OLIVER WYMAN

Consulting Actuaries

Volume 10 | **SPRING 2019**

GETTING THE MOST OUT OF AXIS™

IMPROVEMENTS IS THE NAME OF THE GAME

Editor's words: Welcome to the Spring 2019 edition of our AXIS newsletter. This issue outlines the latest developments in AXIS related to FASB's 2018 US GAAP Targeted Improvements guidance for long duration contracts and also discusses a potential model automation structure to improve operational efficiency. You will find helpful tips and tricks for navigating the system and highlights of new features in recent AXIS releases. We hope you enjoy the newsletter.

IN THIS ISSUE

MARSH & McLENNAN COMPANIES

EXECUTIVE CORNER

FASB ACCOUNTING STANDARDS UPDATE FOR LONG DURATION CONTRACTS: WHAT YOU NEED TO KNOW

INTRODUCTION

FASB's 2018 US GAAP ("GAAP") Targeted Improvements ("GTI") guidance for long duration contracts (also commonly referred to as "LDTI"), which will take effect in 2021, represents the most significant revision to life insurance GAAP reporting in decades. There are substantial changes to calculations used in the measurement of long duration liabilities as well as significant increases to the disclosures accompanying them. Implementing changes of this magnitude requires effective management of updates to the end-to-end actuarial model architecture and financial reporting processes.

This article summarizes GTI changes and how Moody's Analytics is updating AXIS to handle them.

"US GAAP Targeted Improvements [...] represents the most significant revision to life insurance GAAP reporting in decades"

SUMMARY OF CHANGES

Exhibit 1 summarizes key changes in GAAP accounting standards for long duration contracts. As indicated in Exhibit 1, FASB's goal is to make the measurement of long duration liabilities less complex and financial statements more transparent. Given the expanded disclosures and current measurement approaches, there will be a considerable impact on data requirements for both the actuarial models and post-model processes. For instance, if a company elects a full retrospective transition approach, all historical experience data as of contract inception is required.

"Given the expanded disclosures and new measurement approaches, there will be a considerable impact on data requirements for both the actuarial models and post-model processes"

Exhibit 1: Summary of GTI key changes

	LIABILITY FOR FUTURE POLICY BENEFITS	DAC	MARKET RISK BENEFITS ("MRBs")	DISCLOSURES
	• Assumptions no longer locked in for FAS 60 business • Discount rate uses an upper-medium grade security with the net effect to flow through other comprehensive income ("OCI") • No provisions for adverse deviation and maintenance expenses for FAS 60 • Loss recognition no longer required for traditional contracts – net premium ratio cap is 100%	• DAC amortized on straight line basis • Amortization based on individual contract or a grouped contract method • Interest no longer credited to balance • Sales inducement assets, unearned revenue also follow new DAC guidance • Terminal dividends accrued and recognized at constant rate basis • DAC is no longer subject to an impairment test	• Protection for more than nominal capital market risk categorized as MRBs • Fair value for all MRBs (including GMIB and GMDB)[1] • Fair value changes attributable to instrument-specific credit risk reported in OCI • Contracts that are not MRBs but meet the embedded derivative ("ED") definition follow ED FV guidance	• Disaggregated roll forwards required for FPBs, PABs, DAC, MRBs and SIAs with disclosures on interest, crediting rates, NAR[2] • Separate presentation of remeasurement gains/losses required in I/S[3] • MRB liabilities reported separately in the B/S and I/S • Additional disclosures required on liability assumptions, judgments and methodology • Qualitative and quantitative disclosures required for transition adjustments
Trad life (term, WL, LTC, DI)	✓	✓	✗	✓
Non trad life (UL, VUL, IUL)	✗	✓	✗	✓
Payout annuities	✓	✓	✗	✓
Variable annuities	✗	✓	✓	✓
Fixed annuities	✗	✓	✓	✓

1 GMIB: guaranteed minimum income benefit, GMDB: guaranteed minimum death benefit
2 FPBs: future policy benefits, PABs: policyholder account balances, SIAs: sales inducement assets, NAR: net amount at risk
3 B/S: balance sheet, I/S: income statement

AXIS UPDATES

Moody's Analytics is updating AXIS to support the changes described in the previous section. In addition to enhancing the existing Base Liability Modules, new US GAAP Link Modules are being created to address some of the new challenges presented by GTI, such as current period reporting. Exhibit 2 contains an overview of key AXIS enhancements.

Exhibit 2: Key AXIS GTI-related enhancements

Framework redesign	GAAP calculations	GAAP reporting
• Creation of a US GAAP Cohort object, which will warehouse assumptions, methods and available options for each cohort • Enhancements to the US GAAP section of Cells and Subfunds to handle new methodology requirements while retaining current functionality • Expansion of the US GAAP tab in Dataset Parameters • Creation of US GAAP Link Module to manage cohort historic data and assumptions	• Updates to calculation methodology of liabilities for future policy benefits, DAC and MRBs • Applying fair value calculations to FIAs and a broader group of variable annuities (the necessary assumptions and stochastic calculations already exist within AXIS)	• Enhancements to support transparency of reporting under GTI • Changing presentation of GAAP financials to reflect GTI • Automation of data warehouse updates and disclosure reporting under the new guidance

Those planning GTI implementation timelines should be cognizant of the above enhancements. Exhibit 3 illustrates how these enhancements could fit within an insurer's scheduled GTI implementation.

We recommend reviewing and testing new US GAAP functionality as it becomes available in AXIS and communicating proactively with Moody's Analytics to ensure your feedback and requirements are met by planned system enhancements.

TIPS & TRICKS

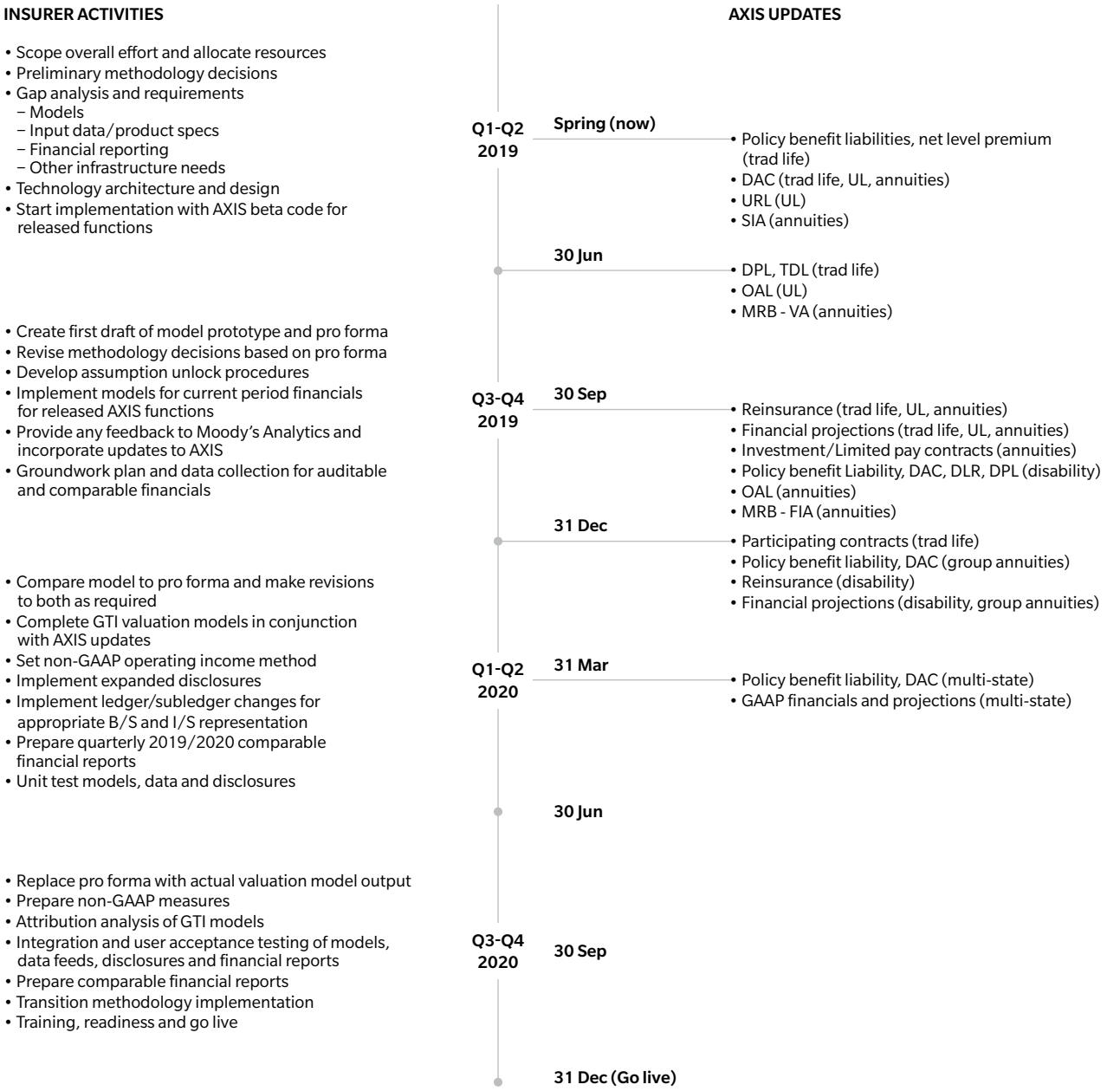
Performing Dataset comparison in Version Control Project

An AXIS Version Control Project allows users to keep an audit trail for updates made to a Dataset. The functionality allows users to save past versions of the Dataset without having to store multiple copies of the Dataset. This functionality can be leveraged as part of a broader model development control process.

To review differences between two versions, the user can perform a Dataset comparison with the following steps:

1. Select the Dataset in the Version Control Project and right-click to select "View History"
2. Choose the two Dataset versions to compare by holding down the 'Ctrl' key while selecting Datasets (both Datasets need to be on the same AXIS version)
3. Right-click and select "Compare versions"; AXIS will generate a report showing differences between the two versions of the Dataset

Exhibit 3: Aligning the GTI implementation timeline with AXIS updates



CONCLUSION

The next few months are crucial as insurers complete planning, requirements and gap analysis as they prepare for mandatory implementation of the changes by 2021.

While FASB's new guidance will significantly impact insurers' modeling and reporting activities, a plan that is aligned with Moody's Analytics' strategy for enhancing AXIS will help ensure the overall efficiency of the implementation while also creating a sustainable environment for future model or process enhancements.

IN THE SPOTLIGHT

AUTOMATING AXIS PROCESSING: A CASE STUDY

INTRODUCTION

Model development and maintenance become increasingly complex as companies seek to centralize modeling functions. When multiple actuarial functions rely on the same model, this necessitates parameterization of the model for each use case.

"When multiple actuarial functions rely on the same model, this necessitates parameterization of the model for each use case"

Certain model users may be unfamiliar with the technical details of the platform and desire a way to manage model processing externally via a front-end, user-friendly tool. Further, developers may desire a way to externalize certain inputs to increase process flexibility and scalability, or to reduce model size and complexity.

This article examines a potential development path to achieve such goals. In the case study below, an Excel file is used to create multiple comma-delimited text files (.csv) that are saved with the Dataset and read by AXIS through Dataset Formula Batches to execute specific functions. Excel represents one of various potential front-end tools and serves as a convenient illustrative example given the prevalence of its use; however, many other viable approaches exist including Moody's functionality available through FormulaLink and EnterpriseLink jobs.

CASE STUDY BACKGROUND

Company ABC has a centralized model development team. Users from different actuarial functions would like a single Excel interface where the following components can be managed. The following section outlines a framework for developing model automation and subsequent sections address the three specific components noted below.

COMPONENTS	DETAILS
A. Settings	End users would like to manage Dataset Parameters, specifically, the Valuation Date and Reinvestment Frequency, outside of AXIS
B. Inputs	Company ABC has a master Excel spreadsheet where it warehouses the final version of all valuation and projection assumptions. End users would like to be able to read these tables into MS Access and then import them into AXIS
C. Calculations and processing	Certain end users care only about certain lines of business and processing steps. They would like to turn on/off specific model commands (such as running DataLink or processing certain lines of business) or reorder steps, as necessary, for purposes of model attribution

AUTOMATION FRAMEWORK

The following steps are performed for each component to be automated.

STEP	DESCRIPTION
1	Identify commands in Dataset Formula Batch code that can be run to perform the desired actions
2	Create an Excel file template where users can enter necessary information to execute commands
3	Specify a storage location and file names for .csv files to be created from Excel
4	Write code to read the required information and execute the identified commands

READING INFORMATION INTO AXIS

Before discussing the Excel interface in further detail, a primer will be provided on reading information from a .csv file into AXIS. Exhibit 1 below provides sample code that can be utilized to check if the specified .csv file can be accessed and read. Then each line of the .csv file is read sequentially.

Exhibit 1: Sample code to read .csv

```
1
2  dim v_path as String
3  dim v_name as String
4  dim i_array(#) as string            'array size should be 1 less than # of columns used in Excel
5
6  v_path = "$PATH\$DATASET\DOCS\"     'path should end with a backslash
7  v_name = "$DatasetParameters.csv"
8
9  'Ensure access to file
10 IF CanReadFile(v_path & v_name) THEN
11
12             'Enable read ability
13             USING MyReader as New FileIO.TextFieldParser(v_path & v_name)
14
15                         'Set paramters for file being read
16                         MyReader.TextFieldType = FileIO.FieldType.Delimited
17                         MyReader.SetDelimiters(",")
18
19                         'Read file line by line
20                         WHILE NOT MyReader.EndOfData
21                                    i_array = MyReader.ReadFields()
22
23                                    '''Code to execute based on information in csv'''
24
25                         END WHILE
26
27             END USING
28
29 END IF
```

Additional code snippets shown in article placed here

COMPONENT A: SETTINGS

STEP 1

Exhibit 2 shows the specified commands and their input description and requirements.

Exhibit 2: Sample commands for setting Dataset Parameters

DATASET PARAMETER	FUNCTION	INPUT DESCRIPTION
Valuation Date	SetValuationDate(ValuationDate)	Sets the Valuation Date to the parameter specified in YYYYMM format
Reinvestment Frequency	SetReinvFrequency(RFrequency, Months, Quarters)	The Reinvestment Frequency can be entered as an integer or utilizing specific strings (this example will utilize "RFREQ_M_Q" which corresponds to "Monthly for M months, quarterly for Q quarters, then annual"). Months (M) and Quarters (Q) are entered numerically

STEP 2

An Excel file can allow users to input their desired Valuation Date and Reinvestment Frequency.

Exhibit 3: Sample Excel input file for setting Dataset Parameters

	A	B	C	D
1	Valuation date	2018	12	
2	Reinvestment frequency	120	0	
3				
4				
5				
6				

STEP 3

A location is specified to save the above Excel tab as a .csv file. In this example, the location will be the DOCS folder within the Dataset and AXIS will be able to identify the path to this location simply as "$PATH\$DATASET\DOCS\".

STEP 4

The code shown in Exhibit 4 should be placed in the section identified in Exhibit 1. Note that the array defined in this code should be parameterized correctly. Many other Dataset Parameters can be similarly set in this manner. See the Action section in the leftmost browser of the DataLink Formula Batch.

Exhibit 4: Sample execution code for setting Dataset Parameters

```
'0 : Dataset Parameter
'1 : Input variable 1
'2 : Input variable 2

SELECT CASE i_array(0)
    CASE IS = "Valuation date"
        'Date should have been entered in Excel as YYYY and MM
        Call SetValuationDate(i_array(1) & i_array(2))
    CASE IS = "Reinvestment frequency"
        'Months and quarters should have been entered in Excel
        Call SetReinvFrequency(RFREQ_M_Q, i_array(1), i_array(2))
END SELECT
```

TIPS & TRICKS

AXIS EnterpriseLink scripted jobs

EnterpriseLink scripted jobs ("Jobs") unlock the potential for many model updates to be made within a production environment. They allow for increased automation and improved controls, refocusing actuarial staff towards analytical efforts instead of manual activities. Conceptually, Jobs are analogous to a Formula Batch that exists outside a single Dataset, allowing the user to interact with and manipulate multiple Datasets during a single procedure.

There are a number of sample Jobs available in the EnterpriseLink environment that demonstrate some of the functionality available to the user. These can be accessed through the "Jobs" section in the lower left-hand menu, immediately below "Versions". An existing Job Set, called "Sample Jobs", contains examples written in VB.NET.

Examples include options such as "Open a Dataset and run a Batch" and "Manipulate Excel File", which demonstrates how Jobs can interact with Excel. The sample Jobs can be copied and modified to fit the user's end purpose. Benefits of using Jobs are compounded with an efficient AXIS Dataset structure. For example, Datasets that use consistent naming conventions for Batches, Cells, Tables, etc., allow for simpler script development when interacting with multiple Datasets. Leveraging Global Parameters will also facilitate the streamlining of routine model updates.

COMPONENT B: INPUTS

First, an Import/Export Macro Batch is created in AXIS with a single "read tables" action. This Batch is called "READ TABLES" in this example. Then, a second Import/Export Macro Batch is created in AXIS with a single "import" action. This Batch is called "IMPORT" in this example. Then, the previously outlined automation framework is followed.

STEP 1

Exhibit 5 shows the specified commands and their input description and requirements.

Exhibit 5: Sample commands for table imports

FUNCTION	INPUT DESCRIPTION
SetImpExpMacroActionFileTypeName (BatchName As String, ActionNo As Long, FileType As Integer, FileName As String)	Set File Type and File Name for Action number ActionNo in the Import/Export Macro Batch
SetImpExpMacroActionWorksheetOrTableName (BatchName As String, ActionNo As Long, WorksheetOrTableName As String)	Set Worksheet or Table name for Action number ActionNo in the Import/Export Macro Batch
SetImpExpMacroActionDatabaseLink (BatchName As String, ActionNo As Long, DatabaseLink As String)	Set DatabaseLink for Action number ActionNo in the Import/Export Macro Batch
SetImpExpMacroActionDatabaseTableName (BatchName As String, ActionNo As Long, DatabaseTableName As String)	Set Database Table name for Action number ActionNo in the Import/Export Macro Batch
RunBatch(BatchName as String)	Runs a Batch in a currently open Dataset

STEP 2

The Excel file allows users to input any required information.

Column A specifies what action will be taken: "0" indicates the line includes header information and should be skipped, "1" specifies the information should be read from Excel into MS Access, and "2" indicates the information should be imported from MS Access into AXIS.

Exhibit 6: Sample Excel input file for table imports

	A	B	C	D	E
1	0	File Path	Worksheet Tab Name	MS Access Database	MS Access Table
2	1	C:\ExamplePath\File1.xlsx	ExampleTab1	DatabaseName1	AccessTable1
3	1	C:\ExamplePath\File2.xlsx	ExampleTab2	DatabaseName2	AccessTable2
4	2	n/a	n/a	DatabaseName1	AccessTable1
5	2	n/a	n/a	DatabaseName2	AccessTable2
6					

STEP 3

A location is specified to save the above Excel tabs as .csv files. For this example, the location will be the DOCS folder within the Dataset and AXIS will be able to identify the path to this location simply as "$PATH\$DATASET\DOCS\".

STEP 4

The code shown in Exhibit 7 below should be placed in the section identified in Exhibit 1. Note that the array defined in this code should be parameterized correctly.

Exhibit 7: Sample execution code for table imports

```
'0 : Skip/Read/Import (0/1/2)
'1 : Full file path (including name)
'2 : Tab name
'3 : Database name
'4 : Database table

SELECT CASE i_array(0)
     'Skip row from csv
     CASE IS = 0

     'Read Action
     CASE IS = 1
          'Set File and type for Read Action
          Call SetImpExpMacroActionFileTypeName("READ TABLES", 1, IMPEXP_EXCEL2007, i_array(1))

          'Set Import Sheet for Read Action
          Call SetImpExpMacroActionWorksheetOrTableName("READ TABLES", 1, i_array(2))

          'Set Database Link for Read Action
          Call SetImpExpMacroActionDatabaseLink("READ TABLES", 1, i_array(3))

          'Set Database table name for Read Action
          Call SetImpExpMacroActionDatabaseTableName("READ TABLES", 1, i_array(4))

          'Read into database
          Call RunBatch("READ TABLES")

     'Import Action
     CASE IS = 2
          'Set Database Link for Import Action
          Call SetImpExpMacroActionDatabaseLink("Import", 1, i_array(3))

          'Set Database Table name for Import Action
          Call SetImpExpMacroActionDatabaseTableName("Import", 1, i_array(4))

          'Set Database Link for Read Action
          Call RunBatch("Import")

END SELECT
```

COMPONENT C: CALCULATIONS AND PROCESSING

The framework outlined earlier in the article can be leveraged again to manage model runs with one exception: AXIS does not allow the command "RunBatch" to be executed unless the actual Batch name is written explicitly in the code. We can utilize a subroutine that swaps a variable with a hardcoded Batch name to obtain the desired result; this additional element will be added to the code shown in Exhibit 1.

STEP 1

The command "RunBatch" from Exhibit 5 will be utilized.

STEP 2

An Excel file can include a tab that allows the user to set the order of model processing similar to what was shown in the previous Component B: Inputs section.

STEP 3

A location is specified to save the Excel tab as a .csv. For this example, the location will be the DOCS folder within the Dataset and AXIS will be able to identify the path to this location simply as "$PATH\$DATASET\DOCS\".

STEP 4

Insert the code in Exhibit 8 below into the wrapper code from Exhibit 1. This snippet checks if the Batch has been set to process (i.e., the run indicator is set to 1) and then runs the code from Exhibit 9 to swap the variable Batch name with a hardcoded Batch name.

Exhibit 8: Sample execution code to check Batch run indicator

```
'0 : Run Indicator
'1 : Batch Name

IF i_array(0) = 0 THEN
    Call s_BatchSwitch(i_array(1))
END IF
```

Add the code from Exhibit 9 below after the modified Exhibit 1 code. The addition of this subroutine is required since AXIS does not allow the action "RunBatch" to have a variable as an input; rather, the Batch name must be coded explicitly. The subroutine is used as a switch; hence, it is important that the "CASE IS" statements align with actual Batch names and that the Excel file contains identical Batch names.

Exhibit 9: Sample execution code for setting Dataset Parameters

```
'Subroutine that runs Batch based on string input
Sub s_BatchSwitch(i_BatchName as String)
    SELECT CASE i_BatchName
        CASE IS = "Update Dataset Parameters"
            Call RunBatch("Update Dataset Parameters")
        CASE IS = "Read and Import"
            Call RunBatch("Read and Import")
        CASE IS = "Run DataLink"
            Call RunBatch("Run DataLink")
        CASE IS = "Process Product X"
            Call RunBatch("Process Product X")
        CASE ELSE
     END SELECT
END SUB
```

CONCLUSION

After integrating the above design elements, the developer can produce an Excel interface to manage Dataset Parameters, table imports and model processing.

It is possible to externalize not only the model components presented in this article but also others. It is important to understand whether the functionality presented in this article fits within the modeling standards for your organization. If so, it may be possible to realize significant benefits across different modeling functions and users from externalizing certain model inputs.

WHAT'S NEW IN AXIS

EXCLUDE FORCED ANNUITIZATION BENEFITS FROM SOP 03-1

Description

- For the Annuity module, in the US GAAP section, a new option "2 - Include elective annuitization benefits in excess of fund" has been added to the SOP 03-1 settings "Income benefit definition" switch
- When the new option is selected, income benefits in the SOP 03-1 liability will only include benefits arising from elective annuitizations and exclude benefits from forced annuitizations

Details

- Version 20191602

Learn more

- https://www.ggy.com/bugenhance/updatedetail/26485/

MODAL POLICY FEE FACTOR AS SERIATIM OPTIONAL FIELD

Description

- In the Regular Life, Par Products and Disability modules, a Seriatim Optional Field "Modal Policy Fee Factor" has been added to the Policy Information Table
- The new Optional Field allows the user to assign Seriatim level modal policy fee factors and override the "Policy fee with each premium" and "Modal Policy Fee Factors" fields in the Product Features section of the Cell

Details

- Version 20190601

Learn more

- https://www.ggy.com/bugenhance/updatedetail/25904/

EXCLUDE REINSURANCE IN BLOCK PROJECTIONS

Description

- For Stochastic Processing, in each projection, the "Exclude reinsurance" switch has been added to exclude all reinsurance in each block projection without rerunning the Liability Cells

Details

- Version 20190601

Learn more

- https://www.ggy.com/bugenhance/updatedetail/25896/

ABOUT OLIVER WYMAN

Oliver Wyman is a global leader in management consulting. With offices in 60 cities across 29 countries, Oliver Wyman combines deep industry knowledge with specialized expertise in strategy, operations, risk management, and organization transformation. The firm has more than 5,000 professionals around the world who work with clients to optimize their business, improve their operations and risk profile, and accelerate their organizational performance to seize the most attractive opportunities. Oliver Wyman is a wholly owned subsidiary of Marsh & McLennan Companies [NYSE: MMC], the world's leading professional services firm in the areas of risk, strategy and people. Marsh is a leader in insurance broking and risk management; Guy Carpenter is a leader in providing risk and reinsurance intermediary services; Mercer is a leader in health, wealth and career consulting; and Oliver Wyman is a leader in management consulting. With annual revenue of more than $14 billion and more than 65,000 colleagues worldwide, Marsh & McLennan Companies provides analysis, advice and transactional capabilities to clients in more than 130 countries.

The Actuarial Practice of Oliver Wyman has life, health, and property & casualty actuaries that advise financial institutions, insurance companies, regulators, and self-insured entities across a broad spectrum of risk management issues. With almost 350 professionals in over 20 offices across North America, the Caribbean and Europe, the firm's consulting actuaries provide independent, objective advice, combining a wide range of expertise with specialized knowledge of specific risks.

For more information, please contact:

Dean Kerr, FSA, ACIA, MAAA
Partner
Dean.Kerr@oliverwyman.com
+1 416 868 7061

Bryan Lindsley, FSA, MAAA
Principal
Bryan.Lindsley@oliverwyman.com
+1 860 723 5633

Chris Whitney, FSA, MAAA
Principal
Christopher.Whitney@oliverwyman.com
+1 860 723 5605

David Weinsier, FSA, MAAA
Partner
David.Weinsier@oliverwyman.com
+1 404 239 6431

www.oliverwyman.com

OLIVER WYMAN