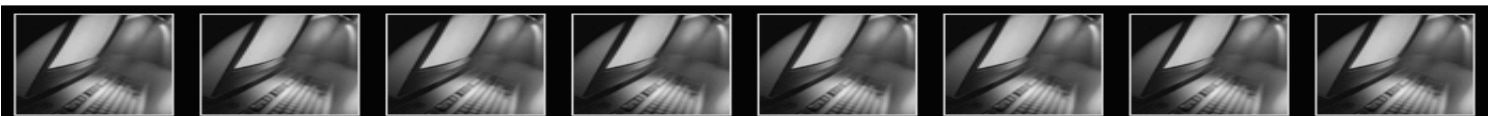


Rates and Re-pricing User's Guide

Oracle Insurance Policy Administration - Life
Release 8.1 or previous
E14444-01
May 2009



Oracle® Insurance Policy Administration

Copyright © 2009, Oracle. All rights reserved.

License Restrictions & Warranty Disclaimer

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

Restricted Rights Notice

U.S. GOVERNMENT RIGHTS

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

Third Party Web Sites, Content, Products, and Services Disclaimer

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

Trademark Notice

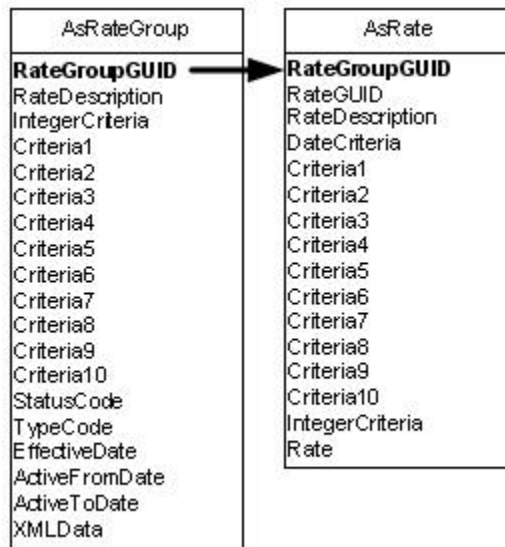
Oracle, JD Edwards, and PeopleSoft are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Table of Contents

Rate Overview	4
The Rate Table Interface	5
Rate Group	6
Steps to Create a Rate Group	7
Rates	8
Types of Rates That Can be Uploaded	8
Aggregate Tables	9
Select Tables	13
Select with Ultimate	16
Steps to Upload Select with Ultimate Rates	16
Configuration using Rates	20
RATEARRAY Configuration	22
RATE ARRAY XML Example	23
Policy Print Example	24
Manipulating an Array of Rates	24
Repricing	25
Repricing for New Policies –vs – Repricing for Existing and New Policies	26
Example One: Two Sets of Rates are Open	26
Example Two: Only One Set of Rates is Open for BaseRates	28
Appendix A - Entering Original RateGroup and Rates	30
Appendix B - Re-priced Rates with Original Rate Group Not Closed.....	33
Appendix C - Original BaseRates Entry After Group Closure.....	36
Appendix D - XML Used for Rate Retrieval Examples	39
Appendix E - Rates Used for Rate Retrieval Examples	40
Appendix F - Pseudo SQL that is Generated.....	41

Rate Overview

Rates are uploaded into the Oracle Insurance Policy Administration (OIPA) system using excel spreadsheets that populate the rate tables in the database. In order to upload rates, a rate group that describes the rates must first be created. Then the actual rates may be uploaded. The two rate tables that are populated are AsRateGroups and AsRate.



Rate Tables Populated in Database

When a new plan is configured for OIPA, a rate group must initially be created via the Tables>Rates Table menu. A rate group bands rates together according to their designated purpose and provides a means of describing and retrieving rates. Usually rates are created for a specific plan. After you create a rate group, you can upload the actuarial rates, which should have been provided in an Excel spreadsheet by the business, client or party responsible for supplying rate information for a plan.

The Rate Table Interface

The screenshot shows the 'Rate Table' interface. At the top, there's a title bar 'Rate Table' with a key icon, a help icon, and a close icon. Below the title bar, there are several input fields and buttons. The 'Rate List' field contains '(NewRates)' and has a dropdown arrow. A red box highlights this field with the text 'Search existing rate groups and associated rates'. Below 'Rate List' are 'Name:' and 'Date:' fields. To the right of these are 'Criteria 3:', 'Criteria 4:', 'Criteria 5:', 'Criteria 6:', 'Criteria 8:', and 'Criteria 10:' fields. Below 'Name:' is an 'Intgr Criteria:' field. Below 'Date:' is a 'Rate:' field. A red box highlights the 'Rate:' field with the text 'Create a New Rate Group'. Below the 'Rate:' field is a 'Description' button. A red box highlights the 'Upload Rates' button. Below the 'Description' and 'Upload Rates' buttons is a 'Close' button. At the bottom of the interface is a table header with four columns: 'Rate Description', 'Rate Activation Date', 'Transaction From Date', and 'Transaction To Date'.

There are three steps involved in using rates in OIPA.

1. Create Rate Group
2. Upload Rates
3. Retrieve Rates for Configuration Purposes

Rate Group

Rate groups help organize and associate rates together in the AsRateGroup table. Once a rate group is created a unique GUID is created and used to link the rate group to rates. The rate group stores a description for rates, the rating indexing method and rate criteria(s). These key pieces of information are required for the rate group and drive how the system will use the associated rates for configuration and policy events.

The screenshot shows a 'Rate Group' configuration window with the following fields:

- Rate Group:** New Rate Group (dropdown)
- Name:** ABCTerm
- Integer Criteria:** Duration
- Rate Activation Date:** 01/01/2009
- Transaction From Date:** 01/01/2009
- Transaction To Date:** (empty)
- Criteria 1:** Gender
- Criteria 2:** Tobacco
- Criteria 3:** Issue Age
- Criteria 4:** (empty)
- Criteria 5:** (empty)
- Criteria 6:** (empty)
- Criteria 7:** (empty)
- Criteria 8:** (empty)
- Criteria 9:** (empty)
- Criteria 10:** (empty)

Buttons: Save, Close

AsRateGroups
RateGroupGUID
RateDescription
IntegerCriteria
Criteria1
Criteria2
Criteria3
Criteria4
Criteria5
Criteria6
Criteria7
Criteria8
Criteria9
Criteria10
EffectiveDate
ActiveFromDate
ActiveToDate
XMLDate

Rate Group Information

Key information required for rate groups:

- **RateGroupGUID**- A system generated unique GUID assigned to the RateGroup and RateDescription.
- **Name:** A descriptive name that is used to identify the rates. This should be the name provided with the rate table. This is stored in the **RateDescription** column in AsRateGroup.
- **Integer Criteria:** This is used as a means of identifying the name of the primary method of rate indexing. Rate indexing is a mechanism employed by insurance companies to track the amount of time or change affecting an insurance rate. Main indexes are generally **age** and policy **duration** (in years), which are both integer indexes. The actual index values, such as 1, 2, 3, etc..., are actually stored in AsRate. The name of the index is stored in **IntegerCriteria** of the AsRateGroup table.
- **Rate Activation Date:** This date is used in conjunction with a policy's effective date to identify when a set of rates is active for a policy. The policy must be effective **on or after** the date listed here to use the set of rates. It is this date that is used as a way to distinguish multiple sets of rates tied to a single rate group name. Rates may change due to **re-pricing** after the launch of a plan or for various other reasons. While all other factors may remain, by creating an identical rate group with a different rate activation date; you can manage another set of rates for the same product.

Note: Rate Activation Date is stored as **EffectiveDate** in the AsRateGroup table.

- **Transaction From Date:** A date used in conjunction with transaction that requires the use of rates; if a transaction occurs **on or after** this date, but prior to the ActiveToDate. This is stored in the **ActiveFromDate** column in AsRateGroup.

Important: Transaction From Date cannot change after it has been set, as the system would see this as a need for undo/redo or backdating.

- **Transaction To Date:** A date used in conjunction with transaction that requires the use of rates; if a transaction occurs **before** this date. This is stored in the **ActiveToDate** column in AsRateGroup.

Important: Transaction To Date can not change after it has been set, as the system would see this as a need for undo/redo or backdating.

- **Criteria 1 through 10:** In these fields you will enter the criteria descriptions for the Excel spreadsheet with the rates. Criteria names must remain constant for all rate groups in the system. For example, if UWClass is used, then there should not also be entries for Underwriting Class or UWclass. Keep the syntax and spelling the same for criteria with the same meaning. These criteria names will correlate to columns in your rate table.

The system is based on XML and is case sensitive so your criteria casing must match. Gender and gender are interpreted differently.

Note: In AsRateGroup, the StatusCode, TypeCode and XMLData columns are not used.

Steps to Create a Rate Group

1. Select **Table>Rate Tables**.
2. Select the **Description** button.
3. Enter the rate group: (Descriptions have been provided above for each field.)
 - a. Name
 - b. Integer Criteria
 - c. Rate Activation Date
 - d. Transaction From Date
 - e. Transaction To Date (optional)
 - f. Enter the criteria.
4. Select the **Save** button.

Rates

After a rate group is created for rates, the rates must be uploaded via an Excel spreadsheet. Rates are associated with a rate group via the RateGroupGUID and criteria are the same as in the RateGroup, but they are stored by their code value and the IntegerCriteria is the actual integer value.

AsRate	
RateGroupGUID	RateGroupGUID is the RateGroupGUID of the RateGroup that the Rate is attached to. RateGUID is system-generated - a unique GUID for each rate. RateDescription is the RateDescription of the RateGroup that the Rate is attached to Criteria are the same as in the RateGroup, but they are stored by their code value. For example, the Criteria in AsRateGroup may be Gender. When loading the rates, the codes for the applicable Gender must be entered such as 01 for male, 02 for female or 03 for unisex. IntegerCriteria is the actual integer value according to the description name stored in the rate group. Rate is the individual rate according to all the criteria in its record. DateCriteria is not filled in when uploading rates.
RateGUID	
RateDescription	
DateCriteria	
Criteria1	
Criteria2	
Criteria3	
Criteria4	
Criteria5	
Criteria6	
Criteria7	
Criteria8	
Criteria9	
Criteria10	
IntegerCriteria	
Rate	

Types of Rates That Can be Uploaded

Rates come in different forms depending on the plan requirements. OIPA supports three types of rates. The types of rates that can be uploaded are named according to OIPA naming convention and not industry standards. You will need to read about the three types of rates to understand which type of upload you will need to perform.

The three types of rates that can be uploaded are:

1. Aggregate
2. Select
3. Select with Ultimate

Aggregate Tables

Aggregate tables use only the specified IntegerCriteria, such as age, (or policy duration) to index the rates. In the case of Aggregate tables, the IntegerCriteria will be age regardless of whether you are using issue or attained age indexed rates. The table only stores the integer numbers and uses them as the index. Configuration will handle whether to use the issue or attained age. The same follows if you use duration; whether the business requirements state that it should be the duration policy is in effect or duration of time the product has been on the market.

An aggregate table must be in the correct OIPA format to upload. The specified IntegerCriteria must be placed in the left-most column of your table. All other criteria must be placed in the columns directly to the right of the Integer Criteria column. Examples of additional criteria include gender, age, tobacco use, etc.

Example Aggregate Rate Table

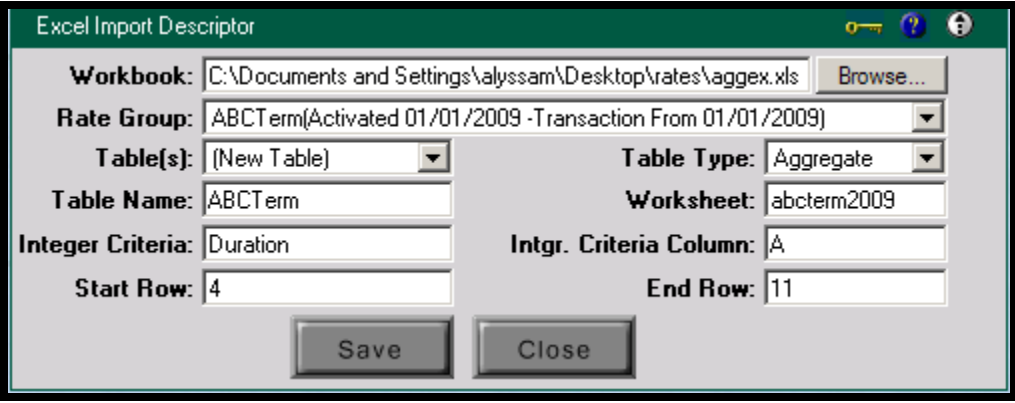
	A	B	C	D	E	Criteria2	Criteria1
1							
2		Male		Female		Unisex	
3	AGE	Smoke	Nonsmoke	Smoker	Nonsmoke	Smoker	Nonsmoker
4	0	11.53	11.53	13.99	13.99	11.95	11.95
5	1	11.59	11.59	13.97	13.97	11.99	11.99
6	2	11.27	11.27	13.58	13.58	1.66	1.66
7	3	10.94	10.94	13.19	13.19	11.33	11.33
8	4	10.63	10.63	12.81	12.81	10.99	10.99
9	5	10.34	10.34	12.42	12.42	10.67	10.67
10	6	10.05	10.05	12.05	12.05	10.34	10.34
11	7	9.69	9.69	11.68	11.68	10.02	10.02
12							

Annotations in the image:

- Criteria2** (green box) points to the header of the 6th column.
- Criteria1** (red box) points to the header of the 7th column.
- Rate** (purple box) points to the value 11.27 in row 6, column C.
- IntegerCriteria** (blue box) points to the value 10 in row 9, column A.
- Worksheet Name** (orange box) points to the tab name 'abcterm2009' at the bottom.

Steps to Upload Aggregate Rates

1. Select **Tables>Rate Tables**.
2. Select the **Upload** button.
3. Select the **Browse** button for Workbook and select the Excel spreadsheet containing the rates you want to upload.
4. Select the [Rate Group](#) that is associated with these rates.
5. For Table, make sure New Table is selected.
6. For Table Type, select Aggregate.
7. For Table Name, enter the name of the table you are going to create to support your rates. This should be the name of the rate group.
8. For the Worksheet field, enter the literal worksheet name exactly as it appears in the Excel spreadsheet.
9. **Enter the spreadsheet column** identifier containing the Integer Criteria for the intended indexing strategy.
10. Enter the starting and ending row numbers of the Rates to be uploaded.



The image shows a dialog box titled "Excel Import Descriptor". It contains the following fields and controls:

- Workbook:** A text field containing "C:\Documents and Settings\alyssam\Desktop\rates\aggex.xls" and a "Browse..." button.
- Rate Group:** A dropdown menu showing "ABCTerm(Activated 01/01/2009 -Transaction From 01/01/2009)".
- Table(s):** A dropdown menu showing "(New Table)".
- Table Type:** A dropdown menu showing "Aggregate".
- Table Name:** A text field containing "ABCTerm".
- Worksheet:** A text field containing "abcterm2009".
- Integer Criteria:** A text field containing "Duration".
- Intgr. Criteria Column:** A text field containing "A".
- Start Row:** A text field containing "4".
- End Row:** A text field containing "11".
- At the bottom, there are "Save" and "Close" buttons.

Excel Import Screen

11. Select the **Save** button.

Note: You may get a SadServer if your SystemInformation.PROPERTIES file has the incorrect value for UploadedFilePath. The value should be a valid path your system can access.

12. Enter each column and associated code value for each criteria from the spreadsheet. Select the **Save** button to create another row in the Descriptor Criteria.
13. Select the **Upload** button.

Descriptor Criteria section appears

Excel Import Descriptor

Workbook: C:\BOSS VERSION 8\bin\AdminServer\AB4B6860-1F91-D50C-DF7D-ED4F2FB43689aggex.xls

Rate Group: ABCTerm(Activated 01/01/2009 - Transaction From 01/01/2009)

Table(s): ABCTerm

Table Name: ABCTerm

Integer Criteria: Duration

Intgr. C:

End Row: 11

Save Delete Close

The Descriptor Criteria section opens and must be filled out.

Changes to where the Excel file was uploaded. Upload path is identified in the SystemInformation.PROPERTIES file.

Descriptor Criteria

Column	Gender	Tobacco	IssueAge	Criteria 4	Criteria 5	>>

New Save Upload

Code values for criteria on Spreadsheet

2		Male(01)		Female(02)		Unisex(03)	
3	AGE	Smoke(01)	Nonsmoker(00)	Smoke(01)	Nonsmoker(00)	Smoke(01)	Nonsmoker(00)
4	0	11.53	11.53	13.99	13.99	11.95	11.95
5	1	14.53	14.53	13.97	13.97	11.99	11.99
6	2	Codes for Criteria		13.58	13.58	1.66	1.66
7	3	10.94	10.94	13.19	13.19	11.33	11.33
8	4	10.63	10.63	12.81	12.81	10.99	10.99

Enter all Columns and Criteria then select the Upload button.

Descriptor Criter

Column	Gender	Tobacco	Criteria 3	Criteria 4	Criteria 5	>>
F	01	01				
G	01	00				
H	02	01				
I	02	00				
J	03	01				
K	03	00				

New Save Upload

Now from the main Rate page, you can select the name from the Rate List and view the rates that were uploaded into AsRate.

Rate Table

Rate List: ABCTerm 1. Select the name of the Rate Group

Name: ABCTerm

Date:

Age:

Rate:

Criteria 5:

Criteria 7:

Criteria 9:

Tobacco:

Criteria 4:

Criteria 6:

Criteria 8:

Criteria 10:

Rate Description **Rate Activation Date** **Transaction From** **Ratio**

ABCTerm 01/01/2009 01/01/2009 2. Select the rates according to dates

Rate Value(s)

Page 1 of 5 Page 1 2 3 4 5 Maximum Results: 20

Rate	Date	Age	Rate	Gender	Tobacco	Criteria 3	Criteria 4	>>
ABCTerm		0	11.53	01	00			
ABCTerm		0	11.53	01	00			
ABCTerm		1	11.59	01	00			
ABCTerm		1	11.59	01	00			
ABCTerm		2	11.27	01	00			
ABCTerm		2	11.27	01	00			
ABCTerm		3	10.94	01	00			
ABCTerm		3	10.94	01	00			
ABCTerm		4	10.63	01	00			
ABCTerm		4	10.63	01	00			
ABCTerm		5	10.31	01	00			

3. Rates appear

Select Tables

Select table is used when two indexes are needed for the rating strategy. For example, a rate table that uses both duration and age as methods for indexing rates. In this scenario, the IntegerCriteria for the Select Table is Duration and it is always specified in the left-most column of your table. Age is then the secondary index and is specified as the row header for each additional column. All Age indices must be placed in the columns directly to the right of the IntegerCriteria column.

Select Table Example

	A	B	C	D	E	F	G	H
1	XYZ Life Insurance							
	Criteria: Male - NonSmoker							
4	Index 1				Index 2			
4	Duration			AGE -->				
5		18	19	20	21	22	23	24
6	1	0.0002440	0.0002440	0.0002341	0.0002242	0.0002103	0.0001944	0.0001745
7	2	0.0002559	0.0002539	0.0002461	0.0002362	0.0002224	0.0002125	0.0002026
8	3	0.0002638	0.0002637	0.0002578	0.0002480	0.0002421	0.0002342	0.0002302
9	4	0.0002753	0.0002753	0.0002694	0.0002654	0.0002614	0.0002594	0.0002612
10	5	0.0002847	0.0002866	0.0002865	0.0002825	0.0002804	0.0002803	0.0002860
11	6	0.0002959	0.0003053	0.0003052	0.0002974	0.0002972	0.0003028	0.0003084
12	7	0.0003114	0.0003188	0.0003186	0.0003128	0.0003107	0.0003162	0.0003273
13	8	0.0003281	0.0003298	0.0003298	0.0003238	0.0003254	0.0003344	0.0003472

Steps to Upload Select Rates

1. Select **Tables>Rate Tables**.
2. Select the **Upload** button.
3. Select the **Browse** button for Workbook and select the Excel spreadsheet containing the rates you want to upload.
4. Select the **Rate Group** that is associated with these rates.

Rate Group

Rate Group: XYZLife(Activated 01/01/2009 - Transaction From 01/01/200

Name: XYZLife

Integer Criteria: Duration

Rate Activation Date: 01/01/2009

Transaction From Date: 01/01/2009

Transaction To Date:

Criteria 1: Gender

Criteria 2: Tobacco

Criteria 3: Age

Criteria 5:

Criteria 7:

Criteria 9:

Criteria 4: Secondary Index

Criteria 6:

Criteria 8:

Criteria 10:

Save

Close

Note: You must add the secondary index as a criterion for the Rate Group.

5. For Table, make sure New Table is selected.
6. For Table Type, select **Select**.
7. For Table Name, enter a name for the table you are going to create to support your rates. This should be the name of the rate group.
8. For the Worksheet field, enter the literal worksheet name exactly as it appears on the Excel spreadsheet.
9. Enter the name of the index used for Integer Criteria.

10. Enter the spreadsheet column identifier containing the Integer Criteria for the intended indexing strategy.
11. Enter the starting and ending row numbers for the Rates to be uploaded.
12. Enter the name for the **Secondary Index**, exactly the way it was spelled for the Rate Group.
13. Enter the row with the Secondary Index numbers.
14. Enter the starting and ending column letters for the secondary index.
15. Select the correct **Select Period**. Either Duration, Maturity Age, Maturity Age Max or Maturity Age Min.
16. Enter the total number of duration years in the **Duration Period** field, which should be the **End Row** minus the **Start Row**.
17. Select the **Save** button. The **Descriptor Criteria** section appears.
18. Enter in the code values for the **Descriptor Criteria**.
19. Enter **SI** into the criteria field that is the Secondary Index.
20. Select the **Save** button.
21. Select the **Upload** button. (You must select Save before Upload or you will get a Sad Sever.)

Uploading the Worksheet and Row and Column information

Excel Import Descriptor

Workbook: C:\WBOSS VERSION 7\bin\AdminServer0EA12870-52E0-4497-5FB1-2B1014B7B130rates.xls

Rate Group: XYZLife(Activated 01/01/2009 - Transaction From 01/01/2009)

Table(s): XYZLife

Table Name: XYZLife

Integer Criteria: Duration

Start Row: 3

Secondary Index: Age

Start Column: B

Select Period: Duration

Table Type: Select

Worksheet: select

Intgr. Criteria Column: A

End Row: 105

Secondary Index Row: 2

End Column: Q

Duration Period: 103

Save Delete Close

Descriptor Criteria

Column	Gender	Tobacco	Age
	01	01	SI

New Save Upload

Enter the codes for the criteria for the rate table

Rate Group Criteria created for the Secondary Index. You must enter SI to indicate it is the Secondary Index.

Rates in the OIPA

Rate Table

Rate List: XYZLife 1. Select Rate List
 Name: XYZLife
 Date:
 Duration:
 Rate:

Gender:
 Age:
 Criteria 5:
 Criteria 7:
 Criteria 9:

Tobacco:
 Criteria 4:
 Criteria 6:
 Criteria 8:
 Criteria 10:

2. Select Find

Rate Description	Rate Activation Date	Transaction Date	Action To Date
XYZLife	01/01/2009	01/01/2009	3. Select Rates according to date if applicable <input type="button" value="Action To Date"/>

Rate Value(s)

Page 1 of 83 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 Maximum Results: 20

Rate	Date	Duration	Rate	Gender	Tobacco	Age	Criteria 4	>>
XYZLife	1	2.44E-4	01	01	18			<input type="button" value="View Rates"/> 4. View Rates
XYZLife	2	2.559E-4						
XYZLife	3	2.638E-4						Rates are not stored in exponential format in the database
XYZLife	4	2.753E-4						
XYZLife	5	2.847E-4	01	01	18			

Rates in the Database

```
Select * from asrate where RateGroupGUID='9E63086D-C22F-E9C8-FDFA-EE4DF89B4D43'
```

RateDescription	Rate	IntegerCriteria	Criterial	Criteria2	Criteria3	Crit
XYZLife	0.000244	1 01	01	18		(null)
XYZLife	0.0002559	2 01	01	18		(null)
XYZLife	0.0002638	3 01	01	18		(null)
XYZLife	0.0002753	4 01	01	18		(null)
XYZLife	0.0002847	5 01	01	18		(null)

Select with Ultimate

Select with Ultimate is a hybrid strategy that uses two indexes for the rating. The final row of rates is called the **Ultimate Row** and once attained will end IntegerCriteria index incrementing. Once the ultimate rate is reached that row will always be used for retrieving rates and only the column index method will increment for retrieving rates.

For example, the rates below use the duration of a policy years as the IntegerCriteria index and age as a secondary index. Both indexes are needed to locate the appropriate rate. Once a policy reaches its 10th year the duration index will always be what is listed in the Ultimate Row, the age index will be the only index used to locate proper rate.

	A	B	C	D	E	F	G
1		Index 1	AGE -->	Index 2			
2		duration	18	19	20	21	22
3		1	0.0001270	0.0001191	0.0001098	0.0001006	0.0000901
4		2	0.0001261	0.0001195	0.0001103	0.0001024	0.0000931
5		3	0.0001251	0.0001198	0.0001120	0.0001041	0.0000975
6		4	0.0001255	0.0001202	0.0001124	0.0001059	0.0001019
7		5	0.0001257	0.0001205	0.0001141	0.0001102	0.0001075
8		6	0.0001260	0.0001221	0.0001182	0.0001142	0.0001116
9		7	0.0001251	0.0001237	0.0001210	0.0001172	0.0001158
10		8	0.0001277	0.0001264	0.0001226	0.0001199	0.0001198
11		9	0.0001303	0.0001277	0.0001251	0.0001237	0.0001236
12		10	0.0001314	0.0001289	0.0001287	0.0001286	0.0001308
23		Ultimate Row	0.0002008	0.0002135	0.0002285	0.0002466	0.0002701
24							
25		ultimate age	28		30	31	32

Worksheet Name

Note: In this general scenario, the IntegerCriteria for a Select with Ultimate rates is the Duration and it is always specified in the left-most column of your table. Age is then specified as the secondary index for each additional column. All Age criteria must be placed in the columns directly to the right of the IntegerCriteria column. Also the last row of all columns must be the Ultimate Rate value in order to identify the point at which the indexing strategy will shift to use Age.

Steps to Upload Select with Ultimate Rates

1. Select **Tables>Rate Tables**.
2. Select the **Upload** button.
3. Select the **Browse** button for Workbook and select the Excel spreadsheet containing the rates you want to upload.
4. Select the **Rate Group** that is associated with these rates.

Rate Group

Rate Group: New Rate Group

Name: KLMLife

Integer Criteria: Duration

Rate Activation Date: 01/01/2009

Transaction From Date: 01/01/2009

Transaction To Date:

Criteria 1: Gender

Criteria 2: Tobacco

Criteria 3: Age

Criteria 5:

Criteria 7:

Criteria 9:

Criteria 6:

Criteria 8:

Criteria 10:

Save Close

Note: You must add the secondary index as a criterion for the Rate Group.

5. For Table, make sure **New Table** is selected.
6. For Table Type, select **Ultimate**.
7. For Table Name, enter a name for the table you are going to create to support your rates. This should be the name of the rate group.
8. For the Worksheet field, enter the literal worksheet name exactly as it appears in the Excel spreadsheet.
9. Enter the name of the index used for Integer Criteria.
10. **Enter the spreadsheet column** identifier containing the Integer Criteria for the intended indexing strategy.
11. Enter the starting and ending row numbers for the Rates to be uploaded.
12. Enter the name for the **Secondary Index**, exactly the way it was spelled for the Rate Group.
13. Enter the row with the Secondary Index numbers.
14. Enter the starting and ending column letters for the secondary index.
15. Select the correct **Select Period**. Either Duration, Maturity Age, Maturity Age Max or Maturity Age Min.
16. Enter the total number of duration years in the **Duration Period** field, which should be the **End Row** minus the **Start Row**.
17. Select the **Save** button. The **Descriptor Criteria** section appears.
18. Enter the code values for the **Descriptor Criteria**.
19. Enter **SI** into the criteria field that is the Secondary Index.
20. Select the **Save** button.
21. Select the **Upload** button. (You must select Save before Upload or you will get a Sad Sever.)

Uploading the Worksheet and Row and Column information

Excel Import Descriptor

Workbook: C:\BOSS VERSION 7\bin\AdminServer3695A92B-D876-8688-70BA-4B2D1FD087FErates.xls

Rate Group: KLMLife(Activated 01/01/2009 -Transaction From 01/01/2009)

Table(s): KLMLife

Table Name: KLMLife

Integer Criteria: Duration

Start Row: 3

Secondary Index: Age

Start Column: C

Select Period: Duration

Table Type: Ultimate

Worksheet: ultimate

Intgr. Criteria Column: B

End Row: 13

Secondary Index Row: 2

End Column: BR

Duration Period: 10

Enter the codes for the criteria for the rate table

Save Delete Close

Descriptor Criteria

Gender	Tobacco	Age
01	01	SI

Rate Group Criteria created for the Secondary Index. You must enter SI to indicate it is the Secondary Index.

New Upload

Select with Ultimate Rates in the OIPA

Rates loaded into OIPA

Rate Table

Rate List: KLMLife

Name: KLMLife

Find

Rate Description: KLMLife

Rate Activation Date: 01/01/2009

Rate Value(s)

Page 1 of 135

Page 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33

Rates from Excel Spreadsheet

Index 1	Index 2	AGE -->	duration
1	18	19	20
2	0.0001270	0.0001191	0.0001098
3	0.0001261	0.0001195	0.0001103
4	0.0001251	0.0001198	0.0001120
5	0.0001255	0.0001202	0.0001124
6	0.0001257	0.0001205	0.0001141
7	0.0001260	0.0001221	0.0001182
8	0.0001251	0.0001237	0.0001210
9	0.0001277	0.0001264	0.0001226
10	0.0001303	0.0001277	0.0001251
11	0.0001314	0.0001289	0.0001287
12	0.0002008	0.0002135	0.0002285
13	0.0002135	0.0002285	0.0002466

Ultimate Row

Rate	Date	Duration	Rate	Gender	Tobacco	Age	Criteria 4
KLMLife		1	1.27E-4	01	01	18	
KLMLife		2	1.261E-4	01	01	18	
KLMLife		3	1.251E-4	01	01	18	
KLMLife		4	1.255E-4	01	01	18	
KLMLife		5	1.257E-4	01	01	18	
KLMLife		6	1.26E-4	01	01	18	
KLMLife		7	1.251E-4	01	01	18	
KLMLife		8	1.277E-4	01	01	18	
KLMLife		9	1.303E-4	01	01	18	
KLMLife		10	1.314E-4	01	01	18	
KLMLife		11	2.008E-4	01	01	18	
KLMLife		12	2.135E-4	01	01	18	
KLMLife		13	2.285E-4	01	01	18	
KLMLife		14	2.466E-4	01	01	18	

Configuration using Rates

The RATEARRAY function in OIPA is used to return a collection (array) of rates from the rate related tables based on the parameters it is passed. RATEARRAY is a type of MathVariable with its own attributes. Using the RATEARRAY function performs better than using SQL query because it results in faster response times. The two primary tables used in such a query are:

AsRateGroup – The table that stores the criteria information, indexing method, date and other descriptive rate information as described earlier in this guide.

AsRate – The table that stores the criteria values from AsCodes, rating and the rates.

The AsRate and AsRateGroup tables can be joined, for query purposes, via the RateGroupGUID or RateDescription fields.

Query joining AsRateGroup and AsRate

```
3 SELECT * FROM ASRATE where RateGroupGUID='C6496E59-7EF7-4719-959B-2C0065CA4EF9'
4 AND Criteria1='01' male
5 AND Criteria3='03' risk something
6 AND Criteria4='39' year of age
```

	RateGroupGUID	RateDescription	IntegerCriteria	Criteria1	Criteria2
1	C6496E59-7EF7-4719-959B-2C0065CA4EF9	Term_20_Premium	Duration	Gender	Tobacco

	Criteria3	Criteria4	Criteria5	Criteria6	Criteria7	Criteria8	Criteria9	Criteria10
89	01	01	(null)	(null)	(null)	(null)	(null)	(null)
89	02	02	(null)	(null)	(null)	(null)	(null)	(null)
89	03	01	(null)	(null)	(null)	(null)	(null)	(null)
89	03	39	02	(null)	(null)	(null)	(null)	(null)

Codes from AsCodes

Codes: AsCodeGender		
Code Value(s)	Short Description	Long Description
02	F	Female
01	M	Male
03	U	Unisex

Codes: AsCodeTobacco		
Code Value(s)	Short Description	Long Description
01	Current Tobacco Use	Current Tobacco Use
02	Qualifies Tobacco Free	Qualifies Tobacco Free

The Integer Criteria column of the AsRate table is used as the main method for rate indexing. Rate indexing is a mechanism used by insurance companies to track the amount of time or change affecting an insurance rate. This form of indexing should not be confused with array indexes. For example, a company can use Integer Criteria as the method of rating because they use “duration in years a policy is in effect” as a rate index. The database does not use this as an index.

```

1 SELECT * FROM ASRATEGROUP where RateGroupGUID='C6496E59-7EF7-4719-959B-2C0065CA4EF9'
2 GO
3 SELECT * FROM ASRATE where RateGroupGUID='C6496E59-7EF7-4719-959B-2C0065CA4EF9'

```

5:33 INS

AsRateGroup

	RateDescription	IntegerCriteria	Criterial	Criteria2	Criteria3	Criteria4	Criteria5	EffectiveDate	
1	Term_20_Premium	Duration	Gender	Tobacco	Risk	IssueAge	Band	1/1/2002 12:00:00 AM	1

AsRate

	RateDescription	DateCriteria	Criterial	Criteria2	Criteria3	Criteria4	Criteria5	IntegerCriteria1	Rate
1	Term_20_Premium	(null)	01	02	03	39	01	1	1.43
2	Term_20_Premium	(null)	01	02	03	39	02	1	1.43
3	Term_20_Premium	(null)	01	02	03	39	02	2	1.43
4	Term_20_Premium	(null)	01	02	03	39	01	2	1.43
5	Term_20_Premium	(null)	01	02	03	39	02	3	1.43
6	Term_20_Premium	(null)	01	02	03	39	01	3	1.43
7	Term_20_Premium	(null)	01	02	03	39	02	4	1.43
8	Term_20_Premium	(null)	01	02	03	39	01	4	1.43
9	Term_20_Premium	(null)	01	02	03	39	02	5	1.43
10	Term_20_Premium	(null)	01	02	03	39	01	5	1.43
11	Term_20_Premium	(null)	01	02	03	39	02	6	1.43

Note: There is also a MathVariable type named RATE, which functions the same as RATEARRAY.

RATEARRAY Configuration

All criteria needed for any call to the RATEARRAY function must be included in the query parameters and taken from what was entered for the rates. The <Parameter> tag values must be defined in the MathVariables section of configuration prior to use in the RATEARRAY function. Examples of commonly used parameters are RateDescription, ActiveFromDate and EffectiveDate.

Note: If you list a parameter, the value must be the correct datatype or it will process in error. You will also receive an error if the value is null.

Parameters

- **REQUIRED: RateDescription** from AsRateGroup defines the set of rates associated to that description. There must be a perfect match between the value of the MathVariable and the data in the table column. There can be no wildcard searches or partial name searches.
- **REQUIRED: EffectiveDate** from AsRateGroup, which is the Rate Activation Date used to group rates by date.

Note: The variable value will be greater than or equal to the column data that results from the execution of RATEARRAY. For multiple entries in AsRateGroup with the same name, at no time will two sets of rates be available for a given rate active date. And at no time can there be a given rate active date without a rate set. The possible scenarios are below in the “Examples of EffectiveDate/ActiveDate Combinations”

- **ActiveDate:** ActiveToDate from AsRateGroup. The variable value will be greater than or equal to the column data that results from the execution of RATEARRAY. The possible scenarios are below in the “Examples of EffectiveDate/ActiveDate Combinations”
- **RateCriteria:** The parameters are from AsRateGroup and provide an additional way to filter rate data. The names of the parameters are stored in AsRateGroup, the values are in AsRate. For example, the value of “Gender” for Criteria1, “Tobacco” for Criteria2 and “Age” for IntegerCriteria. Therefore, the three NAME attributes value in any order will be “Gender”, “Tobacco” and “Age”. The data type for the variables feeding data to all of the criteria must be TEXT except for IntegerCriteria, as it must be an INTEGER.

Start and End Index Attribute

The RATEARRAY function has a start and end index attribute that determines the records that are retrieved based on either the integer or date criteria start and end index. Your start and end index can be configured to best meet the needs of the business. Hence if you only wanted to retrieve rates for ages 30 to 39, you should use these attributes.

The StartIndex variable indicates what IntegerCriteria in AsRate you are going to begin with when retrieving values for your RateArray. The array index for identification begins with 0. The StartIndex has no effect on the actual array index.

The EndIndex indicates the last value in IntegerCriteria that will be used in the function. The EndIndex does not include the last value to be retrieved so you need to set the EndIndex value to be one greater than the length of the array.

RATE ARRAY XML Example

```
<MathVariables>
  <MathVariable VARIABLENAME="RateTableDescription" TYPE="VALUE">ABCTerm</MathVariable>
  <MathVariable VARIABLENAME="ActivityEffectiveDate" TYPE="FIELD">Activity:EffectiveDate</MathVariable>
  <MathVariable VARIABLENAME="RateActivationDate" TYPE="VALUE">01/01/2008</MathVariable>

  <MathVariable VARIABLENAME="RateArray" TYPE="RATEARRAY" STARTINDEX="StartIndex" ENDINDEX="EndIndex">
    <Parameters>
      <RateDescription>RateTableDescription</RateDescription> REQUIRED
      <EffectiveDate>ActivityEffectiveDate</EffectiveDate> REQUIRED
      <ActiveDate>RateActivationDate</ActiveDate> REQUIRED
      <RateCriteria NAME="Gender">Gender</RateCriteria>
      <RateCriteria NAME="Tobacco">Tobacco</RateCriteria>
    </Parameters>
  </MathVariable>
</MathVariables>
```

Policy Print Example

```
<MathVariable VARIABLENAME="EffectiveDate" TYPE="FIELD">Activity.EffectiveDate</MathVariable>
<MathVariable VARIABLENAME="SegmentRateActiveDate" TYPE="SEGMENTFIELD"
SOURCEARRAY="BaseSegmentGUIDs">SegmentRateActiveDate</MathVariable>
<MathVariable VARIABLENAME="CVTableDescription" TYPE="VALUE">PPT10ROPUGCVRatesHigh</MathVariable>
<MathVariable VARIABLENAME="ROPRoleCollection" TYPE="COLLECTION">SELECT AsRoleField.FieldName,CASE
AsRoleField.FieldTypeCode WHEN '01' THEN FNCFORMATDATE(AsRoleField.DateValue) WHEN '02' THEN AsRoleField.TextValue
WHEN '03' THEN CAST(AsRoleField.IntValue AS CHAR(12)) WHEN '04' THEN CAST(AsRoleField.FloatValue AS CHAR(12)) END FROM
AsRoleField JOIN AsRole ON AsRole.RoleGUID = AsRoleField.RoleGUID AND AsRole.RoleCode = '37' JOIN AsSegment ON
AsSegment.SegmentGUID = AsRole.SegmentGUID AND AsSegment.PolicyGUID = '[PolicyGUID]' JOIN AsSegmentName ON
AsSegmentName.SegmentNameGUID = AsSegment.SegmentNameGUID AND AsSegmentName.TypeCode = '50' JOIN AsSegmentField
ON AsSegmentField.SegmentGUID = AsSegment.SegmentGUID AND AsSegmentField.FieldName = 'SegmentActiveCode' AND
AsSegmentField.TextValue NOT IN ('04') WHERE AsRoleField.FieldName IN
('InsuredIssueAge','InsuredRateGender','InsuredUWClass')</MathVariable>
<MathVariable VARIABLENAME="ROPUWClass" TYPE="COLLECTIONVALUE"
KEY="InsuredUWClass">ROPRoleCollection</MathVariable>
<MathVariable VARIABLENAME="ROPRateGender" TYPE="COLLECTIONVALUE"
KEY="InsuredRateGender">ROPRoleCollection</MathVariable>
<MathVariable VARIABLENAME="ROPIssueAge" TYPE="COLLECTIONVALUE"
KEY="InsuredIssueAge">ROPRoleCollection</MathVariable>
<MathVariable VARIABLENAME="EndIndex" TYPE="EXPRESSION"
ROUND="0">CalendarYearsDiffOf(IssueDate,ROPEndowmentDate)</MathVariable>
<MathVariable VARIABLENAME="CVRateArray" TYPE="RATEARRAY" STARTINDEX="PolicyYear" ENDINDEX="EndIndex + 1">
  <Parameters>
    <RateDescription>CVTableDescription</RateDescription>
    <EffectiveDate>EffectiveDate</EffectiveDate>
    <ActiveDate>SegmentRateActiveDate</ActiveDate>
    <RateCriteria NAME="IssueAge">ROPIssueAge</RateCriteria>
    <RateCriteria NAME="Gender">ROPRateGender</RateCriteria>
    <RateCriteria NAME="UWClass">ROPUWClass</RateCriteria>
  </Parameters>
</MathVariable>
```

Manipulating an Array of Rates

After retrieving an array of rates, you may want to further manipulate the array. OIPA has various functions that can be used in conjunction with RATEARRAY.

The available array functions in v7 are

- AggregateFunction
- ArrayFunction
- Expand
- InsertItem & InsertItems
- RemoveItem & RemoveItems
- Transform

For further information regarding how these functions operate, please enroll in the Oracle Insurance Policy Administration Configuration IV course offered by Oracle University.

Repricing

Sometimes a re-price may occur after the initial product is launched. The processes and structure of the product may not change, but the re-price could change some aspects of the coverage such as the base coverage rates.

In order for OIPA to use a set of different rates from the original product, you must use a combination of the **EffectiveDate**, **ActiveFromDate** and **ActiveToDate** in the AsRateGroup table. Everything else in the AsRateGroup table will remain the same. The date information, along with the Rate Description and criteria for the Rate Group, is used by a configurator to point to the correct stored values in the AsRate table. Regardless of how many entries for a single Rate Description are entered into AsRateGroup, the differentiating rate group information needs to be entered to ensure that only one set of rates will be valid for any policy.

EffectiveDate will determine the set of rates to retrieve based on the RateDescription. It also determines for a specific policy which rates are “active,” meaning that they are valid based on a set date value. This is most commonly a PolicyField or a SegmentField. In the ACME products, it is a SegmentField called SegmentActiveDate. OIPA will refer to the policy and plan effective dates when identifying what RateGroups are applicable according to the EffectiveDate.

Once a set of RateGroups is retrieved that have the required EffectiveDate (there may be a few at this point), the list is further narrowed down based on the Activity date, which will use ActiveFromDate and ActiveToDate to determine the applicable rates.

RateGroup selection SQL Pseudo Code that is generated:

```
SELECT * FROM AsRateGroup
WHERE rateDescription = [RateDescription]
AND effectiveDate <= [EffectiveDate]
AND activeFromDate <= [ActiveDate]
AND ( activeToDate >= [ActiveDate] OR activeToDate is null )
```

Repricing for New Policies –vs – Repricing for Existing and New Policies

By using Rate Groups, actuaries are given the opportunity to do one of the following:

- Re-price a product for an existing benefit that sets new rates for a product, but the new rates are **used only for new policies after a given date**. Existing policies will still use the old set of rates. See Example 1 below.
- Re-price a product where the old rates are no longer used from a certain date forward, and **both old and new policies use a set of rates from that given date**. See Example 2 below.

In either case, only one set of rates is valid for a policy at any one time.

Example One: Two Sets of Rates are Open

Re-price a product for an existing benefit that sets new rates for a product, but the new rates are **used only for new policies after a given date**. Existing policies will still use the old set of rates.

- For example, a base segment was priced using rate set A with an original effective date of 1/1/2000 and no backdating. The Rate Group name is **BaseRates**. It is determined four years later that the rates should be increased to be more profitable with a re-price effective date of 1/1/2004, again with no backdating. Further suppose that it is decided that already existing policies should not be affected and they should continue to use the existing BaseRates rates that have been loaded.
- In this case, BaseRates is the appropriate Rate Description for the rates and by continuing to use the same description, configuration to perform the rate lookup will not need to change as the RateDescription will stay the same.
- In the database, there will exist two entries in AsRateGroup, both called BaseRates but with different EffectiveDates also known as Rate Activation Date.
- The second set of rates will follow the same Rate Upload procedures as above. The RateDescription used for the second set of rates will be BaseRates the same name as the original set of rates, but will be loaded with a different Rate Activation Date and a different ActiveFromDate
- After entering the Rate Group information, AsRateGroup would hold the following information. For this example, no criteria is used other than IntegerCriteria. If additional criteria were used, all criteria names must match between the old and new rates.

Mapping it Out

Line one is used for RateDescription=**BaseRates** where policies have an EffectiveDate from 1/1/2000 (inclusive) to 1/1/2004 (exclusive).

Line two is used for RateDescription=**BaseRates** where policies have an EffectiveDate from 1/1/2004 (inclusive) forward.

Line	RateDescription	EffectiveDate / RateActivationDate	ActiveFromDate	ActiveToDate
1	BaseRates	1/1/2000	1/1/2000	
2	BaseRates	1/1/2004	1/1/2004	

Backdating

Should backdating of activities be allowed, then the ActiveFromDate would not match the EffectiveDate/RateActivationDate. An example where three months of backdating is allowed would look like this:

Line	RateDescription	EffectiveDate / RateActivationDate	ActiveFromDate	ActiveToDate
1	BaseRates	1/1/2000	10/1/1999	
2	BaseRates	1/1/2004	10/1/2003	

Example 1 Illustration of the AsRateGroup Table in the OIPA Database

	RateGroupGUID	RateDescription	IntegerCriteria
1	B2225C5B-8CF0-056F-1184-02BCDAA322B6	BaseRates	Age
2	4DBF1E7F-1777-3928-76EB-F1CDA976A26A	BaseRates	Age

	Criteria1	Criteria2	Criteria3	Criteria4	Criteria5
1	(null)	(null)	(null)	(null)	(null)
2	(null)	(null)	(null)	(null)	(null)

	Criteria6	Criteria7	Criteria8	Criteria9	Criteria10	StatusCode
1	(null)	(null)	(null)	(null)	(null)	(null)
2	(null)	(null)	(null)	(null)	(null)	(null)

	TypeCode	EffectiveDate	ActiveFromDate	ActiveToDate	XMLData
1	(null)	1/1/2000 12:00:00 AM	1/1/2000 12:00:00 AM	(null)	(null)
2	(null)	1/1/2004 12:00:00 AM	1/1/2004 12:00:00 AM	(null)	(null)

- For Activities on policies whose EffectiveDate is between 1/1/2000 and 12/31/2003 in this case, the original rates will be used.
- For Activities on all policies whose EffectiveDate is on or after 1/1/2004, the new re-priced rates will be used.
- Appendix A (Original Rates) and Appendix B (re-priced rates)** shows examples using the information in AsRateGroup as given above where two set of rates for a rate group are open at one time.

Example Two: Only One Set of Rates is Open for BaseRates

Re-price a product where the old rates are no longer used from a certain date forward, and **both old and new policies use a set of rates from that given date.**

- For example, a base segment was priced using rate set A and named the Rate Group **BaseRates**. It is determined four years later that the rates should be increased to be more profitable. Further suppose that it is decided that already existing policies should also have their rates increased and their current rates will no longer be valid at a given date.
- In this case, BaseRates is the appropriate Rate Description for the rates. Since you are continuing to use the same description, you will configure the rate lookup the same way.
- In the database, there will exist two entries in AsRateGroup, both called BaseRates but with different EffectiveDates. In addition, the rates that will no longer be in affect will have the ActiveToDate filled in.
- After entering the Rate Group information, AsRateGroup will hold the following information. For this example, no criteria is used other than IntegerCriteria. If criteria were used, all criteria names must match between the old and new rates.

Mapping it Out

Line one is used from RateDescription=BaseRates where policies have an EffectiveDate from 1/1/2000 (inclusive) forward but only for activities with effective dates between 1/1/2000 (inclusive) and 12/31/2003 (inclusive).

Line two is used from RateDescription=BaseRates where policies have an EffectiveDate from 1/1/2000 (inclusive) forward but only for activities with effective dates on or after 1/1/2004.

Line	RateDescription	EffectiveDate / RateActivationDate	ActiveFromDate	ActiveToDate
1	BaseRates	1/1/2000	1/1/2000	12/31/2003
2	BaseRates	1/1/2000	1/1/2004	

Example 2 Illustration of the AsRateGroup Table in the OIPA Database

	RateGroupGUID	RateDescription	IntegerCriteria
1	B2225C5B-8CF0-056F-1184-02BCDAA322B6	BaseRates	Age
2	4DBF1E7F-1777-3928-76EB-F1CDA976A26A	BaseRates	Age

	Criteria1	Criteria2	Criteria3	Criteria4	Criteria5
1	(null)	(null)	(null)	(null)	(null)
2	(null)	(null)	(null)	(null)	(null)

	Criteria6	Criteria7	Criteria8	Criteria9	Criteria10	StatusCode
1	(null)	(null)	(null)	(null)	(null)	(null)
2	(null)	(null)	(null)	(null)	(null)	(null)

	TypeCode	EffectiveDate	ActiveFromDate	ActiveToDate	XMLData
1	(null)	1/1/2000 12:00:00 AM	12/1/1999 12:00:00 AM	12/31/2003 12:00:00 AM	(null)
2	(null)	1/1/2000 12:00:00 AM	1/1/2004 12:00:00 AM	(null)	(null)

- Activities on policies between 1/1/2000 and 12/31/2003 whose RateActivationDate is between 1/1/2000 and 12/31/2003 in this case will use the original rates.
- Activities on all policies that occur on or after 1/1/2004 will use the new re-priced rates.
- **Appendix C (One set of rates open at any one time)** shows examples using the information in AsRateGroup as given above.

Appendix A - Entering Original RateGroup and Rates

The Original BaseRates Rate Group was entered as follows:

Rate Group

Rate Group: BaseRates(Activated 01/01/2000 - Transaction From 01/01/2000)

Name: BaseRates

Integer Criteria: Age

Rate Activation Date: 01/01/2000

Transaction From Date: 01/01/2000

Transaction To Date:

Criteria 1: Criteria 2: Criteria 3: Criteria 4: Criteria 5: Criteria 6: Criteria 7: Criteria 8: Criteria 9: Criteria 10:

Save Close

The Original BaseRates were loaded as indicated in Appendix E. A subset of those rates is given here. For this example, it is assumed that only one RateGroup for BaseRates has been loaded.

Rate Table

Rate List: BaseRates

Name: BaseRates

Date:

Intgr Criteria:

Rate:

Criteria 1: Criteria 2: Criteria 3: Criteria 4: Criteria 5: Criteria 6: Criteria 7: Criteria 8: Criteria 9: Criteria 10:

Description Upload Close

Rate Description	Rate Activation Date	Transaction From Date	Transaction To Date
BaseRates	01/01/2000	01/01/2000	

ORIGINAL RateGroup		
Age	Rate	(index)
5	0.1035	0
6	0.1035	1
7	0.1035	2
8	0.1035	3
9	0.1035	4
10	0.1035	5
11	0.1035	6
12	0.1035	7
13	0.1035	8
14	0.1035	9

ORIGINAL RateGroup		
Age	Rate	(index)
19	0.1035	14
20	0.1035	15
21	0.1035	16
22	0.1035	17
23	0.1035	18
24	0.1035	19
25	0.1035	20
26	0.1035	21
27	0.1035	22
28	0.1035	23

15	0.1035	10	29	0.1035	24
16	0.1035	11	30	0.1035	25
17	0.1035	12	31	0.115	26
18	0.1035	13	32	0.115	27

A policy input for an original rate group example is given below. The **Issue Date** of the policy will be used as the policy's "rate activation date" and will match up to the EffectiveDate column in AsRateGroup.

Note: The Test XML that was used appears in Appendix D.

Case One:

The following was retrieved for an activity on 2/2/2000. The Search Name is BaseRates and this is the Original Test Policy case.

Calculated Fields			
	Field	Type	Value
	RateTableDescription		BaseRates
	ActivityEffectiveDate		02/02/2000
	PolicyIssueDate		02/02/2000
	StartIndex		0
	EndIndex		65
	BaseRatesArray		[Array]
	FirstValue		0.1035
	TwentySeventhValue		0.115

The PolicyIssueDate MathVariable below will be used to represent the policy's EffectiveDate (RateActivationDate) while the ActivityEffectiveDate will be used for the ActiveDate (ActiveFromDate/ActiveToDate test).

Case Two:

The following was retrieved for an activity on 2/21/2004. The Search Name was BaseRates and this is for the Original Test Policy case.

Calculated Fields		
Field	Type	Value
RateTableDescription		BaseRates
ActivityEffectiveDate		02/21/2004
PolicyIssueDate		02/02/2000
StartIndex		0
EndIndex		65
BaseRatesArray		[Array]
FirstValue		0.1035
TwentySeventhValue		0.115

Appendix B - Re-priced Rates with Original Rate Group Not Closed

The re-priced BaseRates Rate Group was entered as follows:

Rate Group

Rate Group: BaseRates(Activated 01/01/2004 - Transaction From 01/01/2004)

Name: BaseRates Criteria 1: Criteria 2: Criteria 3: Criteria 4: Criteria 5: Criteria 6: Criteria 7: Criteria 8: Criteria 9: Criteria 10:

Integer Criteria: Age

Rate Activation Date: 01/01/2004

Transaction From Date: 01/01/2004

Transaction To Date:

Save Close

The re-priced BaseRates were loaded as indicated in Appendix E. A subset of those rates is given here. Now, two RateGroups exist for BaseRates.

Rate Table

Rate List: BaseRates Criteria 1: Criteria 2: Criteria 3: Criteria 4: Criteria 5: Criteria 6: Criteria 7: Criteria 8: Criteria 9: Criteria 10:

Name: BaseRates

Date:

Intgr Criteria:

Rate:

Description Upload Close

Rate Description	Rate Activation Date	Transaction From Date	Transaction To Date
BaseRates	01/01/2000	01/01/2000	
BaseRates	01/01/2004	01/01/2004	

ORIGINAL RateGroup			Re-Priced RateGroup	
Age	Rate	(index)	Age	Rate
5	0.1035	0	5	0.102155
6	0.1035	1	6	0.102155
7	0.1035	2	7	0.102155
8	0.1035	3	8	0.102155
9	0.1035	4	9	0.102155
10	0.1035	5	10	0.102155
11	0.1035	6	11	0.102155
12	0.1035	7	12	0.102155

ORIGINAL RateGroup			Re-Priced RateGroup	
Age	Rate	(index)	Age	Rate
19	0.1035	14	19	0.102155
20	0.1035	15	20	0.102155
21	0.1035	16	21	0.102155
22	0.1035	17	22	0.102155
23	0.1035	18	23	0.102155
24	0.1035	19	24	0.102155
25	0.1035	20	25	0.102155
26	0.1035	21	26	0.102155

13	0.1035	8	13	0.102155
14	0.1035	9	14	0.102155
15	0.1035	10	15	0.102155
16	0.1035	11	16	0.102155
17	0.1035	12	17	0.102155
18	0.1035	13	18	0.102155

27	0.1035	22	27	0.102155
28	0.1035	23	28	0.102155
29	0.1035	24	29	0.102155
30	0.1035	25	30	0.102155
31	0.115	26	31	0.113505
32	0.115	27	32	0.113505

A Policy Input for a re-price policy example is given below. The **Issue Date** of the policy will be used as the policy's "rate activation date" and will match up to the EffectiveDate column in AsRateGroup.

Policy	Allocations	Clients	Roles	Segments	Activities	Policy Values
Company: Acme Life Plan Group: Variable Plan Group Plan: Variable Annuity Entry Date: 02/21/2004 Search Name: BaseRates Reprice Policy Status: Pending Issue State: CT Policy Number: AVA3101020939						
Statutory Company: Acme Life Option Groups: ContractBSeries Effective Date: Sweep Date: GMAB Rider: <input checked="" type="radio"/> No <input type="radio"/> Yes Qual Type: IRA Issue Date: 02/21/2004 FreeLook End Date:						
<div> <div>Save</div> <div>Activity</div> <div>Allocate</div> <div>Values</div> <div>New</div> <div>Close</div> </div>						

Note: The Test XML that was used appears in Appendix D.

Case One - Re-Priced Policy:

For any activity prior to its RateActivationDate, no values would be retrieved (i.e. passing in this policy's EffectiveDate of 2/21/2004 with an ActiveDate (activity effective date) prior to the effective date of the rate group, such as 12/31/2003).

Case Two - Re-Priced Policy:

The following was retrieved for an activity on 2/21/2004: The Search Name is BaseRates and this is the Re-price Test Policy case.

Calculated Fields		
Field	Type	Value
RateTableDescription		BaseRates
ActivityEffectiveDate		02/21/2004
PolicyIssueDate		02/21/2004
StartIndex		0
EndIndex		65
BaseRatesArray		[Array]
FirstValue		0.1021545
TwentySeventhValue		0.113505

The PolicyIssueDate MathVariable will be used to represent the policy's EffectiveDate (RateActivationDate) while the ActivityEffectiveDate will be used for the ActiveDate (ActiveFromDate/ActiveToDate test).

Appendix C - Original BaseRates Entry After Group Closure

After the Original BaseRates Rate Group was closed as of 1/1/2004, the RateGroup information looked as follows:

Rate Group

Rate Group: BaseRates(Activated 01/01/2000 - Transaction From 01/01/2000)

Name: BaseRates

Integer Criteria: Age

Rate Activation Date: 01/01/2000

Transaction From Date: 01/01/2000

Transaction To Date: 12/31/2003

Criteria 1: Criteria 2: Criteria 3: Criteria 4: Criteria 5: Criteria 6: Criteria 7: Criteria 8: Criteria 9: Criteria 10:

Save Close

The Re-priced RateGroup information was loaded as:

Rate Group

Rate Group: BaseRates(Activated 01/01/2000 - Transaction From 01/01/2004)

Name: BaseRates

Integer Criteria: Age

Rate Activation Date: 01/01/2000

Transaction From Date: 01/01/2004

Transaction To Date:

Criteria 1: Criteria 2: Criteria 3: Criteria 4: Criteria 5: Criteria 6: Criteria 7: Criteria 8: Criteria 9: Criteria 10:

Save Close

Summary View:

Rate Table

Rate List: BaseRates

Name: BaseRates

Date:

Intgr Criteria:

Rate:

Criteria 1: Criteria 2: Criteria 3: Criteria 4: Criteria 5: Criteria 6: Criteria 7: Criteria 8: Criteria 9: Criteria 10:

Description Upload Close

Rate Description	Rate Activation Date	Transaction From Date	Transaction To Date
BaseRates	01/01/2000	01/01/2000	12/31/2003
BaseRates	01/01/2000	01/01/2004	

The Original and Re-priced BaseRates were loaded as given in Appendix E (the same rates as used in previous examples).

A Policy Input for an Original example is given below. The Issue Date of the policy will be used as the policy's "rate activation date" and will match up to the EffectiveDate column in AsRateGroup.

Splash Page	Policy	Allocations	Clients	Roles	Segments	Activities	Policy Values
Company: Acme Life Plan Group: Variable Plan Group Plan: Variable Annuity Entry Date: 02/02/2000 Search Name: BaseRates Orig Test Policy Status: Pending Issue State: CT Policy Number: AVA3101020938							
Statutory Company: Acme Life Option Groups: ContractBSeries Effective Date: Sweep Date: Qual Type: IRA Issue Date: 02/02/2000 FreeLook End Date: GMAB Rider: <input checked="" type="radio"/> No <input type="radio"/> Yes							
<div>Save</div> <div>Activity</div> <div>Allocate</div> <div>Values</div> <div>New</div> <div>Close</div>							

Case One:

The following was retrieved for an activity on 2/2/2000. The Search Name is BaseRates and is the Original Test Policy case.

The PolicyIssueDate MathVariable below will be used to represent the policy's EffectiveDate (RateActivationDate) while the ActivityEffectiveDate will be used for the ActiveDate (ActiveFromDate/ActiveToDate test).

Calculated Fields			
	Field	Type	Value
	RateTableDescription		BaseRates
	ActivityEffectiveDate		02/02/2000
	PolicyIssueDate		02/02/2000
	StartIndex		0
	EndIndex		65
	BaseRatesArray		[Array]
	FirstValue		0.1035
	TwentySeventhValue		0.115

Case Two:

The following was retrieved for an activity on 2/21/2004. The Search Name is BaseRates for the Original Test Policy case.

Calculated Fields			
	Field	Type	Value
	RateTableDescription		BaseRates
	ActivityEffectiveDate		02/21/2004
	PolicyIssueDate		02/02/2000
	StartIndex		0
	EndIndex		65
	BaseRatesArray		[Array]
	FirstValue		0.1021545
	TwentySeventhValue		0.113505

For the Re-price Policy (Search Name: BaseRates Orig Test Policy), the following would occur:

Splash Page	Policy	Allocations	Clients	Roles	Segments	Activities	Policy Values
<p> Company: Acme Life Plan Group: Variable Plan Group Plan: Variable Annuity Entry Date: 02/21/2004 Search Name: BaseRates Reprice Policy Status: Pending </p> <p> Issue State: CT Policy Number: AVA3101020939 </p>							
<p>Policy</p> <p> Statutory Company: Acme Life Option Groups: ContractBSeries Effective Date: Sweep Date: GMAB Rider: <input checked="" type="radio"/> No <input type="radio"/> Yes </p> <p> Qual Type: IRA Issue Date: 02/21/2004 FreeLook End Date: </p> <p> <input type="button" value="Save"/> <input type="button" value="Activity"/> <input type="button" value="Allocate"/> <input type="button" value="Values"/> <input type="button" value="New"/> <input type="button" value="Close"/> </p>							

Case One - Re-Priced Policy:

The following was retrieved for an activity on 12/31/2003:
The Search Name is BaseRates and this is the Original Test Policy case with the old rates

Calculated Fields			
	Field	Type	Value
	RateTableDescription		BaseRates
	ActivityEffectiveDate		12/31/2003
	PolicyIssueDate		02/21/2004
	StartIndex		0
	EndIndex		65
	BaseRatesArray		[Array]
	FirstValue		0.1035
	TwentySeventhValue		0.115

Case Two - Re-Priced Policy:

The following was retrieved for an activity on 2/21/2004.
The Search Name is BaseRates and this is the Original Test Policy case with new rates.

Calculated Fields			
	Field	Type	Value
	RateTableDescription		BaseRates
	ActivityEffectiveDate		02/21/2004
	PolicyIssueDate		02/21/2004
	StartIndex		0
	EndIndex		65
	BaseRatesArray		[Array]
	FirstValue		0.1021545
	TwentySeventhValue		0.113505

Appendix D - XML Used for Rate Retrieval Examples

Note: The PolicyIssueDate MathVariable below will be used to represent the policy's EffectiveDate (RateActivationDate) while the ActivityEffectiveDate will be used for the ActiveDate (ActiveFromDate/ActiveToDate test).

```
<Transaction>
  <EffectiveDate STATUS="Enabled" TYPE="SYSTEM"/>
  <Math>
    <MathVariables>
      <MathVariable VARIABLENAME="RateTableDescription" TYPE="VALUE" DATATYPE="TEXT">BaseRates</MathVariable>
      <MathVariable VARIABLENAME="ActivityEffectiveDate" TYPE="EXPRESSION"
DATATYPE="DATE">Activity:EffectiveDate</MathVariable>
      <MathVariable VARIABLENAME="PolicyIssueDate" TYPE="POLICYFIELD" DATATYPE="DATE">PolicyIssueDate</MathVariable>
      <MathVariable VARIABLENAME="StartIndex" TYPE="VALUE" DATATYPE="INTEGER">0</MathVariable>
      <MathVariable VARIABLENAME="EndIndex" TYPE="VALUE" DATATYPE="INTEGER">65</MathVariable>
      <!-- initialize returns -->
      <MathVariable VARIABLENAME="BaseRatesArray" TYPE="RATEARRAY" STARTINDEX="StartIndex" ENDINDEX="EndIndex"
DATATYPE="DECIMAL">
        <Parameters>
          <RateDescription>RateTableDescription</RateDescription>
          <EffectiveDate> PolicyIssueDate</EffectiveDate>
          <ActiveDate> ActivityEffectiveDate </ActiveDate>
        </Parameters>
      </MathVariable>
      <MathVariable VARIABLENAME="FirstValue" TYPE="AGGREGATEFUNCTION" DATATYPE="DECIMAL" METHOD="INDEX"
INDEX="0">BaseRatesArray</MathVariable>
      <MathVariable VARIABLENAME="TwentySeventhValue" TYPE="AGGREGATEFUNCTION" DATATYPE="DECIMAL"
METHOD="INDEX" INDEX="26">BaseRatesArray</MathVariable>
    </MathVariables>
  </Math>
</Transaction>
```

Appendix E - Rates Used for Rate Retrieval Examples

ORIGINAL RateGroup			Re-Priced RateGroup	
Age	Rate	(index)	Age	Rate
5	0.1035	0	5	0.102155
6	0.1035	1	6	0.102155
7	0.1035	2	7	0.102155
8	0.1035	3	8	0.102155
9	0.1035	4	9	0.102155
10	0.1035	5	10	0.102155
11	0.1035	6	11	0.102155
12	0.1035	7	12	0.102155
13	0.1035	8	13	0.102155
14	0.1035	9	14	0.102155
15	0.1035	10	15	0.102155
16	0.1035	11	16	0.102155
17	0.1035	12	17	0.102155
18	0.1035	13	18	0.102155
19	0.1035	14	19	0.102155
20	0.1035	15	20	0.102155
21	0.1035	16	21	0.102155
22	0.1035	17	22	0.102155
23	0.1035	18	23	0.102155
24	0.1035	19	24	0.102155
25	0.1035	20	25	0.102155
26	0.1035	21	26	0.102155
27	0.1035	22	27	0.102155
28	0.1035	23	28	0.102155
29	0.1035	24	29	0.102155
30	0.1035	25	30	0.102155
31	0.115	26	31	0.113505
32	0.115	27	32	0.113505
33	0.115	28	33	0.113505
34	0.115	29	34	0.113505
35	0.115	30	35	0.113505
36	0.115	31	36	0.113505
37	0.115	32	37	0.113505
38	0.115	33	38	0.113505
39	0.115	34	39	0.113505
40	0.115	35	40	0.113505

ORIGINAL RateGroup			Re-Priced RateGroup	
Age	Rate	(index)	Age	Rate
41	0.115	36	41	0.113505
42	0.115	37	42	0.113505
43	0.115	38	43	0.113505
44	0.115	39	44	0.113505
45	0.115	40	45	0.113505
46	0.1265	41	46	0.124856
47	0.1265	42	47	0.124856
48	0.1265	43	48	0.124856
49	0.1265	44	49	0.124856
50	0.1265	45	50	0.124856
51	0.138	46	51	0.136206
52	0.138	47	52	0.136206
53	0.138	48	53	0.136206
54	0.138	49	54	0.136206
55	0.138	50	55	0.136206
56	0.138	51	56	0.136206
57	0.138	52	57	0.136206
58	0.138	53	58	0.136206
59	0.138	54	59	0.136206
60	0.138	55	60	0.136206
61	0.1495	56	61	0.147557
62	0.1495	57	62	0.147557
63	0.1495	58	63	0.147557
64	0.1495	59	64	0.147557
65	0.1495	60	65	0.147557

Appendix F - Pseudo SQL that is Generated:

```
SELECT *  
FROM AsRateGroup  
WHERE rateDescription = [RateDescription]  
AND effectiveDate = (select max(effectivedate)  
FROM asrategroup where ratedescription = [RateDescription]  
AND effectivedate <= [EffectiveDate])  
AND activeFromDate <= [ActiveDate]  
AND ( activeToDate >= [ActiveDate]  
OR activeToDate is null )
```