD & DD/MM/YYYY)
Attribute	currentStatus	-	No	New/NoChange/Updated/Deleted
Attribute	dataType	-	Yes	String/Decimal/Integer/Date

XML EXAMPLE

…

```
<ResultVariable name="TotalPremium" id="CB6DAABA-5647-4D94-B242-E377DEEB4F7F" dateModified="8-1-2012" currentStatus="New" dataType="Decimal"/>
```

…

Where: … indicates XML contents.

PROGRAM NODE

The Program node is not required unless local elements, table data, sequencing and/or result groups are being used. There are no attributes for this node.

The program node identifies the elements used at the local level. Local elements are used in only one program. The local level can contain table variables, calculated variables, driver assignments, algorithms, sequencing and result groups.

All table data for every table variable used in the program is located in the table data node found in the program node. Even for global level table variables. Table data is separted by table variable name. Sequencing and result groups are also at the local level. By default any sequence listed will be checked as used.

TYPE	NAME	OCCURANCE	REQUIRED	RESTRICTIONS/FORMATS
Node	Progam	1	No	-

XML EXAMPLE

```
<OIRUProgramImport effectiveDate="01-01-2013" versionDescription="Jan V01"
versionID="1">
   <LineOfBusiness name="Home">
      <Global>

      …

      </Global>
      <Program>

      …

      </Program>
   </LineOfBusiness>
</OIRUProgramImport>
```

Where: … indicates XML contents.

TABLEVARIABLE NODE

The TableVariable node is not required but can be used to import large tables. Insbridge uses table variables to perform value lookups based on one or more criteria. Items represented as tables in rating manuals, for example Territory Base Premium, Form Factor, Protection/Construction Factor, Key Factor, etc., can be converted to Insbridge table variables that are used by other variables including calculated variables, algorithms, as well as other table variables.

Local table varibles are located under the program node and global table variables are located under the global node. All table data, even for a global table variable will be located in a single table data node under the program node.

Table variables can have up to 10 criteria. Criteria are defined in a criteria node directly under the table variable node. Under the criteria node is a term node that defines the how lookup is to be performed.

CRITERIA	CRITERIA	CRITERIA	CRITERIA	Table Var	Element type
PolicyTypeCd	ConstructionCd	EQTerritoryCd	EQDeductiblePct	EQCov	Element name
=	=	=	=		Term Value
02	F	21	5	0.21	Data row
02	MY	21	5	0.23	Data row
02	R	21	5	0.25	Data row
02	SMNC	21	5	0.27	Data row
02	SNC	21	5	0.29	Data row
02	V	21	5	0.31	Data row

DATA

In this example, the table variable, EQCov, has four criteria; PolicyTypeCd, ConstructionCd, EQTerritoryCd and EQDeductiblePct. The term value for each of the criteria is equals (=). The remaining rows are the data lookup values.

In order to lookup the the value of the table variable EQCov, a value would have to be entered for each of the four criteria. If values were entered for the PolicyTypeCd = 02, the ConstructionCd=R, the EQTerritoryCd = 21 and the EQDeductiblePct=5, RateManager would return a value of 0.25 for the EQCov table variable.

NOTE: *For release 4.7, Linked variables are not supported. Any linked variables will have to be manually entered.*

TableVariable Attributes

Table variables can be defined at either a **Global** or **Local** level. Local table variables can use global variables, but global table variables cannot use local variables.

Prior to creating a table variable, the exact category ID must be obtained. This is the category id as defined in the XML file.

The information required to define an Insbridge table variable are:

- **name** [required] – the table variable name that clearly describes the item and its intent.
- **id** [required] – a consistent, universally unique identifier.
- **dateModified** – the date the table variable definition was last modified. The date must be in yyyy-mm-dd format. If the dateModifed attribute is not included, the default is the date the file was imported in to RateManager
- **currentStatus** – the current status of the table varaible in this version of the XML file. If a currentStatus is not included, the apply process looks for the id in the database. If the id exists, the apply process compares the current item defined with the existing id and determines whether the status is NoChange or Updated. If the id does not exist in the database, the import process sets the status to New.
    - New
    - NoChange (i.e., no changes for this release, rules, circular)
    - Updated
    - Deleted
- **version** – a numeric ID of the table variable revision. This is an information field and is visible in the XML file only. When the file is applied, the revision number will be defaulted to 1.
- **dataType** [required] – the table variable result data type.
    - String
    - Decimal
    - Integer
    - Date
- **defaultValue** – a specified value that complies with the data type. The default value is used if no matching criteria are found.
- **categoryID** [required] – the previously defined numeric id of the category where this table variable will be executed. This is the Insbridge working category. The category and category id must be defined in the XML file.
- **categoryName** – the category ID name where this table variable will be executed. This field is primarily for informational purposes and intended to provide descriptive information for persons creating or reviewing the XML file.

TYPE	NAME	OCCURANCE	REQUIRED	RESTRICTIONS
Node	TableVariables	1	No	-
Node	TableVariable	Unbounded	No	-
Attribute	Name	-	Yes	String type. Does not allow some special characters.

Attribute	Id	-	Yes	Allows String type
Attribute	dateModified	-	No	Two types of Date Formats are allowed (i.e., YYYY-MM-DD & DD/MM/YYYY)
Attribute	currentStatus	-	No	New/NoChange/Updated/Deleted
Attribute	Version	-	No	Integer type
Attribute	dataType	-	Yes	String/Decimal/Integer/Date
Attribute	defaultValue	-	No	The value should comply with the defined dataType.
Attribute	categoryID	-	Yes	Allows String type
Attribute	categoryName	-	No	String type. Does not allow some special characters.

Criteria and Term Attributes

Table variable **Criteria** must include:

- **criterionNumber** [required] – an integer value (1-10) that indicates the order in which this criteria definition appears in the table.

NOTE: *A maximum of ten criteria can be added to a table variable.*

The criterion **Terms** – that define the lookup value(s). Each **Term** is defined by:

- **termType** [required] – describes the purpose of the term. Table variables can include these special types:
  - **mask** – allows users 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, the masking function can be used to read only the first five characters of the value. Mask can only be applied to String and Date Criteria data types. Masking is described in more detail in Masking.
  - **wildcard** – a binary (1,0) yes/no indicator used when certain table data is only relevant in specific cases. For example, if Territory Code is determined by Garaging Zip and, in some cases, Residence Type, a value or an empty field could be used in the Residence Type 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. If not included, the default value will be no (0)
  - **interpolate** – a binary (1,0) yes/no indicator used when interpolation should be applied in table value lookups. For example, if the key factor table contains values for a dwelling limits of $10,000 (2.00) and $20,000 (3.00), use interpolation to determine the key factors for limits that are not specified in the table. If not included, the default value will be no (0)

Interpolation Examples

Coverage Limit	**$15,000**	**$13,500**
Table Lower Limit	$10,000	$10,000
Table Next Higher Limit	$20,000	$20,000
*Difference / 1000	10	10
Lower Limit Key Factor	2.000	2.000
Next Higher Limit Key Factor	3.000	3.000
Upper/Lower Key Factor Difference	1.000	1.000
÷ by High/Low Limit *Difference / 1000	0.100	0.100
x Coverage Limit - Lower Limit #of 1000's	5.000	3.500
Key Factor Increase	0.500	0.350
+ Lower Key Factor	2.000	2.000
= Interpolated Key Factor	**2.500**	**2.350**

- **value** [required] – the value of the termtype selected. Varies depending on the term type.
- **valueName** – describes the value. This field is primarily for informational purposes intended to provide descriptive information for persons creating or reviewing the XML file.

Term types and values are described in more detail in the Term Types/Codes section.

TYPE	NAME	OCCURANCE	REQUIRED	RESTRICTIONS
Node	Criteria	1	No	-
Node	Criterion	maxOccurance=10	No	-
Attribute	criterionNumber	-	Yes	Integer Type
Node	Term	Unbounded	No	-
Attribute	termType	-	Yes	Integer Type
Attribute	value	-	Yes	String Type
Attribute	valueName	-	No	String Type

XML EXAMPLE

```
…

   <TableVariables>
      <TableVariable name="AccountsReceivableLimit" id="47D92217-3E01-
40D1-A2B0-A19D875E43CF" dateModified="9-1-2012" dataType="Integer"
currentStatus="New" version="1" defaultValue="" categoryID="A43375C0-8899-
4B45-B0E1-697D60661BD1" categoryName="Policy">
         <Criteria>
            <Criterion criterionNumber="1">
               <Term termType="Input" value="1C977EA3-8CD1-4E59-B7C9-
64465428228E" valueName="BusnPrsnlPropLimit"/>
               <Term termType="operator" value="5" valueName="equals"/>
            </Criterion>
            <Criterion criterionNumber="2">
               <Term termType="Input" value="3D82F49D-A52C-4A97-AF6C-
227278511F3C" valueName="FunctlBusnPrsnlPropValtnApply"/>
               <Term termType="operator" value="5" valueName="equals"/>
            </Criterion>
         </Criteria>
      </TableVariable>

…
```

Where: … indicates XML contents.

TABLEDATA NODE

The TableData node contains the 'lookup' values that are retrieved from table variables in the rating/rules execution process. There is only one table data node and it is located in the program node.

The table data node contains the name and id of the table variable where the data is to be placed. This information can be obtained from the table variable node defined in the XML file. Once the table variable is named, a row node and data node are placed underneath. The row node is a count of the number of rows to be entered. The data node contains the data.

The table data node contains all the table data for every table in the XML file. Each entry is separated by a table variable node that defined that table where the data is to be placed.

TableData Attributes

The information required to define an Insbridge table data are:

- **name** – the previously defined table variable name. This field is primarily for informational purposes intended to provide descriptive information for persons creating or reviewing the XML file.
- **id** [required] – the previously defined universally unique identifier of the target table.

Row and Data Attributes

The Row attribute identifies the previously defined target table variable, and criteria. Each **Row** is defined by:

- **num** – the number of the row in the order they should be in inserted into the table.

The Data attribute represents each data element in a cell (intersection of row/column) in the table variable. Each **Data** is defined by:

- **columnName** – the previously defined name of the target column. This field is primarily for informational purposes intended to provide descriptive information for persons creating or reviewing the XML file.
- **columnNum** [required] – the numeric value that represents the order that the previously defined table variable, and criteria appear in the table.
- **id** [required] – the previously defined universally unique identifier of the target

TYPE	NAME	OCCURANCE	REQUIRED	RESTRICTIONS
Node	TableData	1	No	-
Node	TableVariable	1	No	-

Attribute	name	-	No	String type. Does not allow some special characters.
Attribute	id	-	Yes	Allows String type
Node	Row	Unbounded	No	-
Attribute	num	-	No	Integer Type
Node	Data	Unbounded	No	-
Attribute	columnName	-	No	String type. Does not allow some special characters.
Attribute	columnNum	-	Yes	Integer Type
Attribute	id	-	Yes	Allows String Type

XML EXAMPLE

…

```
        <TableData>
           <TableVariable name="AccountsReceivableLimit" id="47D92217-3E01-
40D1-A2B0-A19D875E43CF">
              <Row num="1">
                 <Data columnNum="1" columnName="AccountsReceivableLimit"
id="">10000 through 999999999</Data>
                 <Data columnNum="2" columnName="BusnPrsnlPropLimit" id="">1
through 999999999</Data>
                 <Data columnNum="3"
columnName="FunctlBusnPrsnlPropValtnApply" id="">Yes</Data>
              </Row>
              <Row num="2">
                 <Data columnNum="1" columnName="AccountsReceivableLimit"
id="">10000 through 999999999</Data>
                 <Data columnNum="2" columnName="BusnPrsnlPropLimit" id="">1
through 999999999</Data>
                 <Data columnNum="3"
columnName="FunctlBusnPrsnlPropValtnApply" id="">No</Data>
              </Row>
              <Row num="3">
                 <Data columnNum="1" columnName="AccountsReceivableLimit"
id="">10000 through 999999999</Data>
                 <Data columnNum="2" columnName="BusnPrsnlPropLimit"
id="">0</Data>
                 <Data columnNum="3"
columnName="FunctlBusnPrsnlPropValtnApply" id="">Yes</Data>
              </Row>
              <Row num="4">
                 <Data columnNum="1" columnName="AccountsReceivableLimit"
id="">0</Data>
```

```
                    <Data columnNum="2" columnName="BusnPrsnlPropLimit"
id="">0</Data>
                    <Data columnNum="3"
columnName="FunctlBusnPrsnlPropValtnApply" id="">No</Data>
                </Row>
             </TableVariable>
          </TableData>
```

…

Where: … indicates XML contents.

CALCULATEDVARIABLE NODE

The CalculatedVariable node is not required. 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, a calculated variable would be used to calculate driver age from 'effective date' and 'driver date of birth' inputs.

A calculated variable supports advanced logic, such as arithmetic and if/then functions. Logic and functions within a calculated variable are created as steps. These steps are then added together to form a sequence of calculation.

Local calculated varibles are located under the program node and global calculated variables are located under the global node. Each Calculated Variable node contains step node(s) that define the step type and list the step order and term node(s) that defines the action that needs to be performed.

Calculated Variable Attributes

Calculated variables can be defined at either a **Global** or **Local** level. Local calculated variables can use global variables, but global calculated variables cannot use local variables.

The information required to define an Insbridge calculated variable are:

- **name** – the calculated variable name that clearly describes the item and its intent. For example, KeyPremium, 301BasePremium, etc.
- **id** [required] – a consistent, universally unique identifier.
- **dateModified** – the date the calculated variable definition was last modified. The date must be in yyyy-mm-dd format. If the dateModifed attribute is not included, the default is the date the file was imported in to RateManager
- **currentStatus** – the current status of the calculated variable in this version of the XML file. If a currentStatus is not included, the apply process looks for the id in the database. If the id exists, the apply process compares the current item defined with the existing id and determines whether the status is NoChange or Updated. If the id does not exist in the database, the import process sets the status to New.
    - New
    - NoChange (i.e., no changes for this release, rules, circular)
    - Updated
    - Deleted
- **version** – a numeric ID of the calculated variable revision. This is an information field and is visible in the XML file only. When the file is applied, the revision number will be defaulted to 1.
- **dataType** [required] – the calculated variable result data type
    - String
    - Decimal
    - Integer
    - Date

- **categoryID** [required] – the previously defined numeric id of the category where this calculated variable will be executed. This is the Insbridge working category. The category and category id must be defined in the XML file.
- **categoryName** – describes the category ID where this calculated variable will be executed. This field is primarily for informational purposes intended to provide descriptive information for persons creating or reviewing the XML file.

TYPE	NAME	OCCURANCE	REQUIRED	RESTRICTIONS
Node	CalculatedVariables	1	No	-
Node	CalculatedVariable	Unbounded	No	-
Attribute	name	-	No	String type. Does not allow some special characters.
Attribute	id	-	Yes	Allows String type
Attribute	dateModified	-	No	Two types of Date Formats are allowed (i.e., YYYY-MM-DD & DD/MM/YYYY)
Attribute	currentStatus	-	No	New/NoChange/Updated/Deleted
Attribute	version	-	No	Integer type
Attribute	dataType	-	Yes	String/Decimal/Integer/Date
Attribute	categoryID	-	Yes	Allows String type
Attribute	categoryName	-	No	String type. Does not allow some special characters.

Step and Term Attributes

The step and terms are defined by:

The calculated variable Steps. Each **Step** must include:

- **stepNumber** [required] – an integer value that indicates the order in which this step will execute in relation to the other steps in this calculated variable
- **stepType** [required] – codes that define the purpose of the step are described in detail in the Step Types section.
- **stepName** – describes the step type. This field is primarily for informational purposes intended to provide descriptive information for persons creating or reviewing the XML file.

The step Terms - interim operations within each step in order of execution. Each **Term** is defined by:

- **termType** [required] – describes the purpose of the term
- **value** [required] – varies depending on the term type
- **valueName** – describes the value. This field is primarily for informational purposes intended to provide descriptive information for persons creating or reviewing the XML file.

Term types and values are described in more detail in the Term Types/Codes section.

TYPE	NAME	OCCURANCE	REQUIRED	RESTRICTIONS
Node	Steps	1	No	-
Node	Step	Unbounded	No	-
Attribute	stepNumber	-	Yes	Interger Type (Allows Unique Value)
Attribute	stepType	-	Yes	Decimal Type
Attribute	stepName	-	No	String Type
Node	Term	Unbounded	No	-
Attribute	termType	-	Yes	Integer Type
Attribute	value	-	Yes	String Type
Attribute	valueName	-	No	String Type

XML EXAMPLE

…

```
         <CalculatedVariables>
            <CalculatedVariable name="KeyPremium" id="66C33473-AFAB-437B-
8A81-4A1D2D9D5F03" dateModified="9-1-2012" currentStatus="New" version="1"
dataType="Decimal" categoryID="0C70E248-EC76-4799-B7F8-24EEA4D4F065"
categoryName="Dwelling">
               <Steps>
                  <Step stepNumber="1" stepType="5.1" stepName="IF">
                     <Term termType="expression" value="1" valueName="IF"/>
                     <Term termType="Input" value="E4E9B263-8673-4C0E-9341-
73FA350586A6" valueName="RatingPropertyType"/>
                     <Term termType="operator" value="5" valueName="equals"/>
                     <Term termType="constant" value="2" valueName="HO Form
2"/>
                     <Term termType="expression" value="4" valueName="OR"/>
                     <Term termType="Input" value="E4E9B263-8673-4C0E-9341-
73FA350586A6" valueName="RatingPropertyType"/>
                     <Term termType="operator" value="5" valueName="equals"/>
```

```
                        <Term termType="constant" value="3" valueName="HO Form
3"/>
                        <Term termType="expression" value="4" valueName="OR"/>
                        <Term termType="Input" value="E4E9B263-8673-4C0E-9341-
73FA350586A6" valueName="RatingPropertyType"/>
                        <Term termType="operator" value="5" valueName="equals"/>
                        <Term termType="constant" value="5" valueName="HO Form
5"/>
                        <Term termType="expression" value="4" valueName="OR"/>
                        <Term termType="Input" value="E4E9B263-8673-4C0E-9341-
73FA350586A6" valueName="RatingPropertyType"/>
                        <Term termType="operator" value="5" valueName="equals"/>
                        <Term termType="constant" value="8" valueName="HO Form
8"/>
                        <Term termType="expression" value="2" valueName="THEN"/>
                        <Term termType="nextStep" value="2" valueName="Step 2"/>
                        <Term termType="expression" value="3" valueName="ELSE"/>
                        <Term termType="nextStep" value="4" valueName="Step 4"/>
                     </Step>
                     <Step stepNumber="2" stepType="7.1" stepName="Arithmentic">
                        <Term termType="ResultVariable" value="CB6DAABA-5647-
4D94-B242-E377DEEB4F7F" valueName="TotalPremium"/>
                        <Term termType="operator" value="5" valueName="equals"/>
                        <Term termType="Input" value="1C977EA3-8CD1-4E59-B7C9-
64465428228E" valueName="BusnPrsnlPropLimit"/>
                        <Term termType="operator" value="3"
valueName="multiplied by"/>
                        <Term termType="Input" value="375F1931-578B-4C16-B2B9-
7EF5118400CD" valueName="AutomaticIncreasePct"/>
                        <Term termType="round" value="0" valueName="Round to 0
places"/>
                        <Term termType="nextStep" value="3" valueName="Step 3"/>
                     </Step>
                  </Steps>
               </CalculatedVariable>
            </CalculatedVariables>

      …
```

Where: … indicates XML contents.

ALGORITHMS

The Algorithms node is not required. Insbridge Algorithms are most commonly used in determining a premium or checking an underwriting condition. Algorithms use other variables - inputs, table variables, calculated variables, and result variables - to determine a 'final' premium. The algorithm would typically be the last step(s) in a series of calculations.

Local algorithms are located under the program node and global algorithms are located under the global node. Each algorithm node contains step node(s) that define the step type and list the step order and term node(s) that defines the action that needs to be performed.

For example, the Homeowners 'Base Premium' could consist of a series of calculated variables for:

- Rule 301 Premium (Territory Base, Form, Protection/Construction, Key Factor, Number of Families)
- Loss Settlement Premium/Credit
- Ordinance or Law Premium
- Special Personal Property Premium

In this example "Rule 301 Base Premium", "Loss Settlement Premium/Credit", "Ordinance or Law Premium" and "Special Personal Property Premium" could be calculated variables that are used in an algorithm to determine the final result - "Base Premium". Because it will be used in other calculations (HO6 Increased Coverage A, HO4 Ordinance or Law), "Key Premium" could also be a calculated variable that is subsequently used in the "Rule 301 Base Premium" calculated variable.

Items represented as tables in rating manuals (e.g., Territory Base Premium, Form Factor, Protection/Construction Factor, Key Factor, etc.) can be converted to Insbridge table variables, which can be used by other variables as well as algorithms.

Algorithms and calculated variables are very similar in their features and functionality. An algorithm or calculated variable may require several calculations. Each interim operation may be contained in one or more steps, depending upon the expected results. In Insbridge rounding is defined at the step-level, so in the in the Key Premium and 301 Base Premium examples, each operation must be a separate step to enforce whole dollar rounding after each calculation.

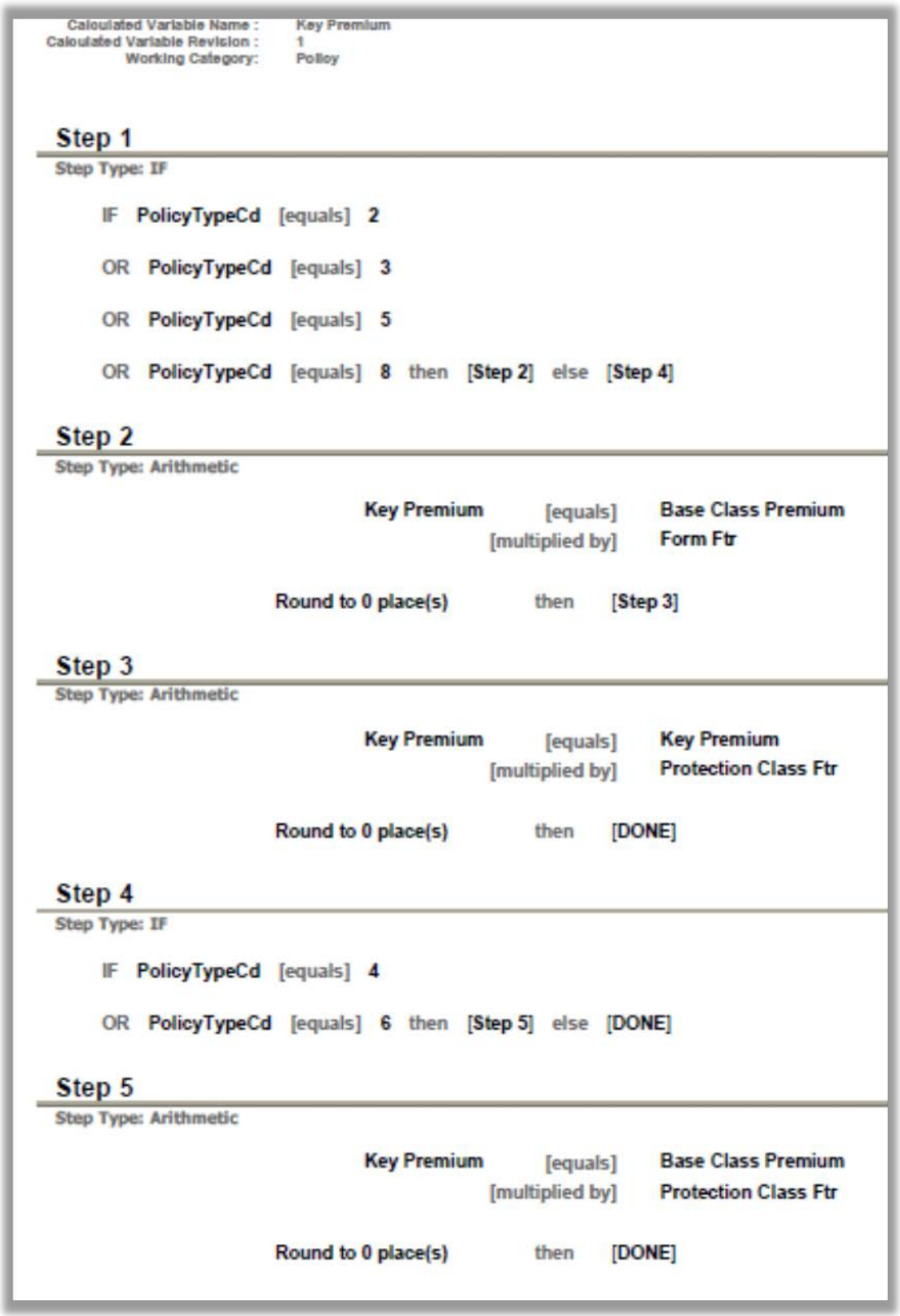
Some calculations may be performed in a single step, such as when rounding applies only at the final operation (or not at all).

SAMPLE HOMEOWNERS BASE PREMIUM - 'MANUAL' COMPUTATION

Territory Base Premium	100.00
x Form Factor	0.98
= (whole dollar rounded - all x HO4,HO6)	98.00
x Protection/Construction Factor	0.95
= Key Premium (whole dollar rounded)	93.00
x Key Factor	**0.6789**
= Base Premium (whole dollar rounded)	63.00
x Number of Families Factor	1.000
= RULE 301 BASE PREMIUM	**63.00**
Special Loss Settlement Factor	0.95
x Base Premium = Loss Settlement Adjusted Base Premium	59.85
- Base Premium	63.00
- Base Premium = Loss Settlement Premium/Credit	-3.15
Loss Settlement Premium/Credit	**-3.00**

Ordinance or Law Factor	1.04
x Base Premium = Ordinance or Law Adjusted Base Premium	65.52
- Base Premium	63.00
- Base Premium = Ordinance or Law Premium	2.52
Ordinance or Law Premium	**3.00**
Factor	1.00
x Base Premium = Special Personal Property Adjusted Base Premium	63.00
- Base Premium	63.00
- Base Premium = Special Personal Property Premium	0.00
Special Personal Property Premium	**0.00**
= BASE PREMIUM	**63.00**

SAMPLE HOMEOWNERS BASE PREMIUM - 'INSBRIDGE' COMPUTATION



Calculated Variable Name : Key Premium
Calculated Variable Revision : 1
Working Category: Policy

Step 1

Step Type: IF

IF **PolicyTypeCd** [equals] **2**

OR **PolicyTypeCd** [equals] **3**

OR **PolicyTypeCd** [equals] **5**

OR **PolicyTypeCd** [equals] **8** then **[Step 2]** else **[Step 4]**

Step 2

Step Type: Arithmetic

Key Premium [equals] **Base Class Premium**
[multiplied by] **Form Ftr**

Round to 0 place(s) then **[Step 3]**

Step 3

Step Type: Arithmetic

Key Premium [equals] **Key Premium**
[multiplied by] **Protection Class Ftr**

Round to 0 place(s) then **[DONE]**

Step 4

Step Type: IF

IF **PolicyTypeCd** [equals] **4**

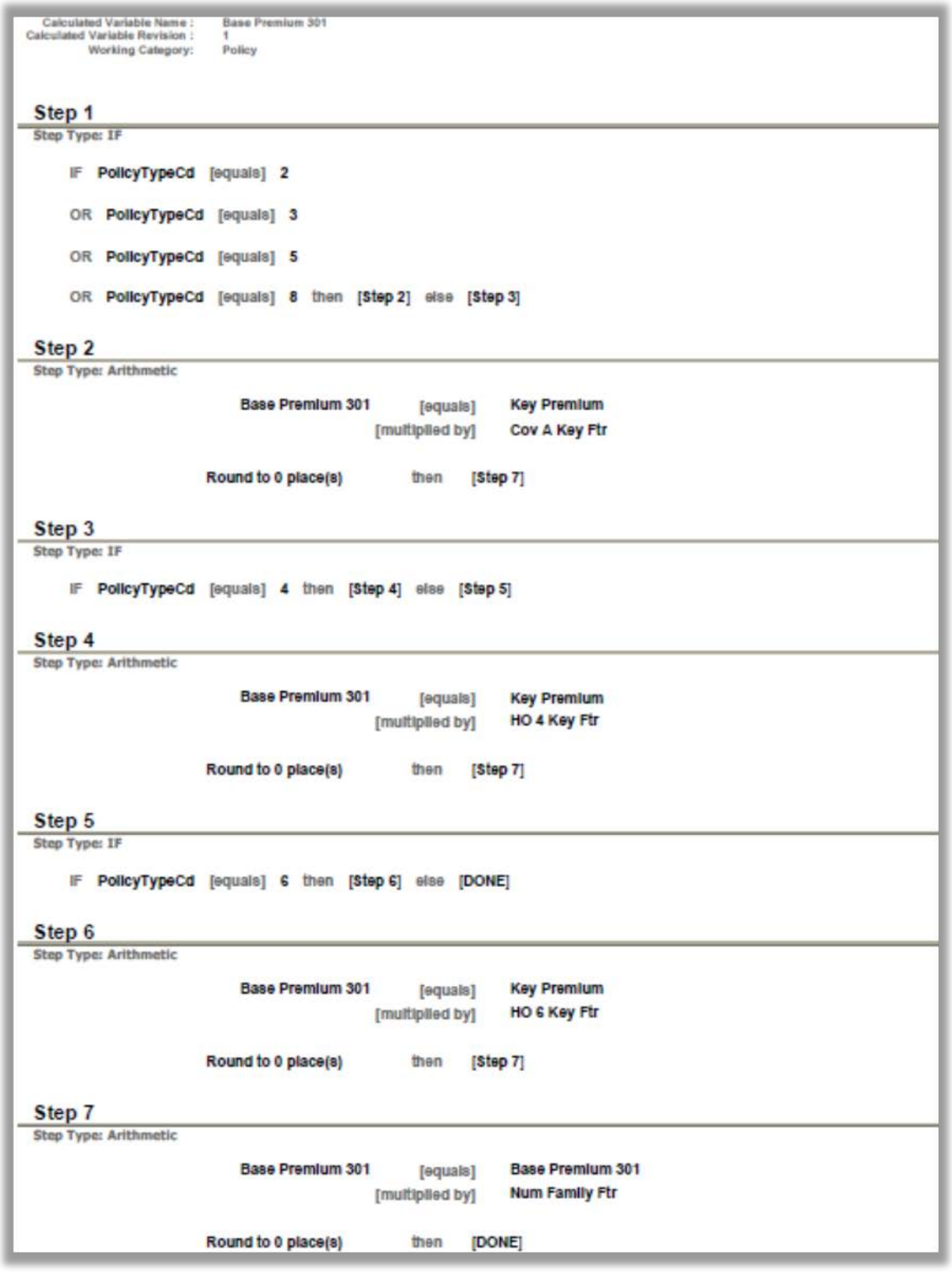
OR **PolicyTypeCd** [equals] **6** then **[Step 5]** else **[DONE]**

Step 5

Step Type: Arithmetic

Key Premium [equals] **Base Class Premium**
[multiplied by] **Protection Class Ftr**

Round to 0 place(s) then **[DONE]**



Calculated Variable Name : Base Premium 301
Calculated Variable Revision : 1
Working Category: Policy

Step 1

Step Type: IF

IF **PolicyTypeCd** [equals] **2**

OR **PolicyTypeCd** [equals] **3**

OR **PolicyTypeCd** [equals] **5**

OR **PolicyTypeCd** [equals] **8** then **[Step 2]** else **[Step 3]**

Step 2

Step Type: Arithmetic

Base Premium 301 [equals] **Key Premium**
[multiplied by] **Cov A Key Ftr**

Round to 0 place(s) then **[Step 7]**

Step 3

Step Type: IF

IF **PolicyTypeCd** [equals] **4** then **[Step 4]** else **[Step 5]**

Step 4

Step Type: Arithmetic

Base Premium 301 [equals] **Key Premium**
[multiplied by] **HO 4 Key Ftr**

Round to 0 place(s) then **[Step 7]**

Step 5

Step Type: IF

IF **PolicyTypeCd** [equals] **6** then **[Step 6]** else **[DONE]**

Step 6

Step Type: Arithmetic

Base Premium 301 [equals] **Key Premium**
[multiplied by] **HO 6 Key Ftr**

Round to 0 place(s) then **[Step 7]**

Step 7

Step Type: Arithmetic

Base Premium 301 [equals] **Base Premium 301**
[multiplied by] **Num Family Ftr**

Round to 0 place(s) then **[DONE]**

Algorithm Attributes

Algorithms can be defined at either a **Global** or **Local** level. Local algorithms can use global variables, but global algorithms cannot use local variables.

The information required to define an Insbridge algorithm are:

- **name** [required] – the algorithm name that clearly describes the item and its intent. For example, TotalPolicyPremium, BasePremium, AdjustedBasePremium, etc.
- **id** [required] – a consistent, universally unique identifier.
- **dateModified** – the date the algorithm definition was last modified. The date must be in yyyy-mm-dd format. If the dateModifed attribute is not included, the default is the date the file was imported in to RateManager.
- **currentStatus** – the current status of the algorithm in this version of the XML file. If a currentStatus is not included, the apply process looks for the id in the database. If the id exists, the apply process compares the current item defined with the existing id and determines whether the status is NoChange or Updated. If the id does not exist in the database, the import process sets the status to New.
    - New
    - NoChange (i.e., no changes for this release, rules, circular)
    - Updated
    - Deleted
- **version** – a numeric ID of the algorithm revision. This is an information field and is visible in the XML file only. When the file is applied, the revision number will be defaulted to 1.
- **dataType** [required] – the algorithm result data type.
    - String
    - Decimal
    - Integer
    - Date

- **algorithmType** [required] – There are two options: Rating or Underwriting
- **categoryID** [required] – the previously defined id of the category where the algorithm will be executed. This is the Insbridge working category. The category and category id must be defined in the XML file.
- **categoryName** – describes the category ID where this algorithm will be executed. This field is primarily for informational purposes intended to provide descriptive information for persons creating or reviewing the XML file.

TYPE	NAME	OCCURANCE	REQUIRED	RESTRICTIONS
Node	Algorithms	1	No	-
Node	Algorithm	Unbounded	No	-
Attribute	name	-	Yes	String type. Does not allow some special characters.
Attribute	id	-	Yes	Allows String type
Attribute	dateModified	-	No	Two types of Date Formats are allowed (i.e., YYYY-MM-DD & DD/MM/YYYY)
Attribute	currentStatus	-	No	New/NoChange/Updated/Deleted
Attribute	version	-	No	Integer type
Attribute	algorithmType	-	Yes	String type. Does not allow some special characters.
Attribute	dataType	-	Yes	String/Decimal/Integer/Date
Attribute	categoryID	-	Yes	Allows String type
Attribute	categoryName	-	No	String type. Does not allow some special characters.

Step and Term Attributes

The algorithm **Steps**. Each **Step** must include:

- **stepNumber** [required] – an integer value that indicates the order in which this step is executed in relation to the other steps in this algorithm.
- **stepType** [required] – codes that define the purpose of the step are described in the Step Types section.

- **stepName** – describes the step type. This field is primarily for informational purposes intended to provide descriptive information for persons creating or reviewing the XML file.

The step **Terms** - interim operations within each step in order of execution. Each **Term** is defined by:

- **termType** [required] – describes the purpose of the term.
- **value** [required] – varies depending on the term type.
- **valueName** – describes the value. This field is primarily for informational purposes intended to provide descriptive information for persons creating or reviewing the XML file.

Term types and values are described in more detail in the Term Types/Codes section.

TYPE	NAME	OCCURANCE	REQUIRED	RESTRICTIONS
Node	Steps	1	No	-
Node	Step	Unbounded	No	-
Attribute	stepNumber	-	Yes	Interger Type (Allows Unique Value)
Attribute	stepType	-	Yes	Decimal Type
Attribute	stepName	-	No	String Type
Node	Term	Unbounded	No	-
Attribute	termType	-	Yes	Integer Type
Attribute	value	-	Yes	String Type
Attribute	valueName	-	No	String Type

XML EXAMPLE

…

```
        <Algorithms>
           <Algorithm name="TerrorismCoveragePremium" id="2615bb44-ffed-
4b7c-b9e8-fb3769cbac2c" dateModified="9-1-2012" currentStatus="New"
version="1" algorithmType="Rating" categoryID="A43375C0-8899-4B45-B0E1-
697D60661BD1" categoryName="Policy">
              <Steps>
                 <Step stepNumber="1" stepType="7.1" stepName="Arithmetic">
                    <Term termType="StepVariable" value="1"
valueName="Results of Step 1"/>
                    <Term termType="operator" value="5" valueName="equals"/>
                    <Term termType="Input" value="99999999" valueName="-No
Input"/>
                    <Term termType="round" value="2" valueName="Round to 2
places"/>
```

```
                    <Term termType="nextStep" value="DONE"
valueName="DONE"/>
                </Step>
            </Steps>
        </Algorithm>
    </Algorithms>

…
```

Where: … indicates XML contents.

SEQUENCE NODE

The Sequence node defines the order in which algorithms will execute within this specific program. Logically algorithms whose results are used in subsequent algorithms must be executed before their dependents.

Sequencing and result groups are also at the local level and will be under the program node. By default any sequence listed will be checked as used.

Sequence Attributes

The Sequence must be defined at the **Local** program level. The sequence can use global and local algorithms.

The information required to define an Insbridge program sequence is:

- **executionOrder** [required] – a numeric (integer) value that indicates the order in which this algorithm will execute in relation to all other algorithms (Global and Local)
- **algorithmType** – Rating or Underwriting, and - for the auto line of business DAScenario, FlagDriver, FlagVehicle, RankDriver or RankVehicle. This field is primarily for informational purposes intended to provide descriptive information for persons creating or reviewing the XML file.
- **name** – the algorithm name that clearly describes the item and its intent. For example TotalPolicyPremium, BasePremium, AdjustedBasePremium, etc. This field is primarily for informational purposes intended to provide descriptive information for persons creating or reviewing the XML file.
- **id** [required] – the previously defined algorithm unique identifier.
- **version** – the Insbridge version number of the algorithm definition. This should be the algorithm revision that is 'active' for this program version. This field is primarily for informational purposes intended to provide descriptive information for persons creating or reviewing the XML file. All element revisions are 1.
- **activeInd**– a binary (1,0) yes/no value used to indicate whether this algorithm is used in this specific version of the program. If not included, the default value is yes (1).

TYPE	NAME	OCCURANCE	REQUIRED	RESTRICTIONS
Node	Sequence	1	No	-
Node	Algorithm	Unbounded	No	-
Attribute	executionOrder	-	Yes	Interger Type
Attribute	algorithmType	-	No	String type. Does not allow some special characters.

Attribute	name	-	No	String type. Does not allow some special characters.
Attribute	id	-	Yes	Integer Type
Attribute	version	-	No	Integer Type
Attribute	activeInd	-	No	BinaryType

XML EXAMPLE

```
      …

         <Sequence>
            <Algorithm executionOrder="1" algorithmType="Rating"
name="TerrorismCoveragePremium" id="2615bb44-ffed-4b7c-b9e8-fb3769cbac2c"
version="1" activeInd="1"/>
            <Algorithm executionOrder="2" algorithmType="Rating"
name="MathFunctions" id="b661b275-23f7-4782-9bc9-100f6db46f2d" version="1"
activeInd="1"/>
            <Algorithm executionOrder="3" algorithmType="Rating"
name="Category Items" id="63e16d41-b1db-4bc6-8377-b7671ba8ea89" version="1"
activeInd="1"/>
            <Algorithm executionOrder="4" algorithmType="Rating" name="Count
Across" id="8611bffa-12c9-46d3-bbbe-b150f3d77a65" version="1" activeInd="1"/>
         </Sequence>
      …
```

Where: … indicates XML contents.

RESULTGROUPS NODE

The ResultGroups node is not required. Result groups are output mappings that define which variable values will be included in the output file. Results are at the local level and the result group node is under the program node. The result group node contains a variable node that defines the result order.

You can create as many result groups as needed. Result group IDs will be changed by the system. When creating result groups, the active ID defaults to 1, meaning the group is active. If the group has a 0 ID, then the results will not be shown for the entrie group. The same goes for elements. If the ID is 1, the result will be shown, if the result is 0 the result will not be displayed.

ResultGroups Attributes

Result Groups must be defined at the **Local** program level. Mappings can include the values of any global or local variable - input, result, table or calculated variable. This might include premiums, rating factors, driver ages, etc.

The information required to define an Insbridge result group is:

- **name** [required] – the result group name, which should clearly describe the item and its intent (AZHomePolicyResults, etc.)
- **dateModified** – the date the algorithm definition was last modified (yyyy-mm-dd format). If not include the default will be the import date.
- **activeInd** – a binary (1,0) yes/no value used to indicate whether this result group is active in this specific version of the program. If not included, the default value is yes (1).
- **defaultInd** [required] – a binary (1,0) yes/no value used to indicate whether this result group is used when no result group is defined in the rating request input file. Only one result group can be defined as the default.
- **separateDriverVehicleOutputInd** – this binary (1,0) yes/no attribute applies to the auto line of business only and is used to indicate that output for drivers and vehicles will be listed separately in the result XML

Variable Attributes

The information required to define an Insbridge result group variable is:

- **resultOrder** [required] – the order in which the output values will appear in the rating result XML file.
- **name** – the previously defined variable name. This field is primarily for informational purposes intended to provide descriptive information for persons creating or reviewing the XML file.
- **id** [required] – the previously defined variable universally unique identifier (GUID)

- **variableType** – describes whether the type of variable being used as the output. This field is primarily for informational purposes intended to provide descriptive information for persons creating or reviewing the XML file.
    - Input
    - CalculatedVariable
    - TableVariable
    - ResultVariable
- **resultID** – a user-defined value, typically used for identification when mapping outside of Insbridge or clarifying the description of the output. For example users may want to more clearly identify an input whose name is “PostalCode” and an output of “US ZIP Code”.
- **categoryName** – describes the category ID under which this variable is defined. This field is primarily for informational purposes intended to provide descriptive information for persons creating or reviewing the XML file.
- **activeInd** – a binary (1,0) yes/no value used to indicate whether this output is used in this specific result group definition. If not included, the default value is yes (1).

NOTE: *Please note that inactive result groups and variables will not be applied in RateManager.*

TYPE	NAME	OCCURANCE	REQUIRED	RESTRICTIONS
Node	ResultGroups	1	No	-
Node	ResultGroup	Unbounded	No	-
Attribute	name	-	Yes	String type. Does not allow some special characters.
Attribute	dateModified	-	No	Two types of Date Formats are allowed (i.e., YYYY-MM-DD & DD/MM/YYYY)
Attribute	activeInd	-	No	Binary Type. Default value = 1 (YES)
Attribute	defaultInd	-	Yes	Binary Type. Only one group can be default.
Attribute	outputCode	-	No	String type. Does not allow some special characters.
Node	Variable	Unbounded	No	-
Attribute	resultOrder	-	Yes	Integer Type
Attribute	name	-	No	String type. Does not allow some special characters.
Attribute	id	-	Yes	String Type
Attribute	variableType	-	No	String type. Does not allow some special characters.

Attribute	resultID	-	No	String type. Does not allow some special characters.
Attribute	categoryName	-	No	String type. Does not allow some special characters.
Attribute	categoryID	-	No	String type. Does not allow some special characters.
Attribute	activeInd	-	No	Binary Type

XML EXAMPLE

```
…

   <ResultGroups>
      <ResultGroup name="RatingResults" dateModified="2012-09-26"
activeInd="1" defaultInd="1" outputCode="12AB3456789">
         <Variable resultOrder="1" name="TotalPremium" id="CB6DAABA-
5647-4D94-B242-E377DEEB4F7F" variableType="ResultVariable" resultID="Total
Premium" categoryID="A43375C0-8899-4B45-B0E1-697D60661BD1"
categoryName="Policy" activeInd="1"/>
         <Variable resultOrder="2" name="AggregateLimit" id="586C578C-
F98F-4D31-838C-C1125F4DA9DD" variableType="Input" resultID="Aggregate Limit"
categoryID="A43375C0-8899-4B45-B0E1-697D60661BD1" categoryName="Policy"
activeInd="1"/>
         <Variable resultOrder="3" name="KeyPremium" id="66C33473-AFAB-
437B-8A81-4A1D2D9D5F03" variableType="CaculatedVariable" resultID="Key
Premium" categoryID="0C70E248-EC76-4799-B7F8-24EEA4D4F065"
categoryName="Dwelling" activeInd="1"/>
      </ResultGroup>
   </ResultGroups>
…
```

Where: … indicates XML contents.

NOTES

The Notes node is optional and can be placed anywhere in the XML file. Notes can be included with any item in the import XML (Program, Category, Input, Algorithm, etc.) to convey additional information or instructions related to the content.

Notes Attributes

The information required to define an Insbridge note is:

- **id** [required] – a consistent, universally unique identifier.
- **dateModified** – the date the note definition was created or last modified (yyyy-mm-dd format). If not include the default will be the import date.
- **author** [required] – the source of the note. This can be a user name or comment.
- **attachmentInd** – a binary (1,0) yes/no value used to indicate whether an attachment is included. The default is No (0).
- **attachmentName** – a string value used to indicate the name of the attachment.
- **description** [required] – the contents of the note (max 200 characters)

TYPE	NAME	OCCURANCE	REQUIRED	RESTRICTIONS
Node	Notes	1	No	-
Node	Note	Unbounded	No	-
Attribute	id	-	Yes	Allows String Type
Attribute	dateModified	-	No	Two types of Date Formats are allowed (i.e., YYYY-MM-DD & DD/MM/YYYY)
Attribute	author	-	Yes	String type. Does not allow some special characters.
Attribute	attachmentInd	-	No	Binary Type. 0=NO, 1=Yes
Attribute	attachmentName	-	No	String Type
Attribute	Description	-	Yes	String Type

XML EXAMPLE

```
…

   <Notes>
      <Note id="8203DC7E-5AF0-4C48-A17B-A7D52D8BBB3C"
dateModified="2012-09-25" author="Public XML Admin" attachmentInd="0"
description="This is a new rating program introducing Public XML content.">
   </Notes>

…
```

Where: … indicates XML contents.

Adding an Attachment

Attachments must be placed in the zip file at the same level as the XML file. No folders are allowed. The attachment name must be entered exactly in the attachmentName field. For example, if you place a file called ‘change_order_1234a.doc’ in the zip file, the Notes node will look like this:

XML EXAMPLE

```
…

   <Notes>
      <Note id="8203DC7E-5AF0-4C48-A17B-A7D52D8BBB3C"
dateModified="2012-09-25" author="Public XML Admin" attachmentInd="0"
description="This is a change order for rating program 1234a."
attachmenName="change_order_1234a.doc">
   </Notes>
…
```

Where: … indicates XML contents.

WORKING WITH PUBLIC XML FILES

Once the Public XML file has been created, it must be zipped. RateManager users can only import zipped files on the Public XML tab.

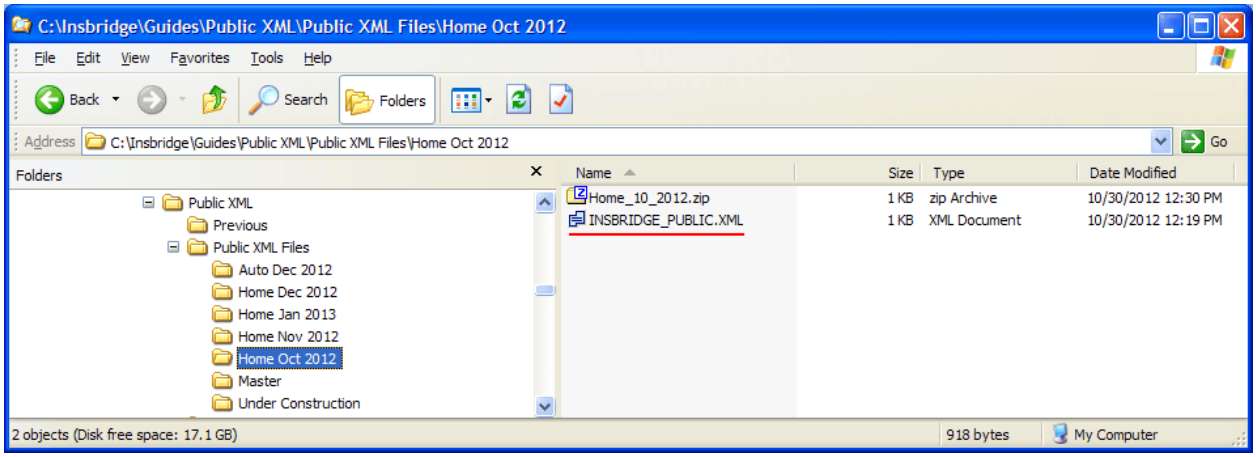
ZIPPING THE FILE

There are a few rules when zipping the XML file:

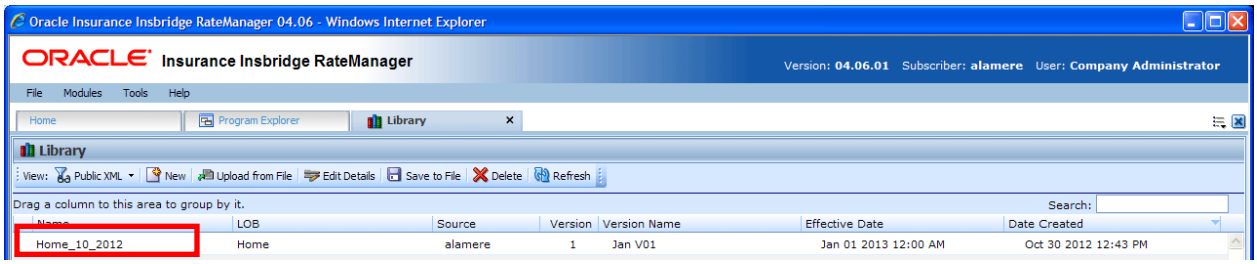
1. The file located inside the zip file must be named: INSBRIDGE_PUBLIC.XML. The name of the zip file can be any name you choose. The zip file name ius the name displayed in RateManager.

2. The file must be on the same level and cannot be in a folder.

3. The zip file name must be unique across the lines of business and sublines. Duplicate names will be rejected in RateManager.



Zip files are the only file type that will be accepted in the Public XML tab. The zip file can use any name format. Once uploaded, the name of the file is the name of the zip file. This can be edited.



RESTRICTIONS / XML SCHEMA

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

Most name fields - variables, algorithms, inputs, etc., are limited to 40 characters including spaces and most cannot contain special characters. Most description fields, notes, change control justifications, are limited to 200 characters including spaces and most cannot contain special characters. If you enter a special character you may receive an error message.

Special Characters Not Allowed

Name	Character		Name	Character
Ampersand	&		Less-Than Sign	<
Apostrophe	'		Percent Sign	%
Asterisk	*		Period	.
At Sign	@		Plus Sign	+
Backslash	\		Pound Sign	#
Caret	^		Question Mark	?
Colon	:		Quotation Mark	"
Comma	,		Right Brace	}
Dollar Sign	$		Right Parentheses	)
Equals Sign	=		Semicolon	;
Exclamation Mark	!		Slash	/
Greater-Than Sign	>		Tick	`
Left Brace	{		Tilde	~
Left Parentheses	(		Vertical Bar	|

Special Characters Allowed

These characters are allowed in any field:

Name	Character
Left Bracket	[
Minus Sign / Dash	-
Right Bracket	]
Underscore	_

In addition to the allowed special characters, the Notes field also allows:

Name	Character
Colon	:
Comma	,
Left Parentheses	(
Period	.

Name	Character
Right Parentheses	)
Slash	/
Vertical Bar	|

In addition to the allowed special characters, the Mask function also allows:

Name	Character
Caret	^
Tilde	~
Vertical Bar	|

In addition to the allowed special characters, the Set Message Step also allows:

Name	Character
Asterisk	*
Colon	:
Comma	,
Exclamation Mark	!
Period	.
Question Mark	?
Semicolon	;

STEP TYPES

Step Group	stepName	Definition	stepType
Category Items steps are used to get or set a specific instance of a variable	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	1.1
	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	1.2
	Get Category Item (All Available)	Items in a subcategory of the current category are looked at as a whole	1.3
	Set Category Item (All Available)	Items in a subcategory of the current category are looked at as a whole	1.4
	Get Ranked Category Item	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.	1.5
	Set Ranked Category Item	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.	1.6
Count Across steps are used to determine how many instances of a specific item exist	Count Across Category (All Available)	Items in a subcategory of the current category are looked at as a whole.	2.1
	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	2.2
Data Type Functions check whether a value is numeric, alphabetic, or a date	IsAlpha	Checks whether a variable holds an alpha character value	3.1
	IsDate	Checks whether a variable holds a date value	3.2
	IsNumeric	checks whether a variable	3.3

Step Group	stepName	Definition	stepType
		holds a numeric value	
Date Functions calculate the difference between two dates in days, months or years, or adds to a date	Date Addition	used to add a specific number of days, months or years to an existing date	4.1
	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.	4.2
	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.	4.3
	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.	4.4
IF Statements are used to test criteria and execute certain steps based on whether the statement evaluates to true or false	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.	5.1
	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.	5.2
	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.	5.3
	IF Any (All Available)	Evaluates to TRUE if the statement is true for any instance of the element and	5.4

Step Group	stepName	Definition	stepType
		FALSE otherwise. Can be used for an element in any category.	
	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, the current category's child categories and the current category's parent categories.	5.5
	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, the current category's child categories and the current category's parent categories.	5.6
	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, the current category's child categories and the current category's parent categories.	5.7
Mask step used to select a portion of a string value	Mask	used to parse, truncate or append a string. For example, if a vehicle's ID number is passed into the system as VEH01, to use only the last two digits, use a mask step to store those digits in another element	6.1
Math Functions perform mathematical operations (arithmetic, absolute value, exponent)	Arithmetic	used to perform basic arithmetic, addition, subtraction, multiplication and division. It also can be used to perform bitwise OR and bitwise AND operations.	7.1
	Absolute Value	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	7.2
	Power	Used to raise a number to a power. The components of the exponent step are Result,	7.3

Step Group	stepName	Definition	stepType
		Base, and Power.	
	Log	Returns the logarithm of a number to the base specified.	7.4
	Log10	Returns the base-10 logarithm of a number	7.5
	Exp	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	7.6
	Rand	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	7.7
	Fact	Returns the factorial of a number. Number values must be positive	7.8
	Sqrt	Returns a positive square root of the number entered. Number values must be positive	7.9
	Ceil	Allows for a number to be rounded up away from zero, to the nearest multiple of significance.	7.10
	Floor	Allows for a number to be rounded down towards zero, to the nearest multiple of significance.	7.11
	Even	Allows for a number to be rounded up to the nearest even integer.	7.12
	Odd	Allows for a number to be rounded up to the nearest odd integer.	7.13
Set Underwriting To Fail is a special algorithm step available only in underwriting type algorithms, used to halt rating because a policy does not meet an underwriting requirement	Set Underwriting To Fail	A set underwriting to fail step is used in an algorithm to halt rating because a policy does not meet an underwriting rule or requirement.	9.1
String Functions are used to create or modify a string data type	Get Length	gets the number of characters in a specified string.	10.1
	Set Message	used to set a specific message such as the reason for an underwriting failure.	10.2
	String Addition	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".	10.3
Sum Across steps are used to determine the sum of all instances of a specific	Sum across Category (All Available)	Items in a subcategory of the current category are looked at as a whole	11.1
	Sum across Category (Use	Items in a subcategory of the	11.2

Step Group	stepName	Definition	stepType
element (input, variable, result)	Current Path)	current category are looked at independently, for each instance of the current category.	

Sample Step XML(s)

```
<Step stepNumber="2" stepType="8.2" stepName="Sum Across Category (All Available)">
  <Term termType="expression" value="7" valueName="USE VARIABLE:"></Term>
  <Term termType="Input" value="114HW492-C758-CD97-9019-CPLIA8752C6"
valueName="DriveHours"></Term>
  <Term termType="nextStep" value="3" valueName="Step 3"></Term>
</Step>
```

Masking

Masking allows users 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, the masking function can be used to read only the first five characters of the value. Mask can only be applied to String and Date Criteria data types.

Masking

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

Masking Examples

Function	Input	Mask	Result
Hold	GA263SX4597	~~~~~	GA263
Remove	VEH01	|||~~	1
Replace	Insbridge	~~~urance	Insurance
Insert	Cat	~^h~^r~	Chart

TERM TYPES/CODES

Valid term types are:

Term Group	termType	valueName	value
variable	Input	The previously assigned name related to the id for the specific variable	id (GUID)
variable	TableVariable	The previously assigned name related to the id for the specific variable	id (GUID)
variable	CalculatedVariable	The previously assigned name related to the id for the specific variable	id (GUID)
variable	ResultVariable	The previously assigned name related to the id for the specific variable	id (GUID)
variable	StepVariable	The previously assigned name related to the id for the specific variable	id (GUID)
variable	Algorithm	The previously assigned name related to the id for the specific variable	id (GUID)
variable	constant	• 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 100, 1000, 10000 • Y, YES • N, NO • **NULL** • (system) Current Category Instance # • (system) Current System Date and Time • Custom Value (any user defined value not included above, excluding values with restricted characters)	
operator	plus		1
operator	minus		2
operator	multiplied by		3
operator	divided by		4
operator	equals		5
operator	not equal to		6
operator	less than		7
operator	less than or equal to		8
operator	greater than		9
operator	greater than or equal to		10
operator	masked by		11
expression	ELSE		3
expression	OR		4
expression	AND		5
math function	log		
math function	log10		
math function	abs		
round	round	Rounds the number using standard rounding rules. Rounding can be set anywhere between	r0-r9

Term Group	termType	valueName	value
		0 and 9 decimal places. (e.g., Round to 2 places)	
round	truncate	All or a portion of the decimal will be dropped. Can be set between 0 and 9 decimal places.	t0-t9
round	round up	Will round the number, or a portion of it, to the next higher number. Can be set between 0 and 9 decimal places.	u0-u9
round	no round	No rounding will occur on the step.	No
date	precision	days	1
date	precision	months	2
date	precision	years	3
nextStep	stepNumber	The numeric value assigned to a step	n-nnnn
nextStep	DONE	there are no subsequent steps	DONE
nextStep	EXIT LOOP		EXIT
Table options	mask	Masking is described in more detail in the Masking section.	
Table options	wildcard	Yes, No	1,0
Table options	interpolate	Yes, No	1,0
Underwriting	underwriting	Set Underwriting to FAIL	1,0

Sample Terms

```
<Term termType="Input" value="[varID/GUID]" valueName=""/>
<Term termType="TableVariable" value="[varID/GUID]" valueName=""/>
<Term termType="CalculatedVariable" value="[varID/GUID]" valueName=""/>
<Term termType="ResultVariable" value="[varID/GUID]" valueName=""/>
<Term termType="stepVariable" value="[varID/GUID]" valueName=""/>

<Term termType="constant" value="[literal constant value]" valueName=""/>

<Term termType="operator" value="1" valueName="plus"/>
<Term termType="operator" value="2" valueName="minus"/>
<Term termType="operator" value="3" valueName="multiplied by"/>
<Term termType="operator" value="4" valueName="divided by"/>
<Term termType="operator" value="5" valueName="equals"/>
<Term termType="operator" value="6" valueName="not equal to"/>
<Term termType="operator" value="7" valueName="less than"/>
<Term termType="operator" value="8" valueName="less than or equal to"/>
<Term termType="operator" value="9" valueName="greater than"/>
<Term termType="operator" value="10" valueName="greater than or equal to"/>

<Term termType="expression" value="3" valueName="ELSE"/>
<Term termType="expression" value="4" valueName="OR"/>
<Term termType="expression" value="5" valueName="AND"/>

<Term termType="round" value="r0" valueName="Round to 0 places"/>
<Term termType="round" value="r1" valueName="Round to 1 places"/>
<Term termType="round" value="r2" valueName="Round to 2 places"/>
```

```
<Term termType="round" value="r3" valueName="Round to 3 places"/>
<Term termType="round" value="r4" valueName="Round to 4 places"/>
<Term termType="round" value="r5" valueName="Round to 5 places"/>
<Term termType="round" value="r6" valueName="Round to 6 places"/>
<Term termType="round" value="r7" valueName="Round to 7 places"/>
<Term termType="round" value="r8" valueName="Round to 8 places"/>
<Term termType="round" value="r9" valueName="Round to 9 places"/>

<Term termType="truncate" value="t0" valueName="Truncate to 0 places"/>
<Term termType="truncate" value="t1" valueName="Truncate to 1 places"/>
<Term termType="truncate" value="t2" valueName="Truncate to 2 places"/>
<Term termType="truncate" value="t3" valueName="Truncate to 3 places"/>
<Term termType="truncate" value="t4" valueName="Truncate to 4 places"/>
<Term termType="truncate" value="t5" valueName="Truncate to 5 places"/>
<Term termType="truncate" value="t6" valueName="Truncate to 6 places"/>
<Term termType="truncate" value="t7" valueName="Truncate to 7 places"/>
<Term termType="truncate" value="t8" valueName="Truncate to 8 places"/>
<Term termType="truncate" value="t9" valueName="Truncate to 9 places"/>

<Term termType="roundUp" value="u0" valueName="RoundUp Up to 0 places"/>
<Term termType="roundUp" value="u1" valueName="RoundUp Up to 1 places"/>
<Term termType="roundUp" value="u2" valueName="RoundUp Up to 2 places"/>
<Term termType="roundUp" value="u3" valueName="RoundUp Up to 3 places"/>
<Term termType="roundUp" value="u4" valueName="RoundUp Up to 4 places"/>
<Term termType="roundUp" value="u5" valueName="RoundUp Up to 5 places"/>
<Term termType="roundUp" value="u6" valueName="RoundUp Up to 6 places"/>
<Term termType="roundUp" value="u7" valueName="RoundUp Up to 7 places"/>
<Term termType="roundUp" value="u8" valueName="RoundUp Up to 8 places"/>
<Term termType="roundUp" value="u9" valueName="RoundUp Up to 9 places"/>

<Term termType="noRound" value="No" valueName="No Rounding"/>
<Term termType="precision" value="1" valueName="days"/>
<Term termType="precision" value="2" valueName="months"/>
<Term termType="precision" value="3" valueName="years/>

<Term termType="nextStep" value="2" valueName="Step 2"/>
<Term termType="nextStep" value="DONE" valueName="DONE"/>
<Term termType="nextStep" value="EXIT" valueName="EXIT LOOP"/>

<Term termType="mask" value="|||~~^i" valueName=""/>
<Term termType="wildcard" value="0" valueName="No"/>
<Term termType="interpolate" value="0" valueName="No"/>
```

FULL XML EXAMPLE

```
<OIRUProgramImport effectiveDate="01-01-2013" versionDescription="Jan V01"
versionID="1">
   <LineOfBusiness name="Home">
      <Global>
         <Categories>
            <Category name="Policy" id="A43375C0-8899-4B45-B0E1-697D60661BD1"
dateModified="9-1-2012" currentStatus="New"/>
               <Input name="BusnPrsnlPropLimit" id="1C977EA3-8CD1-4E59-B7C9-
64465428228E" dateModified="9-1-2012" dataType="Integer"
currentStatus="New"/>
               <ResultVariable name="TotalPremium" id="CB6DAABA-5647-4D94-
B242-E377DEEB4F7F" dateModified="8-1-2012" currentStatus="New"
dataType="Decimal"/>
               <Category name="Dwelling" id="0C70E248-EC76-4799-B7F8-
24EEA4D4F065" dateModified="8-1-2012" currentStatus="New">
                  <Input name="CoverageALimit" id="A10E1A64-E673-472C-82AC-
25BB76DF42E5" dateModified="8-1-2012" currentStatus="New"
dataType="Integer"/>
                  <ResultVariable name="TotalCoverage" id="CABAABA-5600-5D93-
B119-E388DEEB4F7F" dateModified="8-1-2012" currentStatus="New"
dataType="Decimal"/>
               </Category>
            </Category>
         </Categories>
         <CalculatedVariables>
            <CalculatedVariable name="KeyPremium" id="66C33473-AFAB-437B-
8A81-4A1D2D9D5F03" dateModified="9-1-2012" currentStatus="New" version="1"
dataType="Decimal" categoryID="0C70E248-EC76-4799-B7F8-24EEA4D4F065"
categoryName="Dwelling">
               <Steps>
                  <Step stepNumber="1" stepType="5.1" stepName="IF">
                     <Term termType="expression" value="1" valueName="IF"/>
                     <Term termType="Input" value="E4E9B263-8673-4C0E-9341-
73FA350586A6" valueName="RatingPropertyType"/>
                     <Term termType="operator" value="5" valueName="equals"/>
                     <Term termType="constant" value="2" valueName="HO Form
2"/>
                     <Term termType="expression" value="4" valueName="OR"/>
                     <Term termType="Input" value="E4E9B263-8673-4C0E-9341-
73FA350586A6" valueName="RatingPropertyType"/>
                     <Term termType="operator" value="5" valueName="equals"/>
                     <Term termType="constant" value="3" valueName="HO Form
3"/>
                     <Term termType="expression" value="4" valueName="OR"/>
                     <Term termType="Input" value="E4E9B263-8673-4C0E-9341-
73FA350586A6" valueName="RatingPropertyType"/>
                     <Term termType="operator" value="5" valueName="equals"/>
                     <Term termType="constant" value="5" valueName="HO Form
5"/>
                     <Term termType="expression" value="4" valueName="OR"/>
                     <Term termType="Input" value="E4E9B263-8673-4C0E-9341-
73FA350586A6" valueName="RatingPropertyType"/>
                     <Term termType="operator" value="5" valueName="equals"/>
```

```
                        <Term termType="constant" value="8" valueName="HO Form
8"/>
                        <Term termType="expression" value="2" valueName="THEN"/>
                        <Term termType="nextStep" value="2" valueName="Step 2"/>
                        <Term termType="expression" value="3" valueName="ELSE"/>
                        <Term termType="nextStep" value="4" valueName="Step 4"/>
                     </Step>
                     <Step stepNumber="2" stepType="7.1" stepName="Arithmentic">
                        <Term termType="ResultVariable" value="CB6DAABA-5647-
4D94-B242-E377DEEB4F7F" valueName="TotalPremium"/>
                        <Term termType="operator" value="5" valueName="equals"/>
                        <Term termType="Input" value="1C977EA3-8CD1-4E59-B7C9-
64465428228E" valueName="BusnPrsnlPropLimit"/>
                        <Term termType="operator" value="3"
valueName="multiplied by"/>
                        <Term termType="Input" value="375F1931-578B-4C16-B2B9-
7EF5118400CD" valueName="AutomaticIncreasePct"/>
                        <Term termType="round" value="0" valueName="Round to 0
places"/>
                        <Term termType="nextStep" value="3" valueName="Step 3"/>
                     </Step>
                  </Steps>
               </CalculatedVariable>
            </CalculatedVariables>
            <Algorithms>
               <Algorithm name="TerrorismCoveragePremium" id="2615bb44-ffed-
4b7c-b9e8-fb3769cbac2c" dateModified="9-1-2012" currentStatus="New"
version="1" algorithmType="Rating" categoryID="A43375C0-8899-4B45-B0E1-
697D60661BD1" categoryName="Policy">
                  <Steps>
                     <Step stepNumber="1" stepType="7.1" stepName="Arithmetic">
                        <Term termType="StepVariable" value="1"
valueName="Results of Step 1"/>
                        <Term termType="operator" value="5" valueName="equals"/>
                        <Term termType="Input" value="99999999" valueName="-No
Input"/>
                        <Term termType="round" value="2" valueName="Round to 2
places"/>
                        <Term termType="nextStep" value="DONE"
valueName="DONE"/>
                     </Step>
                  </Steps>
               </Algorithm>
            </Algorithms>
         </Global>
         <Program>
            <TableVariables>
               <TableVariable name="AccountsReceivableLimit" id="47D92217-3E01-
40D1-A2B0-A19D875E43CF" dateModified="9-1-2012" dataType="Integer"
currentStatus="New" version="1" defaultValue="" categoryID="A43375C0-8899-
4B45-B0E1-697D60661BD1" categoryName="Policy">
                  <Criteria>
                     <Criterion criterionNumber="1">
                        <Term termType="Input" value="1C977EA3-8CD1-4E59-B7C9-
64465428228E" valueName="BusnPrsnlPropLimit"/>
                        <Term termType="operator" value="5" valueName="equals"/>
                     </Criterion>
```

```
                  <Criterion criterionNumber="2">
                     <Term termType="Input" value="3D82F49D-A52C-4A97-AF6C-
227278511F3C" valueName="FunctlBusnPrsnlPropValtnApply"/>
                     <Term termType="operator" value="5" valueName="equals"/>
                  </Criterion>
               </Criteria>
            </TableVariable>
         </TableVariables>
         <TableData>
            <TableVariable name="AccountsReceivableLimit" id="47D92217-3E01-
40D1-A2B0-A19D875E43CF">
               <Row num="1">
                  <Data columnNum="1" columnName="AccountsReceivableLimit"
id="">10000 through 999999999</Data>
                  <Data columnNum="2" columnName="BusnPrsnlPropLimit" id="">1
through 999999999</Data>
                  <Data columnNum="3"
columnName="FunctlBusnPrsnlPropValtnApply" id="">Yes</Data>
               </Row>
               <Row num="2">
                  <Data columnNum="1" columnName="AccountsReceivableLimit"
id="">10000 through 999999999</Data>
                  <Data columnNum="2" columnName="BusnPrsnlPropLimit" id="">1
through 999999999</Data>
                  <Data columnNum="3"
columnName="FunctlBusnPrsnlPropValtnApply" id="">No</Data>
               </Row>
               <Row num="3">
                  <Data columnNum="1" columnName="AccountsReceivableLimit"
id="">10000 through 999999999</Data>
                  <Data columnNum="2" columnName="BusnPrsnlPropLimit"
id="">0</Data>
                  <Data columnNum="3"
columnName="FunctlBusnPrsnlPropValtnApply" id="">Yes</Data>
               </Row>
               <Row num="4">
                  <Data columnNum="1" columnName="AccountsReceivableLimit"
id="">0</Data>
                  <Data columnNum="2" columnName="BusnPrsnlPropLimit"
id="">0</Data>
                  <Data columnNum="3"
columnName="FunctlBusnPrsnlPropValtnApply" id="">No</Data>
               </Row>
            </TableVariable>
         </TableData>
         <Algorithms>
            <Algorithm name="DriverAssignmentScenario1" id="67879432-20B7-
450D-85A9-418D858A836A" dateModified="2011-07-15" currentStatus="New"
version="1" dataType="decimal" executionOrder="1" algorithmType="DAScenario"
categoryID="4CD357C1-DDB7-47F5-987D-83153104391A" categoryName="Policy">
               <Steps>
                  <Step stepNumber="0" stepType="30.1" stepName="Vehicle
Usage Option">
                     <Term termType="Vehicle Usage Option" value="13.4"
valueName="Vehicle Usage by VehicleID (Inputs: 299/368)"/>
                     <Term termType="nextStep" value="1" valueName="Step 1"/>
                  </Step>
```

```
                  <Step stepNumber="1" stepType="32.1" stepName="Rank All
Dvrs (Low-to-High)">
                     <Term termType="expression" value="" valueName="USE
ALGORITHM:"/>
                     <Term termType="Algorithm" value="25F81111-0A19-4242-
8628-BCAE516E622C" valueName="rankDriverByID"/>
                     <Term termType="nextStep" value="2" valueName="Step 2"/>
                  </Step>
                  <Step stepNumber="2" stepType="32.9" stepName="Rank All
Vehs (Low-to-High)">
                     <Term termType="expression" value="6" valueName="USE
ALGORITHM:"/>
                     <Term termType="Algorithm" value="314CF402-C698-4983-
99E0-6727033650C4" valueName="rankVehicleByID"/>
                     <Term termType="nextStep" value="3" valueName="Step 3"/>
                  </Step>
                  <Step stepNumber="3" stepType="33.1" stepName="Assign 1st
Ranked Veh to 1st Ranked Drv">
                     <Term termType="nextStep" value="DONE"
valueName="DONE"/>
                  </Step>
               </Steps>
            </Algorithm>
         </Algorithms>
         <Sequence>
            <Algorithm executionOrder="1" algorithmType="Rating"
name="TerrorismCoveragePremium" id="2615bb44-ffed-4b7c-b9e8-fb3769cbac2c"
version="1" activeInd="1"/>
            <Algorithm executionOrder="2" algorithmType="Rating"
name="MathFunctions" id="b661b275-23f7-4782-9bc9-100f6db46f2d" version="1"
activeInd="1"/>
            <Algorithm executionOrder="3" algorithmType="Rating"
name="Category Items" id="63e16d41-b1db-4bc6-8377-b7671ba8ea89" version="1"
activeInd="1"/>
            <Algorithm executionOrder="4" algorithmType="Rating" name="Count
Across" id="8611bffa-12c9-46d3-bbbe-b150f3d77a65" version="1" activeInd="1"/>
         </Sequence>
         <ResultGroups>
            <ResultGroup name="RatingResults" dateModified="2012-09-26"
activeInd="1" defaultInd="1" outputCode="12AB3456789">
               <Variable resultOrder="1" name="TotalPremium" id="CB6DAABA-
5647-4D94-B242-E377DEEB4F7F" variableType="ResultVariable" resultID="Total
Premium" categoryID="A43375C0-8899-4B45-B0E1-697D60661BD1"
categoryName="Policy" activeInd="1"/>
               <Variable resultOrder="2" name="AggregateLimit" id="586C578C-
F98F-4D31-838C-C1125F4DA9DD" variableType="Input" resultID="Aggregate Limit"
categoryID="A43375C0-8899-4B45-B0E1-697D60661BD1" categoryName="Policy"
activeInd="1"/>
               <Variable resultOrder="3" name="KeyPremium" id="66C33473-AFAB-
437B-8A81-4A1D2D9D5F03" variableType="CaculatedVariable" resultID="Key
Premium" categoryID="0C70E248-EC76-4799-B7F8-24EEA4D4F065"
categoryName="Dwelling" activeInd="1"/>
            </ResultGroup>
         </ResultGroups>
      </Program>
   </LineOfBusiness>
</OIRUProgramImport>
```

CONTACTING SUPPORT

If you need assistance with an Oracle Insurance Insbridge Enterprise Rating System product, please log a Service Request using My Oracle Support at https://support.oracle.com/.

Oracle customers have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Address any additional inquiries to:

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com

Index

V

X

Z