

---

## RiskAgility FM Audit Report

---

Solution Name	RA_Unify_Life_v97
Project Name	RA_Unify_Life_v97
Job Name	Base (14)
Report Created By	CLAL-INS\arikt
Report Created On	7/24/2025 12:43:28 PM (UTC+03:00)

Willis Towers Watson makes no warranty or representation as to the accuracy or reliability of this document or that it is suitable for any particular purpose.

# Table of Contents

<b>1</b>	<b>Summary .....</b>	<b>4</b>
1.1	Job Information.....	4
1.2	Output File Details .....	4
1.3	Job Details.....	5
1.3.1	Pre Sub Loops .....	5
1.3.2	Sub Loops .....	5
1.3.3	Post Sub Loops.....	5
1.3.4	No Sub Loops .....	5
1.4	Job Submission Settings .....	14
1.5	Team Edition Settings .....	14
1.5.1	Pending Changes.....	14
<b>2</b>	<b>Input Manager .....</b>	<b>15</b>
2.1	Input Manager: Input Manager.....	15
2.1.1	Input Page: Annuity .....	15
2.1.2	Input Page: Charges .....	19
2.1.3	Input Page: Claims.....	20
2.1.4	Input Page: Commission .....	23
2.1.5	Input Page: DAC .....	24
2.1.6	Input Page: Economic .....	26
2.1.7	Input Page: Expenses .....	33
2.1.8	Input Page: Lapses .....	34
2.1.9	Input Page: Margins .....	40
2.1.10	Input Page: Mortality .....	42
2.1.11	Input Page: Premium .....	52
2.1.12	Input Page: Product Details .....	58
2.1.13	Input Page: Reinsurance.....	65

	2.1.14	Input Page: Reserve .....	68
	2.1.15	Input Page: Setup .....	70
	2.1.16	Data Page: Ann_Data .....	73
	2.1.17	Data Page: life_Data .....	74
	2.1.18	Data Page: riders_Data.....	122
	2.1.19	External Sources.....	127
<b>3</b>	<b>Referenced Files .....</b>	<b>153</b>	
<b>4</b>	<b>Output Manager .....</b>	<b>154</b>	
	<b>4.1 OM Main .....</b>	<b>154</b>	
	4.1.1	Output Definition: Cashflow - Unify .....	154
<b>5</b>	<b>Code Manager .....</b>	<b>164</b>	
	<b>5.1 Model Tree .....</b>	<b>164</b>	
	<b>5.2 Project Variables .....</b>	<b>165</b>	
	<b>5.3 Model Classes .....</b>	<b>165</b>	
	5.3.1	ann_cflow .....	165
	5.3.2	fund_cflow .....	277
	5.3.3	life_cflow .....	299
	5.3.4	sub_2_cflow .....	870
	5.3.5	sub_array .....	921
	5.3.6	sub1_cflow .....	930
	<b>5.4 Custom Outputs .....</b>	<b>970</b>	
	<b>5.5 Externs and DLLs/Libs .....</b>	<b>971</b>	
<b>6</b>	<b>Appendix .....</b>	<b>971</b>	
	<b>6.1 Formulas .....</b>	<b>971</b>	
	6.1.1	Model Classes.....	971
	6.1.2	Externs .....	1387
	<b>6.2 Lookup Settings .....</b>	<b>1388</b>	
	6.2.1	Input Manager: Input Manager.....	1388

# 1 Summary

## 1.1 Job Information

Job Name:	Base (14)
Submitted On:	7/24/2025 12:43:19 PM (UTC+03:00)
Completed On:	7/24/2025 12:43:21 PM (UTC+03:00)
Submitted By:	CLAL-INS\arikt
Job Status:	Completed
Project Name:	RA_Unify_Life_v97
Project Folder:	C:\RAFM\Life Source Control
Edition:	Team
Execution Engine:	Gen2
RiskAgility FM Version:	3.7

## 1.2 Output File Details

Output Type	File Location	Modified On	File Size
Projection Output	\\vismoses01\moses\$\Data Files\Output\2412\Life\Model v97\Base (14)\Solvency_Base~life.csv	7/24/2025 12:43:20 PM (UTC+03:00)	896.5 KB (918033 Bytes)
Individual File	\\vismoses01\moses\$\Data Files\Output\2412\Life\Model v97\Base (14)\Solvency_Base~life@indiv.csv	7/24/2025 12:43:20 PM (UTC+03:00)	5.1 KB (5194 Bytes)

### 1.3 Job Details

Projection Set Loops                      None

#### 1.3.1            Pre Sub Loops

Task Name	Order	Status	Projection Task Loops	Dependencies	Input Manager	Target Object	Parameter Sets		
							Goal Seek	Output	Run

#### 1.3.2            Sub Loops

Projection Set Sub Loops                      None

Task Name	Order	Status	Projection Task Loops	Dependencies	Input Manager	Target Object	Parameter Sets		
							Goal Seek	Output	Run

#### 1.3.3            Post Sub Loops

Task Name	Order	Status	Projection Task Loops	Dependencies	Input Manager	Target Object	Parameter Sets		
							Goal Seek	Output	Run

#### 1.3.4            No Sub Loops

Task Name	Order	Status	Projection Task Loops	Dependencies	Input Manager	Target Object	Parameter Sets		
							Goal Seek	Output	Run
Base (Solvency)	1	Completed	None	None	Input Manager	life.profit_net_vif _pv,age_last,allo c_units,bonus_s himur,expense_ clm,coverage_u nits,expense_init ,claims_insuranc	None	OPS_Unify	RPS_main

Task Name	Order	Status	Projection Task Loops	Dependencies	Input Manager	Target Object	Parameter Sets		
							Goal Seek	Output	Run
						e,expense_ren,b e_retire,int_cred, cal_month,cal_y ear,comm_reg,c omm_profit,clai ms_annuity_gt, mgt_fees_prem, prem_insurance, prem_savings,u nits_for_takeup, cashflow_b,servi ce_units,cashflo w_e,service_unit s_pv,cashflow_p v,cashflow_re_b, cashflow_re_e,c ashflow_re_pv,c harges_premium ,charges_premiu m_pv,claims_an nuity,claims_ann uity_nogt,claims _annuity_pv,clai ms_death,claims _death_pv,claim s_disability,claim s_disability_pv,cl aims_maturity,cl aims_maturity_p v,claims_pv,clai ms_re,claims_su rrender,claims_s urrender_pv,clai ms_total,comm_ clawback,comm _clawback_pv,c omm_hekef,com m_nihul,comm_ prize,comm_pv, comm_re,comm _re_prof,comm_			

Task Name	Order	Status	Projection Task Loops	Dependencies	Input Manager	Target Object	Parameter Sets		
							Goal Seek	Output	Run
						regular,comm_r enewal,comm_r eserve,comm_re serve_pv,comm _supervisor,com m_total,cover_c harge,dac_book, dac_tax,death_b enefit,death_clai m_si,death_clai m_units,death_r ate,expense_clai ms_pv,expense _inflation,expens e_initial_fix,expe nse_initial_perc, expense_invest ment,expense_i nvestment_pv,e xpense_pv,expe nse_ren_charge, expense_ren_ch arge_pv,expens e_ren_fix,expen se_ren_perc,exp _total,expense_ var_pv,interest_r e,interest_re_pv, investment_inco me,investment_i ncome_chetz,inv estment_income _chetz_pv,invest ment_income_p v,lapse_rate_act _prm,lapse_total _prm,managem ent_fees,manag ement_fee_pv,cl aims_annuity_n ogt_pv,pol_fee,p			

Task Name	Order	Status	Projection Task Loops	Dependencies	Input Manager	Target Object	Parameter Sets		
							Goal Seek	Output	Run
						ol_fee_pv,premium,premium_disc,premium_disc_pv,premium_extra,premium_gross,premium_if_b,premium_if_rides,premium_pv,premium_re,profit_book_active_vif,profit_bk_act_vif_pv,profit_book_vif_pv,profit_book_vif,profit_net_vif,profit_re_pv,proj_month,proj_year,pup_rate_prm,rein_claims_pv,rein_comm_pv,rein_prem_pv,res_ann_deficiency,reserve,reserve_annuity,reserve_basic,reserve_claims,reserve_extra,reserve_increase,surr_value,reserve_increase_pv,reserve_re,re,re_increase,re_increase_pv,sum_insured,sum_insured_if_e,surv_prm,units_e,units_bon,premium_if_b_total,cashflow_b_bef_ret,cashflow_b_post_ret,profit_book_vif_pv_pos,			



Task Name	Order	Status	Projection Task Loops	Dependencies	Input Manager	Target Object	Parameter Sets		
							Goal Seek	Output	Run
						management_fees_fixed_ann,reserve_pv,manage_fees_fixed_ann_pv,management_fees_var_active,management_fees_var_ann,manage_fees_var_ann_pv,manage_fees_fixe_active_pv,manage_fees_var_active_pv,management_fees_fixed_active,capital_at_risk,capital_at_risk_rm,ber_retire_rm,bor_acc_pup,claims_annuity_pv_rm,claims_death_pv_rm,claims_disability_pv_rm,expense_pv_rm,inv_income_chetz_pv_rm,profit_book_vif_pv_pos_rm,rid_cashflow_pv,comm_renewal_pv,premium_gross_fix,premium_gross_var,pol_month,pol_year,expense_total_pre_ret,reserve_increase_bef_ret,investment_income_bef_ret,claims_lrc_q1,claims_lrc_yr2plus,bor_acc,bor_r			

Task Name	Order	Status	Projection Task Loops	Dependencies	Input Manager	Target Object	Parameter Sets		
							Goal Seek	Output	Run
						eturn,bor_return _pup,comm_hek ef_net,cashflow_ pv_e,claims_lrc_ q2,claims_lrc_q3 ,claims_lrc_q4,cl aims_re_lrc_q1, claims_re_lrc_q 2,claims_re_lrc_ q3,claims_re_lrc_ _q4,claims_re_lr c_yr2plus,expen se_claims_lrc_q 1,expense_claim s_lrc_q2,expens e_claims_lrc_q3, expense_claims_ _lrc_q4,expense_ _claims_lrc_yr2p lus,riskadj_gross _rel_q1,riskadj_ gross_rel_q2,risk adj_gross_rel_ q3,riskadj_gross _rel_q4,riskadj_ gross_rel_total,ri skadj_gross_rel_ _yr2plus,riskadj_ re_rel_q1,riskadj_ _re_rel_q2,riska dj_re_rel_q3,risk adj_re_rel_q4,ris kadj_re_rel_total ,riskadj_re_rel_y r2plus,fvui,lapse _rate_act_cnt,la pse_rate_act_ba l,lapse_rate_pup _prm,lapse_rate_ _pup_cnt,pup_ra te_cnt,pup_rate_			

Task Name	Order	Status	Projection Task Loops	Dependencies	Input Manager	Target Object	Parameter Sets		
							Goal Seek	Output	Run
						bal,surv_bal,riskadj_gross,riskadj_net,coverage_units_re,profit_book_vif_gross,profit_book_vif_gross_pv,surv_cnt,claim_cost,claim_cost_pv,claim_cost_pv_rm,claim_cost_re_pv,claim_cost_re_pv_rm,rein_claims_pv_rm,cover_charge_pv,income_b,income_e,income_pv,outgo_b,outgo_e,outgo_pv,cashflow,expense_pv_active,expense_pv_ann,expense_investment_pv_bef_ret,expense_investment_pv_post_ret,expense_pv_active_no_inv,comm_not_res_pv,investment_income_pv_active,reserve_increase_pv_active,profit_book_vif_pv_active,claims_maturity_ret_pv,units_b,management_fee_variable,sum_insured_occ_gross,sum_insured_occ_retent,claims_			

Task Name	Order	Status	Projection Task Loops	Dependencies	Input Manager	Target Object	Parameter Sets		
							Goal Seek	Output	Run
						retent,reserve_cl aims_retent,pre mium_disc_shim ur,premium_disc _shimur_pv,tot _bor_acc_pv,tot al_bor_return_p v,prem_savings _pv,cashflow_pv _chetz,nogt_ann pv,claims_lrc_q1 _pv,claims_lrc_q 2_pv,claims_lrc_ q3_pv,claims_lrc _q4_pv,claims_lr c_yr2plus_pv,ex pense_claims_lr c_q1_pv,expens e_claims_lrc_q2 _pv,expense_cla ims_lrc_q3_pv,e xpense_claims_l rc_q4_pv,expen se_claims_lrc_yr 2plus_pv,claims _re_lrc_q1_pv,cl aims_re_lrc_q2_ pv,claims_re_lrc _q3_pv,claims_r e_lrc_q4_pv,clai ms_re_lrc_yr2pl us_pv,riskadj_gr oss_rel_q1_pv,ri skadj_gross_rel _q2_pv,riskadj_ gross_rel_q3_pv ,riskadj_gross_r el_q4_pv,riskadj _gross_rel_total _pv,riskadj_gros s_rel_yr2plus_p			

Task Name	Order	Status	Projection Task Loops	Dependencies	Input Manager	Target Object	Parameter Sets		
							Goal Seek	Output	Run
						v,riskadj_re_rel_q1_pv,riskadj_re_rel_q2_pv,riskadj_re_rel_q3_pv,riskadj_re_rel_q4_pv,riskadj_re_rel_total_pv,riskadj_re_rel_yr2plus_pv			

## 1.4 Job Submission Settings

Create subfolder for Job Name	Yes
Overwrite existing files	Yes
Validate if External Source files exist	Yes
Wildcard Set	Single Policy
Distribution Method	None
Clear job working folder	Clear only successful jobs
Profile wanted	No
Rebasing checks	No
Write column to log	No
Write temporary table	No
Track accessed files	No
Odometer	100
Compiler settings	Microsoft® Visual Studio® 2017, Version 14.16.27045.0, 64-bit
Memory Check Mode	No
Output Validation	No
Treat code from Additional Include Directories as non-mutating by Gen2	No
Checking Level	NoChecks
Lookups executed in Gen2 Classic	No
Scalars executed in Gen2 Classic	No
Extractions executed in Gen2 Classic	No

## 1.5 Team Edition Settings

Team Foundation Server Path: [https://tfsprod/tfs/clalbitcollection\\$/RAFM/Development Life/Team/v43](https://tfsprod/tfs/clalbitcollection$/RAFM/Development Life/Team/v43)

### 1.5.1 Pending Changes

Document Type	Document Name	Pending Changes	Changeset
Project File	RA_Unify_Life_v97.msproj	No	708231
Referenced Files	Referenced Files.rfs	No	560957
Output Manager	OM Sens.opm	No	707207
Output Manager	OM Main.opm	No	707022
Output Manager	OM ESG.opm	No	706687
Input Manager	Input Manager.ipm	No	708231
Code Manager	Code Manager.mmf	No	708231
Run Manager	RM Life runs.rmr	No	708229
Output Parameter Set	OPS_ESG_0.ops	No	667645
Output Parameter Set	OPS_ESG.ops	No	666729
Output Parameter Set	OPS_ESG_Q_Yr1.ops	No	706687
Output Parameter Set	OPS_IFRS.ops	No	627535
Output Parameter Set	OPS_Sens.ops	No	706659
Output Parameter Set	OPS_Unify.ops	No	605491
Output Parameter Set	OPS_All.ops	No	604124
Output Parameter Set	OPS_ESG_ind.ops	No	640667
Output Parameter Set	OPS_ESG_Q.ops	No	677320
Run Parameter Set	RPS_ESG_5.rps	No	670036

Run Parameter Set	RPS_ESG_1.rps	No	702853
Run Parameter Set	RPS_ESG_5000.rps	No	692824
Run Parameter Set	RPS_quick.rps	No	604124
Run Parameter Set	RPS_model_points_NB.rps	No	629976
Run Parameter Set	RPS_NB.rps	No	604124
Run Parameter Set	RPS_model_points.rps	No	604124
Run Parameter Set	RPS_main.rps	No	696562
Wildcards	Wildcards.wcd	No	708207

## 2 Input Manager

### 2.1 Input Manager: Input Manager

#### 2.1.1 Input Page: Annuity

##### 2.1.1.1 Assumption Set: Base

Description:

Help:

Top Model Object:

life

Modified On:

12/11/2022 1:24:34 PM (UTC+02:00)

Modified By:

CLAL-INS\ahuvaa

Assumption Set Links:

Unlinked

##### 2.1.1.1.1 Input Variable: *res\_prop\_data*

Description:

Proportion of old money from data

Help:

Associated Code Variables:

res\_prop\_old\_data,res\_prop\_newtag\_data,res\_prop  
\_piz\_data,res\_prop\_prat\_data

Modified On:

12/11/2022 2:20:00 PM (UTC+02:00)

Modified By:

CLAL-INS\ahuvaa

Validation Failure Behaviour:

Error

Variable Type:

Floating Point Number

Valid Range From:

0

Valid Range To:

1

Choice List:

Value Type:

External Source

Value:

Tzeva Kesef

##### 2.1.1.1.2 Input Variable: *piz\_antiselection\_adj*

Description:

Adjustment to anti-selection i.r.o. pizuim due to tax-  
exempt limit

Help:

Associated Code Variables:

piz\_antiselection\_adj

Modified On:

12/11/2022 1:25:31 PM (UTC+02:00)

Modified By:

CLAL-INS\ahuvaa

Validation Failure Behaviour:

Error

Variable Type:	Floating Point Number
Valid Range From:	0
Valid Range To:	100
Choice List:	
Value Type:	External Source
Value:	AnnuityTU

#### **2.1.1.1.3 Input Variable: *prat\_antiselection\_adj***

Description:	Adjustment to anti-selection i.r.o. private funds
Help:	
Associated Code Variables:	prat_antiselection_adj
Modified On:	2/22/2021 10:44:43 AM (UTC+02:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	0
Valid Range To:	100
Choice List:	
Value Type:	External Source
Value:	AnnuityTU

#### **2.1.1.1.4 Input Variable: *old\_antiselection\_adj***

Description:	Adjustment to anti-selection i.r.o. old tagmulim
Help:	
Associated Code Variables:	old_antiselection_adj
Modified On:	2/22/2021 10:44:47 AM (UTC+02:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	0
Valid Range To:	100
Choice List:	
Value Type:	External Source
Value:	AnnuityTU

#### **2.1.1.1.5 Input Variable: *gimla\_table***

Description:	
Help:	PV of annuities at maturity (or at 'vesting date'). Deducted from single premium table.
Associated Code Variables:	gimla_table
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	
Valid Range To:	
Choice List:	



Value Type:	External Source
Value:	gimla

#### **2.1.1.1.6 Input Variable: *takeup\_age***

Description:	Take-up age for annuities
Help:	
Associated Code Variables:	takeup_age
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Integer Number
Valid Range From:	0
Valid Range To:	100
Choice List:	
Value Type:	External Source
Value:	Annuity

#### **2.1.1.1.7 Input Variable: *Types of Annuity Prop***

Description:	Table of policy ann factors & assumptions
Help:	
Associated Code Variables:	gtee_ppn,no_gtee_ppn,joint_life_ppn
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	0
Valid Range To:	0
Choice List:	
Value Type:	External Source
Value:	AnnuityTU

#### **2.1.1.1.8 Input Variable: *Retirement rate***

Description:	Percentage retiring at current age
Help:	
Associated Code Variables:	retirement_rate,prem_termination_rate
Modified On:	9/5/2019 11:19:02 AM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	
Valid Range To:	
Choice List:	
Value Type:	External Source
Value:	AnnuityTU

#### **2.1.1.1.9 Input Variable: *annuity\_value\_res\_tbl***

Description:	Reserve Deficiency table of annuity values at
--------------	---

Help:	maturity Value of annuity of 100 per month, by calender year at maturity and sex_maturity-age_discount-rate.
Associated Code Variables:	annuity_value_res_tbl
Modified On:	6/13/2021 12:26:32 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	0
Valid Range To:	0
Choice List:	
Value Type:	External Source
Value:	T_Factors

#### **2.1.1.1.10      *Input Variable: annuity\_detail\_gtee\_tbl***

Description:	Table of policy ann factors & assumptions
Help:	
Associated Code Variables:	ann_fac_gtee_value
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	0
Valid Range To:	0
Choice List:	
Value Type:	External Source
Value:	AnnuityDetails

#### **2.1.1.1.11      *Input Variable: annuity\_details\_tbl***

Description:	Table of policy ann factors & assumptions
Help:	
Associated Code Variables:	freeinv_res_ann,freeinv_res_ann_inpay,res_ann_mort_fac,int_res_ann,res_ann_exp,gtee_prd,int_tarif,mgt_fee_fixed,mgt_fee_var,mgt_fee_max,life2_ppn,age_diff,redn_factor,base_year,ann_fac_dthben,ann_fac_joint,ann_fac_no_gtee
Modified On:	9/12/2021 4:59:04 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	0
Valid Range To:	0
Choice List:	
Value Type:	External Source
Value:	AnnuityDetails

#### **2.1.1.1.12      *Input Variable: Retirement rate ann***

Description:	Percentage retiring at current age
--------------	------------------------------------

Help:	
Associated Code Variables:	retirement_rate
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	
Valid Range To:	
Choice List:	
Value Type:	External Source
Value:	AnnuityTU

#### **2.1.1.1.13      *Input Variable: annuity\_details\_temp\_tbl***

Description:	Table of policy ann factors & assumptions
Help:	
Associated Code Variables:	annuity_details_temp_tbl
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	0
Valid Range To:	0
Choice List:	
Value Type:	External Source
Value:	AnnuityDetails

### **2.1.2              Input Page: Charges**

#### **2.1.2.1 Assumption Set: Base**

Description:	
Help:	
Top Model Object:	life
Modified On:	12/10/2023 11:11:51 AM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Assumption Set Links:	Unlinked

##### **2.1.2.1.1 *Input Variable: mgtfee\_tbl***

Description:	Management fee table by format no.
Help:	
Associated Code Variables:	mgtfee_age_after,mgtfee_acc_after,mgtfee_dthben,mgtfee_senior,mgtfee_disc_mth,mgtfee_disc_after,mgtfee_orig,mgtfee_age,mgtfee_acc,mgtfee_from_dthben,mgtfee_max_dthben,mgtfee_from_senior,mgtfee_floor
Modified On:	4/24/2023 4:52:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Validation Failure Behaviour:	Error
Variable Type:	Integer Number

Valid Range From:  
 Valid Range To:  
 Choice List:  
 Value Type: External Source  
 Value: format\_mgtfee

#### **2.1.2.1.2 Input Variable: mgt\_deficit\_perc**

Description: Management fee surplus/deficit as % of accumulation  
 Help:  
 Associated Code Variables: mgt\_deficit\_perc  
 Modified On: 1/6/2020 11:51:04 AM (UTC+02:00)  
 Modified By: CLAL-INS\NinaB  
 Validation Failure Behaviour: Error  
 Variable Type: Floating Point Number  
 Valid Range From:  
 Valid Range To:  
 Choice List:  
 Value Type: External Source  
 Value: Economic

### **2.1.3 Input Page: Claims**

#### **2.1.3.1 Assumption Set: Base**

Description:  
 Help:  
 Top Model Object: life  
 Modified On: 3/16/2023 10:57:31 AM (UTC+02:00)  
 Modified By: CLAL-INS\ahuvaa  
 Assumption Set Links: Unlinked

##### **2.1.3.1.1 Input Variable: claim\_cost\_factors\_tbl**

Description: Claims Cost Table for Profil Riders  
 Help: Annuity Factors applied to claims of type PHI/FIB/LTC etc.  
 Factors are in respect of 1 shekel monthly benefit.  
 Associated Code Variables: claim\_cost\_factors\_tbl  
 Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
 Modified By: CLAL-INS\NinaB  
 Validation Failure Behaviour: Error  
 Variable Type: Floating Point Number  
 Valid Range From: 0  
 Valid Range To: 0  
 Choice List:  
 Value Type: External Source  
 Value: profil\_rider\_claims\_annuity\_fac

**2.1.3.1.2 Input Variable: *claim\_rates\_tbl***

Description:	claim rates table for Profil riders
Help:	
Associated Code Variables:	claim_rates_tbl
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	0
Valid Range To:	0
Choice List:	
Value Type:	External Source
Value:	profil_decrement_rates

**2.1.3.1.3 Input Variable: *claims\_cost\_factors\_tbl***

Description:	claim cost factors table (annuity factors)
Help:	PV of future claim payments (here: as multiplier of the sum assured).
Associated Code Variables:	claims_cost_factors_tbl
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	0
Valid Range To:	0
Choice List:	
Value Type:	External Source
Value:	claim_cost_phi12_ltc07

**2.1.3.1.4 Input Variable: *clms\_mult\_infl***

Description:	Claims Multiplier table (%)
Help:	
Associated Code Variables:	clms_mult_infl
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	0
Valid Range To:	0
Choice List:	
Value Type:	External Source
Value:	clms_mult

**2.1.3.1.5 Input Variable: *clms\_mult\_i***

Description:	Claims Multiplier table (%)
Help:	
Associated Code Variables:	clms_mult_i

Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	0
Valid Range To:	0
Choice List:	
Value Type:	External Source
Value:	clms_mult

#### **2.1.3.1.6 Input Variable: clms\_mult\_tt**

Description:	Claims Multiplier table (%)
Help:	
Associated Code Variables:	clms_mult_tt
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	0
Valid Range To:	0
Choice List:	
Value Type:	External Source
Value:	clms_mult

#### **2.1.3.1.7 Input Variable: recovery\_rates\_tbl**

Description:	
Help:	
Associated Code Variables:	recovery_rates_tbl
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	
Valid Range To:	
Choice List:	
Value Type:	External Source
Value:	phi_recover

#### **2.1.3.1.8 Input Variable: Various\_Parameters**

Description:	
Help:	Maximum cumulative claim inflation allowed (for health products). As a percentage. For example, 200 means that the claims inflation will stop if and when the claims reach double the base assumption.
Associated Code Variables:	claim_inflation_max,claim_inflation_max_re,min_ytro n_perc,pizui_prop_pup_stat_c

Modified On:	5/29/2025 11:26:15 AM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	0
Valid Range To:	100000
Choice List:	
Value Type:	External Source
Value:	Parameters

## 2.1.4 Input Page: Commission

### 2.1.4.1 Assumption Set: Base

Description:	
Help:	
Top Model Object:	life
Modified On:	5/4/2022 11:23:31 AM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Assumption Set Links:	Unlinked

#### 2.1.4.1.1 Input Variable: *comm\_extra\_tbl*

Description:	Extra commission table (%)
Help:	Extra and Shimur Tik commissions (manual commns) expressed as a % of New Premium by Production Year or all years (if year is omitted).
Associated Code Variables:	comm_extra_tbl
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	0
Valid Range To:	0
Choice List:	
Value Type:	External Source
Value:	comm_extra

#### 2.1.4.1.2 Input Variable: *comm\_extra\_agent\_tbl*

Description:	Extra commission table (%) by Osek Mureshe
Help:	Super commission expressed as a % of initial regular commissions or % of premium.
Associated Code Variables:	comm_extra_agent_tbl
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	0
Valid Range To:	0
Choice List:	

Value Type:	External Source
Value:	comm_extra_agent

#### **2.1.4.1.3 Input Variable: comm\_claw\_prpn\_tbl**

Description:	Clawback proportion table
Help:	Table of clawback in each policy month as a proportion of the initial commission paid.
Associated Code Variables:	comm_claw_prpn_tbl
Modified On:	6/29/2021 5:03:17 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	0
Valid Range To:	0
Choice List:	
Value Type:	External Source
Value:	commclaw

#### **2.1.4.1.4 Input Variable: comm\_ren\_perc\_prem\_mrtg**

Description:	Renewal Commission (%) for mortgage policies sold after 04/2007 and after 16 yrs vetek
Help:	
Associated Code Variables:	comm_ren_perc_prem_mrtg
Modified On:	1/26/2022 10:06:20 AM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	
Valid Range To:	
Choice List:	
Value Type:	External Source
Value:	Parameters

### **2.1.5 Input Page: DAC**

#### **2.1.5.1 Assumption Set: Base**

Description:	
Help:	
Top Model Object:	life
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Assumption Set Links:	Unlinked

##### **2.1.5.1.1 Input Variable: dac\_amort\_type**

Description:	DAC amortisation type
Help:	DAC amortisation type
Associated Code Variables:	dac_amort_type



Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Character
Valid Range From:	0
Valid Range To:	0
Choice List:	Fixed,Lifetime
Value Type:	External Source
Value:	Parameters

#### **2.1.5.1.2 Input Variable: *dac\_cap\_apply***

Description:	Apply capital requirement of x% of DAC
Help:	The percentage of the DAC for books that has to be retained in the company's capital
Associated Code Variables:	dac_cap_apply
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Character
Valid Range From:	0
Valid Range To:	100
Choice List:	Y,N
Value Type:	External Source
Value:	Parameters

#### **2.1.5.1.3 Input Variable: *dac\_book\_adj\_factor***

Description:	Adjustment factor for Dac book (to scale up DAC to actuals)
Help:	
Associated Code Variables:	dac_book_adj_factor_input
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	
Valid Range To:	
Choice List:	
Value Type:	External Source
Value:	Economic

#### **2.1.5.1.4 Input Variable: *dac\_tax\_adj\_factor***

Description:	Adjustment factor for Dac tax (to scale up DAC to actuals)
Help:	
Associated Code Variables:	dac_tax_adj_factor_input
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error

Variable Type:	Floating Point Number
Valid Range From:	
Valid Range To:	
Choice List:	
Value Type:	External Source
Value:	Economic

## 2.1.6 Input Page: Economic

### 2.1.6.1 Assumption Set: Solv\_Base

Description:	
Help:	
Top Model Object:	life
Modified On:	5/29/2025 11:37:54 AM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Assumption Set Links:	Unlinked

#### 2.1.6.1.1 Input Variable: fund\_rates\_tbl

Description:	Table of parameters by fund
Help:	Model fund name/number, based on actual fund number from data file
Associated Code Variables:	fund_name,par_npar,dactype,invinc,var_mgt_fee,fixe d_mgt_fee,intres,intres_puresav,dac_book_fac,dac_t ax_fac,cap_req_perc_premium,cap_req_perc_reserv e,mort_res,mort_addn,ann_series,fixed_mgt_fee_ter m,tat_shnatiut_assum,max_chetz
Modified On:	6/27/2024 12:45:16 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	0
Valid Range To:	0
Choice List:	
Value Type:	External Source
Value:	fundrate

#### 2.1.6.1.2 Input Variable: fund\_rates\_code\_tbl

Description:	Table of parameters by fund
Help:	Read in set_exp_variables.
Associated Code Variables:	fund_rates_code_tbl
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	0
Valid Range To:	0
Choice List:	
Value Type:	External Source

Value:

fundrate

**2.1.6.1.3 Input Variable: inv\_rates**

Description:

Investment income rate (%)

Help:

Annual investment income rate on free assets.

This is average with the fund-specific rate on special bonds, to obtain inv\_rate\_mth\_w which is used in the projection.

This may vary by calender year. The array index = calender year - valn\_year + 1. In the fund rate table inv\_free may be entered for a specific calender year.

Associated Code Variables:

inv\_rate\_rollup

Modified On:

8/25/2022 12:48:07 PM (UTC+03:00)

Modified By:

CLAL-INS\ahuvaa

Validation Failure Behaviour:

Error

Variable Type:

Floating Point Array

Valid Range From:

-100

Valid Range To:

100

Choice List:

Value Type:

External Source

Value:

Economic

**2.1.6.1.4 Input Variable: Yield\_pre\_ret**

Description:

Risk free investment &amp; discount rates

Help:

Associated Code Variables:

inv\_rate\_m

Modified On:

8/25/2022 11:48:32 AM (UTC+03:00)

Modified By:

CLAL-INS\ahuvaa

Validation Failure Behaviour:

Error

Variable Type:

Floating Point Number

Valid Range From:

0

Valid Range To:

0

Choice List:

Value Type:

External Source

Value:

RFR\_Solv

**2.1.6.1.5 Input Variable: Yield\_post\_ret**

Description:

Risk free investment &amp; discount rates

Help:

Associated Code Variables:

ann\_inv\_rate\_m

Modified On:

8/25/2022 11:48:49 AM (UTC+03:00)

Modified By:

CLAL-INS\ahuvaa

Validation Failure Behaviour:

Error

Variable Type:

Floating Point Number

Valid Range From:

0

Valid Range To:

0

Choice List:

Value Type:

External Source

Value: RFR\_Solv

#### **2.1.6.1.6 Input Variable: Discounting\_pre**

Description: Risk free investment & discount rates  
Help:  
Associated Code Variables: disc\_rate\_m  
Modified On: 8/25/2022 2:05:11 PM (UTC+03:00)  
Modified By: CLAL-INS\ahuvaa  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From: 0  
Valid Range To: 0  
Choice List:  
Value Type: External Source  
Value: RFR\_Solv

#### **2.1.6.1.7 Input Variable: Yield\_pre\_ret\_ifrs**

Description: Risk free investment & discount rates IFRS  
Help:  
Associated Code Variables: inv\_rate\_m\_ifrs  
Modified On: 6/27/2024 12:45:49 PM (UTC+03:00)  
Modified By: CLAL-INS\arikt  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From: 0  
Valid Range To: 0  
Choice List:  
Value Type: External Source  
Value: RFR\_IFRS

#### **2.1.6.1.8 Input Variable: Yield\_post\_ret\_ifrs**

Description: Risk free investment & discount rates IFRS  
Help:  
Associated Code Variables: ann\_inv\_rate\_m\_ifrs  
Modified On: 6/27/2024 12:45:54 PM (UTC+03:00)  
Modified By: CLAL-INS\arikt  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From: 0  
Valid Range To: 0  
Choice List:  
Value Type: External Source  
Value: RFR\_IFRS

#### **2.1.6.1.9 Input Variable: Discounting\_pos**

Description: Risk free investment & discount rates  
Help:

Associated Code Variables:	ann_disc_rate_m
Modified On:	8/25/2022 2:05:14 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	0
Valid Range To:	0
Choice List:	
Value Type:	External Source
Value:	RFR_Solv

#### **2.1.6.1.10      *Input Variable: Discounting\_NoVA***

Description:	Risk free investment & discount rates
Help:	
Associated Code Variables:	inv_rate_rm_m
Modified On:	8/25/2022 11:48:57 AM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	0
Valid Range To:	0
Choice List:	
Value Type:	External Source
Value:	RFR_Solv

#### **2.1.6.1.11      *Input Variable: tax\_rate***

Description:	Lookup value coode variable wildcard
Help:	
Associated Code Variables:	tax_rate
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	
Valid Range To:	
Choice List:	
Value Type:	External Source
Value:	tax_rates

#### **2.1.6.1.12      *Input Variable: CU\_Discounted***

Description:	Discount coverage units
Help:	
Associated Code Variables:	cu_discounted
Modified On:	8/5/2024 3:40:52 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Validation Failure Behaviour:	Error
Variable Type:	Character

Valid Range From:  
 Valid Range To:  
 Choice List:  
 Value Type: External Source  
 Value: Parameters

#### **2.1.6.1.13      *Input Variable: fund\_group***

Description: Fund Group  
 Help:  
 Associated Code Variables: fund\_group  
 Modified On: 8/5/2024 4:42:41 PM (UTC+03:00)  
 Modified By: CLAL-INS\arikt  
 Validation Failure Behaviour: Error  
 Variable Type: Character  
 Valid Range From:  
 Valid Range To:  
 Choice List: P  
 Value Type: External Source  
 Value: fundrate

#### **2.1.6.1.14      *Input Variable: fundgroup\_manual***

Description: Fund Group  
 Help:  
 Associated Code Variables: fundgroup\_manual  
 Modified On: 8/5/2024 4:50:02 PM (UTC+03:00)  
 Modified By: CLAL-INS\arikt  
 Validation Failure Behaviour: Default  
 Variable Type: Character  
 Valid Range From:  
 Valid Range To:  
 Choice List:  
 Value Type: External Source  
 Value: fundrate

#### **2.1.6.1.15      *Input Variable: free\_inv\_ratio\_tbl***

Description: Free investment ratio by fund  
 Help:  
 Associated Code Variables: free\_inv\_ratio\_tbl  
 Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
 Modified By: CLAL-INS\NinaB  
 Validation Failure Behaviour: Error  
 Variable Type: Floating Point Number  
 Valid Range From: 0  
 Valid Range To: 0  
 Choice List:  
 Value Type: External Source  
 Value: FreeInvRatio

**2.1.6.1.16      *Input Variable: fund\_rates\_tbl\_yesodi***

Description:	Table of parameters by fund
Help:	
Associated Code Variables:	par_npar_yesodi
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	0
Valid Range To:	0
Choice List:	
Value Type:	External Source
Value:	fundrate

**2.1.6.1.17      *Input Variable: Economic\_Char***

Description:	Economic_Assumptions - Character
Help:	EV Dscout rate to use for the run: Vector= uses the input discount vector (v-month_t) Single = replaces discount vector based on the single input rate
Associated Code Variables:	ev_discount_rate_type,start_int_proj_after_rollup,esg_run,phi_res_discount_rate_type
Modified On:	1/18/2024 2:21:28 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Validation Failure Behaviour:	Error
Variable Type:	Character
Valid Range From:	0
Valid Range To:	0
Choice List:	Single,Vector,Earned
Value Type:	External Source
Value:	Economic

**2.1.6.1.18      *Input Variable: Economic\_Num***

Description:	Economic_Assumptions - Number
Help:	Investment margin used as discount rate for annuity deficiency reserve
Associated Code Variables:	ann_def_res_inv_margin,ann_def_res_inv_margin_par,ev_disc_rate,vat,mortg_int
Modified On:	5/10/2022 3:04:05 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	0
Valid Range To:	100
Choice List:	
Value Type:	External Source
Value:	Economic

**2.1.6.1.19      *Input Variable: Cols\_of\_Money\_Prop***

Description:

Help:

Associated Code Variables: ann\_tu\_old,ann\_tu\_old\_res,ann\_tu\_piz,ann\_tu\_piz\_res,ann\_tu\_prat,ann\_tu\_prat\_res,ann\_tu\_newtag\_res,ann\_tu\_newtag

Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)

Modified By: CLAL-INS\NinaB

Validation Failure Behaviour: Error

Variable Type: Floating Point Number

Valid Range From:

Valid Range To:

Choice List:

Value Type: External Source

Value: AnnuityTU

**2.1.6.1.20      *Input Variable: temp\_fund\_rates\_tbl ann***

Description: Table of parameters by fund

Help:

Associated Code Variables: temp\_fund\_rates\_tbl

Modified On: 5/18/2023 9:47:12 AM (UTC+03:00)

Modified By: CLAL-INS\ahuvaa

Validation Failure Behaviour: Error

Variable Type: Floating Point Number

Valid Range From: 0

Valid Range To: 0

Choice List:

Value Type: External Source

Value: fundrate

**2.1.6.1.21      *Input Variable: fund\_t\_factor***

Description: Table of parameters by fund

Help:

Associated Code Variables: fund\_t\_factor

Modified On: 5/18/2023 9:56:01 AM (UTC+03:00)

Modified By: CLAL-INS\ahuvaa

Validation Failure Behaviour: Error

Variable Type: Floating Point Number

Valid Range From: 0

Valid Range To: 0

Choice List:

Value Type: External Source

Value: fundrate

**2.1.6.1.22      *Input Variable: fund\_rates\_tbl temp***

Description: Table of parameters by fund

Help:



Associated Code Variables:	ann_series_temp
Modified On:	6/30/2021 10:14:45 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	0
Valid Range To:	0
Choice List:	
Value Type:	External Source
Value:	fundrate

### **2.1.6.1.23      *Input Variable: prop\_gteedint\_post\_maturity***

Description:	Prop of participating policies getting Guar int rate post maturity
Help:	
Associated Code Variables:	prop_gteedint_post_maturity
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	0
Valid Range To:	1
Choice List:	
Value Type:	Code Default
Value:	0

## **2.1.7              Input Page: Expenses**

### **2.1.7.1 Assumption Set: Base**

Description:	
Help:	
Top Model Object:	life
Modified On:	10/26/2021 1:26:22 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Assumption Set Links:	Unlinked

#### **2.1.7.1.1 *Input Variable: exp\_dac\_perc***

Description:	Proportion of initial expenses deferred (in DAC)
Help:	Proportion (%) of initial expenses that are deferrable in the DAC.
Associated Code Variables:	exp_dac_perc
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	0
Valid Range To:	100

Choice List:

Value Type:

Value:

External Source

Parameters

**2.1.7.1.2 Input Variable: exp\_mult\_tbl**

Description:

Help:

Associated Code Variables:

Modified On:

Modified By:

Validation Failure Behaviour:

Variable Type:

Valid Range From:

Valid Range To:

Choice List:

Value Type:

Value:

Expense multipliers (%)

Expense multipliers table for initial and renewal expenses.

exp\_mult\_tbl

8/27/2019 4:00:59 PM (UTC+03:00)

CLAL-INS\NinaB

Error

Floating Point Number

0

100

External Source

exp\_mult

**2.1.7.1.3 Input Variable: expense\_tbl**

Description:

Help:

Associated Code Variables:

Modified On:

Modified By:

Validation Failure Behaviour:

Variable Type:

Valid Range From:

Valid Range To:

Choice List:

Value Type:

Value:

Expense table

i\_perpol\_sp,i\_single,m\_pup,i\_perpol,i\_prem,m\_prem,  
m\_perpol,m\_clms,m\_ann\_pmt,m\_res\_par,m\_res\_no  
npar,m\_res,i\_percov,m\_percov,i\_percov\_sp,m\_perc  
ov\_sp,m\_clms\_cov,exp\_mada

4/28/2022 4:42:37 PM (UTC+03:00)

CLAL-INS\ahuvaa

Error

Floating Point Number

0

0

External Source

expense

**2.1.8 Input Page: Lapses****2.1.8.1 Assumption Set: Solv\_Base**

Description:

Help:

Top Model Object:

Modified On:

Modified By:

Assumption Set Links:

life

12/23/2024 8:54:37 PM (UTC+02:00)

CLAL-INS\ahuvaa

Unlinked

**2.1.8.1.1 Input Variable: various parameters**

Description:	various parameters
Help:	For "achrayut le'chaim" product. The percentage of normal lapses experienced by policies that continue after a claim. (the first claim is from the group that allows the policy to continue at 50%).
Associated Code Variables:	secondary_lapse_mult,fix_term_end_age_limit,fix_term_curr_age_max,fix_term_new_end_age,fix_term_curr_age_above_max_add_months
Modified On:	11/29/2021 7:44:32 AM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	0
Valid Range To:	1000
Choice List:	
Value Type:	External Source
Value:	Parameters

**2.1.8.1.2 Input Variable: sur\_val\_method**

Description:	various parameters
Help:	Different ways to calculate surrender values: "sv_table" = SV looked up from table. "perc_res" = SV is a percentage of reserve. Percentages come from array variable "sur_val_perc" by policy year. This method is useful for old products that do not have tables available.
Associated Code Variables:	sur_val_method
Modified On:	3/27/2023 4:00:52 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Validation Failure Behaviour:	Error
Variable Type:	Character
Valid Range From:	0
Valid Range To:	1000
Choice List:	
Value Type:	External Source
Value:	Parameters

**2.1.8.1.3 Input Variable: lapse\_factor\_y1**

Description:	Agent Lapse factors table
Help:	
Associated Code Variables:	lapse_factor_y1
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	0

Valid Range To: 0  
Choice List:  
Value Type: External Source  
Value: lapse\_factor

#### **2.1.8.1.4 Input Variable: lapse\_clawback\_factor**

Description: Agent Lapse factors table  
Help:  
Associated Code Variables: lapse\_clawback\_factor  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From: 0  
Valid Range To: 0  
Choice List:  
Value Type: External Source  
Value: lapse\_factor

#### **2.1.8.1.5 Input Variable: lapse\_factor\_yplus**

Description: Agent Lapse factors table  
Help:  
Associated Code Variables: lapse\_factor\_yplus  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From: 0  
Valid Range To: 0  
Choice List:  
Value Type: External Source  
Value: lapse\_factor

#### **2.1.8.1.6 Input Variable: lapse\_rate\_im**

Description: Lapse rates table (%)  
Help:  
Associated Code Variables: lapse\_rate\_im  
Modified On: 1/15/2023 2:58:01 PM (UTC+02:00)  
Modified By: CLAL-INS\joshm  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From: 0  
Valid Range To: 0  
Choice List:  
Value Type: External Source  
Value: lapse

**2.1.8.1.7 Input Variable: lapse\_rider\_profil\_dth**

Description:	Lapse rates table (%)
Help:	
Associated Code Variables:	lapse_rider_profil_dth
Modified On:	1/15/2023 2:55:17 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	0
Valid Range To:	0
Choice List:	
Value Type:	External Source
Value:	lapse

**2.1.8.1.8 Input Variable: lapse\_rider\_other**

Description:	Lapse rates table (%)
Help:	
Associated Code Variables:	lapse_rider_other
Modified On:	1/15/2023 3:02:51 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	0
Valid Range To:	0
Choice List:	
Value Type:	External Source
Value:	lapse

**2.1.8.1.9 Input Variable: lapse\_rate\_pup\_im**

Description:	Lapse rates table (%)
Help:	
Associated Code Variables:	lapse_rate_pup_im
Modified On:	1/15/2023 2:58:16 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	0
Valid Range To:	0
Choice List:	
Value Type:	External Source
Value:	lapse

**2.1.8.1.10 Input Variable: puv\_09\_tbl**

Description:	Composite external source
Help:	
Associated Code Variables:	puv_09_tbl
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)

Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	
Valid Range To:	
Choice List:	
Value Type:	External Source
Value:	puv_composite

#### **2.1.8.1.11      *Input Variable: masslaps\_tbl***

Description:	Lapse rate table by flag_code for solvency scenario
Help:	
Associated Code Variables:	masslaps_tbl
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	0
Valid Range To:	0
Choice List:	
Value Type:	External Source
Value:	mass_lapse_tab

#### **2.1.8.1.12      *Input Variable: surr\_chg\_tbl***

Description:	Surrender charges table
Help:	Contains penalty rates on surrender.
Associated Code Variables:	surr_chg_tbl
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	0
Valid Range To:	0
Choice List:	
Value Type:	External Source
Value:	surr_chg

#### **2.1.8.1.13      *Input Variable: pup\_ltc\_tbl***

Description:	
Help:	
Associated Code Variables:	pup_ltc_tbl
Modified On:	9/12/2021 4:59:54 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	
Valid Range To:	

Choice List:  
Value Type: External Source  
Value: pup

#### **2.1.8.1.14      *Input Variable: pup\_ltc\_tbl\_next***

Description:  
Help:  
Associated Code Variables: pup\_ltc\_tbl\_next  
Modified On: 9/12/2021 4:59:58 PM (UTC+03:00)  
Modified By: CLAL-INS\joshm  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:  
Choice List:  
Value Type: External Source  
Value: pup

#### **2.1.8.1.15      *Input Variable: lapse\_factor\_proj***

Description: Lapse factor by proj\_yr  
Help:  
Associated Code Variables: lapse\_factor\_proj  
Modified On: 12/19/2024 3:34:09 PM (UTC+02:00)  
Modified By: CLAL-INS\ahuvaa  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From: 0  
Valid Range To: 0  
Choice List:  
Value Type: External Source  
Value: lapse\_factor\_proj

#### **2.1.8.1.16      *Input Variable: lapse\_factor\_proj\_rider***

Description: Lapse factor for rider by proj\_yr  
Help:  
Associated Code Variables: lapse\_factor\_proj\_rider  
Modified On: 12/19/2024 3:35:42 PM (UTC+02:00)  
Modified By: CLAL-INS\ahuvaa  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From: 0  
Valid Range To: 0  
Choice List:  
Value Type: External Source  
Value: lapse\_factor\_proj

**2.1.8.1.17 Input Variable: lapse\_factor\_profil\_rider**

Description:	Lapse factor for rider by proj_yr
Help:	
Associated Code Variables:	lapse_factor_profil_rider
Modified On:	12/23/2024 8:58:12 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	0
Valid Range To:	0
Choice List:	
Value Type:	External Source
Value:	lapse_factor_proj

**2.1.9 Input Page: Margins****2.1.9.1 Assumption Set: Solv\_Base**

Description:	
Help:	
Top Model Object:	life
Modified On:	3/1/2023 11:13:36 AM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Assumption Set Links:	Unlinked

**2.1.9.1.1 Input Variable: asset\_shock**

Description:	Asset Shock (to replace investment income)
Help:	
Associated Code Variables:	asset_shock
Modified On:	12/4/2022 8:58:06 AM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	-100
Valid Range To:	100
Choice List:	
Value Type:	External Source
Value:	Asset_Shocks

**2.1.9.1.2 Input Variable: prem\_disc\_scenario**

Description:	Prem Disc for Scenario
Help:	
Associated Code Variables:	prem_disc_scenario
Modified On:	11/29/2020 3:48:35 PM (UTC+02:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number



Valid Range From: 0  
 Valid Range To: 100  
 Choice List:  
 Value Type: External Source  
 Value: Discount\_Scenarios

#### **2.1.9.1.3 Input Variable: mgt\_fee\_disc**

Description: Discount on management fees  
 Help:  
 Associated Code Variables: mgt\_fee\_disc  
 Modified On: 11/29/2020 3:49:36 PM (UTC+02:00)  
 Modified By: CLAL-INS\NinaB  
 Validation Failure Behaviour: Error  
 Variable Type: Floating Point Number  
 Valid Range From: 0  
 Valid Range To: 100  
 Choice List:  
 Value Type: External Source  
 Value: Discount\_Scenarios

#### **2.1.9.1.4 Input Variable: Margins\_Char**

Description: Scenarios Sets - Character variables  
 Help: Add margins to assumptions Y/N ?  
 (expenses, mortality and lapses)  
 eg for DAC Recoverability test.  
 Associated Code Variables: margin\_add,margin\_add\_asset,margin\_add\_discount,margin\_add\_cat,prem\_disc\_shimur\_flag  
 Modified On: 2/11/2024 4:57:37 PM (UTC+02:00)  
 Modified By: CLAL-INS\Arikt  
 Validation Failure Behaviour: Error  
 Variable Type: Character  
 Valid Range From:  
 Valid Range To:  
 Choice List: Y,N  
 Value Type: External Source  
 Value: margins

#### **2.1.9.1.5 Input Variable: Margins\_Number**

Description: Scenarios Sets - Number variables  
 Help: only added if margin\_add = Y  
 Associated Code Variables: margin\_exp\_ini\_fix,margin\_exp\_ini\_pc,margin\_exp\_ren\_fix,margin\_exp\_ren\_pc,margin\_lapses,margin\_mort\_pc,margin\_claims,margin\_ann\_mort\_pc,margin\_annuity\_takeup,margin\_recover,margin\_1styr\_clms\_add,infl\_rate\_expenses,cat\_risk,lapse\_force\_month,lapse\_force\_rate\_input,margin\_claim\_cost\_mitriya,margin\_res\_ann\_mort\_fac  
 Modified On: 3/17/2024 1:54:44 PM (UTC+02:00)

Modified By:	CLAL-INS\Arikt
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	-100
Valid Range To:	1000
Choice List:	
Value Type:	External Source
Value:	margins

## 2.1.10 Input Page: Mortality

### 2.1.10.1 Assumption Set: Base

Description:	
Help:	
Top Model Object:	life
Modified On:	7/23/2024 3:33:58 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Assumption Set Links:	Unlinked

#### 2.1.10.1.1 Input Variable: *death\_rates\_tbl*

Description:	
Help:	
Associated Code Variables:	death_rates_tbl
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	
Valid Range To:	
Choice List:	
Value Type:	External Source
Value:	death_rates_comp

#### 2.1.10.1.2 Input Variable: *sv\_09\_tbl*

Description:	
Help:	
Associated Code Variables:	sv_09_tbl
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	
Valid Range To:	
Choice List:	
Value Type:	External Source
Value:	sv_composite

**2.1.10.1.3      *Input Variable: sv\_09\_tbl\_check***

Description:	
Help:	
Associated Code Variables:	sv_09_tbl_check
Modified On:	10/17/2021 4:46:15 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	
Valid Range To:	
Choice List:	
Value Type:	External Source
Value:	sv_composite

**2.1.10.1.4      *Input Variable: decrement rates***

Description:	Decrement rate tables
Help:	Decrement table by sex and smoker status. If set_by_prodcod = "Y" then table is set based on "risk_rates" in prod assumptions table.
	Not relevant for health and death benefits.
Associated Code Variables:	decrem_rates,decrem_rates_res
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	0
Valid Range To:	0
Choice List:	
Value Type:	External Source
Value:	decrem_rates_com

**2.1.10.1.5      *Input Variable: decrement rates check***

Description:	Decrement rate tables
Help:	Decrement table by sex and smoker status. If set_by_prodcod = "Y" then table is set based on "risk_rates" in prod assumptions table.
	Not relevant for health and death benefits.
Associated Code Variables:	decrem_rates_check
Modified On:	7/27/2021 11:56:06 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	0
Valid Range To:	0
Choice List:	
Value Type:	External Source

Value: decrem\_rates\_com

#### **2.1.10.1.6      *Input Variable: decrement rates by UW date***

Description: Decrement rate tables  
 Help: Decrement table by sex and smoker status.  
 If set\_by\_procode = "Y" then table is set based on "risk\_rates" in prod assumptions table.  
 Not relevant for health and death benefits.  
 Associated Code Variables: decrem\_rates\_uw, decrem\_rates\_uw\_res  
 Modified On: 3/19/2024 7:30:07 PM (UTC+02:00)  
 Modified By: CLAL-INS\Arikt  
 Validation Failure Behaviour: Error  
 Variable Type: Floating Point Number  
 Valid Range From: 0  
 Valid Range To: 0  
 Choice List:  
 Value Type: External Source  
 Value: decrem\_rates\_uw\_com

#### **2.1.10.1.7      *Input Variable: decrem\_mult\_tbl***

Description: decrement multiplier table (%)  
 Help: Percentage of basic decrement table used for all lives, split by M/F, S/NS/A, occupation class and product type.  
 Associated Code Variables: decrem\_mult\_tbl  
 Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
 Modified By: CLAL-INS\NinaB  
 Validation Failure Behaviour: Error  
 Variable Type: Floating Point Number  
 Valid Range From:  
 Valid Range To:  
 Choice List:  
 Value Type: External Source  
 Value: decrmult

#### **2.1.10.1.8      *Input Variable: mort\_mult\_end\_age***

Description: Age from which to phase-out mortality multiplier  
 Help: At this age the mortality multiplier will gradually be phased-out (i.e. approach 100%) until the omega-age, so that at very old ages the base mortality table is less effected by the multiplier.  
 Associated Code Variables: mort\_mult\_end\_age  
 Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
 Modified By: CLAL-INS\NinaB  
 Validation Failure Behaviour: Error  
 Variable Type: Integer Number  
 Valid Range From: 0

Valid Range To: 75  
 Choice List:  
 Value Type: External Source  
 Value: Parameters

#### **2.1.10.1.9      *Input Variable: antisel\_margin***

Description: Antiselection margin for Annuity Old money  
 Help:  
 Associated Code Variables: antisel\_margin  
 Modified On: 6/30/2024 2:51:20 PM (UTC+03:00)  
 Modified By: CLAL-INS\arikt  
 Validation Failure Behaviour: Error  
 Variable Type: Integer Number  
 Valid Range From: 0  
 Valid Range To: 75  
 Choice List:  
 Value Type: External Source  
 Value: Parameters

#### **2.1.10.1.10      *Input Variable: mort\_mult\_tbl***

Description: Mortality multiplier table (%)  
 Help: Percentage of basic mortality table used for all lives, split by M/F, S/NS/A, and product type.  
 Associated Code Variables: mort\_mult\_tbl  
 Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
 Modified By: CLAL-INS\NinaB  
 Validation Failure Behaviour: Error  
 Variable Type: Floating Point Number  
 Valid Range From: 0  
 Valid Range To: 0  
 Choice List:  
 Value Type: External Source  
 Value: mortmult

#### **2.1.10.1.11      *Input Variable: survive\_tbl***

Description: survival factors table for extra annuity reserve  
 Help: nPx factors (survival to age 65) used for extra annuity reserve, based on adjusted table 4a1 as calculated in Clal's annuity reserve spreadsheet.  
 Associated Code Variables: survive\_tbl  
 Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
 Modified By: CLAL-INS\NinaB  
 Validation Failure Behaviour: Error  
 Variable Type: Floating Point Number  
 Valid Range From: 0  
 Valid Range To: 1  
 Choice List:  
 Value Type: External Source

Value: Survival\_Rates

#### **2.1.10.1.12      *Input Variable: select\_periods***

Description: Mortality select periods  
 Help:  
 Associated Code Variables: select\_periods  
 Modified On: 1/9/2023 11:57:48 AM (UTC+02:00)  
 Modified By: CLAL-INS\joshm  
 Validation Failure Behaviour: Error  
 Variable Type: Floating Point Number  
 Valid Range From:  
 Valid Range To:  
 Choice List:  
 Value Type: External Source  
 Value: Parameters

#### **2.1.10.1.13      *Input Variable: omega\_age***

Description: omega age  
 Help: Highest age in mortality table. Internal logic variable calculated in calc\_omega\_age.  
 Associated Code Variables: omega\_age,omega\_age\_dec,omega\_age\_cmi  
 Modified On: 1/11/2023 11:20:18 AM (UTC+02:00)  
 Modified By: CLAL-INS\joshm  
 Validation Failure Behaviour: Error  
 Variable Type: Floating Point Number  
 Valid Range From:  
 Valid Range To:  
 Choice List:  
 Value Type: External Source  
 Value: Parameters

#### **2.1.10.1.14      *Input Variable: death\_rates\_ann\_m\_1***

Description: Male Annuiants death rate table  
 Help: Death-only rate table  
 Associated Code Variables: death\_rates\_ann\_m\_1  
 Modified On: 7/26/2021 2:51:58 PM (UTC+03:00)  
 Modified By: CLAL-INS\joshm  
 Validation Failure Behaviour: Error  
 Variable Type: Floating Point Number  
 Valid Range From: 0  
 Valid Range To: 0  
 Choice List:  
 Value Type: External Source  
 Value: ann\_mort\_08\_M\_BE

#### **2.1.10.1.15      *Input Variable: sel\_ret\_qx\_im\_dth\_1***

Description:

Help:	Death-only rate table
Associated Code Variables:	sel_ret_qx_im_dth_1
Modified On:	7/23/2024 3:32:55 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	
Valid Range To:	
Choice List:	
Value Type:	External Source
Value:	Sel_Ret_Qx

#### **2.1.10.1.16      *Input Variable: sel\_ret\_qx\_im\_dth\_2***

Description:	
Help:	Death-only rate table
Associated Code Variables:	sel_ret_qx_im_dth_2
Modified On:	7/23/2024 3:37:21 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	
Valid Range To:	
Choice List:	
Value Type:	External Source
Value:	Sel_Ret_Qx

#### **2.1.10.1.17      *Input Variable: death\_rates\_ann\_f\_1***

Description:	Female Annuiants death rate table
Help:	Death-only rate table
Associated Code Variables:	death_rates_ann_f_1
Modified On:	7/26/2021 2:52:01 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	0
Valid Range To:	0
Choice List:	
Value Type:	External Source
Value:	ann_mort_08_F_BE

#### **2.1.10.1.18      *Input Variable: death\_rates\_ann\_m\_2***

Description:	Male Annuiants death rate table
Help:	Death-only rate table
Associated Code Variables:	death_rates_ann_m_2
Modified On:	7/26/2021 2:57:43 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Validation Failure Behaviour:	Error

Variable Type:	Floating Point Number
Valid Range From:	0
Valid Range To:	0
Choice List:	
Value Type:	External Source
Value:	ann_mort_08_M_BE

#### **2.1.10.1.19      *Input Variable: death\_rates\_ann\_f\_2***

Description:	Female Annuants death rate table
Help:	Death-only rate table
Associated Code Variables:	death_rates_ann_f_2
Modified On:	7/26/2021 2:57:58 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	0
Valid Range To:	0
Choice List:	
Value Type:	External Source
Value:	ann_mort_08_F_BE

#### **2.1.10.1.20      *Input Variable: death\_rates\_ann\_m\_b3\_2***

Description:	Male Annuants death rate table - B3 mortality
Help:	Death-only rate table
Associated Code Variables:	death_rates_ann_m_b3_2
Modified On:	7/26/2021 2:58:08 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	0
Valid Range To:	0
Choice List:	
Value Type:	External Source
Value:	ann_mort_b3_08_M_BE

#### **2.1.10.1.21      *Input Variable: death\_rates\_ann\_f\_b3\_2***

Description:	Female Annuants death rate table - B3 mortality
Help:	Death-only rate table
Associated Code Variables:	death_rates_ann_f_b3_2
Modified On:	7/26/2021 2:58:15 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	0
Valid Range To:	0
Choice List:	
Value Type:	External Source



Value: ann\_mort\_b3\_08\_F\_BE

#### **2.1.10.1.22      *Input Variable: death\_rates\_ann\_m\_res\_1***

Description: Male Annuants death rate table  
 Help: Death-only rate table  
 Associated Code Variables: death\_rates\_ann\_m\_res\_1  
 Modified On: 7/26/2021 2:59:04 PM (UTC+03:00)  
 Modified By: CLAL-INS\joshm  
 Validation Failure Behaviour: Error  
 Variable Type: Floating Point Number  
 Valid Range From: 0  
 Valid Range To: 0  
 Choice List:  
 Value Type: External Source  
 Value: ann\_mort\_08\_M\_res

#### **2.1.10.1.23      *Input Variable: death\_rates\_ann\_m\_res\_2***

Description: Male Annuants death rate table  
 Help: Death-only rate table  
 Associated Code Variables: death\_rates\_ann\_m\_res\_2  
 Modified On: 7/26/2021 3:01:13 PM (UTC+03:00)  
 Modified By: CLAL-INS\joshm  
 Validation Failure Behaviour: Error  
 Variable Type: Floating Point Number  
 Valid Range From: 0  
 Valid Range To: 0  
 Choice List:  
 Value Type: External Source  
 Value: ann\_mort\_08\_M\_res

#### **2.1.10.1.24      *Input Variable: death\_rates\_ann\_m\_res\_tt***

Description: Male Annuants death rate table  
 Help: Death-only rate table  
 Associated Code Variables: death\_rates\_ann\_m\_res\_tt  
 Modified On: 7/26/2021 3:08:57 PM (UTC+03:00)  
 Modified By: CLAL-INS\joshm  
 Validation Failure Behaviour: Error  
 Variable Type: Floating Point Number  
 Valid Range From: 0  
 Valid Range To: 0  
 Choice List:  
 Value Type: External Source  
 Value: ann\_mort\_08\_M\_res

#### **2.1.10.1.25      *Input Variable: death\_rates\_ann\_f\_res\_1***

Description: Female Annuants death rate table  
 Help: Death-only rate table

Associated Code Variables: death\_rates\_ann\_f\_res\_1  
Modified On: 7/26/2021 2:59:09 PM (UTC+03:00)  
Modified By: CLAL-INS\joshm  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From: 0  
Valid Range To: 0  
Choice List:  
Value Type: External Source  
Value: ann\_mort\_08\_F\_res

#### **2.1.10.1.26      *Input Variable: death\_rates\_ann\_f\_res\_2***

Description: Female Annuiants death rate table  
Help: Death-only rate table  
Associated Code Variables: death\_rates\_ann\_f\_res\_2  
Modified On: 7/26/2021 3:01:21 PM (UTC+03:00)  
Modified By: CLAL-INS\joshm  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From: 0  
Valid Range To: 0  
Choice List:  
Value Type: External Source  
Value: ann\_mort\_08\_F\_res

#### **2.1.10.1.27      *Input Variable: death\_rates\_ann\_f\_res\_tt***

Description: Female Annuiants death rate table  
Help: Death-only rate table  
Associated Code Variables: death\_rates\_ann\_f\_res\_tt  
Modified On: 7/26/2021 3:09:03 PM (UTC+03:00)  
Modified By: CLAL-INS\joshm  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From: 0  
Valid Range To: 0  
Choice List:  
Value Type: External Source  
Value: ann\_mort\_08\_F\_res

#### **2.1.10.1.28      *Input Variable: death\_rates\_ann\_m\_res\_b3\_2***

Description: Male Annuiants death rate table - B3 Mortality  
Help: Death-only rate table  
Associated Code Variables: death\_rates\_ann\_m\_res\_b3\_2  
Modified On: 7/26/2021 3:01:36 PM (UTC+03:00)  
Modified By: CLAL-INS\joshm  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number

Valid Range From: 0  
Valid Range To: 0  
Choice List:  
Value Type: External Source  
Value: ann\_mort\_b3\_08\_M\_res

#### **2.1.10.1.29      *Input Variable: death\_rates\_ann\_m\_res\_b3\_tt***

Description: Male Annuiants death rate table - B3 Mortality  
Help: Death-only rate table  
Associated Code Variables: death\_rates\_ann\_m\_res\_b3\_tt  
Modified On: 7/26/2021 3:09:16 PM (UTC+03:00)  
Modified By: CLAL-INS\joshm  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From: 0  
Valid Range To: 0  
Choice List:  
Value Type: External Source  
Value: ann\_mort\_b3\_08\_M\_res

#### **2.1.10.1.30      *Input Variable: death\_rates\_ann\_f\_res\_b3\_2***

Description: Female Annuiants death rate table - B3 mortality  
Help: Death-only rate table  
Associated Code Variables: death\_rates\_ann\_f\_res\_b3\_2  
Modified On: 7/26/2021 3:01:29 PM (UTC+03:00)  
Modified By: CLAL-INS\joshm  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From: 0  
Valid Range To: 0  
Choice List:  
Value Type: External Source  
Value: ann\_mort\_b3\_08\_F\_res

#### **2.1.10.1.31      *Input Variable: death\_rates\_ann\_f\_res\_b3\_tt***

Description: Female Annuiants death rate table - B3 mortality  
Help: Death-only rate table  
Associated Code Variables: death\_rates\_ann\_f\_res\_b3\_tt  
Modified On: 7/26/2021 3:09:21 PM (UTC+03:00)  
Modified By: CLAL-INS\joshm  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From: 0  
Valid Range To: 0  
Choice List:  
Value Type: External Source  
Value: ann\_mort\_b3\_08\_F\_res

**2.1.10.1.32      *Input Variable: death\_rates\_res\_tbl***

Description:

Help:

Associated Code Variables:

death\_rates\_res\_tbl

Modified On:

8/27/2019 4:00:59 PM (UTC+03:00)

Modified By:

CLAL-INS\NinaB

Validation Failure Behaviour:

Error

Variable Type:

Floating Point Number

Valid Range From:

Valid Range To:

Choice List:

Value Type:

External Source

Value:

death\_rates\_res\_comp

**2.1.11      Input Page: Premium****2.1.11.1      Assumption Set: Base**

Description:

Help:

Top Model Object:

life

Modified On:

2/11/2024 2:47:13 PM (UTC+02:00)

Modified By:

CLAL-INS\Arikt

Assumption Set Links:

Unlinked

**2.1.11.1.1      *Input Variable: Various\_Parameters***

Description:

Help:

Associated Code Variables:

prem\_newtag\_prop,netprem\_max,pol\_fee\_disc\_perc  
,prem\_risk\_max,tagmulim\_perc,health\_occ\_perc\_min

Modified On:

3/13/2023 3:14:06 PM (UTC+02:00)

Modified By:

CLAL-INS\ahuvaa

Validation Failure Behaviour:

Error

Variable Type:

Floating Point Number

Valid Range From:

0

Valid Range To:

100000

Choice List:

Value Type:

External Source

Value:

Parameters

**2.1.11.1.2      *Input Variable: prem\_rates\_series\_end\_im***

Description:

Prem Rates Im

Help:

Associated Code Variables:

prem\_rates\_series\_end\_im

Modified On:

8/27/2019 4:00:59 PM (UTC+03:00)

Modified By:

CLAL-INS\NinaB

Validation Failure Behaviour:

Error

Variable Type: Integer Number  
Valid Range From:  
Valid Range To:  
Choice List:  
Value Type: External Source  
Value: prem\_rates

#### **2.1.11.1.3      *Input Variable: prem\_rates\_series***

Description: Premium rate table  
Help:  
Associated Code Variables: prem\_rates\_series  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Integer Number  
Valid Range From: 0  
Valid Range To: 10000  
Choice List:  
Value Type: External Source  
Value: prem\_rates

#### **2.1.11.1.4      *Input Variable: prem\_key\_temp\_rates***

Description: Premium rate table  
Help:  
Associated Code Variables: prem\_rates\_temp\_series\_end  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Integer Number  
Valid Range From: 0  
Valid Range To: 10000  
Choice List:  
Value Type: External Source  
Value: prem\_rates

#### **2.1.11.1.5      *Input Variable: prem\_rates\_charge\_tt***

Description: Premium rate table  
Help:  
Associated Code Variables: prem\_rates\_charge\_tt  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Integer Number  
Valid Range From: 0  
Valid Range To: 10000  
Choice List:  
Value Type: External Source

Value: prem\_rates

#### **2.1.11.1.6      *Input Variable: prem\_if\_rates***

Description: Premium rate table  
Help:  
Associated Code Variables: prem\_if\_rates  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Integer Number  
Valid Range From: 0  
Valid Range To: 10000  
Choice List:  
Value Type: External Source  
Value: prem\_rates

#### **2.1.11.1.7      *Input Variable: prem\_rates\_others***

Description: Premium rate table  
Help:  
Associated Code Variables: base  
Modified On: 5/3/2022 4:29:44 PM (UTC+03:00)  
Modified By: CLAL-INS\ahuvaa  
Validation Failure Behaviour: Error  
Variable Type: Integer Number  
Valid Range From: 0  
Valid Range To: 10000  
Choice List:  
Value Type: External Source  
Value: prem\_rates

#### **2.1.11.1.8      *Input Variable: prem\_rates\_si***

Description: Premium rate table  
Help:  
Associated Code Variables: prem\_rates\_si  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Integer Number  
Valid Range From: 0  
Valid Range To: 10000  
Choice List:  
Value Type: External Source  
Value: prem\_rates

#### **2.1.11.1.9      *Input Variable: sal\_tbl***

Description: Tables for salary increases  
Help:

Associated Code Variables:	sal_tbl
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	0
Valid Range To:	0
Choice List:	
Value Type:	External Source
Value:	sal_inc

#### **2.1.11.1.10      *Input Variable: sal\_rider\_tbl***

Description:	Tables for salary increases
Help:	
Associated Code Variables:	sal_rider_tbl
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	0
Valid Range To:	0
Choice List:	
Value Type:	External Source
Value:	sal_inc

#### **2.1.11.1.11      *Input Variable: prem\_rates\_extra\_prm***

Description:	
Help:	
Associated Code Variables:	prem_rates_extra_prm
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	
Valid Range To:	
Choice List:	
Value Type:	External Source
Value:	prem_rates_extra

#### **2.1.11.1.12      *Input Variable: prem\_rates\_risk\_1***

Description:	
Help:	
Associated Code Variables:	prem_rates_risk_1
Modified On:	5/2/2022 11:43:31 AM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number

Valid Range From:  
Valid Range To:  
Choice List:  
Value Type: External Source  
Value: prem\_rates\_risk\_comp

#### **2.1.11.1.13 Input Variable: prem\_rates\_risk\_2**

Description:  
Help:  
Associated Code Variables: prem\_rates\_risk\_2  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:  
Choice List:  
Value Type: External Source  
Value: prem\_rates\_risk\_comp

#### **2.1.11.1.14 Input Variable: prem\_rates\_risk\_rider**

Description:  
Help:  
Associated Code Variables: prem\_rates\_risk\_rider  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:  
Choice List:  
Value Type: External Source  
Value: prem\_rates\_risk\_rider

#### **2.1.11.1.15 Input Variable: prem\_code\_map\_tbl**

Description: health prem\_code\_map\_tbl  
Help:  
Associated Code Variables: prem\_code\_map\_tbl  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Character  
Valid Range From: 0  
Valid Range To: 0  
Choice List:  
Value Type: External Source  
Value: prem\_code\_map



**2.1.11.1.16      *Input Variable: prate\_level\_tbl***

Description:	Prate Level Tbl
Help:	level premium rate rate table. looked up by age and benefit-term (NOT premium-term)
Associated Code Variables:	prate_level_tbl
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	
Valid Range To:	
Choice List:	
Value Type:	External Source
Value:	prem_rates_level_comp

**2.1.11.1.17      *Input Variable: zillmer\_pr\_tbl***

Description:	Zillmer rates table (% of premium)
Help:	Table of zillmer premium rates by policy type and dac purpose (book or taxe). Set for the 10 first policy years ( = 0 after).
Associated Code Variables:	zillmer_pr_tbl
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	0
Valid Range To:	0
Choice List:	
Value Type:	External Source
Value:	zillmer_prm

**2.1.11.1.18      *Input Variable: prem\_disc\_shimur***

Description:	
Help:	
Associated Code Variables:	prem_disc_shimur_im
Modified On:	2/11/2024 3:24:57 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	
Valid Range To:	
Choice List:	
Value Type:	External Source
Value:	Shimur_disc

## 2.1.12 Input Page: Product Details

### 2.1.12.1 Assumption Set: Base

Description:  
Help:  
Top Model Object: life  
Modified On: 2/12/2024 12:20:57 PM (UTC+02:00)  
Modified By: CLAL-INS\Arikt  
Assumption Set Links: Unlinked

#### 2.1.12.1.1 Input Variable: *prod\_assumpt\_tbl*

Description: Product code specific assumptions  
Help:  
Associated Code Variables: comm\_set,clms\_mult\_set,clwback\_set,exp\_mult\_set,alloc\_rate\_set,surr\_chg\_set,lapse\_set\_riders,decrem\_mult\_set,exp\_set\_cvr,prodcode\_par\_nonpar,sal\_inc\_set  
Modified On: 11/12/2024 4:34:45 PM (UTC+02:00)  
Modified By: CLAL-INS\arikt  
Validation Failure Behaviour: Error  
Variable Type: Character  
Valid Range From: 0  
Valid Range To: 0  
Choice List:  
Value Type: External Source  
Value: prod\_ass

#### 2.1.12.1.2 Input Variable: *savings\_pol\_prod\_code*

Description: Product code specific assumptions  
Help:  
Associated Code Variables: savings\_pol\_prod\_code  
Modified On: 3/23/2023 11:07:17 AM (UTC+02:00)  
Modified By: CLAL-INS\ahuvaa  
Validation Failure Behaviour: Error  
Variable Type: Character  
Valid Range From: 0  
Valid Range To: 0  
Choice List:  
Value Type: External Source  
Value: prod\_ass

#### 2.1.12.1.3 Input Variable: *prod\_assumpt\_base\_tbl*

Description: Product code specific assumptions  
Help:  
Associated Code Variables: lapse\_set,exp\_set\_pol,savings\_pol,mort\_mult\_set  
Modified On: 11/12/2024 4:35:11 PM (UTC+02:00)  
Modified By: CLAL-INS\arikt

Validation Failure Behaviour:	Error
Variable Type:	Character
Valid Range From:	0
Valid Range To:	0
Choice List:	
Value Type:	External Source
Value:	prod_ass

#### **2.1.12.1.4      *Input Variable: prod\_assumpt\_key\_tbl***

Description:	Product code specific assumptions
Help:	Product code specific assumptions. Used only if lookup_by_prodcode = "Y".
Associated Code Variables:	prod_assumpt_key_tbl
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Character
Valid Range From:	0
Valid Range To:	0
Choice List:	
Value Type:	External Source
Value:	prod_ass

#### **2.1.12.1.5      *Input Variable: prod\_assumpt\_rider\_exp\_tbl***

Description:	Product code specific assumptions
Help:	Product code specific assumptions. Used only if lookup_by_prodcode = "Y".
Associated Code Variables:	prod_assumpt_rider_exp_tbl
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Character
Valid Range From:	0
Valid Range To:	0
Choice List:	
Value Type:	External Source
Value:	prod_ass

#### **2.1.12.1.6      *Input Variable: sal\_inc\_set\_rider***

Description:	Product code specific assumptions
Help:	
Associated Code Variables:	sal_inc_set_rider
Modified On:	12/29/2022 5:38:30 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Validation Failure Behaviour:	Error
Variable Type:	Character
Valid Range From:	0
Valid Range To:	0

Choice List:  
 Value Type: External Source  
 Value: prod\_ass

#### **2.1.12.1.7      *Input Variable: prod\_assumpt\_rider\_lapse\_tbl***

Description: Product code specific assumptions  
 Help: Product code specific assumptions. Used only if lookup\_by\_prodcode = "Y".  
 Associated Code Variables: prod\_assumpt\_rider\_lapse\_tbl  
 Modified On: 6/15/2022 9:39:18 AM (UTC+03:00)  
 Modified By: CLAL-INS\ahuvaa  
 Validation Failure Behaviour: Error  
 Variable Type: Character  
 Valid Range From: 0  
 Valid Range To: 0  
 Choice List:  
 Value Type: External Source  
 Value: prod\_ass

#### **2.1.12.1.8      *Input Variable: prod\_assumpt\_rider\_clms\_tbl***

Description: Product code specific assumptions  
 Help:  
 Associated Code Variables: prod\_assumpt\_rider\_clms\_tbl  
 Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
 Modified By: CLAL-INS\NinaB  
 Validation Failure Behaviour: Error  
 Variable Type: Character  
 Valid Range From: 0  
 Valid Range To: 0  
 Choice List:  
 Value Type: External Source  
 Value: prod\_ass

#### **2.1.12.1.9      *Input Variable: prod\_spec\_term***

Description: prod spec term  
 Help:  
 Associated Code Variables: res\_basis,prem\_lookup,prem\_lookup\_freq,adjust\_prem\_and\_claims,prem\_init\_different,dd\_prop\_cont  
 Modified On: 6/2/2022 2:15:27 PM (UTC+03:00)  
 Modified By: CLAL-INS\ahuvaa  
 Validation Failure Behaviour: Error  
 Variable Type: Character  
 Valid Range From: 0  
 Valid Range To: 0  
 Choice List:  
 Value Type: External Source  
 Value: prod\_spec\_term

**2.1.12.1.10      *Input Variable: prod\_spec\_trad***

Description:	prod_spec_trad
Help:	
Associated Code Variables:	procdold,matan_perc,prem_inc,prem_age,sum_inc
Modified On:	5/16/2023 10:08:39 AM (UTC+03:00)
Modified By:	CLAL-INS\Arikt
Validation Failure Behaviour:	Error
Variable Type:	Character
Valid Range From:	0
Valid Range To:	0
Choice List:	
Value Type:	External Source
Value:	prod_spec_trad

**2.1.12.1.11      *Input Variable: prem\_lookup\_trad***

Description:	prem_lookup_trad
Help:	
Associated Code Variables:	prem_lookup_trad
Modified On:	8/16/2021 11:05:58 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Validation Failure Behaviour:	Error
Variable Type:	Character
Valid Range From:	0
Valid Range To:	0
Choice List:	
Value Type:	External Source
Value:	prod_spec_trad

**2.1.12.1.12      *Input Variable: prem\_lookup\_freq\_trad***

Description:	prem_lookup_freq_trad
Help:	
Associated Code Variables:	prem_lookup_freq_trad
Modified On:	8/29/2021 12:10:42 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Validation Failure Behaviour:	Error
Variable Type:	Character
Valid Range From:	0
Valid Range To:	0
Choice List:	
Value Type:	External Source
Value:	prod_spec_trad

**2.1.12.1.13      *Input Variable: pup\_sv\_charge\_rebate***

Description:	pup_sv_charge_rebate
Help:	
Associated Code Variables:	pup_sv_charge_rebate
Modified On:	8/29/2021 11:39:59 AM (UTC+03:00)

Modified By:	CLAL-INS\joshm
Validation Failure Behaviour:	Error
Variable Type:	Character
Valid Range From:	0
Valid Range To:	0
Choice List:	
Value Type:	External Source
Value:	prod_spec_unit

#### **2.1.12.1.14      *Input Variable: prod\_specs\_max\_perc***

Description:	prod_specs_max_perc
Help:	
Associated Code Variables:	prod_specs_max
Modified On:	8/29/2021 11:50:17 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Validation Failure Behaviour:	Error
Variable Type:	Character
Valid Range From:	0
Valid Range To:	0
Choice List:	
Value Type:	External Source
Value:	prod_spec_unit

#### **2.1.12.1.15      *Input Variable: prod\_spec\_risk\_code***

Description:	prod_spec_risk_code
Help:	
Associated Code Variables:	prod_spec_risk_code
Modified On:	8/29/2021 12:25:42 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Validation Failure Behaviour:	Error
Variable Type:	Character
Valid Range From:	0
Valid Range To:	0
Choice List:	
Value Type:	External Source
Value:	prod_spec_unit

#### **2.1.12.1.16      *Input Variable: prod\_specs\_rider***

Description:	prod_specs_rider
Help:	
Associated Code Variables:	prod_specs_rider
Modified On:	8/29/2021 12:40:17 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Validation Failure Behaviour:	Error
Variable Type:	Character
Valid Range From:	0
Valid Range To:	0

Choice List:  
 Value Type: External Source  
 Value: prod\_spec\_unit

#### **2.1.12.1.17 Input Variable: rider\_tarif\_tbl**

Description: Table to map Profil Riders Tarif code to product code  
 Help:  
 Associated Code Variables: rider\_tarif\_tbl  
 Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
 Modified By: CLAL-INS\NinaB  
 Validation Failure Behaviour: Error  
 Variable Type: Character  
 Valid Range From: 0  
 Valid Range To: 0  
 Choice List:  
 Value Type: External Source  
 Value: profil\_rider\_tarif\_map

#### **2.1.12.1.18 Input Variable: tarif\_spec**

Description: set tarif spec. var. with string value  
 Help:  
 Associated Code Variables: pitzui\_shichrur,prem\_key\_start,premkey\_endage,pre  
 mkey\_sex,premkey\_smoker,premkey\_insured,prem\_  
 series\_year,claims\_cost\_key\_start,claimskey\_endag  
 e,claimskey\_sex,claims\_series\_year,claims\_factor,w  
 aiting\_period\_modeled,incidencerate\_key,prem\_fact  
 or,reins\_key\_start,premkey\_occ,lapse\_tarif\_set,blue  
 \_white  
 Modified On: 7/15/2024 3:53:50 PM (UTC+03:00)  
 Modified By: CLAL-INS\arikt  
 Validation Failure Behaviour: Error  
 Variable Type: Character  
 Valid Range From: 0  
 Valid Range To: 0  
 Choice List:  
 Value Type: External Source  
 Value: tarif\_spec

#### **2.1.12.1.19 Input Variable: tarif\_spec\_lookup\_freq**

Description: set tarif spec. var. with string value  
 Help:  
 Associated Code Variables: tarif\_spec\_lookup\_freq  
 Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
 Modified By: CLAL-INS\NinaB  
 Validation Failure Behaviour: Error  
 Variable Type: Character  
 Valid Range From: 0  
 Valid Range To: 0

Choice List:  
Value Type: External Source  
Value: tarif\_spec

#### **2.1.12.1.20      *Input Variable: alloc\_rate\_stri***

Description: Allocation rate table  
Help:  
Associated Code Variables: alloc\_rate\_stri  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From: 0  
Valid Range To: 0  
Choice List:  
Value Type: External Source  
Value: alloc

#### **2.1.12.1.21      *Input Variable: suminisba\_tbl***

Description:  
Help:  
Associated Code Variables: suminisba\_tbl  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:  
Choice List:  
Value Type: External Source  
Value: suminisba

#### **2.1.12.1.22      *Input Variable: claims\_factor\_occ***

Description:  
Help:  
Associated Code Variables: claims\_factor\_occ  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Character  
Valid Range From:  
Valid Range To:  
Choice List: 0  
Value Type: External Source  
Value: tarif\_spec\_occ



### 2.1.12.1.23 *Input Variable: bonus\_tbl*

Description:	
Help:	Table with persistency bonus rates by policy month. This is always used (even if read_from_tables = "N") - looked up by surr_charge_set
Associated Code Variables:	bonus_tbl
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	
Valid Range To:	
Choice List:	
Value Type:	External Source
Value:	bonus5

### 2.1.12.1.24 *Input Variable: gorem\_mult*

Description:	
Help:	
Associated Code Variables:	gorem_mult
Modified On:	12/10/2023 11:57:18 AM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	
Valid Range To:	
Choice List:	
Value Type:	External Source
Value:	prod_ass

## 2.1.13 **Input Page: Reinsurance**

### 2.1.13.1 **Assumption Set: Base**

Description:	
Help:	
Top Model Object:	life
Modified On:	7/15/2024 1:13:24 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Assumption Set Links:	Unlinked

#### 2.1.13.1.1 *Input Variable: prem\_rates\_re*

Description:	reinsurance premium rates for life
Help:	
Associated Code Variables:	prem_rates_re
Modified On:	7/22/2021 9:48:29 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Validation Failure Behaviour:	Error

Variable Type:	Floating Point Number
Valid Range From:	0
Valid Range To:	0
Choice List:	
Value Type:	External Source
Value:	premium_rates_rein_life

#### **2.1.13.1.2      *Input Variable: reinsur\_w***

Description:	table reinsurance treaties assumptions
Help:	
Associated Code Variables:	exp_re_nom,prof_comm,interest,retention,madad,typ e,prem_extra,quota_share_ppn,prem_re_mult,comm _by_cal
Modified On:	1/5/2025 3:04:01 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Validation Failure Behaviour:	Error
Variable Type:	Character
Valid Range From:	0
Valid Range To:	0
Choice List:	
Value Type:	External Source
Value:	life_treaty_details

#### **2.1.13.1.3      *Input Variable: rein\_series\_end\_key\_temp***

Description:	table reinsurance treaties assumptions
Help:	
Associated Code Variables:	rein_series_end_key_temp
Modified On:	10/25/2021 3:08:53 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Validation Failure Behaviour:	Error
Variable Type:	Character
Valid Range From:	0
Valid Range To:	0
Choice List:	
Value Type:	External Source
Value:	life_treaty_details

#### **2.1.13.1.4      *Input Variable: reinsur\_kod\_tavla***

Description:	table reinsurance treaties assumptions
Help:	
Associated Code Variables:	reinsur_kod_tavla
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Character
Valid Range From:	0
Valid Range To:	0
Choice List:	

Value Type: External Source  
Value: life\_treaty\_details

#### **2.1.13.1.5      *Input Variable: reinsur\_comm***

Description: table reinsurance treaties assumptions  
Help:  
Associated Code Variables: reinsur\_comm  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Character  
Valid Range From: 0  
Valid Range To: 0  
Choice List:  
Value Type: External Source  
Value: life\_treaty\_details

#### **2.1.13.1.6      *Input Variable: reinsur\_simple\_perc***

Description:  
Help:  
Associated Code Variables: reinsur\_simple\_perc  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:  
Choice List:  
Value Type: External Source  
Value: LifeReins

#### **2.1.13.1.7      *Input Variable: reinsur\_simple\_cost***

Description:  
Help:  
Associated Code Variables: reinsure\_simple\_cost  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:  
Choice List:  
Value Type: External Source  
Value: LifeReins

#### **2.1.13.1.8      *Input Variable: reinsur\_simple\_rider\_cost***

Description:

Help:  
 Associated Code Variables: reinsur\_simple\_rider\_cost  
 Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
 Modified By: CLAL-INS\NinaB  
 Validation Failure Behaviour: Error  
 Variable Type: Floating Point Number  
 Valid Range From:  
 Valid Range To:  
 Choice List:  
 Value Type: External Source  
 Value: LifeReins

## 2.1.14 Input Page: Reserve

### 2.1.14.1 Assumption Set: Base

Description:  
 Help:  
 Top Model Object: life  
 Modified On: 11/14/2024 4:52:04 PM (UTC+02:00)  
 Modified By: CLAL-INS\arikt  
 Assumption Set Links: Unlinked

#### 2.1.14.1.1 Input Variable: *err\_sar\_perc*

Description: ERR as % of sum at risk  
 Help: extra-ordinary reserve  
 Associated Code Variables: err\_sar\_perc  
 Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
 Modified By: CLAL-INS\NinaB  
 Validation Failure Behaviour: Error  
 Variable Type: Floating Point Number  
 Valid Range From: 0  
 Valid Range To: 0.3  
 Choice List:  
 Value Type: External Source  
 Value: Parameters

#### 2.1.14.1.2 Input Variable: *err\_spread\_period*

Description: years to build up err  
 Help: number of years over which err is built up  
 Associated Code Variables: err\_spread\_period  
 Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
 Modified By: CLAL-INS\NinaB  
 Validation Failure Behaviour: Error  
 Variable Type: Floating Point Number  
 Valid Range From: 0  
 Valid Range To: 30  
 Choice List:

Value Type:	External Source
Value:	Parameters

### **2.1.14.1.3      *Input Variable: zeroise\_res***

Description:	Zeroise negative reserves (Y/N)?
Help:	Y = Individual negative reserves are set to zero
Associated Code Variables:	zeroise_res
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Character
Valid Range From:	0
Valid Range To:	0
Choice List:	Y,N
Value Type:	External Source
Value:	Parameters

### **2.1.14.1.4      *Input Variable: reserve\_factors\_tbl***

Description:	reserve factors table
Help:	
Associated Code Variables:	reserve_factors_tbl
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	0
Valid Range To:	0
Choice List:	
Value Type:	External Source
Value:	Reserve_Factors

### **2.1.14.1.5      *Input Variable: zillmer\_adj\_factor***

Description:	Adjustment factor for Zillmer (to scale up to actuals)
Help:	The percentage applied to the DAC tax value taken from the data file to adjust it according to the actual DAC held.
Associated Code Variables:	zillmer_adj_factor
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	0
Valid Range To:	1000
Choice List:	
Value Type:	External Source
Value:	Economic

**2.1.14.1.6      *Input Variable: AnnuitySets***

Description:	Reserves Annuities Sets
Help:	variable linked with the kitzba reserve field from the inforce file
Associated Code Variables:	res_anndef_lapse,res_anndef_lapse_par
Modified On:	11/9/2022 9:40:50 AM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Validation Failure Behaviour:	Error
Variable Type:	Integer Number
Valid Range From:	0
Valid Range To:	100
Choice List:	
Value Type:	External Source
Value:	Annuity

**2.1.14.1.7      *Input Variable: Res\_Adj\_Factor***

Description:	Reserves adjustment factor
Help:	
Associated Code Variables:	res_adj_factor
Modified On:	8/5/2024 3:51:52 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	
Valid Range To:	
Choice List:	
Value Type:	External Source
Value:	Reserve_Manual

**2.1.14.1.8      *Input Variable: Comm\_reserves\_AddVAT***

Description:	Add VAT to commres (Y/N)?
Help:	
Associated Code Variables:	commres_addvat
Modified On:	11/14/2024 4:53:42 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Validation Failure Behaviour:	Error
Variable Type:	Character
Valid Range From:	0
Valid Range To:	0
Choice List:	Y,N
Value Type:	External Source
Value:	Parameters

**2.1.15      *Input Page: Setup*****2.1.15.1      *Assumption Set: Solv\_Base***

Description:

Help:  
 Top Model Object: life  
 Modified On: 12/3/2024 5:14:30 PM (UTC+02:00)  
 Modified By: CLAL-INS\ahuvaa  
 Assumption Set Links: Unlinked

### 2.1.15.1.1 *Input Variable: RunControl\_Char*

Description:  
 Help: Internal logic variable set in startup. Y = In negative periods gross up survivorship.  
 Associated Code Variables: gross\_up\_historic,reinsurance,comm\_extra\_agent\_use,projection\_type,projection\_type\_int,decrements\_apply,mort\_sel\_status,chetz\_be\_ind,interest\_re\_calculate,reserve\_re\_increase\_calculate,freeinv\_res\_ann\_tarif  
 Modified On: 11/13/2024 10:47:00 PM (UTC+02:00)  
 Modified By: CLAL-INS\arikt  
 Validation Failure Behaviour: Error  
 Variable Type: Character  
 Valid Range From:  
 Valid Range To:  
 Choice List:  
 Value Type: External Source  
 Value: Run\_Control

### 2.1.15.1.2 *Input Variable: RunControl\_Num*

Description:  
 Help: Valuation occurs at end of valn\_month in valn\_year  
 Associated Code Variables: valn\_month,valn\_year,madad\_current,rollup\_period,qx\_sd\_random,qx\_sd\_comp,qx\_sd\_random\_res,qx\_sd\_comp\_res,chetz\_be\_ind\_yrs  
 Modified On: 6/21/2023 1:43:54 PM (UTC+03:00)  
 Modified By: CLAL-INS\ahuvaa  
 Validation Failure Behaviour: Error  
 Variable Type: Floating Point Number  
 Valid Range From:  
 Valid Range To:  
 Choice List:  
 Value Type: External Source  
 Value: Run\_Control

### 2.1.15.1.3 *Input Variable: Param\_Switch*

Description:  
 Help: policy type switches - original or current  
 Associated Code Variables: pol\_type\_lapse\_switch,pol\_type\_sal\_inc\_switch,pol\_type\_expenses\_switch,pol\_type\_phi\_incidence\_switch,pol\_type\_annuity\_tu\_switch,pol\_type\_comm\_hekef\_switch,pol\_type\_lapse\_rider\_switch,pol\_type\_recovery\_rates\_switch,use\_tat\_shnatiut\_assum,zeroise\_a

Modified On:	nn_def 6/17/2025 11:11:00 AM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Validation Failure Behaviour:	Error
Variable Type:	Character
Valid Range From:	
Valid Range To:	
Choice List:	
Value Type:	External Source
Value:	Parameters

#### **2.1.15.1.4      *Input Variable: RA\_factor***

Description:	Risk Adjustment factor
Help:	
Associated Code Variables:	ra_fact_mort_gross,ra_fact_tu_gross,ra_fact_lapse_gross,ra_fact_exp_gross,ra_fact_long_gross,ra_fact_dis_incid_gross,ra_fact_dis_termi_gross,ra_fact_exp_reins,ra_fact_lapse_reins,ra_fact_mort_reins,ra_fact_long_reins,ra_fact_tu_reins,ra_fact_dis_incid_reins,ra_fact_dis_termi_reins
Modified On:	3/17/2024 9:46:07 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	-10000
Valid Range To:	10000
Choice List:	
Value Type:	External Source
Value:	RA_Factor

#### **2.1.15.1.5      *Input Variable: Serv\_Units\_Dur***

Description:	
Help:	
Associated Code Variables:	serv_units_dur
Modified On:	6/11/2023 10:39:08 AM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	
Valid Range To:	
Choice List:	
Value Type:	External Source
Value:	Serv_Units_Dur

#### **2.1.15.1.6      *Input Variable: dump\_vars***

Description:	Output all variables to logfile?
Help:	If Y, the program will output all variables to the log stream after the startup has been executed in each



Associated Code Variables:	model. NB Should only be set to Y for testing purposes, as will create HUGE amounts of output in the log file when run with a large policy file.
Modified On:	dump_vars
Modified By:	8/27/2019 4:00:59 PM (UTC+03:00)
Validation Failure Behaviour:	CLAL-INS\NinaB
Variable Type:	Error
Valid Range From:	Character
Valid Range To:	0
Choice List:	0
Value Type:	Y,N
Value:	External Source
	Run_Control

### 2.1.15.1.7 *Input Variable: dump\_vars (2)*

Description:	Output all variables to logfile?
Help:	If Y, the program will output all variables to the log stream after the startup has been executed in each model. NB Should only be set to Y for testing purposes, as will create HUGE amounts of output in the log file when run with a large policy file.
Associated Code Variables:	dump_vars
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Character
Valid Range From:	0
Valid Range To:	0
Choice List:	Y,N
Value Type:	External Source
Value:	Run_Control

## 2.1.16 Data Page: Ann\_Data

### 2.1.16.1 Assumption Set: Ann\_Data

Description:	
Help:	
Top Model Object:	life
Modified On:	7/22/2021 5:22:38 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Data Page Model Object:	annuity
Assumption Set Links:	Unlinked
External Source:	retirement_ages
Model Point Extraction:	None
Model Points:	
Apply Model Point Weight	No
Weighting File	
Weighted Data Field	

**2.1.16.1.1      *Input Variable: takeup\_age***

Description:	Take-up age for annuities
Help:	
Associated Code Variables:	takeup_age
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Integer Number
Valid Range From:	
Valid Range To:	
Choice List:	
Value Type:	Data
Value:	takeup_age

**2.1.17      Data Page: life\_Data****2.1.17.1      Assumption Set: All Solvency**

Description:	
Help:	
Top Model Object:	life
Modified On:	2/13/2025 3:59:10 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Data Page Model Object:	life
Assumption Set Links:	Unlinked
External Source:	Policy data - Solvency
Model Point Extraction:	Range or specified
Model Points:	1,2
Apply Model Point Weight	No
Weighting File	
Weighted Data Field	

**2.1.17.1.1      *Input Variable: prem\_disc\_dcr3\_m***

Description:	Premium descreasing discount3 months
Help:	Month from policy start when Premium discount period ends.
Associated Code Variables:	prem_disc_dcr3_m
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Integer Number
Valid Range From:	
Valid Range To:	
Choice List:	
Value Type:	Data
Value:	lod_re_3_p

**2.1.17.1.2      Input Variable: *prem\_disc\_dcr4\_m***

Description:	Premium decreasing discount 4 months
Help:	Month from policy start when Premium discount period ends.
Associated Code Variables:	prem_disc_dcr4_m
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Integer Number
Valid Range From:	
Valid Range To:	
Choice List:	
Value Type:	Data
Value:	lod_re_4_p

**2.1.17.1.3      Input Variable: *prem\_disc\_step***

Description:	Premium discount decrease type
Help:	0 = no loading 1 = gorem on yesodi 3 = temp discount on all covers in policy 4 = permanent discount on all covers in policy 5 = temp discount on specific cover 6 = permanent discount on specific cover other = gorem
Associated Code Variables:	prem_disc_step
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Integer Number
Valid Range From:	
Valid Range To:	
Choice List:	
Value Type:	Data
Value:	lod_red_ty

**2.1.17.1.4      Input Variable: *prem\_disc\_dcr1\_r***

Description:	Premium decreasing discount 1 rate
Help:	Premium discount as a percentage of premium . Applied during a defined period (see Prem_disc_month).
Associated Code Variables:	prem_disc_dcr1_r
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	

Valid Range To:

Choice List:

Value Type:

Data

Value:

lod\_re\_1\_a

#### **2.1.17.1.5      *Input Variable: prem\_disc\_dcr1\_m***

Description:

Premium decreasing discount1 months

Help:

Month from policy start when Premium discount period ends.

Associated Code Variables:

prem\_disc\_dcr1\_m

Modified On:

8/27/2019 4:00:59 PM (UTC+03:00)

Modified By:

CLAL-INS\NinaB

Validation Failure Behaviour:

Error

Variable Type:

Integer Number

Valid Range From:

Valid Range To:

Choice List:

Value Type:

Data

Value:

lod\_re\_1\_p

#### **2.1.17.1.6      *Input Variable: prem\_disc\_dcr4\_r***

Description:

Premium decreasing discount4 rate

Help:

Premium discount as a percentage of premium .  
Applied during a defined period (see Prem\_disc\_month).

Associated Code Variables:

prem\_disc\_dcr4\_r

Modified On:

8/27/2019 4:00:59 PM (UTC+03:00)

Modified By:

CLAL-INS\NinaB

Validation Failure Behaviour:

Error

Variable Type:

Floating Point Number

Valid Range From:

Valid Range To:

Choice List:

Value Type:

Data

Value:

lod\_re\_4\_a

#### **2.1.17.1.7      *Input Variable: prem\_disc\_dcr3\_r***

Description:

Premium decreasing discount3 rate

Help:

Premium discount as a percentage of premium .  
Applied during a defined period (see Prem\_disc\_month).

Associated Code Variables:

prem\_disc\_dcr3\_r

Modified On:

8/27/2019 4:00:59 PM (UTC+03:00)

Modified By:

CLAL-INS\NinaB

Validation Failure Behaviour:

Error

Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:  
  
Choice List:  
Value Type: Data  
Value: lod\_re\_3\_a

#### **2.1.17.1.8      *Input Variable: prem\_disc\_dcr2\_r***

Description: Premium decreasing discount2 rate  
Help: Premium discount as a percentage of premium .  
Applied during a defined period (see  
Prem\_disc\_month).  
  
Associated Code Variables: prem\_disc\_dcr2\_r  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:  
  
Choice List:  
Value Type: Data  
Value: lod\_re\_2\_a

#### **2.1.17.1.9      *Input Variable: prem\_disc\_dcr5\_m***

Description: Premium decreasing discount5 months  
Help: Month from policy start when Premium discount  
period ends.  
  
Associated Code Variables: prem\_disc\_dcr5\_m  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Integer Number  
Valid Range From:  
Valid Range To:  
  
Choice List:  
Value Type: Data  
Value: lod\_re\_5\_p

#### **2.1.17.1.10      *Input Variable: prem\_disc\_dcr5\_r***

Description: Premium decreasing discount5 rate  
Help: Premium discount as a percentage of premium .  
Applied during a defined period (see  
Prem\_disc\_month).  
  
Associated Code Variables: prem\_disc\_dcr5\_r  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)

Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:

Choice List:  
Value Type: Data  
Value: lod\_re\_5\_a

#### **2.1.17.1.11      *Input Variable: prem\_disc\_dcr2\_m***

Description: Premium decreasing discount 2 months  
Help: Month from policy start when Premium discount period ends.  
Associated Code Variables: prem\_disc\_dcr2\_m  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Integer Number  
Valid Range From:  
Valid Range To:

Choice List:  
Value Type: Data  
Value: lod\_re\_2\_p

#### **2.1.17.1.12      *Input Variable: prem\_disc\_type\_2***

Description: Premium discount type  
Help: 0 = no loading  
1 = gorem on yesodi  
3 = temp discount on all covers in policy  
4 = permanent discount on all covers in policy  
5 = temp discount on specific cover  
6 = permanent discount on specific cover  
other = gorem  
Associated Code Variables: prem\_disc\_type\_2  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Integer Number  
Valid Range From:  
Valid Range To:

Choice List:  
Value Type: Data  
Value: lod\_type\_2

**2.1.17.1.13 Input Variable: prem\_disc\_type**

Description:	Premium discount type
Help:	0 = no loading 1 = gorem on yesodi 3 = temp discount on all covers in policy 4 = permanent discount on all covers in policy 5 = temp discount on specific cover 6 = permanent discount on specific cover other = gorem
Associated Code Variables:	prem_disc_type
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Integer Number
Valid Range From:	
Valid Range To:	
Choice List:	
Value Type:	Data
Value:	lod_type_1

**2.1.17.1.14 Input Variable: prem\_disc\_month\_2**

Description:	Premium discount period (last month)
Help:	
Associated Code Variables:	prem_disc_month_2_input
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	
Valid Range To:	
Choice List:	
Value Type:	Data
Value:	lod_pe_r_2

**2.1.17.1.15 Input Variable: prem\_disc\_month**

Description:	Premium discount period (last month)
Help:	
Associated Code Variables:	prem_disc_month_input
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	
Valid Range To:	
Choice List:	

Value Type: Data  
Value: lod\_pe\_r\_1

#### **2.1.17.1.16      *Input Variable: prem\_disc\_perc\_2***

Description: Premium discount as a percentage  
Help:  
Associated Code Variables: prem\_disc\_perc\_2\_input  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:

Choice List:  
Value Type: Data  
Value: lod\_amt\_2

#### **2.1.17.1.17      *Input Variable: prem\_disc\_perc***

Description: Premium discount as a percentage  
Help:  
Associated Code Variables: prem\_disc\_perc\_input  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:

Choice List:  
Value Type: Data  
Value: lod\_amt\_1

#### **2.1.17.1.18      *Input Variable: amala\_nihul\_6***

Description: commission field from data file  
Help:  
Associated Code Variables: amala\_nihul\_6  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:

Choice List:  
Value Type: Data  
Value: amla\_ni\_6



**2.1.17.1.19      Input Variable: amala\_pikuach\_1**

Description: supervisor commission field from data file  
Help:  
Associated Code Variables: amala\_pikuach\_1  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:  
  
Choice List:  
Value Type: Data  
Value: amla\_pk\_1

**2.1.17.1.20      Input Variable: amala\_pikuach\_0**

Description: supervisor commission field from data file  
Help:  
Associated Code Variables: amala\_pikuach\_0  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:  
  
Choice List:  
Value Type: Data  
Value: amla\_pk\_0

**2.1.17.1.21      Input Variable: amala\_nihul\_0**

Description: commission field from data file  
Help:  
Associated Code Variables: amala\_nihul\_0  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:  
  
Choice List:  
Value Type: Data  
Value: amla\_ni\_0

**2.1.17.1.22      Input Variable: amala\_nihul\_1**

Description: commission field from data file

Help:  
Associated Code Variables: amala\_nihul\_1  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:

Choice List:  
Value Type: Data  
Value: amla\_ni\_1

#### **2.1.17.1.23      *Input Variable: amala\_nihul\_2***

Description: commission field from data file  
Help:  
Associated Code Variables: amala\_nihul\_2  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:

Choice List:  
Value Type: Data  
Value: amla\_ni\_2

#### **2.1.17.1.24      *Input Variable: amala\_nihul\_3***

Description: commission field from data file  
Help:  
Associated Code Variables: amala\_nihul\_3  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:

Choice List:  
Value Type: Data  
Value: amla\_ni\_3

#### **2.1.17.1.25      *Input Variable: amala\_nihul\_4***

Description: commission field from data file  
Help:  
Associated Code Variables: amala\_nihul\_4

Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:

Choice List:  
Value Type: Data  
Value: amla\_ni\_4

#### **2.1.17.1.26      *Input Variable: amala\_nihul\_5***

Description: commission field from data file  
Help:  
Associated Code Variables: amala\_nihul\_5  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:

Choice List:  
Value Type: Data  
Value: amla\_ni\_5

#### **2.1.17.1.27      *Input Variable: amala\_nihul\_7***

Description: commission field from data file  
Help:  
Associated Code Variables: amala\_nihul\_7  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:

Choice List:  
Value Type: Data  
Value: amla\_ni\_7

#### **2.1.17.1.28      *Input Variable: amala\_nihul\_8***

Description: commission field from data file  
Help:  
Associated Code Variables: amala\_nihul\_8  
Modified On: 4/9/2024 5:41:00 PM (UTC+03:00)  
Modified By: CLAL-INS\Arikt

Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:

Choice List:  
Value Type: Data  
Value: AMLA\_NI\_8

#### **2.1.17.1.29      *Input Variable: amala\_nihul\_9***

Description: commission field from data file  
Help:  
Associated Code Variables: amala\_nihul\_9  
Modified On: 4/9/2024 5:41:13 PM (UTC+03:00)  
Modified By: CLAL-INS\Arikt  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:

Choice List:  
Value Type: Data  
Value: AMLA\_NI\_9

#### **2.1.17.1.30      *Input Variable: amala\_nihul\_10***

Description: commission field from data file  
Help:  
Associated Code Variables: amala\_nihul\_10  
Modified On: 4/9/2024 5:41:19 PM (UTC+03:00)  
Modified By: CLAL-INS\Arikt  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:

Choice List:  
Value Type: Data  
Value: AMLA\_NI\_10

#### **2.1.17.1.31      *Input Variable: amala\_nihul\_11***

Description: commission field from data file  
Help:  
Associated Code Variables: amala\_nihul\_11  
Modified On: 4/9/2024 5:41:25 PM (UTC+03:00)  
Modified By: CLAL-INS\Arikt  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number

Valid Range From:

Valid Range To:

Choice List:

Value Type:

Data

Value:

AMLA\_NI\_11

#### **2.1.17.1.32      *Input Variable: amala\_nihul\_12***

Description:

commission field from data file

Help:

Associated Code Variables:

amala\_nihul\_12

Modified On:

4/9/2024 5:41:39 PM (UTC+03:00)

Modified By:

CLAL-INS\Arikt

Validation Failure Behaviour:

Error

Variable Type:

Floating Point Number

Valid Range From:

Valid Range To:

Choice List:

Value Type:

Data

Value:

AMLA\_NI\_12

#### **2.1.17.1.33      *Input Variable: amala\_nihul\_13***

Description:

commission field from data file

Help:

Associated Code Variables:

amala\_nihul\_13

Modified On:

4/9/2024 5:41:46 PM (UTC+03:00)

Modified By:

CLAL-INS\Arikt

Validation Failure Behaviour:

Error

Variable Type:

Floating Point Number

Valid Range From:

Valid Range To:

Choice List:

Value Type:

Data

Value:

AMLA\_NI\_13

#### **2.1.17.1.34      *Input Variable: amala\_nihul\_14***

Description:

commission field from data file

Help:

Associated Code Variables:

amala\_nihul\_14

Modified On:

4/9/2024 5:41:52 PM (UTC+03:00)

Modified By:

CLAL-INS\Arikt

Validation Failure Behaviour:

Error

Variable Type:

Floating Point Number

Valid Range From:

Valid Range To:

Choice List:  
Value Type: Data  
Value: AMLA\_NI\_14

#### **2.1.17.1.35      *Input Variable: amala\_nihul\_15***

Description: commission field from data file  
Help:  
Associated Code Variables: amala\_nihul\_15  
Modified On: 4/9/2024 5:41:58 PM (UTC+03:00)  
Modified By: CLAL-INS\Arikt  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:

Choice List:  
Value Type: Data  
Value: AMLA\_NI\_15

#### **2.1.17.1.36      *Input Variable: amala\_nihul\_16***

Description: commission field from data file  
Help:  
Associated Code Variables: amala\_nihul\_16  
Modified On: 4/9/2024 5:42:05 PM (UTC+03:00)  
Modified By: CLAL-INS\Arikt  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:

Choice List:  
Value Type: Data  
Value: AMLA\_NI\_16

#### **2.1.17.1.37      *Input Variable: amala\_15***

Description: commission field from data file  
Help:  
Associated Code Variables: amala\_15  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:

Choice List:

Value Type: Data  
Value: amla\_sh\_15

#### **2.1.17.1.38      *Input Variable: amala\_9***

Description: commission field from data file  
Help:  
Associated Code Variables: amala\_9  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:

Choice List:  
Value Type: Data  
Value: amla\_sh\_9

#### **2.1.17.1.39      *Input Variable: amala\_8***

Description: commission field from data file  
Help:  
Associated Code Variables: amala\_8  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:

Choice List:  
Value Type: Data  
Value: amla\_sh\_8

#### **2.1.17.1.40      *Input Variable: amala\_7***

Description: commission field from data file  
Help:  
Associated Code Variables: amala\_7  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:

Choice List:  
Value Type: Data  
Value: amla\_sh\_7

**2.1.17.1.41      Input Variable: amala\_6**

Description: commission field from data file  
Help:  
Associated Code Variables: amala\_6  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:  
  
Choice List:  
Value Type: Data  
Value: amla\_sh\_6

**2.1.17.1.42      Input Variable: amala\_5**

Description: commission field from data file  
Help:  
Associated Code Variables: amala\_5  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:  
  
Choice List:  
Value Type: Data  
Value: amla\_sh\_5

**2.1.17.1.43      Input Variable: amala\_4**

Description: commission field from data file  
Help:  
Associated Code Variables: amala\_4  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:  
  
Choice List:  
Value Type: Data  
Value: amla\_sh\_4

**2.1.17.1.44      Input Variable: amala\_3**

Description: commission field from data file



Help:  
Associated Code Variables: amala\_3  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:

Choice List:  
Value Type: Data  
Value: amla\_sh\_3

#### **2.1.17.1.45      *Input Variable: amala\_2***

Description: commission field from data file  
Help:  
Associated Code Variables: amala\_2  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:

Choice List:  
Value Type: Data  
Value: amla\_sh\_2

#### **2.1.17.1.46      *Input Variable: amala\_16***

Description: commission field from data file  
Help:  
Associated Code Variables: amala\_16  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:

Choice List:  
Value Type: Data  
Value: amla\_sh\_16

#### **2.1.17.1.47      *Input Variable: amala\_1***

Description: commission field from data file  
Help:  
Associated Code Variables: amala\_1

Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:

Choice List:  
Value Type: Data  
Value: amla\_sh\_1

#### **2.1.17.1.48      *Input Variable: amala\_10***

Description: commission field from data file  
Help:  
Associated Code Variables: amala\_10  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:

Choice List:  
Value Type: Data  
Value: amla\_sh\_10

#### **2.1.17.1.49      *Input Variable: amala\_11***

Description: commission field from data file  
Help:  
Associated Code Variables: amala\_11  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:

Choice List:  
Value Type: Data  
Value: amla\_sh\_11

#### **2.1.17.1.50      *Input Variable: amala\_12***

Description: commission field from data file  
Help:  
Associated Code Variables: amala\_12  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB

Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:

Choice List:  
Value Type: Data  
Value: amla\_sh\_12

#### **2.1.17.1.51      *Input Variable: amala\_13***

Description: commission field from data file  
Help:  
Associated Code Variables: amala\_13  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:

Choice List:  
Value Type: Data  
Value: amla\_sh\_13

#### **2.1.17.1.52      *Input Variable: comm\_perc\_res\_b***

Description: Commissions as % of reserves (Pure savin  
Help:  
Associated Code Variables: comm\_perc\_res\_b\_input  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:

Choice List:  
Value Type: Data  
Value: a\_zvra\_sav

#### **2.1.17.1.53      *Input Variable: comm\_perc\_res\_a***

Description: Commissions as % of reserves (basic)  
Help:  
Associated Code Variables: comm\_perc\_res\_a\_input  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number

Valid Range From:

Valid Range To:

Choice List:

Value Type:

Data

Value:

amla\_zvira

#### **2.1.17.1.54      *Input Variable: comm\_ren\_perc\_sav***

Description:

Renewal commission (%) for pure saving

Help:

Renewal commission expressed as a % of pure saving premium income. Used when benef class is adif

Associated Code Variables:

comm\_ren\_perc\_sav

Modified On:

8/27/2019 4:00:59 PM (UTC+03:00)

Modified By:

CLAL-INS\NinaB

Validation Failure Behaviour:

Error

Variable Type:

Floating Point Number

Valid Range From:

Valid Range To:

Choice List:

Value Type:

Data

Value:

d\_gvia\_sav

#### **2.1.17.1.55      *Input Variable: comm\_ren\_perc\_prem***

Description:

Renewal commission (%)

Help:

Renewal commission expressed as a % of premium income.

Associated Code Variables:

comm\_ren\_perc\_prem

Modified On:

8/27/2019 4:00:59 PM (UTC+03:00)

Modified By:

CLAL-INS\NinaB

Validation Failure Behaviour:

Error

Variable Type:

Floating Point Number

Valid Range From:

Valid Range To:

Choice List:

Value Type:

Data

Value:

d\_gvia\_prc

#### **2.1.17.1.56      *Input Variable: comm\_prof***

Description:

Renewal commission (%)

Help:

Renewal commission expressed as a % of premium income.

Associated Code Variables:

comm\_prof

Modified On:

1/11/2023 9:27:26 AM (UTC+02:00)

Modified By:

CLAL-INS\joshm

Validation Failure Behaviour:

Error

Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:

Choice List:  
Value Type: Data  
Value: comm\_prof

#### **2.1.17.1.57      *Input Variable: amala\_14***

Description: commission field from data file  
Help:  
Associated Code Variables: amala\_14  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:

Choice List:  
Value Type: Data  
Value: amla\_sh\_14

#### **2.1.17.1.58      *Input Variable: comm\_renewal\_year***

Description: First year from when renewal commission is paid  
Help:  
Associated Code Variables: comm\_renewal\_year\_input  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:

Choice List:  
Value Type: Data  
Value: d\_gvia\_msh

#### **2.1.17.1.59      *Input Variable: amala\_0***

Description: commission field from data file  
Help:  
Associated Code Variables: amala\_0  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:

Valid Range To:

Choice List:

Value Type:

Data

Value:

amla\_sh\_0

#### **2.1.17.1.60      *Input Variable: prod\_code\_base***

Description:

Product code of the base (Yessodi) cover

Help:

read in from inforce file

Associated Code Variables:

prod\_code\_base

Modified On:

8/27/2019 4:00:59 PM (UTC+03:00)

Modified By:

CLAL-INS\NinaB

Validation Failure Behaviour:

Error

Variable Type:

Character

Valid Range From:

Valid Range To:

Choice List:

Value Type:

Data

Value:

procdcd\_yes

#### **2.1.17.1.61      *Input Variable: tarif***

Description:

tarif

Help:

Associated Code Variables:

tarif

Modified On:

8/27/2019 4:00:59 PM (UTC+03:00)

Modified By:

CLAL-INS\NinaB

Validation Failure Behaviour:

Error

Variable Type:

Integer Number

Valid Range From:

Valid Range To:

Choice List:

Value Type:

Data

Value:

tarif

#### **2.1.17.1.62      *Input Variable: sub\_model***

Description:

sub model to run

Help:

Associated Code Variables:

submodel

Modified On:

8/27/2019 4:00:59 PM (UTC+03:00)

Modified By:

CLAL-INS\NinaB

Validation Failure Behaviour:

Error

Variable Type:

Character

Valid Range From:

Valid Range To:

Choice List:  
Value Type: Data  
Value: submodel

#### **2.1.17.1.63      *Input Variable: risk\_code***

Description: product code of risk rider with Meitav (Managers)  
Help: This is the product code of the risk rider (Sapir) that shares the total premium with the policy (Meitav Managers) being run.  
  
Associated Code Variables: risk\_code  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Character  
Valid Range From:  
Valid Range To:

Choice List:  
Value Type: Data  
Value: risk\_code

#### **2.1.17.1.64      *Input Variable: prog\_name***

Description: prog\_name field from data file  
Help: Used for classifying reserves between health and life  
Associated Code Variables: prog\_name  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Character  
Valid Range From:  
Valid Range To:

Choice List:  
Value Type: Data  
Value: prog\_name

#### **2.1.17.1.65      *Input Variable: prod\_code***

Description: Product code  
Help: read in from inforce file  
Associated Code Variables: prod\_code  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Character  
Valid Range From:  
Valid Range To:

Choice List:

Value Type: Data  
Value: prod\_code

#### **2.1.17.1.66      *Input Variable: benefit\_term***

Description: Policy benefit term (months)  
Help:  
Associated Code Variables: benefit\_term\_input  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:

Choice List:  
Value Type: Data  
Value: term\_ben

#### **2.1.17.1.67      *Input Variable: ben\_class***

Description: benefit class  
Help:  
Associated Code Variables: ben\_class\_input  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Character  
Valid Range From:  
Valid Range To:

Choice List:  
Value Type: Data  
Value: benclass

#### **2.1.17.1.68      *Input Variable: prod\_group\_yessodi\_portfolio***

Description: benefit class  
Help:  
Associated Code Variables: prod\_group\_yessodi\_portfolio  
Modified On: 6/8/2023 2:17:56 PM (UTC+03:00)  
Modified By: CLAL-INS\ahuvaa  
Validation Failure Behaviour: Error  
Variable Type: Character  
Valid Range From:  
Valid Range To:

Choice List:  
Value Type: Data  
Value: prod\_group\_yessodi\_portfolio



**2.1.17.1.69      *Input Variable: channel***

Description: channel  
Help:  
Associated Code Variables: channel  
Modified On: 1/9/2022 12:31:40 PM (UTC+02:00)  
Modified By: CLAL-INS\joshm  
Validation Failure Behaviour: Error  
Variable Type: Character  
Valid Range From:  
Valid Range To:  
  
Choice List:  
Value Type: Data  
Value: channel

**2.1.17.1.70      *Input Variable: rein\_set***

Description: Reinsurance assumptions  
Help:  
Associated Code Variables: rein\_set\_input  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Character  
Valid Range From:  
Valid Range To:  
  
Choice List:  
Value Type: Data  
Value: sug\_sikun

**2.1.17.1.71      *Input Variable: retention\_perc***

Description: Retention Ratio  
Help: Proportion reinsured for surplus reinsurance :  
calculated in startup  
Associated Code Variables: retention\_perc  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:  
  
Choice List:  
Value Type: Data  
Value: retention

**2.1.17.1.72      *Input Variable: mgtfee\_format***

Description: Management fee format number  
Help:  
Associated Code Variables: mgtfee\_format  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Integer Number  
Valid Range From:  
Valid Range To:  
  
Choice List:  
Value Type: Data  
Value: format\_dn

**2.1.17.1.73      *Input Variable: mgt\_fee\_fixed***

Description: Fixed Management fee %  
Help:  
Associated Code Variables: mgt\_fee\_fixed\_input  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:  
  
Choice List:  
Value Type: Data  
Value: d\_nihul\_h

**2.1.17.1.74      *Input Variable: mgt\_fee\_variable***

Description: variable management fee proportion (%)  
Help:  
Associated Code Variables: mgt\_fee\_variable\_input  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:  
  
Choice List:  
Value Type: Data  
Value: d\_nihul\_z

**2.1.17.1.75      *Input Variable: recordno***

Description: record number

Help:	index row num
Associated Code Variables:	index_row_num
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Character
Valid Range From:	
Valid Range To:	
Choice List:	
Value Type:	Data
Value:	recordno

#### **2.1.17.1.76      Input Variable: ind\_nb**

Description:	NB indicator
Help:	1 = NB F = Female
Associated Code Variables:	ind_nb
Modified On:	4/11/2024 3:34:33 PM (UTC+03:00)
Modified By:	CLAL-INS\Arikt
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	
Valid Range To:	
Choice List:	
Value Type:	Data
Value:	IND_NB

#### **2.1.17.1.77      Input Variable: ind\_ifrs**

Description:	IFRS indicator
Help:	1 = NB F = Female
Associated Code Variables:	ind_ifrs
Modified On:	6/25/2024 2:07:55 PM (UTC+03:00)
Modified By:	CLAL-INS\Arikt
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	
Valid Range To:	
Choice List:	
Value Type:	Data
Value:	IND_IFRS

#### **2.1.17.1.78      Input Variable: agent\_no**

Description:	agent number and company letter (used as unique index)
--------------	--

Help:	Agent number and first letter of company (a/c) to get unique agent index
Associated Code Variables:	agent_no
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Character
Valid Range From:	
Valid Range To:	
Choice List:	
Value Type:	Data
Value:	aggent

### **2.1.17.1.79      *Input Variable: insured\_id***

Description:	ID no for main insured
Help:	ID no for main insured
Associated Code Variables:	insured_id
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Integer Number
Valid Range From:	
Valid Range To:	
Choice List:	
Value Type:	Data
Value:	id

### **2.1.17.1.80      *Input Variable: movement\_status***

Description:	Movement Status (IF, NB, PU, NC...)
Help:	Code is used to identify special covers/policies. Not used in the model but passed to the output file for summing the results. 1= LTC NB05 old tarif 2= PHI mifali NB05 alone - added to old policy 3= PHI mifali NB05 alone - other new policy 4= PHI mifali NB05 alone - w/o other new policy
Associated Code Variables:	movement_status
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Character
Valid Range From:	
Valid Range To:	
Choice List:	
Value Type:	Data
Value:	mvt_stat

**2.1.17.1.81 Input Variable: movement\_flag**

Description:	Type of Movement (death, surrender, no change...)
Help:	Code is used to identify special covers/policies. Not used in the model but passed to the output file for summing the results. 1= LTC NB05 old tarif 2= PHI mifali NB05 alone - added to old policy 3= PHI mifali NB05 alone - other new policy 4= PHI mifali NB05 alone - w/o other new policy
Associated Code Variables:	movement_flag
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Integer Number
Valid Range From:	
Valid Range To:	
Choice List:	
Value Type:	Data
Value:	mvt_flag

**2.1.17.1.82 Input Variable: movement\_month**

Description:	Calender month of movement
Help:	Code is used to identify special covers/policies. Not used in the model but passed to the output file for summing the results. 1= LTC NB05 old tarif 2= PHI mifali NB05 alone - added to old policy 3= PHI mifali NB05 alone - other new policy 4= PHI mifali NB05 alone - w/o other new policy
Associated Code Variables:	movement_month
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Integer Number
Valid Range From:	
Valid Range To:	
Choice List:	
Value Type:	Data
Value:	mvt_month

**2.1.17.1.83 Input Variable: occ\_perc**

Description:	Extra loadings (occupation only) on premium/qx
Help:	Extra premium loading (percent of basic premium) for the policy for occupation. Will only be added to the premium if variable health_occ_in_prem is set to N. Regardless this should also be used for the claim assumption.

Associated Code Variables:	occ_perc
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	
Valid Range To:	
Choice List:	
Value Type:	Data
Value:	occ_add_p

#### **2.1.17.1.84      *Input Variable: health\_perc***

Description:	Extra loadings (health only) on premium/qx
Help:	Extra premium loading (percent of basic premium) for the policy for health conditions. Will only be added to the premium if variable health_occ_in_prem is set to N. Regardless this should also be used for the claim assumption.
Associated Code Variables:	health_perc
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	
Valid Range To:	
Choice List:	
Value Type:	Data
Value:	morta_ad_p

#### **2.1.17.1.85      *Input Variable: pol\_index***

Description:	Policy index (Pol No, Company, Tarif, Tafkid)
Help:	Policy Index to uniquely identify cover. Made from Policy number, Company code, Tarif and Tafkid
Associated Code Variables:	pol_index
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Character
Valid Range From:	
Valid Range To:	
Choice List:	
Value Type:	Data
Value:	pindex

**2.1.17.1.86      Input Variable: *maasik\_no***

Description:	Maasik number for managers policies
Help:	Agency Number (Osek merushe number)
Associated Code Variables:	maasik_no
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Character
Valid Range From:	
Valid Range To:	
Choice List:	
Value Type:	Data
Value:	maasik

**2.1.17.1.87      Input Variable: *flag\_code***

Description:	flag_code for data record
Help:	Code is used to identify special covers/policies. Not used in the model but passed to the output file for summing the results. 1= LTC NB05 old tarif 2= PHI mifali NB05 alone - added to old policy 3= PHI mifali NB05 alone - other new policy 4= PHI mifali NB05 alone - w/o other new policy
Associated Code Variables:	flag_code
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Integer Number
Valid Range From:	
Valid Range To:	
Choice List:	
Value Type:	Data
Value:	flag_code

**2.1.17.1.88      Input Variable: *agency\_no***

Description:	Osek Merushe number
Help:	Agency Number (Osek merushe number)
Associated Code Variables:	agency_no
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Integer Number
Valid Range From:	
Valid Range To:	
Choice List:	

Value Type: Data  
Value: osekno

### **2.1.17.1.89      *Input Variable: year\_start***

Description: Year of policy start (origi-date)  
Help: For reporting purposes only  
Associated Code Variables: year\_start  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Integer Number  
Valid Range From:  
Valid Range To:

Choice List:  
Value Type: Data  
Value: origi\_date

### **2.1.17.1.90      *Input Variable: year\_prod***

Description: Year of policy production (prod-date)  
Help: For reporting purposes only  
Associated Code Variables: year\_prod  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Integer Number  
Valid Range From:  
Valid Range To:

Choice List:  
Value Type: Data  
Value: pro\_date

### **2.1.17.1.91      *Input Variable: surr\_value\_if***

Description: Surrender value at valn date from IF file  
Help: units at valuation date (accumulation/reserve) per 1 benefit  
Associated Code Variables: surr\_value\_if\_input  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:

Choice List:  
Value Type: Data



Value: sv

### **2.1.17.1.92      *Input Variable: unit\_value\_savings***

Description: Savings unit balance at valn date  
Help: units at valuation date (extra savings account) per 1 benefit  
Associated Code Variables: unit\_value\_savings\_input  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:  
  
Choice List:  
Value Type: Data  
Value: gross\_res

### **2.1.17.1.93      *Input Variable: fund\_yesodi***

Description: Fund (Keren) for main policy  
Help: 10 = yod,..., 1 = aleph  
Associated Code Variables: fund\_yesodi  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Character  
Valid Range From: 0  
Valid Range To: 200  
  
Choice List:  
Value Type: Data  
Value: keren\_yes

### **2.1.17.1.94      *Input Variable: smoker\_stat***

Description: Smoker status  
Help: Smoker status under which the policy has been issued:  
N = Non smoker, S = Smoker or A = Aggregate  
Associated Code Variables: smoker\_stat  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Character  
Valid Range From:  
Valid Range To:  
  
Choice List:  
Value Type: Data

Value: smoke\_stat

### **2.1.17.1.95      *Input Variable: sex***

Description: Sex  
Help: M = Male  
F = Female  
Associated Code Variables: sex  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Character  
Valid Range From:  
Valid Range To:  
  
Choice List:  
Value Type: Data  
Value: sex

### **2.1.17.1.96      *Input Variable: saving\_perc***

Description: Total percentage of savings (basic + extra)  
Help: For Adif: Total percentage of savings (basic + pure savings) .  
For Profil: Percentage of total premium allocated to pure savings (rest goes to normal product).  
Associated Code Variables: saving\_perc  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:  
  
Choice List:  
Value Type: Data  
Value: sav\_perc

### **2.1.17.1.97      *Input Variable: occ\_key***

Description: Occupational key  
Help:  
Associated Code Variables: occ\_key  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Character  
Valid Range From:  
Valid Range To:  
  
Choice List: 1,2,3

Value Type: Data  
Value: sug\_isuk

### **2.1.17.1.98      *Input Variable: risk\_si***

Description: sum insured of risk rider with Meitav (Managers)  
Help: This is the sum insured of the risk rider (Sapir) that shares the total premium with the policy (Meitav Managers) being run.  
  
Associated Code Variables: risk\_si  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:

Choice List:  
Value Type: Data  
Value: risk\_si

### **2.1.17.1.99      *Input Variable: policy\_type***

Description: policy type  
Help:  
Associated Code Variables: policy\_type  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Character  
Valid Range From:  
Valid Range To:

Choice List:  
Value Type: Data  
Value: pol\_type

### **2.1.17.1.100      *Input Variable: groups\_sol***

Description: groups from solvency  
Help:  
Associated Code Variables: groups\_sol  
Modified On: 3/22/2023 2:02:28 PM (UTC+02:00)  
Modified By: CLAL-INS\ahuvaa  
Validation Failure Behaviour: Error  
Variable Type: Character  
Valid Range From:  
Valid Range To:

Choice List:  
Value Type: Data

Value: groups\_sol

#### **2.1.17.1.101 Input Variable: policy\_type\_orig**

Description: original policy type  
Help:  
Associated Code Variables: policy\_type\_orig  
Modified On: 12/26/2022 1:46:27 PM (UTC+02:00)  
Modified By: CLAL-INS\ahuvaa  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:

Choice List:  
Value Type: Data  
Value: sug\_polisa\_makor

#### **2.1.17.1.102 Input Variable: policies\_curr**

Description: Number of policies inforce at valn date  
Help: Current number of in force policies at the valuation date.  
Associated Code Variables: policies\_curr  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:

Choice List:  
Value Type: Data  
Value: policies

#### **2.1.17.1.103 Input Variable: riders\_count\_w**

Description: Number of riders for current Profil policy  
Help:  
Associated Code Variables: riders\_count\_w\_input  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:

Choice List:  
Value Type: Data  
Value: riders\_num

**2.1.17.1.104 Input Variable: pol\_number**

Description:	Policy number
Help:	Policy number.
Associated Code Variables:	pol_number
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Character
Valid Range From:	
Valid Range To:	
Choice List:	
Value Type:	Data
Value:	pol_num

**2.1.17.1.105 Input Variable: insured**

Description:	insured
Help:	for health covers 1 = main 0 = child 2 = partner (used for looking up correct premium rate for non-family tariffs)
Associated Code Variables:	insured
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Integer Number
Valid Range From:	
Valid Range To:	
Choice List:	
Value Type:	Data
Value:	insure

**2.1.17.1.106 Input Variable: fund**

Description:	Fund (Keren)
Help:	10 = yod,..., 1 = aleph
Associated Code Variables:	fund
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Validation Failure Behaviour:	Error
Variable Type:	Character
Valid Range From:	
Valid Range To:	
Choice List:	
Value Type:	Data

Value: keren

### **2.1.17.1.107 Input Variable: error\_code**

Description: error\_code for data record  
Help: 1 and 3 are OK  
any other number will cause the record to be skipped  
Associated Code Variables: error\_code  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Integer Number  
Valid Range From:  
Valid Range To:  
  
Choice List:  
Value Type: Data  
Value: ind\_rec\_so

### **2.1.17.1.108 Input Variable: elapsed\_months\_extra**

Description: months between tarif & origi date  
Help: The number of months, rounded up, from policy inception to the valuation date.  
Associated Code Variables: elapsed\_months\_extra  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Integer Number  
Valid Range From:  
Valid Range To:  
  
Choice List:  
Value Type: Data  
Value: elapmthxt

### **2.1.17.1.109 Input Variable: paid\_up**

Description: Paid up at valuation date? (Y/N)  
Help:  
Associated Code Variables: paid\_up\_input  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Character  
Valid Range From:  
Valid Range To:  
  
Choice List:  
Value Type: Data

Value: pup\_stat

### **2.1.17.1.110 Input Variable: elapsed\_months**

Description: Elapsed months at valn date  
Help: The number of months, rounded up, from policy inception to the valuation date.  
Associated Code Variables: elapsed\_months  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Integer Number  
Valid Range From:  
Valid Range To:  
  
Choice List:  
Value Type: Data  
Value: elaps\_mths

### **2.1.17.1.111 Input Variable: company**

Description: company name  
Help: Use for looking up expenses from expense table. Only used when lookup by prodcode = "Y"  
Associated Code Variables: company  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Character  
Valid Range From:  
Valid Range To:  
  
Choice List:  
Value Type: Data  
Value: compeny

### **2.1.17.1.112 Input Variable: bonus\_inforce**

Description: bonus inforce at valuation date  
Help: variable linked with the reserve field from the inforce file  
Associated Code Variables: bonus\_inforce  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:  
  
Choice List:  
Value Type: Data

Value: bonus

### **2.1.17.1.113    *Input Variable: benefits\_curr***

Description: Number of covers at valuation date  
Help: Current number of in force benefits at the valuation date.  
Associated Code Variables: benefits\_curr  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:  
  
Choice List:  
Value Type: Data  
Value: benefits

### **2.1.17.1.114    *Input Variable: sum\_ins\_curr***

Description: Sum Insured at valn date  
Help:  
Associated Code Variables: sum\_ins\_curr\_input  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:  
  
Choice List:  
Value Type: Data  
Value: sum\_as

### **2.1.17.1.115    *Input Variable: unit\_value\_accum***

Description: Accum unit balance at valn date  
Help:  
Associated Code Variables: unit\_value\_accum\_input  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:  
  
Choice List:  
Value Type: Data  
Value: reserve



**2.1.17.1.116 Input Variable: alloc\_kafuy**

Description:	alloc_kafuy field from data file
Help:	Percentage of premium allocated to units. Each element in the array applies for the number of months specified in the alloc_rate_period array. The first element in the array should be entered for period 1. Read in from alloc_rate_tbl in set_common_variables.
Associated Code Variables:	alloc_kafuy
Modified On:	2/13/2025 3:51:32 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	
Valid Range To:	
Choice List:	
Value Type:	Data
Value:	alloc_kafuy

**2.1.17.1.117 Input Variable: product\_alloc\_rate\_percent**

Description:	product alloc_kafuy field from data file (no benefit)
Help:	Percentage of premium allocated to units. Each element in the array applies for the number of months specified in the alloc_rate_period array. The first element in the array should be entered for period 1. Read in from alloc_rate_tbl in set_common_variables.
Associated Code Variables:	product_alloc_rate_percent
Modified On:	3/13/2025 9:38:53 AM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	
Valid Range To:	
Choice List:	
Value Type:	Data
Value:	PRODUCT_ALLOC_RATE_PERCENT

**2.1.17.1.118 Input Variable: allocation\_limit\_amount**

Description:	maximum DNP - monthly - shekel
Help:	maximum DNP - monthly - shekel
Associated Code Variables:	allocation_limit_amount
Modified On:	3/13/2025 9:39:37 AM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Validation Failure Behaviour:	Error
Variable Type:	Floating Point Number
Valid Range From:	

Valid Range To:

Choice List:

Value Type:

Data

Value:

ALLOCATION\_LIMIT\_AMOUNT

#### **2.1.17.1.119    Input Variable: *imp\_manual\_alloc\_rate\_term\_dt***

Description:

DNP benefit final date

Help:

DNP benefit final date

Associated Code Variables:

imp\_manual\_alloc\_rate\_term\_dt

Modified On:

3/13/2025 9:40:20 AM (UTC+02:00)

Modified By:

CLAL-INS\ahuvaa

Validation Failure Behaviour:

Error

Variable Type:

Floating Point Number

Valid Range From:

Valid Range To:

Choice List:

Value Type:

Data

Value:

IMP\_MANUAL\_ALLOC\_RATE\_TERM\_DT

#### **2.1.17.1.120    Input Variable: *aml\_a\_hishtatfut\_dnp***

Description:

Commission as % of DNP - With no VAT

Help:

Commission as % of DNP - With no VAT

Associated Code Variables:

aml\_a\_hishtatfut\_dnp

Modified On:

3/13/2025 9:40:45 AM (UTC+02:00)

Modified By:

CLAL-INS\ahuvaa

Validation Failure Behaviour:

Error

Variable Type:

Floating Point Number

Valid Range From:

Valid Range To:

Choice List:

Value Type:

Data

Value:

AMLA\_HISHTATFUT\_DNP

#### **2.1.17.1.121    Input Variable: *age\_exact\_issue***

Description:

Age at issue

Help:

Age at issue.

Can be age last in years (e.g. 35) or exact age using decimals (e.g. 35.75).

This age is rounded down in the model which is based on age last, incrementing by one year each policy anniversary.

Associated Code Variables:

age\_at\_issue

Modified On:

8/27/2019 4:00:59 PM (UTC+03:00)

Modified By:

CLAL-INS\NinaB

Validation Failure Behaviour:

Error

Variable Type: Floating Point Number  
 Valid Range From:  
 Valid Range To:

Choice List:  
 Value Type: Data  
 Value: age

### **2.1.17.1.122    *Input Variable: Chilean***

Description: Chilean indicator for gimla  
 Help: Can be age last in years (e.g. 35) or exact age using decimals (e.g. 35.75).  
 This age is rounded down in the model which is based on age last, incrementing by one year each policy anniversary.

Associated Code Variables: chilean  
 Modified On: 5/15/2023 1:48:12 PM (UTC+03:00)  
 Modified By: CLAL-INS\Arikt  
 Validation Failure Behaviour: Error  
 Variable Type: Floating Point Number  
 Valid Range From:  
 Valid Range To:

Choice List:  
 Value Type: Data  
 Value: Chilean

### **2.1.17.1.123    *Input Variable: foreign\_id***

Description: Foreign / citizen insured identification  
 Help: Can be age last in years (e.g. 35) or exact age using decimals (e.g. 35.75).  
 This age is rounded down in the model which is based on age last, incrementing by one year each policy anniversary.

Associated Code Variables: foreign\_id  
 Modified On: 5/15/2023 1:48:16 PM (UTC+03:00)  
 Modified By: CLAL-INS\Arikt  
 Validation Failure Behaviour: Error  
 Variable Type: Floating Point Number  
 Valid Range From:  
 Valid Range To:

Choice List:  
 Value Type: Data  
 Value: foreign\_id

### **2.1.17.1.124    *Input Variable: tat\_shnatiut\_input***

Description: Foreign / citizen insured identification

Help: modal loading percentage. Set from fund\_rate\_tbl in set exp variables if "read\_from\_table" = Y;  
 Associated Code Variables: tat\_shnatiut\_input  
 Modified On: 12/7/2023 12:03:09 PM (UTC+02:00)  
 Modified By: CLAL-INS\Arikt  
 Validation Failure Behaviour: Error  
 Variable Type: Floating Point Number  
 Valid Range From:  
 Valid Range To:  
  
 Choice List:  
 Value Type: Data  
 Value: TAT\_SHNATIUT

### **2.1.17.1.125    Input Variable: prem\_curr**

Description: Annual gross premium  
 Help:  
 Associated Code Variables: prem\_curr\_input  
 Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
 Modified By: CLAL-INS\NinaB  
 Validation Failure Behaviour: Error  
 Variable Type: Floating Point Number  
 Valid Range From:  
 Valid Range To:  
  
 Choice List:  
 Value Type: Data  
 Value: gross\_prem

### **2.1.17.1.126    Input Variable: prem\_orig**

Description: Original premium at valn date  
 Help: Current in force annual premium per policy at the valuation date.  
 Associated Code Variables: prem\_orig  
 Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
 Modified By: CLAL-INS\NinaB  
 Validation Failure Behaviour: Error  
 Variable Type: Floating Point Number  
 Valid Range From:  
 Valid Range To:  
  
 Choice List:  
 Value Type: Data  
 Value: gprem\_orig

### **2.1.17.1.127    Input Variable: promil**

Description: promil from data file  
 Help: Used for free covers (zero premium) as criteria to

Associated Code Variables: skip or project the cover.  
promil  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:

Choice List:  
Value Type: Data  
Value: promil

#### **2.1.17.1.128    *Input Variable: policy\_fee***

Description: Annual policy fee  
Help:  
Associated Code Variables: policy\_fee\_input  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:

Choice List:  
Value Type: Data  
Value: gorem

#### **2.1.17.1.129    *Input Variable: prem\_term***

Description: Policy premium term (months)  
Help:  
Associated Code Variables: prem\_term\_input  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:

Choice List:  
Value Type: Data  
Value: term\_prem

#### **2.1.17.1.130    *Input Variable: res\_kitzba***

Description: reserve from inforce for kitzba  
Help: variable linked with the kitzba reserve field from the inforce file

Associated Code Variables: res\_kitzba\_input  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:  
  
Choice List:  
Value Type: Data  
Value: res\_kiz

#### **2.1.17.1.131    *Input Variable: dac\_tax\_inforce***

Description: DAC tax or Zillmer from inforce  
Help:  
Associated Code Variables: dac\_tax\_inforce\_input  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:  
  
Choice List:  
Value Type: Data  
Value: taxdac\_zil

#### **2.1.17.1.132    *Input Variable: dac\_book\_inforce***

Description: DAC books from inforce  
Help:  
Associated Code Variables: dac\_book\_inforce\_input  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:  
  
Choice List:  
Value Type: Data  
Value: book\_dac

#### **2.1.17.1.133    *Input Variable: resinforce***

Description: reserve from inforce  
Help: variable linked with the reserve field from the inforce file  
Associated Code Variables: resinforce\_input

Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:  
  
Choice List:  
Value Type: Data  
Value: reserve

#### **2.1.17.1.134    *Input Variable: profit\_weighting***

Description: profit weighting for IFRS  
Help: Maximum cumulative claim inflation allowed (for health products).  
As a percentage.  
For example, 200 means that the claims inflation will stop if and when the claims reach double the base assumption.  
  
Associated Code Variables: profit\_weighting  
Modified On: 5/23/2022 11:42:19 AM (UTC+03:00)  
Modified By: CLAL-INS\ahuvaa  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:  
  
Choice List:  
Value Type: Data  
Value: profit\_weighting

#### **2.1.17.1.135    *Input Variable: profit\_weighting\_re***

Description: profit weighting for IFRS  
Help: Maximum cumulative claim inflation allowed (for health products).  
As a percentage.  
For example, 200 means that the claims inflation will stop if and when the claims reach double the base assumption.  
  
Associated Code Variables: profit\_weighting\_re  
Modified On: 5/8/2023 9:44:26 AM (UTC+03:00)  
Modified By: CLAL-INS\Arikt  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:  
  
Choice List:  
Value Type: Data

Value: PROFIT\_WEIGHTING\_RE

#### **2.1.17.1.136    *Input Variable: profil\_dynamic***

Description: Profil dynamic model (0=No, 1=Yes)  
Help: prem type (0=out, 1=in) per profil rider.  
Associated Code Variables: profil\_dynamic  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Integer Array  
Valid Range From:  
Valid Range To:

Choice List:  
Value Type: Data  
Value: dynamic

#### **2.1.17.1.137    *Input Variable: profil\_dyn\_child\_term***

Description: Profil dynamic model-Risk reduction term for child  
Help: prem type (0=out, 1=in) per profil rider.  
Associated Code Variables: profil\_dyn\_child\_term  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Integer Array  
Valid Range From:  
Valid Range To:

Choice List:  
Value Type: Data  
Value: term\_child

#### **2.1.17.1.138    *Input Variable: profil\_dyn\_child\_sa***

Description: Profil dynamic model-Amount of SA for child  
Help: prem type (0=out, 1=in) per profil rider.  
Associated Code Variables: profil\_dyn\_child\_sa  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Integer Array  
Valid Range From:  
Valid Range To:

Choice List:  
Value Type: Data  
Value: sa\_child



**2.1.17.1.139 Input Variable: profil\_dyn\_spous\_term**

Description: Profil dynamic model-Risk reduction term for spous  
Help: prem type (0=out, 1=in) per profil rider.  
Associated Code Variables: profil\_dyn\_spous\_term  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Integer Array  
Valid Range From:  
Valid Range To:  
  
Choice List:  
Value Type: Data  
Value: term\_spous

**2.1.17.1.140 Input Variable: profil\_dyn\_spous\_sa**

Description: Profil dynamic model-Amount of SA for spous  
Help: prem type (0=out, 1=in) per profil rider.  
Associated Code Variables: profil\_dyn\_spous\_sa  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Integer Array  
Valid Range From:  
Valid Range To:  
  
Choice List:  
Value Type: Data  
Value: sa\_spouse

**2.1.17.1.141 Input Variable: pol\_number\_i**

Description: Pol Number I  
Help:  
Associated Code Variables: pol\_number\_i  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Character  
Valid Range From:  
Valid Range To:  
  
Choice List: 0  
Value Type: Data  
Value: pol\_num

## 2.1.18 Data Page: riders\_Data

### 2.1.18.1 Assumption Set: Base

Description:

Help:

Top Model Object:

life

Modified On:

8/19/2021 8:49:51 AM (UTC+03:00)

Modified By:

CLAL-INS\joshm

Data Page Model Object:

riders

Assumption Set Links:

Unlinked

External Source:

Riders

Model Point Extraction:

None

Model Points:

Apply Model Point Weight

No

Weighting File

Weighted Data Field

#### 2.1.18.1.1 Input Variable: aml\_ni\_1\_6

Description:

Aml Ni 1 6

Help:

Associated Code Variables:

aml\_ni\_1\_6

Modified On:

8/27/2019 4:00:59 PM (UTC+03:00)

Modified By:

CLAL-INS\NinaB

Validation Failure Behaviour:

Error

Variable Type:

Floating Point Number

Valid Range From:

Valid Range To:

Choice List:

Value Type:

Data

Value:

aml\_ni\_1\_6

#### 2.1.18.1.2 Input Variable: amla\_1\_6

Description:

Amla 1 6

Help:

Associated Code Variables:

amla\_1\_6

Modified On:

8/27/2019 4:00:59 PM (UTC+03:00)

Modified By:

CLAL-INS\NinaB

Validation Failure Behaviour:

Error

Variable Type:

Floating Point Number

Valid Range From:

Valid Range To:

Choice List:

Value Type:

Data

Value:

amla\_1\_6

**2.1.18.1.3      Input Variable: amla\_7**

Description: Amla 7  
Help:  
Associated Code Variables: amla\_7  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:  
  
Choice List:  
Value Type: Data  
Value: amla\_7

**2.1.18.1.4      Input Variable: amla\_ni\_7**

Description: Amla Ni 7  
Help:  
Associated Code Variables: amla\_ni\_7  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:  
  
Choice List:  
Value Type: Data  
Value: amla\_ni\_7

**2.1.18.1.5      Input Variable: dynamic**

Description: Dynamic  
Help:  
Associated Code Variables: dynamic  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Integer Number  
Valid Range From:  
Valid Range To:  
  
Choice List:  
Value Type: Data  
Value: dynamic

**2.1.18.1.6      Input Variable: lod\_amt\_1**

Description: Lod Amt 1

Help:  
Associated Code Variables: lod\_amt\_1  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:

Choice List:  
Value Type: Data  
Value: lod\_amt\_1

#### **2.1.18.1.7      *Input Variable: lod\_pe\_r\_1***

Description: Lod Pe R 1  
Help:  
Associated Code Variables: lod\_pe\_r\_1  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Integer Number  
Valid Range From:  
Valid Range To:

Choice List:  
Value Type: Data  
Value: lod\_pe\_r\_1

#### **2.1.18.1.8      *Input Variable: pol\_number\_i***

Description: Pol Number I  
Help:  
Associated Code Variables: pol\_number\_i  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Character  
Valid Range From:  
Valid Range To:

Choice List: 0  
Value Type: Data  
Value: keyfield

#### **2.1.18.1.9      *Input Variable: pr\_cov\_cal***

Description: Pr Cov Cal  
Help:  
Associated Code Variables: pr\_cov\_cal

Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Integer Number  
Valid Range From:  
Valid Range To:

Choice List:  
Value Type: Data  
Value: pr\_cov\_cal

#### **2.1.18.1.10      *Input Variable: prm\_in\_ppn***

Description: Prm In Ppn  
Help: is set to 0 or 1  
0 = rider premium is out  
1 = rider premium is in  
Associated Code Variables: prm\_in\_ppn  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Integer Number  
Valid Range From:  
Valid Range To:

Choice List:  
Value Type: Data  
Value: prm\_in\_ppn

#### **2.1.18.1.11      *Input Variable: retention***

Description: Retention  
Help:  
Associated Code Variables: retention  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Floating Point Number  
Valid Range From:  
Valid Range To:

Choice List:  
Value Type: Data  
Value: retention

#### **2.1.18.1.12      *Input Variable: rid\_sex***

Description: Rid Sex  
Help:  
Associated Code Variables: rid\_sex

Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Integer Number  
Valid Range From:  
Valid Range To:  
  
Choice List:  
Value Type: Data  
Value: rid\_sex

#### **2.1.18.1.13      *Input Variable: risk\_type***

Description: Risk Type  
Help:  
Associated Code Variables: risk\_type  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Integer Number  
Valid Range From:  
Valid Range To:  
  
Choice List:  
Value Type: Data  
Value: risk\_type

#### **2.1.18.1.14      *Input Variable: sum\_as***

Description: Sum As  
Help:  
Associated Code Variables: sum\_as  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Validation Failure Behaviour: Error  
Variable Type: Integer Number  
Valid Range From:  
Valid Range To:  
  
Choice List:  
Value Type: Data  
Value: sum\_as

#### **2.1.18.1.15      *Input Variable: tarif***

Description: Tarif  
Help:  
Associated Code Variables: tarif  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB

Validation Failure Behaviour: Error  
 Variable Type: Integer Number  
 Valid Range From:  
 Valid Range To:

Choice List:  
 Value Type: Data  
 Value: tarif

### **2.1.18.1.16      *Input Variable: prem\_cover***

Description: prem\_cover  
 Help:  
 Associated Code Variables: prem\_cover\_input  
 Modified On: 8/19/2021 8:49:51 AM (UTC+03:00)  
 Modified By: CLAL-INS\joshm  
 Validation Failure Behaviour: Error  
 Variable Type: Character  
 Valid Range From:  
 Valid Range To:

Choice List: 0  
 Value Type: Data  
 Value: prem\_cover

## **2.1.19            External Sources**

### **2.1.19.1           External Sources used in Input Pages**

#### **2.1.19.1.1        *Excel External Sources***

##### **2.1.19.1.1.1      alloc**

Modified On: 1/6/2022 5:45:42 PM (UTC+02:00)  
 Modified By: CLAL-INS\joshm  
 Selection Type: Named Range  
 Selection: alloc  
 Cell Range:

##### **2.1.19.1.1.1.1    T:\RiskAgilityFM\Fixed tables\Life\_fixed\_assumptions.xlsx**

Projections used in: Base  
 File Size: 3.24 MB (3392547 Bytes)  
 File Date Modified: 2/12/2025 10:42:36 AM (UTC+02:00)

##### **2.1.19.1.1.2      ann\_mort\_08\_F\_BE**

Modified On: 1/6/2022 5:45:42 PM (UTC+02:00)  
 Modified By: CLAL-INS\joshm  
 Selection Type: Named Range  
 Selection: ann\_mort\_08\_F\_BE  
 Cell Range:

**2.1.19.1.1.2.1 T:\RiskAgilityFM\Fixed tables\\Life\_fixed\_assumptions.xlsx**

Projections used in: Base  
File Size: 3.24 MB (3392547 Bytes)  
File Date Modified: 2/12/2025 10:42:36 AM (UTC+02:00)

**2.1.19.1.1.3 ann\_mort\_08\_F\_res**

Modified On: 1/6/2022 5:45:42 PM (UTC+02:00)  
Modified By: CLAL-INS\joshm  
Selection Type: Named Range  
Selection: ann\_mort\_08\_F\_res  
Cell Range:

**2.1.19.1.1.3.1 T:\RiskAgilityFM\Fixed tables\\Life\_fixed\_assumptions.xlsx**

Projections used in: Base  
File Size: 3.24 MB (3392547 Bytes)  
File Date Modified: 2/12/2025 10:42:36 AM (UTC+02:00)

**2.1.19.1.1.4 ann\_mort\_08\_M\_BE**

Modified On: 1/6/2022 5:45:42 PM (UTC+02:00)  
Modified By: CLAL-INS\joshm  
Selection Type: Named Range  
Selection: ann\_mort\_08\_M\_BE  
Cell Range:

**2.1.19.1.1.4.1 T:\RiskAgilityFM\Fixed tables\\Life\_fixed\_assumptions.xlsx**

Projections used in: Base  
File Size: 3.24 MB (3392547 Bytes)  
File Date Modified: 2/12/2025 10:42:36 AM (UTC+02:00)

**2.1.19.1.1.5 ann\_mort\_08\_M\_res**

Modified On: 1/6/2022 5:45:42 PM (UTC+02:00)  
Modified By: CLAL-INS\joshm  
Selection Type: Named Range  
Selection: ann\_mort\_08\_M\_res  
Cell Range:

**2.1.19.1.1.5.1 T:\RiskAgilityFM\Fixed tables\\Life\_fixed\_assumptions.xlsx**

Projections used in: Base  
File Size: 3.24 MB (3392547 Bytes)  
File Date Modified: 2/12/2025 10:42:36 AM (UTC+02:00)

**2.1.19.1.1.6 ann\_mort\_b3\_08\_F\_BE**

Modified On: 1/6/2022 5:45:42 PM (UTC+02:00)  
Modified By: CLAL-INS\joshm  
Selection Type: Named Range  
Selection: ann\_mort\_b3\_08\_F\_BE  
Cell Range:



**2.1.19.1.1.6.1 T:\RiskAgilityFM\Fixed tables\\Life\_fixed\_assumptions.xlsx**

Projections used in: Base  
File Size: 3.24 MB (3392547 Bytes)  
File Date Modified: 2/12/2025 10:42:36 AM (UTC+02:00)

**2.1.19.1.1.7 ann\_mort\_b3\_08\_F\_res**

Modified On: 1/6/2022 5:45:42 PM (UTC+02:00)  
Modified By: CLAL-INS\joshm  
Selection Type: Named Range  
Selection: ann\_mort\_b3\_08\_F\_res  
Cell Range:

**2.1.19.1.1.7.1 T:\RiskAgilityFM\Fixed tables\\Life\_fixed\_assumptions.xlsx**

Projections used in: Base  
File Size: 3.24 MB (3392547 Bytes)  
File Date Modified: 2/12/2025 10:42:36 AM (UTC+02:00)

**2.1.19.1.1.8 ann\_mort\_b3\_08\_M\_BE**

Modified On: 1/6/2022 5:45:42 PM (UTC+02:00)  
Modified By: CLAL-INS\joshm  
Selection Type: Named Range  
Selection: ann\_mort\_b3\_08\_M\_BE  
Cell Range:

**2.1.19.1.1.8.1 T:\RiskAgilityFM\Fixed tables\\Life\_fixed\_assumptions.xlsx**

Projections used in: Base  
File Size: 3.24 MB (3392547 Bytes)  
File Date Modified: 2/12/2025 10:42:36 AM (UTC+02:00)

**2.1.19.1.1.9 ann\_mort\_b3\_08\_M\_res**

Modified On: 1/6/2022 5:45:42 PM (UTC+02:00)  
Modified By: CLAL-INS\joshm  
Selection Type: Named Range  
Selection: ann\_mort\_b3\_08\_M\_res  
Cell Range:

**2.1.19.1.1.9.1 T:\RiskAgilityFM\Fixed tables\\Life\_fixed\_assumptions.xlsx**

Projections used in: Base  
File Size: 3.24 MB (3392547 Bytes)  
File Date Modified: 2/12/2025 10:42:36 AM (UTC+02:00)

**2.1.19.1.1.10 Annuity**

Modified On: 6/7/2023 10:06:35 AM (UTC+03:00)  
Modified By: CLAL-INS\ahuvaa  
Selection Type: Worksheet  
Selection: Annuity  
Cell Range:

**2.1.19.1.1.10.1 S:\Data Files\Input\2412\Life\Model v97\Assumptions\Main assumptions - variable.xlsx**

Projections used in:	Base
File Size:	16.36 MB (17151815 Bytes)
File Date Modified:	6/18/2025 5:00:59 PM (UTC+03:00)

**2.1.19.1.1.11 AnnuityDetails**

Modified On:	6/7/2023 4:49:53 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Selection Type	Worksheet
Selection	AnnuityDetails
Cell Range	

**2.1.19.1.1.11.1 S:\Data Files\Input\2412\Life\Model v97\Assumptions\Main assumptions - variable.xlsx**

Projections used in:	Base
File Size:	16.36 MB (17151815 Bytes)
File Date Modified:	6/18/2025 5:00:59 PM (UTC+03:00)

**2.1.19.1.1.12 AnnuityTU**

Modified On:	6/7/2023 4:50:27 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Selection Type	Worksheet
Selection	AnnuityTU
Cell Range	

**2.1.19.1.1.12.1 S:\Data Files\Input\2412\Life\Model v97\Assumptions\Main assumptions - variable.xlsx**

Projections used in:	Base
File Size:	16.36 MB (17151815 Bytes)
File Date Modified:	6/18/2025 5:00:59 PM (UTC+03:00)

**2.1.19.1.1.13 Asset\_Shocks**

Modified On:	6/7/2023 4:51:30 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Selection Type	Worksheet
Selection	Asset_Shocks
Cell Range	

**2.1.19.1.1.13.1 S:\Data Files\Input\2412\Life\Model v97\Assumptions\Main assumptions - variable.xlsx**

Projections used in:	Base
File Size:	16.36 MB (17151815 Bytes)
File Date Modified:	6/18/2025 5:00:59 PM (UTC+03:00)

**2.1.19.1.1.14 bonus5**

Modified On:	1/6/2022 5:45:43 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm

Selection Type	Named Range
Selection	bonus5
Cell Range	

#### 2.1.19.1.1.14.1 T:\RiskAgilityFM\Fixed tables\Life\_fixed\_assumptions.xlsx

Projections used in:	Base
File Size:	3.24 MB (3392547 Bytes)
File Date Modified:	2/12/2025 10:42:36 AM (UTC+02:00)

#### 2.1.19.1.1.15 claim\_cost\_phi12\_ltc07

Modified On:	6/7/2023 4:51:00 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Selection Type	Worksheet
Selection	claim_cost_phi12_ltc07
Cell Range	

#### 2.1.19.1.1.15.1 S:\Data Files\Input\2412\Life\Model v97\Assumptions\Main assumptions - variable.xlsx

Projections used in:	Base
File Size:	16.36 MB (17151815 Bytes)
File Date Modified:	6/18/2025 5:00:59 PM (UTC+03:00)

#### 2.1.19.1.1.16 clms\_mult

Modified On:	6/7/2023 4:51:39 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Selection Type	Worksheet
Selection	clms_mult
Cell Range	

#### 2.1.19.1.1.16.1 S:\Data Files\Input\2412\Life\Model v97\Assumptions\Main assumptions - variable.xlsx

Projections used in:	Base
File Size:	16.36 MB (17151815 Bytes)
File Date Modified:	6/18/2025 5:00:59 PM (UTC+03:00)

#### 2.1.19.1.1.17 comm\_extra

Modified On:	6/7/2023 4:52:33 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Selection Type	Worksheet
Selection	comm_extra
Cell Range	

#### 2.1.19.1.1.17.1 S:\Data Files\Input\2412\Life\Model v97\Assumptions\Main assumptions - variable.xlsx

Projections used in:	Base
File Size:	16.36 MB (17151815 Bytes)
File Date Modified:	6/18/2025 5:00:59 PM (UTC+03:00)

**2.1.19.1.1.18 comm\_extra\_agent**

Modified On:	6/7/2023 4:52:54 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Selection Type	Worksheet
Selection	comm_extra_agent
Cell Range	

**2.1.19.1.1.18.1 S:\Data Files\Input\2412\Life\Model v97\Assumptions\Main assumptions - variable.xlsx**

Projections used in:	Base
File Size:	16.36 MB (17151815 Bytes)
File Date Modified:	6/18/2025 5:00:59 PM (UTC+03:00)

**2.1.19.1.1.19 commclaw**

Modified On:	6/7/2023 4:53:07 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Selection Type	Worksheet
Selection	commclaw
Cell Range	

**2.1.19.1.1.19.1 S:\Data Files\Input\2412\Life\Model v97\Assumptions\Main assumptions - variable.xlsx**

Projections used in:	Base
File Size:	16.36 MB (17151815 Bytes)
File Date Modified:	6/18/2025 5:00:59 PM (UTC+03:00)

**2.1.19.1.1.20 decrmult**

Modified On:	6/7/2023 4:53:49 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Selection Type	Worksheet
Selection	decrmalt
Cell Range	

**2.1.19.1.1.20.1 S:\Data Files\Input\2412\Life\Model v97\Assumptions\Main assumptions - variable.xlsx**

Projections used in:	Base
File Size:	16.36 MB (17151815 Bytes)
File Date Modified:	6/18/2025 5:00:59 PM (UTC+03:00)

**2.1.19.1.1.21 Discount\_Scenarios**

Modified On:	6/7/2023 4:53:52 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Selection Type	Worksheet
Selection	Discount_Scenarios
Cell Range	

**2.1.19.1.1.21.1 S:\Data Files\Input\2412\Life\Model v97\Assumptions\Main assumptions - variable.xlsx**

Projections used in:	Base
File Size:	16.36 MB (17151815 Bytes)
File Date Modified:	6/18/2025 5:00:59 PM (UTC+03:00)

**2.1.19.1.1.22 Economic**

Modified On:	6/7/2023 4:53:56 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Selection Type	Worksheet
Selection	Economic
Cell Range	

**2.1.19.1.1.22.1 S:\Data Files\Input\2412\Life\Model v97\Assumptions\Main assumptions - variable.xlsx**

Projections used in:	Base
File Size:	16.36 MB (17151815 Bytes)
File Date Modified:	6/18/2025 5:00:59 PM (UTC+03:00)

**2.1.19.1.1.23 exp\_mult**

Modified On:	6/7/2023 4:54:07 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Selection Type	Worksheet
Selection	exp_mult
Cell Range	

**2.1.19.1.1.23.1 S:\Data Files\Input\2412\Life\Model v97\Assumptions\Main assumptions - variable.xlsx**

Projections used in:	Base
File Size:	16.36 MB (17151815 Bytes)
File Date Modified:	6/18/2025 5:00:59 PM (UTC+03:00)

**2.1.19.1.1.24 expense**

Modified On:	6/7/2023 4:54:13 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Selection Type	Worksheet
Selection	expense
Cell Range	

**2.1.19.1.1.24.1 S:\Data Files\Input\2412\Life\Model v97\Assumptions\Main assumptions - variable.xlsx**

Projections used in:	Base
File Size:	16.36 MB (17151815 Bytes)
File Date Modified:	6/18/2025 5:00:59 PM (UTC+03:00)

**2.1.19.1.1.25 format\_mgtfee**

Modified On:	6/7/2023 4:54:50 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa

Selection Type	Worksheet
Selection	format_mgtfee
Cell Range	

**2.1.19.1.1.25.1 S:\Data Files\Input\2412\Life\Model v97\Assumptions\Main assumptions - variable.xlsx**

Projections used in:	Base
File Size:	16.36 MB (17151815 Bytes)
File Date Modified:	6/18/2025 5:00:59 PM (UTC+03:00)

**2.1.19.1.1.26 FreeInvRatio**

Modified On:	6/7/2023 4:54:57 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Selection Type	Worksheet
Selection	FreeInvRatio
Cell Range	

**2.1.19.1.1.26.1 S:\Data Files\Input\2412\Life\Model v97\Assumptions\Main assumptions - variable.xlsx**

Projections used in:	Base
File Size:	16.36 MB (17151815 Bytes)
File Date Modified:	6/18/2025 5:00:59 PM (UTC+03:00)

**2.1.19.1.1.27 fundrate**

Modified On:	6/7/2023 4:55:21 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Selection Type	Worksheet
Selection	fundrate
Cell Range	

**2.1.19.1.1.27.1 S:\Data Files\Input\2412\Life\Model v97\Assumptions\Main assumptions - variable.xlsx**

Projections used in:	Base
File Size:	16.36 MB (17151815 Bytes)
File Date Modified:	6/18/2025 5:00:59 PM (UTC+03:00)

**2.1.19.1.1.28 gimla**

Modified On:	1/6/2022 5:45:43 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Selection Type	Named Range
Selection	GIMLA
Cell Range	

**2.1.19.1.1.28.1 T:\RiskAgilityFM\Fixed tables\Life\_fixed\_assumptions.xlsx**

Projections used in:	Base
File Size:	3.24 MB (3392547 Bytes)
File Date Modified:	2/12/2025 10:42:36 AM (UTC+02:00)

**2.1.19.1.1.29 lapse**

Modified On: 6/7/2023 4:55:28 PM (UTC+03:00)  
Modified By: CLAL-INS\ahuvaa  
Selection Type: Worksheet  
Selection: lapse  
Cell Range:

**2.1.19.1.1.29.1 S:\Data Files\Input\2412\Life\Model v97\Assumptions\Main assumptions - variable.xlsx**

Projections used in: Base  
File Size: 16.36 MB (17151815 Bytes)  
File Date Modified: 6/18/2025 5:00:59 PM (UTC+03:00)

**2.1.19.1.1.30 lapse\_factor**

Modified On: 6/7/2023 4:55:40 PM (UTC+03:00)  
Modified By: CLAL-INS\ahuvaa  
Selection Type: Worksheet  
Selection: lapse\_factor  
Cell Range:

**2.1.19.1.1.30.1 S:\Data Files\Input\2412\Life\Model v97\Assumptions\Main assumptions - variable.xlsx**

Projections used in: Base  
File Size: 16.36 MB (17151815 Bytes)  
File Date Modified: 6/18/2025 5:00:59 PM (UTC+03:00)

**2.1.19.1.1.31 lapse\_factor\_proj**

Modified On: 12/19/2024 1:25:27 PM (UTC+02:00)  
Modified By: CLAL-INS\arikt  
Selection Type: Worksheet  
Selection: lapse\_factor\_proj  
Cell Range:

**2.1.19.1.1.31.1 S:\Data Files\Input\2412\Life\Model v97\Assumptions\Main assumptions - variable.xlsx**

Projections used in: Base  
File Size: 16.36 MB (17151815 Bytes)  
File Date Modified: 6/18/2025 5:00:59 PM (UTC+03:00)

**2.1.19.1.1.32 life\_treaty\_details**

Modified On: 6/7/2023 4:56:22 PM (UTC+03:00)  
Modified By: CLAL-INS\ahuvaa  
Selection Type: Worksheet  
Selection: life\_treaty\_details  
Cell Range:

**2.1.19.1.1.32.1 S:\Data Files\Input\2412\Life\Model v97\Assumptions\Main assumptions - variable.xlsx**

Projections used in:	Base
File Size:	16.36 MB (17151815 Bytes)
File Date Modified:	6/18/2025 5:00:59 PM (UTC+03:00)

**2.1.19.1.1.33 LifeReins**

Modified On:	6/7/2023 4:56:33 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Selection Type	Worksheet
Selection	LifeReins
Cell Range	

**2.1.19.1.1.33.1 S:\Data Files\Input\2412\Life\Model v97\Assumptions\Main assumptions - variable.xlsx**

Projections used in:	Base
File Size:	16.36 MB (17151815 Bytes)
File Date Modified:	6/18/2025 5:00:59 PM (UTC+03:00)

**2.1.19.1.1.34 margins**

Modified On:	6/7/2023 4:56:38 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Selection Type	Worksheet
Selection	Margins
Cell Range	

**2.1.19.1.1.34.1 S:\Data Files\Input\2412\Life\Model v97\Assumptions\Main assumptions - variable.xlsx**

Projections used in:	Base
File Size:	16.36 MB (17151815 Bytes)
File Date Modified:	6/18/2025 5:00:59 PM (UTC+03:00)

**2.1.19.1.1.35 mass\_lapse\_tab**

Modified On:	1/6/2022 5:45:43 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Selection Type	Named Range
Selection	MassLapTab
Cell Range	

**2.1.19.1.1.35.1 T:\RiskAgilityFM\Fixed tables\Life\_fixed\_assumptions.xlsx**

Projections used in:	Base
File Size:	3.24 MB (3392547 Bytes)
File Date Modified:	2/12/2025 10:42:36 AM (UTC+02:00)

**2.1.19.1.1.36 mortmult**

Modified On:	6/7/2023 4:56:55 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Selection Type	Worksheet



Selection mortmult  
Cell Range

**2.1.19.1.1.36.1 S:\Data Files\Input\2412\Life\Model v97\Assumptions\Main assumptions - variable.xlsx**

Projections used in: Base  
File Size: 16.36 MB (17151815 Bytes)  
File Date Modified: 6/18/2025 5:00:59 PM (UTC+03:00)

**2.1.19.1.1.37 Parameters**

Modified On: 6/7/2023 4:57:17 PM (UTC+03:00)  
Modified By: CLAL-INS\ahuvaa  
Selection Type: Worksheet  
Selection: Parameters  
Cell Range

**2.1.19.1.1.37.1 S:\Data Files\Input\2412\Life\Model v97\Assumptions\Main assumptions - variable.xlsx**

Projections used in: Base  
File Size: 16.36 MB (17151815 Bytes)  
File Date Modified: 6/18/2025 5:00:59 PM (UTC+03:00)

**2.1.19.1.1.38 phi\_recover**

Modified On: 6/7/2023 4:57:41 PM (UTC+03:00)  
Modified By: CLAL-INS\ahuvaa  
Selection Type: Worksheet  
Selection: phi\_recover  
Cell Range

**2.1.19.1.1.38.1 S:\Data Files\Input\2412\Life\Model v97\Assumptions\Main assumptions - variable.xlsx**

Projections used in: Base  
File Size: 16.36 MB (17151815 Bytes)  
File Date Modified: 6/18/2025 5:00:59 PM (UTC+03:00)

**2.1.19.1.1.39 prem\_code\_map**

Modified On: 6/7/2023 4:58:14 PM (UTC+03:00)  
Modified By: CLAL-INS\ahuvaa  
Selection Type: Worksheet  
Selection: prem\_code\_map  
Cell Range

**2.1.19.1.1.39.1 S:\Data Files\Input\2412\Life\Model v97\Assumptions\Main assumptions - variable.xlsx**

Projections used in: Base  
File Size: 16.36 MB (17151815 Bytes)  
File Date Modified: 6/18/2025 5:00:59 PM (UTC+03:00)

**2.1.19.1.1.40 prem\_rates**

Modified On: 6/7/2023 4:58:26 PM (UTC+03:00)  
Modified By: CLAL-INS\ahuvaa  
Selection Type: Worksheet  
Selection: prem\_rates  
Cell Range:

**2.1.19.1.1.40.1 S:\Data Files\Input\2412\Life\Model v97\Assumptions\Main assumptions - variable.xlsx**

Projections used in: Base  
File Size: 16.36 MB (17151815 Bytes)  
File Date Modified: 6/18/2025 5:00:59 PM (UTC+03:00)

**2.1.19.1.1.41 premium\_rates\_rein\_life**

Modified On: 6/7/2023 4:58:36 PM (UTC+03:00)  
Modified By: CLAL-INS\ahuvaa  
Selection Type: Worksheet  
Selection: premium\_rates\_rein\_life  
Cell Range:

**2.1.19.1.1.41.1 S:\Data Files\Input\2412\Life\Model v97\Assumptions\Main assumptions - variable.xlsx**

Projections used in: Base  
File Size: 16.36 MB (17151815 Bytes)  
File Date Modified: 6/18/2025 5:00:59 PM (UTC+03:00)

**2.1.19.1.1.42 prod\_ass**

Modified On: 6/7/2023 4:58:45 PM (UTC+03:00)  
Modified By: CLAL-INS\ahuvaa  
Selection Type: Worksheet  
Selection: prod\_ass  
Cell Range:

**2.1.19.1.1.42.1 S:\Data Files\Input\2412\Life\Model v97\Assumptions\Main assumptions - variable.xlsx**

Projections used in: Base  
File Size: 16.36 MB (17151815 Bytes)  
File Date Modified: 6/18/2025 5:00:59 PM (UTC+03:00)

**2.1.19.1.1.43 prod\_spec\_term**

Modified On: 6/7/2023 4:58:55 PM (UTC+03:00)  
Modified By: CLAL-INS\ahuvaa  
Selection Type: Worksheet  
Selection: prod\_spec\_term  
Cell Range:

**2.1.19.1.1.43.1 S:\Data Files\Input\2412\Life\Model v97\Assumptions\Main assumptions - variable.xlsx**

Projections used in:	Base
File Size:	16.36 MB (17151815 Bytes)
File Date Modified:	6/18/2025 5:00:59 PM (UTC+03:00)

**2.1.19.1.1.44 prod\_spec\_trad**

Modified On:	6/7/2023 4:59:03 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Selection Type	Worksheet
Selection	prod_spec_trad
Cell Range	

**2.1.19.1.1.44.1 S:\Data Files\Input\2412\Life\Model v97\Assumptions\Main assumptions - variable.xlsx**

Projections used in:	Base
File Size:	16.36 MB (17151815 Bytes)
File Date Modified:	6/18/2025 5:00:59 PM (UTC+03:00)

**2.1.19.1.1.45 prod\_spec\_unit**

Modified On:	6/7/2023 4:59:11 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Selection Type	Worksheet
Selection	prod_spec_unit
Cell Range	

**2.1.19.1.1.45.1 S:\Data Files\Input\2412\Life\Model v97\Assumptions\Main assumptions - variable.xlsx**

Projections used in:	Base
File Size:	16.36 MB (17151815 Bytes)
File Date Modified:	6/18/2025 5:00:59 PM (UTC+03:00)

**2.1.19.1.1.46 profil\_decrement\_rates**

Modified On:	1/6/2022 5:45:42 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Selection Type	Named Range
Selection	profil_decrement_rates_11
Cell Range	

**2.1.19.1.1.46.1 T:\RiskAgilityFM\Fixed tables\Life\_fixed\_assumptions.xlsx**

Projections used in:	Base
File Size:	3.24 MB (3392547 Bytes)
File Date Modified:	2/12/2025 10:42:36 AM (UTC+02:00)

**2.1.19.1.1.47 profil\_rider\_claims\_annuity\_fac**

Modified On:	6/7/2023 4:59:22 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Selection Type	Worksheet

Selection  
Cell Range

profil\_rider\_claims\_annuity\_fac

**2.1.19.1.1.47.1 S:\Data Files\Input\2412\Life\Model v97\Assumptions\Main assumptions - variable.xlsx**

Projections used in: Base  
File Size: 16.36 MB (17151815 Bytes)  
File Date Modified: 6/18/2025 5:00:59 PM (UTC+03:00)

**2.1.19.1.1.48 profil\_rider\_tarif\_map**

Modified On: 6/7/2023 4:59:33 PM (UTC+03:00)  
Modified By: CLAL-INS\ahuvaa  
Selection Type: Worksheet  
Selection: profil\_rider\_tarif\_map  
Cell Range:

**2.1.19.1.1.48.1 S:\Data Files\Input\2412\Life\Model v97\Assumptions\Main assumptions - variable.xlsx**

Projections used in: Base  
File Size: 16.36 MB (17151815 Bytes)  
File Date Modified: 6/18/2025 5:00:59 PM (UTC+03:00)

**2.1.19.1.1.49 RA\_Factor**

Modified On: 6/7/2023 4:59:39 PM (UTC+03:00)  
Modified By: CLAL-INS\ahuvaa  
Selection Type: Worksheet  
Selection: RA\_Factor  
Cell Range:

**2.1.19.1.1.49.1 S:\Data Files\Input\2412\Life\Model v97\Assumptions\Main assumptions - variable.xlsx**

Projections used in: Base  
File Size: 16.36 MB (17151815 Bytes)  
File Date Modified: 6/18/2025 5:00:59 PM (UTC+03:00)

**2.1.19.1.1.50 Reserve\_Factors**

Modified On: 6/7/2023 4:59:52 PM (UTC+03:00)  
Modified By: CLAL-INS\ahuvaa  
Selection Type: Worksheet  
Selection: Reserve\_Factors  
Cell Range:

**2.1.19.1.1.50.1 S:\Data Files\Input\2412\Life\Model v97\Assumptions\Main assumptions - variable.xlsx**

Projections used in: Base  
File Size: 16.36 MB (17151815 Bytes)  
File Date Modified: 6/18/2025 5:00:59 PM (UTC+03:00)

**2.1.19.1.1.51 Reserve\_Manual**

Modified On: 8/5/2024 3:30:55 PM (UTC+03:00)  
Modified By: CLAL-INS\arikt  
Selection Type: Worksheet  
Selection: Reserve\_Manual  
Cell Range:

**2.1.19.1.1.51.1 S:\Data Files\Input\2412\Life\Model v97\Assumptions\Main assumptions - variable.xlsx**

Projections used in: Base  
File Size: 16.36 MB (17151815 Bytes)  
File Date Modified: 6/18/2025 5:00:59 PM (UTC+03:00)

**2.1.19.1.1.52 RFR\_IFRS**

Modified On: 6/8/2023 2:25:42 PM (UTC+03:00)  
Modified By: CLAL-INS\ahuvaa  
Selection Type: Worksheet  
Selection: RFR\_IFRS  
Cell Range:

**2.1.19.1.1.52.1 S:\Data Files\Input\2412\Life\Model v97\Assumptions\Main assumptions - variable.xlsx**

Projections used in: Base  
File Size: 16.36 MB (17151815 Bytes)  
File Date Modified: 6/18/2025 5:00:59 PM (UTC+03:00)

**2.1.19.1.1.53 RFR\_Solv**

Modified On: 6/7/2023 5:00:15 PM (UTC+03:00)  
Modified By: CLAL-INS\ahuvaa  
Selection Type: Worksheet  
Selection: RFR\_Solv  
Cell Range:

**2.1.19.1.1.53.1 S:\Data Files\Input\2412\Life\Model v97\Assumptions\Main assumptions - variable.xlsx**

Projections used in: Base  
File Size: 16.36 MB (17151815 Bytes)  
File Date Modified: 6/18/2025 5:00:59 PM (UTC+03:00)

**2.1.19.1.1.54 Run\_Control**

Modified On: 6/7/2023 5:00:32 PM (UTC+03:00)  
Modified By: CLAL-INS\ahuvaa  
Selection Type: Worksheet  
Selection: Run\_Control  
Cell Range:

**2.1.19.1.1.54.1 S:\Data Files\Input\2412\Life\Model v97\Assumptions\Main assumptions - variable.xlsx**

Projections used in:	Base
File Size:	16.36 MB (17151815 Bytes)
File Date Modified:	6/18/2025 5:00:59 PM (UTC+03:00)

**2.1.19.1.1.55 sal\_inc**

Modified On:	6/7/2023 5:00:39 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Selection Type	Worksheet
Selection	sal_inc
Cell Range	

**2.1.19.1.1.55.1 S:\Data Files\Input\2412\Life\Model v97\Assumptions\Main assumptions - variable.xlsx**

Projections used in:	Base
File Size:	16.36 MB (17151815 Bytes)
File Date Modified:	6/18/2025 5:00:59 PM (UTC+03:00)

**2.1.19.1.1.56 Sel\_Ret\_Qx**

Modified On:	7/23/2024 3:22:52 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Selection Type	Worksheet
Selection	Sel_Ret_Qx
Cell Range	

**2.1.19.1.1.56.1 S:\Data Files\Input\2412\Life\Model v97\Assumptions\Main assumptions - variable.xlsx**

Projections used in:	Base
File Size:	16.36 MB (17151815 Bytes)
File Date Modified:	6/18/2025 5:00:59 PM (UTC+03:00)

**2.1.19.1.1.57 Serv\_Units\_Dur**

Modified On:	6/11/2023 10:23:42 AM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Selection Type	Worksheet
Selection	Serv_Units_Dur
Cell Range	

**2.1.19.1.1.57.1 S:\Data Files\Input\2412\Life\Model v97\Assumptions\Main assumptions - variable.xlsx**

Projections used in:	Base
File Size:	16.36 MB (17151815 Bytes)
File Date Modified:	6/18/2025 5:00:59 PM (UTC+03:00)

**2.1.19.1.1.58 Shimur\_disc**

Modified On:	2/11/2024 3:08:52 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt

Selection Type	Worksheet
Selection	Shimur_Disc
Cell Range	

**2.1.19.1.1.58.1 S:\Data Files\Input\2412\Life\Model v97\Assumptions\Main assumptions - variable.xlsx**

Projections used in:	Base
File Size:	16.36 MB (17151815 Bytes)
File Date Modified:	6/18/2025 5:00:59 PM (UTC+03:00)

**2.1.19.1.1.59 surr\_chg**

Modified On:	1/6/2022 5:45:43 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Selection Type	Named Range
Selection	surr_chg
Cell Range	

**2.1.19.1.1.59.1 T:\RiskAgilityFM\Fixed tables\Life\_fixed\_assumptions.xlsx**

Projections used in:	Base
File Size:	3.24 MB (3392547 Bytes)
File Date Modified:	2/12/2025 10:42:36 AM (UTC+02:00)

**2.1.19.1.1.60 Survival\_Rates**

Modified On:	6/7/2023 5:00:45 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Selection Type	Worksheet
Selection	Survival_Rates
Cell Range	

**2.1.19.1.1.60.1 S:\Data Files\Input\2412\Life\Model v97\Assumptions\Main assumptions - variable.xlsx**

Projections used in:	Base
File Size:	16.36 MB (17151815 Bytes)
File Date Modified:	6/18/2025 5:00:59 PM (UTC+03:00)

**2.1.19.1.1.61 T\_Factors**

Modified On:	6/7/2023 5:00:54 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Selection Type	Worksheet
Selection	T_Factors
Cell Range	

**2.1.19.1.1.61.1 S:\Data Files\Input\2412\Life\Model v97\Assumptions\Main assumptions - variable.xlsx**

Projections used in:	Base
File Size:	16.36 MB (17151815 Bytes)
File Date Modified:	6/18/2025 5:00:59 PM (UTC+03:00)

**2.1.19.1.1.62      tarif\_spec**

Modified On:	6/7/2023 5:01:07 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Selection Type	Worksheet
Selection	tarif_spec
Cell Range	

**2.1.19.1.1.62.1      S:\Data Files\Input\2412\Life\Model v97\Assumptions\Main assumptions - variable.xlsx**

Projections used in:	Base
File Size:	16.36 MB (17151815 Bytes)
File Date Modified:	6/18/2025 5:00:59 PM (UTC+03:00)

**2.1.19.1.1.63      tarif\_spec\_occ**

Modified On:	6/7/2023 5:01:13 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Selection Type	Worksheet
Selection	tarif_spec_occ
Cell Range	

**2.1.19.1.1.63.1      S:\Data Files\Input\2412\Life\Model v97\Assumptions\Main assumptions - variable.xlsx**

Projections used in:	Base
File Size:	16.36 MB (17151815 Bytes)
File Date Modified:	6/18/2025 5:00:59 PM (UTC+03:00)

**2.1.19.1.1.64      tax\_rates**

Modified On:	6/7/2023 5:01:20 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Selection Type	Worksheet
Selection	Tax_Rates
Cell Range	

**2.1.19.1.1.64.1      S:\Data Files\Input\2412\Life\Model v97\Assumptions\Main assumptions - variable.xlsx**

Projections used in:	Base
File Size:	16.36 MB (17151815 Bytes)
File Date Modified:	6/18/2025 5:00:59 PM (UTC+03:00)

**2.1.19.1.1.65      zillmer\_prm**

Modified On:	1/6/2022 5:45:43 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Selection Type	Named Range
Selection	zill_prem
Cell Range	

**2.1.19.1.1.65.1      T:\RiskAgilityFM\Fixed tables\\Life\_fixed\_assumptions.xlsx**

Projections used in:	Base
----------------------	------



File Size: 3.24 MB (3392547 Bytes)  
File Date Modified: 2/12/2025 10:42:36 AM (UTC+02:00)

### **2.1.19.1.2 Text External Sources**

#### **2.1.19.1.2.1 Tzeva Kesef**

Modified On: 7/28/2021 1:12:30 PM (UTC+03:00)  
Modified By: CLAL-INS\joshm  
Delimiter Type: Comma  
Treat Multiple Delimiters As One: No  
Avoid Extraction At Job Submission: No

##### **2.1.19.1.2.1.1 S:\Data Files\Input\2412\Life\Model v97\Data\SP\TK prop.csv**

Projections used in: Base  
File Size: 29.41 MB (30839570 Bytes)  
File Date Modified: 1/12/2025 2:23:15 PM (UTC+02:00)

### **2.1.19.1.3 Composite External Sources**

#### **2.1.19.1.3.1 death\_rates\_comp**

Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Projections used in: Base

##### External Sources:

death_rates	External Source
CMI00FN	CMI00FN
CMI00FS	CMI00FS
CMI00MN	CMI00MN
CMI00MS	CMI00MS

#### **2.1.19.1.3.2 death\_rates\_res\_comp**

Modified On: 3/12/2023 12:29:41 PM (UTC+02:00)  
Modified By: CLAL-INS\ahuvaa  
Projections used in: Base

##### External Sources:

death_rates_res	External Source
AMF4952	AMF4952_tbl
AMF80	AMF80_tbl

#### **2.1.19.1.3.3 decrem\_rates\_com**

Modified On: 3/19/2024 5:16:41 PM (UTC+02:00)  
Modified By: CLAL-INS\Arikt  
Projections used in: Base

## External Sources:

decrem_rates_tbl	External Source
phi_decrem_hash_acc	hash_acc
phi_decrem_hash_acc_sick	hash_acc_sick
phi_decrem_PHI_1	PHI_1
phi_decrem_PHI_3	PHI_3
phi_decrem_PHI_6	PHI_6
phi_decrem_PHIMif_3	PHIMif_3
phi_decrem_PHIMif_6	PHIMif_6
ltc	ltc
dd_346_clal	dd_346_clal
dd_347_clal	dd_347_clal
dd_1363	dd_1363
dd_1362	dd_1362
tpd	tpd

**2.1.19.1.3.4      decrem\_rates\_uw\_com**

Modified On:	3/19/2024 5:16:18 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Projections used in:	Base

## External Sources:

decrem_rates_tbl	External Source
adb_100	ADB_100_UW
adi_08	ADI_08_UW

**2.1.19.1.3.5      prem\_rates\_extra**

Modified On:	8/12/2021 9:36:16 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Projections used in:	Base

## External Sources:

prod_code_adif_extra_prem	External Source
a72	a72_prm
a75	a75_prm
a80-00honi	a80-00honi_prm
a80-01hon	a80-01hon_prm
a80-01kitz	a80-01kitz_prm
rsapir1	rsapir1_prm
rsapir5	rsapir5_prm
asav	asav_tbl
sav-r	sav_r_tbl
ariske	ariske_tbl
none	a72_prm

**2.1.19.1.3.6 prem\_rates\_level\_comp**

Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Projections used in: Base

## External Sources:

prate_level	External Source
50_rsapir5	rsapir5_50
50_rsapir1	rsapir1_50
52_rsapir1	rsapir1_52
52_rsapir5	rsapir5_52
20_rsapir1	rsapir1_20
20_rsapir5	rsapir5_20
44_rsapir1	rsapir1_44
44_rsapir5	rsapir5_44
9_rsapir1	rsapir1_9
9_rsapir5	rsapir5_9

**2.1.19.1.3.7 prem\_rates\_risk\_comp**

Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Projections used in: Base

## External Sources:

prem_rates_risk	External Source
50_rsapir5	rsapir5_50
50_rsapir1	rsapir1_50
52_rsapir1	rsapir1_52
52_rsapir5	rsapir5_52
20_rsapir1	rsapir1_20
20_rsapir5	rsapir5_20
44_rsapir1	rsapir1_44
44_rsapir5	rsapir5_44
9_rsapir1	rsapir1_9
9_rsapir5	rsapir5_9

**2.1.19.1.3.8 prem\_rates\_risk\_rider**

Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Projections used in: Base

## External Sources:

prem_rates_risk	External Source
50_rsapir5	rsapir5_50
50_rsapir1	rsapir1_50
52_rsapir1	rsapir1_52
52_rsapir5	rsapir5_52

20_rsapir1	rsapir1_20
20_rsapir5	rsapir5_20
44_rsapir1	rsapir1_44
44_rsapir5	rsapir5_44
9_rsapir1	rsapir1_9
9_rsapir5	rsapir5_9

### 2.1.19.1.3.9 pup

Modified On:	9/12/2021 5:01:14 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Projections used in:	Base

#### External Sources:

pup_ltc_key	External Source
ltc02-3_F	ltc02-3_F
ltc02-3_M	ltc02-3_M
ltc02-5_F	ltc02-5_F
ltc02-5_M	ltc02-5_M
ltc02-wl_F	ltc02-wl_F
ltc02-wl_M	ltc02-wl_M
ltc-5_F	ltc-5_F
ltc-5_M	ltc-5_M
ltc07-3-y_F	ltc07-3-y_F
ltc07-3-y_M	ltc07-3-y_M
ltc07-5-y_F	ltc07-5-y_F
ltc07-5-y_M	ltc07-5-y_M
ltc07-wl-y_F	ltc07-wl-y_F
ltc07-wl-y_M	ltc07-wl-y_M
ltc-mash_F	ltc-mash_F
ltc-mash_M	ltc-mash_M
ltc-shil_F	ltc-shil_F
ltc-shil_M	ltc-shil_M
ltc-wl_F	ltc-wl_F
ltc-wl_M	ltc-wl_M
ltc-yng-5_F	ltc-yng-5_F
ltc-yng-5_M	ltc-yng-5_M
ltc-yng-wl_F	ltc-yng-wl_F
ltc-yng-wl_M	ltc-yng-wl_M
0	ltc02-5_F

### 2.1.19.1.3.10 puv\_composite

Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Projections used in:	Base

#### External Sources:

puv_tbl	External Source
1_puv_100	puv_1_100

---

1_puv_200	puv_1_200
1_puv_202	puv_1_202
1_puv_601_F	puv_1_601_F
1_puv_601_M	puv_1_601_M
1_puv_620_F	puv_1_620_F
1_puv_620_M	puv_1_620_M
4_puv_100	puv_4_100
4_puv_200	puv_4_200
4_puv_202	puv_4_202
4_puv_601_F	puv_4_601_F
4_puv_601_M	puv_4_601_M
4_puv_620_M	puv_4_620_M
4_puv_620_F	puv_4_620_F
6_puv_100	puv_6_100
6_puv_108	puv_6_108
6_puv_147	puv_6_147
6_puv_200	puv_6_200
6_puv_202	puv_6_202
6_puv_601_F	puv_6_601_F
6_puv_601_M	puv_6_601_M
6_puv_620_F	puv_6_620_F
6_puv_620_M	puv_6_620_M
8_puv_100	puv_8_100
8_puv_106	puv_8_106
8_puv_108	puv_8_108
8_puv_147	puv_8_147
8_puv_200	puv_8_200
8_puv_202	puv_8_202
8_puv_601_F	puv_8_601_F
8_puv_601_M	puv_8_601_M
8_puv_620_F	puv_8_620_F
8_puv_620_M	puv_8_620_M
9_puv_100	puv_9_100
9_puv_106	puv_9_106
9_puv_108	puv_9_108
9_puv_141	puv_9_141
9_puv_147	puv_9_147
9_puv_200	puv_9_200
9_puv_202	puv_9_202
9_puv_601_F	puv_9_601_F
9_puv_601_M	puv_9_601_M
9_puv_604_F	puv_9_604_F
9_puv_604_M	puv_9_604_M
9_puv_620_M	puv_9_620_M
9_puv_620_F	puv_9_620_F
20_puv_100	puv_20_100
20_puv_106	puv_20_106
20_puv_141	puv_20_141
20_puv_200	puv_20_200
20_puv_202	puv_20_202
20_puv_601_F	puv_20_601_F
20_puv_601_M	puv_20_601_M
20_puv_604_F	puv_20_604_F

20\_puv\_604\_M puv\_20\_604\_M  
 20\_puv\_620\_F puv\_20\_620\_F  
 20\_puv\_620\_M puv\_20\_620\_M  
 44\_puv\_100 puv\_44\_100  
 44\_puv\_106 puv\_44\_106  
 44\_puv\_141 puv\_44\_141  
 44\_puv\_200 puv\_44\_200  
 44\_puv\_202 puv\_44\_202  
 44\_puv\_601\_F puv\_44\_601\_F  
 44\_puv\_601\_M puv\_44\_601\_M  
 44\_puv\_604\_F puv\_44\_604\_F  
 44\_puv\_604\_M puv\_44\_604\_M  
 44\_puv\_620\_F puv\_44\_620\_F  
 44\_puv\_620\_M puv\_44\_620\_M  
 50\_puv\_100 puv\_50\_100  
 50\_puv\_106 puv\_50\_106  
 50\_puv\_108 puv\_50\_108  
 50\_puv\_141 puv\_50\_141  
 50\_puv\_200 puv\_50\_200  
 50\_puv\_202 puv\_50\_202  
 50\_puv\_601\_F puv\_50\_601\_F  
 50\_puv\_601\_M puv\_50\_601\_M  
 50\_puv\_604\_F puv\_50\_604\_F  
 50\_puv\_604\_M puv\_50\_604\_M  
 50\_puv\_620\_F puv\_50\_620\_F  
 50\_puv\_620\_M puv\_50\_620\_M  
 52\_puv\_100 puv\_52\_100  
 52\_puv\_141 puv\_52\_141  
 52\_puv\_200 puv\_52\_200  
 52\_puv\_202 puv\_52\_202  
 52\_puv\_601\_F puv\_52\_601\_F  
 52\_puv\_601\_M puv\_52\_601\_M

#### 2.1.19.1.3.11 suminisba

Modified On:  
 Modified By:  
 Projections used in:

8/27/2019 4:00:59 PM (UTC+03:00)  
 CLAL-INS\NinaB  
 Base

#### External Sources:

prod_code	External Source
a72	a72_tbl
a75	a75_tbl
a80-00honi	a80_00honi_tbl
a80-01kitz	a80_01kitz_tbl
a80-01hon	a80_01hon_tbl
a100	a100_tbl
ariske	ariske_tbl
asav	asav_tbl
asave	asave_tbl
sav-r	sav_r_tbl

sav-s sav\_s\_tbl

### 2.1.19.1.3.12 sv\_composite

Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Projections used in: Base

#### External Sources:

sv\_tbl External Source

1\_100 surva1\_1\_100  
1\_200 surva1\_1\_200  
1\_202 surva1\_1\_202  
1\_601 surva1\_1\_601  
1\_601\_F surva1\_1\_601\_F  
1\_601\_M surva1\_1\_601\_M  
1\_620\_F surva1\_1\_620\_F  
1\_620\_M surva1\_1\_620\_M  
4\_100 surva1\_4\_100  
4\_200 surva1\_4\_200  
4\_202 surva1\_4\_202  
4\_601 surva1\_4\_601  
4\_601\_F surva1\_4\_601\_F  
4\_601\_M surva1\_4\_601\_M  
4\_620\_M surva1\_4\_620\_M  
4\_620\_F surva1\_4\_620\_F  
6\_100 surva1\_6\_100  
6\_108 surva1\_6\_108  
6\_147 surva1\_6\_147  
6\_200 surva1\_6\_200  
6\_202 surva1\_6\_202  
6\_601 surva1\_6\_601  
6\_601\_F surva1\_6\_601\_F  
6\_601\_M surva1\_6\_601\_M  
6\_620\_F surva1\_6\_620\_F  
6\_620\_M surva1\_6\_620\_M  
8\_100 surva1\_8\_100  
8\_106 surva1\_8\_106  
8\_108 surva1\_8\_108  
8\_147 surva1\_8\_147  
8\_200 surva1\_8\_200  
8\_202 surva1\_8\_202  
8\_601 surva1\_8\_601  
8\_601\_F surva1\_8\_601\_F  
8\_601\_M surva1\_8\_601\_M  
8\_620\_F surva1\_8\_620\_F  
8\_620\_M surva1\_8\_620\_M  
9\_100 surva1\_9\_100  
9\_106 surva1\_9\_106  
9\_108 surva1\_9\_108  
9\_141 surva1\_9\_141

---

9\_147 surval\_9\_147  
9\_200 surval\_9\_200  
9\_202 surval\_9\_202  
9\_601 surval\_9\_601  
9\_601\_F surval\_9\_601\_F  
9\_601\_M surval\_9\_601\_M  
9\_604\_F surval\_9\_604\_F  
9\_604\_M surval\_9\_604\_M  
9\_620\_M surval\_9\_620\_M  
9\_620\_F surval\_9\_620\_F  
20\_100 surval\_20\_100  
20\_106 surval\_20\_106  
20\_141 surval\_20\_141  
20\_200 surval\_20\_200  
20\_202 surval\_20\_202  
20\_601 surval\_20\_601  
20\_601\_F surval\_20\_601\_F  
20\_601\_M surval\_20\_601\_M  
20\_604\_F surval\_20\_604\_F  
20\_604\_M surval\_20\_604\_M  
20\_620\_F surval\_20\_620\_F  
20\_620\_M surval\_20\_620\_M  
44\_100 surval\_44\_100  
44\_106 surval\_44\_106  
44\_141 surval\_44\_141  
44\_200 surval\_44\_200  
44\_202 surval\_44\_202  
44\_601 surval\_44\_601  
44\_601\_F surval\_44\_601\_F  
44\_601\_M surval\_44\_601\_M  
44\_604\_F surval\_44\_604\_F  
44\_604\_M surval\_44\_604\_M  
44\_620\_F surval\_44\_620\_F  
44\_620\_M surval\_44\_620\_M  
50\_100 surval\_50\_100  
50\_106 surval\_50\_106  
50\_108 surval\_50\_108  
50\_141 surval\_50\_141  
50\_200 surval\_50\_200  
50\_202 surval\_50\_202  
50\_601 surval\_50\_601  
50\_601\_F surval\_50\_601\_F  
50\_601\_M surval\_50\_601\_M  
50\_604\_F surval\_50\_604\_F  
50\_604\_M surval\_50\_604\_M  
50\_620\_F surval\_50\_620\_F  
50\_620\_M surval\_50\_620\_M  
52\_100 surval\_52\_100  
52\_141 surval\_52\_141  
52\_200 surval\_52\_200  
52\_202 surval\_52\_202  
52\_601\_F surval\_52\_601\_F  
52\_601\_M surval\_52\_601\_M



## 2.1.19.2 External Sources used in Data Pages

### 2.1.19.2.1 Text External Sources

#### 2.1.19.2.1.1 Policy data - Solvency

Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
 Modified By: CLAL-INS\NinaB

##### 2.1.19.2.1.1.1 S:\Data Files\Input\2412\Life\Model v97\Data\SP\Solvency.csv

Projections used in: Base  
 File Size: 2.1 KB (2165 Bytes)  
 File Date Modified: 6/8/2025 4:36:15 PM (UTC+03:00)  
 Data Page: life\_Data  
 Number of Model Points Extracted at Job Submission: 1

#### 2.1.19.2.1.2 retirement\_ages

Modified On: 7/22/2021 5:23:14 PM (UTC+03:00)  
 Modified By: CLAL-INS\joshm

##### 2.1.19.2.1.2.1 T:\RiskAgilityFM\Fixed tables\multi\_age\_retirement.csv

Projections used in: Base  
 File Size: 100 Bytes (100 Bytes)  
 File Date Modified: 3/14/2024 3:42:15 PM (UTC+02:00)  
 Data Page: Ann\_Data  
 Number of Model Points Extracted at Job Submission: 22

#### 2.1.19.2.1.3 Riders

Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
 Modified By: CLAL-INS\NinaB

##### 2.1.19.2.1.3.1 S:\Data Files\Input\2412\Life\Model v97\Data\SP\Riders.txt

Projections used in: Base  
 File Size: 2.24 MB (2350937 Bytes)  
 File Date Modified: 1/12/2025 2:25:15 PM (UTC+02:00)  
 Data Page: riders\_Data  
 Number of Model Points Extracted at Job Submission: 37807

## 3 Referenced Files

Referenced files are not used in standalone runs or runs with "Process Input/Output locally on Compute Nodes" unchecked. The referenced input and output details are only displayed when "Process Input/Output locally on Compute Nodes" is checked.

## 4 Output Manager

### 4.1 OM Main

#### 4.1.1 Output Definition: Cashflow - Unify

Model Object:life

Modified On:9/17/2024 9:14:55 PM (UTC+03:00)

Modified By:CLAL-INS\arikt

Name	Type	Header
age_last	Column	Header1
alloc_units	Column	Header1
bonus_shimur	Column	Header1
expense_clm	Column	Header1
coverage_units	Column	Header1
expense_init	Column	Header1
claims_insurance	Column	Header1
expense_ren	Column	Header1
be_retire	Column	Header1
int_cred	Column	Header1
cal_month	Column	Header1
cal_year	Column	Header1
comm_reg	Column	Header1
comm_profit	Column	Header1
claims_annuity_gt	Column	Header1
mgt_fees_prem	Column	Header1
prem_insurance	Column	Header1
prem_savings	Column	Header1
units_for_takeup	Column	Header1

---

cashflow_b	Column	Header1
service_units	Column	Header1
cashflow_e	Column	Header1
service_units_pv	Column	Header1
cashflow_pv	Column	Header1
cashflow_re_b	Column	Header1
cashflow_re_e	Column	Header1
cashflow_re_pv	Column	Header1
charges_premium	Column	Header1
charges_premium_pv	Column	Header1
claims_annuity	Column	Header1
claims_annuity_nogt	Column	Header1
claims_annuity_pv	Column	Header1
claims_death	Column	Header1
claims_death_pv	Column	Header1
claims_disability	Column	Header1
claims_disability_pv	Column	Header1
claims_maturity	Column	Header1
claims_maturity_pv	Column	Header1
claims_pv	Column	Header1
claims_re	Column	Header1
claims_surrender	Column	Header1
claims_surrender_pv	Column	Header1
claims_total	Column	Header1
comm_clawback	Column	Header1
comm_clawback_pv	Column	Header1

---

comm_hekef	Column	Header1
comm_nihul	Column	Header1
comm_prize	Column	Header1
comm_pv	Column	Header1
comm_re	Column	Header1
comm_re_prof	Column	Header1
comm_regular	Column	Header1
comm_renewal	Column	Header1
comm_reserve	Column	Header1
comm_reserve_pv	Column	Header1
comm_supervisor	Column	Header1
comm_total	Column	Header1
cover_charge	Column	Header1
dac_book	Column	Header1
dac_tax	Column	Header1
death_benefit	Column	Header1
death_claim_si	Column	Header1
death_claim_units	Column	Header1
death_rate	Column	Header1
expense_claims_pv	Column	Header1
expense_inflation	Column	Header1
expense_initial_fix	Column	Header1
expense_initial_perc	Column	Header1
expense_investment	Column	Header1
expense_investment_pv	Column	Header1
expense_pv	Column	Header1

---

expense_ren_charge	Column	Header1
expense_ren_charge_pv	Column	Header1
expense_ren_fix	Column	Header1
expense_ren_perc	Column	Header1
exp_total	Column	Header1
expense_var_pv	Column	Header1
interest_re	Column	Header1
interest_re_pv	Column	Header1
investment_income	Column	Header1
investment_income_chetz	Column	Header1
investment_income_chetz_pv	Column	Header1
investment_income_pv	Column	Header1
lapse_rate_act_prm	Column	Header1
lapse_total_prm	Column	Header1
management_fees	Column	Header1
management_fee_pv	Column	Header1
claims_annuity_nogt_pv	Column	Header1
pol_fee	Column	Header1
pol_fee_pv	Column	Header1
premium	Column	Header1
premium_disc	Column	Header1
premium_disc_pv	Column	Header1
premium_extra	Column	Header1
premium_gross	Column	Header1
premium_if_b	Column	Header1
premium_if_riders	Column	Header1

---

premium_pv	Column	Header1
premium_re	Column	Header1
profit_book_active_vif	Column	Header1
profit_bk_act_vif_pv	Column	Header1
profit_book_vif_pv	Column	Header1
profit_book_vif	Column	Header1
profit_net_vif	Column	Header1
profit_net_vif_pv	Column	Header1
profit_re_pv	Column	Header1
proj_month	Column	Header1
proj_year	Column	Header1
pup_rate_prm	Column	Header1
rein_claims_pv	Column	Header1
rein_comm_pv	Column	Header1
rein_prem_pv	Column	Header1
res_ann_deficiency	Column	Header1
reserve	Column	Header1
reserve_annuity	Column	Header1
reserve_basic	Column	Header1
reserve_claims	Column	Header1
reserve_extra	Column	Header1
reserve_increase	Column	Header1
surr_value	Column	Header1
reserve_increase_pv	Column	Header1
reserve_re	Column	Header1
reserve_re_increase	Column	Header1

---

reserve_re_increase_pv	Column	Header1
sum_insured	Column	Header1
sum_insured_if_e	Column	Header1
surv_prm	Column	Header1
units_e	Column	Header1
units_bon	Column	Header1
premium_if_b_total	Column	Header1
cashflow_b_bef_ret	Column	Header1
cashflow_b_post_ret	Column	Header1
profit_book_vif_pv_pos	Column	Header1
management_fees_fixed_ann	Column	Header1
reserve_pv	Column	Header1
manage_fees_fixed_ann_pv	Column	Header1
management_fees_var_active	Column	Header1
management_fees_var_ann	Column	Header1
manage_fees_var_ann_pv	Column	Header1
manage_fees_fixe_active_pv	Column	Header1
manage_fees_var_active_pv	Column	Header1
management_fees_fixed_active	Column	Header1
capital_at_risk	Column	Header1
capital_at_risk_rm	Column	Header1
ber_retire_rm	Column	Header1
bor_acc_pup	Column	Header1
claims_annuity_pv_rm	Column	Header1
claims_death_pv_rm	Column	Header1
claims_disability_pv_rm	Column	Header1

---

expense_pv_rm	Column	Header1
inv_income_chetz_pv_rm	Column	Header1
profit_book_vif_pv_pos_rm	Column	Header1
rid_cashflow_pv	Column	Header1
comm_renewal_pv	Column	Header1
premium_gross_fix	Column	Header1
premium_gross_var	Column	Header1
pol_month	Column	Header1
pol_year	Column	Header1
expense_total_pre_ret	Column	Header1
reserve_increase_bef_ret	Column	Header1
investment_income_bef_ret	Column	Header1
claims_lrc_q1	Column	Header1
claims_lrc_yr2plus	Column	Header1
bor_acc	Column	Header1
bor_return	Column	Header1
bor_return_pup	Column	Header1
comm_hekef_net	Column	Header1
cashflow_pv_e	Column	Header1
claims_lrc_q2	Column	Header1
claims_lrc_q3	Column	Header1
claims_lrc_q4	Column	Header1
claims_re_lrc_q1	Column	Header1
claims_re_lrc_q2	Column	Header1
claims_re_lrc_q3	Column	Header1
claims_re_lrc_q4	Column	Header1



---

claims_re_lrc_yr2plus	Column	Header1
expense_claims_lrc_q1	Column	Header1
expense_claims_lrc_q2	Column	Header1
expense_claims_lrc_q3	Column	Header1
expense_claims_lrc_q4	Column	Header1
expense_claims_lrc_yr2plus	Column	Header1
riskadj_gross_rel_q1	Column	Header1
riskadj_gross_rel_q2	Column	Header1
riskadj_gross_rel_q3	Column	Header1
riskadj_gross_rel_q4	Column	Header1
riskadj_gross_rel_total	Column	Header1
riskadj_gross_rel_yr2plus	Column	Header1
riskadj_re_rel_q1	Column	Header1
riskadj_re_rel_q2	Column	Header1
riskadj_re_rel_q3	Column	Header1
riskadj_re_rel_q4	Column	Header1
riskadj_re_rel_total	Column	Header1
riskadj_re_rel_yr2plus	Column	Header1
fvui	Column	Header1
lapse_rate_act_cnt	Column	Header1
lapse_rate_act_bal	Column	Header1
lapse_rate_pup_prm	Column	Header1
lapse_rate_pup_cnt	Column	Header1
pup_rate_cnt	Column	Header1
pup_rate_bal	Column	Header1
surv_bal	Column	Header1

---

riskadj_gross	Column	Header1
riskadj_net	Column	Header1
coverage_units_re	Column	Header1
profit_book_vif_gross	Column	Header1
profit_book_vif_gross_pv	Column	Header1
surv_cnt	Column	Header1
claim_cost	Column	Header1
claim_cost_pv	Column	Header1
claim_cost_pv_rm	Column	Header1
claim_cost_re_pv	Column	Header1
claim_cost_re	Column	Header1
claim_cost_re_pv_rm	Column	Header1
rein_claims_pv_rm	Column	Header1
cover_charge_pv	Column	Header1
income_b	Column	Header1
income_e	Column	Header1
income_pv	Column	Header1
outgo_b	Column	Header1
outgo_e	Column	Header1
outgo_pv	Column	Header1
cashflow	Column	Header1
expense_pv_active	Column	Header1
expense_pv_ann	Column	Header1
expense_investment_pv_bef_ret	Column	Header1
expense_investment_pv_post_ret	Column	Header1
expense_pv_active_no_inv	Column	Header1

---

comm_not_res_pv	Column	Header1
investment_income_pv_active	Column	Header1
reserve_increase_pv_active	Column	Header1
profit_book_vif_pv_active	Column	Header1
claims_maturity_ret_pv	Column	Header1
units_b	Column	Header1
management_fee_variable	Column	Header1
sum_insured_occ_gross	Column	Header1
sum_insured_occ_retent	Column	Header1
claims_retent	Column	Header1
reserve_claims_retent	Column	Header1
premium_disc_shimur	Column	Header1
premium_disc_shimur_pv	Column	Header1
total_bor_acc_pv	Column	Header1
total_bor_return_pv	Column	Header1
prem_savings_pv	Column	Header1
cashflow_pv_chetz	Column	Header1
nogt_annpv	Column	Header1
claims_lrc_q1_pv	Column	Header1
claims_lrc_q2_pv	Column	Header1
claims_lrc_q3_pv	Column	Header1
claims_lrc_q4_pv	Column	Header1
claims_lrc_yr2plus_pv	Column	Header1
expense_claims_lrc_q1_pv	Column	Header1
expense_claims_lrc_q2_pv	Column	Header1
expense_claims_lrc_q3_pv	Column	Header1

expense_claims_lrc_q4_pv	Column	Header1
expense_claims_lrc_yr2plus_pv	Column	Header1
claims_re_lrc_q1_pv	Column	Header1
claims_re_lrc_q2_pv	Column	Header1
claims_re_lrc_q3_pv	Column	Header1
claims_re_lrc_q4_pv	Column	Header1
claims_re_lrc_yr2plus_pv	Column	Header1
riskadj_gross_rel_q1_pv	Column	Header1
riskadj_gross_rel_q2_pv	Column	Header1
riskadj_gross_rel_q3_pv	Column	Header1
riskadj_gross_rel_q4_pv	Column	Header1
riskadj_gross_rel_total_pv	Column	Header1
riskadj_gross_rel_yr2plus_pv	Column	Header1
riskadj_re_rel_q1_pv	Column	Header1
riskadj_re_rel_q2_pv	Column	Header1
riskadj_re_rel_q3_pv	Column	Header1
riskadj_re_rel_q4_pv	Column	Header1
riskadj_re_rel_total_pv	Column	Header1
riskadj_re_rel_yr2plus_pv	Column	Header1

## 5 Code Manager

### 5.1 Model Tree

Model Object	Model Class	Base Model Class
life	life_cflow	
life term	sub1_cflow	
life trad	sub_2_cflow	
life accum	fund_cflow	
life acc_pup	fund_cflow	
life saving	fund_cflow	

Model Object	Model Class	Base Model Class
life saving_pup	fund_cflow	
life annuity	ann_cflow	
life rider	sub_array	

## 5.2 Project Variables

<No Project Variables Exist>

## 5.3 Model Classes

### 5.3.1 ann\_cflow

Description:

Help:

Base Model Class:

none

Model References

All

Read File:

Before Start Up

Modified On:

6/17/2025 11:03:22 AM (UTC+03:00)

Modified By:

CLAL-INS\arikt

#### 5.3.1.1 Variables

##### 5.3.1.1.1 age\_diff

Description:

Lookup value code variable wildcard

Help:

Modified On:

8/27/2019 4:00:59 PM (UTC+03:00)

Modified By:

CLAL-INS\ninab

Category:

Variable Type:

Floating Point Number

Default Value:

0

Length:

0

Number of Decimals:

1

Choice List:

Character Type:

Not Applicable

Valid Range From:

Valid Range To:

Table Format:

Default Row Numbers

Set Value in Input Manager:

All

Variable Sharing:

Not Shared

Category Order:

0

**5.3.1.1.2 ann\_fac\_dthben**

Description:	Look up value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.1.1.3 ann\_fac\_gtee\_value**

Description:	lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.1.1.4 ann\_fac\_joint**

Description:	Look up value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	
Variable Type:	Floating Point Number

Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.1.1.5 *ann\_fac\_no\_gtee***

Description:	Look up value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.1.1.6 *ann\_series\_prop***

Description:	Ann Series Prop
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers

Set Value in Input Manager:  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.1.1.7 ann\_series\_temp**

Description: Ann Series Temp  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category:  
Variable Type: Floating Point Number  
Default Value: 0  
Length: 1  
Number of Decimals: 1  
Choice List:  
Character Type: Not Applicable  
Valid Range From:  
Valid Range To:  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.1.1.8 annuity\_details\_temp\_tbl**

Description: Look up value code variable wildcard  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category:  
Variable Type: Floating Point Number  
Default Value: 0  
Length: 0  
Number of Decimals: 1  
Choice List:  
Character Type: Not Applicable  
Valid Range From:  
Valid Range To:  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.1.1.9 base\_year**

Description: Lookup value code variable wildcard



Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category:  
Variable Type: Floating Point Number  
Default Value: 0  
Length: 0  
Number of Decimals: 1  
Choice List:  
Character Type: Not Applicable  
Valid Range From:  
Valid Range To:  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.1.1.10      *freeinv\_res\_ann***

Description: lookup value code variable wildcard  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category:  
Variable Type: Floating Point Number  
Default Value: 0  
Length: 0  
Number of Decimals: 1  
Choice List:  
Character Type: Not Applicable  
Valid Range From:  
Valid Range To:  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.1.1.11      *freeinv\_res\_ann\_inpay***

Description: lookup value code variable wildcard  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category:  
Variable Type: Floating Point Number  
Default Value: 0  
Length: 0  
Number of Decimals: 1

Choice List:  
Character Type: Not Applicable  
Valid Range From:  
Valid Range To:  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.1.1.12      *freeinv\_res\_ann\_tarif***

Description: Use Int\_Tarif switch  
Help:  
Modified On: 11/13/2024 10:46:08 PM (UTC+02:00)  
Modified By: CLAL-INS\arikt  
Category:  
Variable Type: Character  
Default Value: 0  
Length: 10  
Number of Decimals: 1  
Choice List: 0  
Character Type: Standard  
Valid Range From:  
Valid Range To:  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.1.1.13      *fund\_t\_factor***

Description: Lookup value constant  
Help:  
Modified On: 5/18/2023 9:55:22 AM (UTC+03:00)  
Modified By: CLAL-INS\ahuvaa  
Category:  
Variable Type: Floating Point Number  
Default Value: 0  
Length: 0  
Number of Decimals: 1  
Choice List:  
Character Type: Not Applicable  
Valid Range From:  
Valid Range To:  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

**5.3.1.1.14 gtee\_ppn**

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.1.1.15 gtee\_prd**

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.1.1.16 int\_res\_ann**

Description:	lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	

Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.1.1.17      *int\_tarif***

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.1.1.18      *joint\_life\_ppn***

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	

Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.1.1.19**      ***life2\_ppn***

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.1.1.20**      ***mgt\_fee\_fixed***

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.1.1.21      *mgt\_fee\_max***

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.1.1.22      *mgt\_fee\_var***

Description:	lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.1.1.23      *no\_gtee\_ppn***

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	
Variable Type:	Floating Point Number

Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.1.1.24      *redn\_factor***

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.1.1.25      *res\_ann\_exp***

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers

Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.1.1.26      *res\_ann\_mort\_fac***

Description: lookup value code variable wildcard  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category:  
Variable Type: Floating Point Number  
Default Value: 0  
Length: 0  
Number of Decimals: 1  
Choice List:  
Character Type: Not Applicable  
Valid Range From:  
Valid Range To:  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.1.1.27      *retirement\_rate***

Description: Percentage retiring at current age  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category:  
Variable Type: Floating Point Number  
Default Value: 0  
Length: 0  
Number of Decimals: 1  
Choice List:  
Character Type: Not Applicable  
Valid Range From:  
Valid Range To:  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.1.1.28      *sel\_death\_rate\_col***

Description: Lookup value code variable wildcard



Help:  
Modified On: 7/29/2024 2:10:48 PM (UTC+03:00)  
Modified By: CLAL-INS\arikt  
Category:  
Variable Type: Floating Point Number  
Default Value: 0  
Length: 0  
Number of Decimals: 1  
Choice List:  
Character Type: Not Applicable  
Valid Range From:  
Valid Range To:  
Table Format: Default Row Numbers  
Set Value in Input Manager:  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.1.1.29      *temp\_annuity\_code***

Description: Temp Annuity Code  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category:  
Variable Type: Character  
Default Value: 0  
Length: 10  
Number of Decimals: 1  
Choice List:  
Character Type: Standard  
Valid Range From:  
Valid Range To:  
Table Format: Default Row Numbers  
Set Value in Input Manager:  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.1.1.30      *temp\_fund\_rates\_tbl***

Description: Lookup value constant  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category:  
Variable Type: Floating Point Number  
Default Value: 0  
Length: 0  
Number of Decimals: 1

Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.1.1.31      *zeroise\_ann\_def***

Description:	Zeroise reserve annuity deficiency
Help:	
Modified On:	6/17/2025 11:06:49 AM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	
Variable Type:	Character
Default Value:	N
Length:	10
Number of Decimals:	1
Choice List:	Y,N
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.1.1.32      *ann\_fac\_gtee***

Description:	policy ann factor- gteed
Help:	policy annuity factor of not guaranteed period
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Annuity
Variable Type:	Floating Point Number
Default Value:	0
Length:	6
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1000000
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.1.1.33      *ann\_fac\_joint\_temp***

Description:	policy ann factor-joint life
Help:	policy annuity factor of not guaranteed period
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Annuity
Variable Type:	Floating Point Number
Default Value:	0
Length:	6
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1000000
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.1.1.34      *ann\_fac\_no\_gtee\_temp***

Description:	policy ann factor-no gteed
Help:	policy annuity factor of not guaranteed period
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Annuity
Variable Type:	Floating Point Number
Default Value:	0
Length:	6
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1000000
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.1.1.35      *annuitization\_rate***

Description:	% of maturing policies taking annuity
Help:	Percentage of maturing units that are converted to an annuity.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab

Category:	Annuity
Variable Type:	Floating Point Number
Default Value:	0.1
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.1.1.36 *annuity\_code*

Description:	ann code to find policy ann factor
Help:	This rate should be net of annuity payment expenses, and should be rounded to the nearest basis point. It is used to lookup the annuity value.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Annuity
Variable Type:	Character
Default Value:	5_M_67
Length:	6
Number of Decimals:	0
Choice List:	
Character Type:	Standard
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.1.1.37 *annuity\_takeup\_new\_tag*

Description:	New tagmulim new annuity take up rate at maturity
Help:	This rate should be net of annuity payment expenses, and should be rounded to the nearest basis point. It is used to lookup the annuity value.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Annuity
Variable Type:	Floating Point Number

Default Value:	87.5
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.1.1.38      *annuity\_takeup\_new\_tag\_res***

Description:	New tagmulim new annuity take up rate for reserve calculation
Help:	This rate should be net of annuity payment expenses, and should be rounded to the nearest basis point. It is used to lookup the annuity value.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Annuity
Variable Type:	Floating Point Number
Default Value:	6
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.1.1.39      *annuity\_takeup\_old***

Description:	Old money annuity take up rate at maturity
Help:	This rate should be net of annuity payment expenses, and should be rounded to the nearest basis point. It is used to lookup the annuity value.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Annuity
Variable Type:	Floating Point Number
Default Value:	16.3
Length:	0

Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.1.1.40      *annuity\_takeup\_old\_res***

Description:	Old money annuity take up rate for reserve calculation
Help:	This rate should be net of annuity payment expenses, and should be rounded to the nearest basis point. It is used to lookup the annuity value.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Annuity
Variable Type:	Floating Point Number
Default Value:	6
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.1.1.41      *annuity\_takeup\_piz***

Description:	Pitzuim annuity take up rate at maturity
Help:	This rate should be net of annuity payment expenses, and should be rounded to the nearest basis point. It is used to lookup the annuity value.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Annuity
Variable Type:	Floating Point Number
Default Value:	17.9
Length:	0
Number of Decimals:	2
Choice List:	

Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.1.1.42      *annuity\_takeup\_piz\_res***

Description:	Pitzuim annuity take up rate for reserve calculation
Help:	This rate should be net of annuity payment expenses, and should be rounded to the nearest basis point. It is used to lookup the annuity value.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Annuity
Variable Type:	Floating Point Number
Default Value:	6
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.1.1.43      *annuity\_takeup\_prat***

Description:	Prat money annuity take up rate at maturity
Help:	This rate should be net of annuity payment expenses, and should be rounded to the nearest basis point. It is used to lookup the annuity value.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Annuity
Variable Type:	Floating Point Number
Default Value:	6
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0

Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.1.1.44      *annuity\_takeup\_prat\_res***

Description:	Prat money annuity take up rate for reserve calculation
Help:	This rate should be net of annuity payment expenses, and should be rounded to the nearest basis point. It is used to lookup the annuity value.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Annuity
Variable Type:	Floating Point Number
Default Value:	6
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.1.1.45      *annuity\_value\_res\_tbl***

Description:	Reserve Deficiency table of annuity values at maturity
Help:	Value of annuity of 100 per month, by calender year at maturity and sex_maturity-age_discount-rate.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Annuity
Variable Type:	Floating Point Number
Default Value:	0
Length:	1
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	0
Table Format:	Row Name (numeric)



Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.1.1.46      *takeup\_age***

Description: Take-up age for annuities  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Annuity  
Variable Type: Integer Number  
Default Value: 67  
Length: 0  
Number of Decimals: 0  
Choice List:  
Character Type: Not Applicable  
Valid Range From: 0  
Valid Range To: 100  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.1.1.47      *calyear***

Description: Calyear  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Balance Sheet|Reserves  
Variable Type: Integer Number  
Default Value: 0  
Length: 0  
Number of Decimals: 0  
Choice List:  
Character Type: Not Applicable  
Valid Range From:  
Valid Range To:  
Table Format: Default Row Numbers  
Set Value in Input Manager:  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.1.1.48      *col***

Description: Col

Help:  
 Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
 Modified By: CLAL-INS\ninab  
 Category: Balance Sheet|Reserves  
 Variable Type: Character  
 Default Value: F  
 Length: 10  
 Number of Decimals: 1  
 Choice List: F  
 Character Type: Standard  
 Valid Range From:  
 Valid Range To:  
 Table Format: Default Row Numbers  
 Set Value in Input Manager:  
 Variable Sharing: Not Shared  
 Category Order: 0

#### **5.3.1.1.49      *exp\_res***

Description: Exp Res  
 Help: expense assumption for reserves  
 Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
 Modified By: CLAL-INS\ninab  
 Category: Balance Sheet|Reserves  
 Variable Type: Floating Point Number  
 Default Value: 0.7  
 Length: 0  
 Number of Decimals: 2  
 Choice List:  
 Character Type: Not Applicable  
 Valid Range From: 0  
 Valid Range To: 20  
 Table Format: Default Row Numbers  
 Set Value in Input Manager:  
 Variable Sharing: Not Shared  
 Category Order: 0

#### **5.3.1.1.50      *freeinv\_rate\_res\_ann***

Description: Free Investment rate (%) for reserves  
 Help:  
 Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
 Modified By: CLAL-INS\ninab  
 Category: Balance Sheet|Reserves  
 Variable Type: Floating Point Number  
 Default Value: 4  
 Length: 0  
 Number of Decimals: 2

Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	10
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.1.1.51      *int\_rate\_res\_ann***

Description:	interest rate (%) for reserves
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Balance Sheet Reserves
Variable Type:	Floating Point Number
Default Value:	4
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	10
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.1.1.52      *mort\_fac\_res\_ann***

Description:	mortality factor (%) for reserves
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Balance Sheet Reserves
Variable Type:	Floating Point Number
Default Value:	4
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	10
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.1.1.53      *ann\_series***

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Economic assumptions
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.1.1.54      *ann\_tu\_newtag***

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Economic assumptions
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.1.1.55      *ann\_tu\_newtag\_res***

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Economic assumptions

Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.1.1.56      *ann\_tu\_old***

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Economic assumptions
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.1.1.57      *ann\_tu\_old\_res***

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Economic assumptions
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	

Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.1.1.58      *ann\_tu\_piz***

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Economic assumptions
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.1.1.59      *ann\_tu\_piz\_res***

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Economic assumptions
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.1.1.60      *ann\_tu\_prat***

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Economic assumptions
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.1.1.61      *ann\_tu\_prat\_res***

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Economic assumptions
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.1.1.62      *int\_tarif\_temp***

Description:	interest rate (%) for tarif
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Embedded Value Outgo Claims
Variable Type:	Floating Point Number

Default Value:	4
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	10
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.1.1.63      *mgt\_fee\_fixed\_max***

Description:	maximum fixed mgt fee
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Embedded Value Outgo Claims
Variable Type:	Floating Point Number
Default Value:	4
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	10
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.1.1.64      *mgt\_fee\_fixed\_temp***

Description:	fixed mgt fee
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Embedded Value Outgo Claims
Variable Type:	Floating Point Number
Default Value:	4
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	10
Table Format:	Default Row Numbers



Set Value in Input Manager:  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.1.1.65      *mgt\_fee\_variable***

Description: variable mgt fee  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Embedded Value|Outgo|Claims  
Variable Type: Floating Point Number  
Default Value: 4  
Length: 0  
Number of Decimals: 2  
Choice List:  
Character Type: Not Applicable  
Valid Range From: 0  
Valid Range To: 10  
Table Format: Default Row Numbers  
Set Value in Input Manager:  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.1.1.66      *antisel\_margin***

Description: Anti select % for annuity payment  
Help:  
Modified On: 6/30/2024 2:51:03 PM (UTC+03:00)  
Modified By: CLAL-INS\arikt  
Category: Mortality  
Variable Type: Floating Point Number  
Default Value: 0.03  
Length: 110  
Number of Decimals: 2  
Choice List:  
Character Type: Not Applicable  
Valid Range From: 0  
Valid Range To: 1  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.1.1.67      *death\_rates\_ann\_f\_1***

Description: Female Annuiants death rate table

Help:	Death-only rate table
Modified On:	7/26/2021 2:55:55 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Mortality
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	0
Table Format:	Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.1.1.68      *death\_rates\_ann\_f\_2***

Description:	Female Annuants death rate table
Help:	Death-only rate table
Modified On:	7/26/2021 2:56:54 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Mortality
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	0
Table Format:	Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.1.1.69      *death\_rates\_ann\_f\_b3\_2***

Description:	Female Annuants death rate table - B3 mortality
Help:	Death-only rate table
Modified On:	7/26/2021 2:56:54 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Mortality
Variable Type:	Floating Point Number
Default Value:	0
Length:	0

Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	0
Table Format:	Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.1.1.70      *death\_rates\_ann\_f\_res\_1***

Description:	Female Annuants death rate table
Help:	Death-only rate table
Modified On:	7/26/2021 2:56:14 PM (UTC+03:00)
Modified By:	CLAL-INSjoshm
Category:	Mortality
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	0
Table Format:	Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.1.1.71      *death\_rates\_ann\_f\_res\_2***

Description:	Female Annuants death rate table
Help:	Death-only rate table
Modified On:	7/26/2021 2:56:54 PM (UTC+03:00)
Modified By:	CLAL-INSjoshm
Category:	Mortality
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	0
Table Format:	Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared

Category Order: 0

#### **5.3.1.1.72      *death\_rates\_ann\_f\_res\_b3\_2***

Description: Female Annuants death rate table - B3 mortality  
Help: Death-only rate table  
Modified On: 7/26/2021 2:56:54 PM (UTC+03:00)  
Modified By: CLAL-INS\joshm  
Category: Mortality  
Variable Type: Floating Point Number  
Default Value: 0  
Length: 0  
Number of Decimals: 1  
Choice List:  
Character Type: Not Applicable  
Valid Range From: 0  
Valid Range To: 0  
Table Format: Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.1.1.73      *death\_rates\_ann\_f\_res\_b3\_tt***

Description: Female Annuants death rate table - B3 mortality  
Help: Death-only rate table  
Modified On: 7/26/2021 3:08:00 PM (UTC+03:00)  
Modified By: CLAL-INS\joshm  
Category: Mortality  
Variable Type: Floating Point Number  
Default Value: 0  
Length: 0  
Number of Decimals: 1  
Choice List:  
Character Type: Not Applicable  
Valid Range From: 0  
Valid Range To: 0  
Table Format: Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.1.1.74      *death\_rates\_ann\_f\_res\_tt***

Description: Female Annuants death rate table  
Help: Death-only rate table

Modified On:	7/26/2021 3:07:50 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Mortality
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	0
Table Format:	Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.1.1.75      *death\_rates\_ann\_m\_1***

Description:	Male Annuiants death rate table
Help:	Death-only rate table
Modified On:	7/26/2021 2:56:29 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Mortality
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	0
Table Format:	Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.1.1.76      *death\_rates\_ann\_m\_2***

Description:	Male Annuiants death rate table
Help:	Death-only rate table
Modified On:	7/26/2021 2:56:54 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Mortality
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	

Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	0
Table Format:	Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.1.1.77      *death\_rates\_ann\_m\_b3\_2***

Description:	Male Annuants death rate table - B3 mortality
Help:	Death-only rate table
Modified On:	7/26/2021 2:56:54 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Mortality
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	0
Table Format:	Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.1.1.78      *death\_rates\_ann\_m\_res\_1***

Description:	Male Annuants death rate table
Help:	Death-only rate table
Modified On:	7/26/2021 2:56:42 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Mortality
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	0
Table Format:	Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.1.1.79 death\_rates\_ann\_m\_res\_2**

Description:	Male Annuiants death rate table
Help:	Death-only rate table
Modified On:	7/26/2021 2:56:54 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Mortality
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	0
Table Format:	Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.1.1.80 death\_rates\_ann\_m\_res\_b3\_2**

Description:	Male Annuiants death rate table - B3 Mortality
Help:	Death-only rate table
Modified On:	7/26/2021 2:56:54 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Mortality
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	0
Table Format:	Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.1.1.81 death\_rates\_ann\_m\_res\_b3\_tt**

Description:	Male Annuiants death rate table - B3 Mortality
Help:	Death-only rate table
Modified On:	7/26/2021 3:08:19 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Mortality
Variable Type:	Floating Point Number

Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	0
Table Format:	Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.1.1.82      *death\_rates\_ann\_m\_res\_tt***

Description:	Male Annuiants death rate table
Help:	Death-only rate table
Modified On:	7/26/2021 3:08:13 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Mortality
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	0
Table Format:	Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.1.1.83      *dth\_rts\_m\_row\_key\_tt***

Description:	Lookup value code variable
Help:	
Modified On:	7/26/2021 2:45:33 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Mortality
Variable Type:	Integer Number
Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers



Set Value in Input Manager:  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.1.1.84      *gtee\_ppn\_temp***

Description: pmt prcntg of gteed annuity  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Mortality  
Variable Type: Floating Point Number  
Default Value: 60  
Length: 110  
Number of Decimals: 1  
Choice List:  
Character Type: Not Applicable  
Valid Range From: 0  
Valid Range To: 100  
Table Format: Default Row Numbers  
Set Value in Input Manager:  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.1.1.85      *joint\_life\_ppn\_temp***

Description: pmt prcntg of jointlife annuity  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Mortality  
Variable Type: Floating Point Number  
Default Value: 60  
Length: 110  
Number of Decimals: 1  
Choice List:  
Character Type: Not Applicable  
Valid Range From: 0  
Valid Range To: 100  
Table Format: Default Row Numbers  
Set Value in Input Manager:  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.1.1.86      *life2\_ppn\_temp***

Description: pmt prcntg of life2 forf joint life

Help:  
 Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
 Modified By: CLAL-INS\ninab  
 Category: Mortality  
 Variable Type: Floating Point Number  
 Default Value: 60  
 Length: 110  
 Number of Decimals: 1  
 Choice List:  
 Character Type: Not Applicable  
 Valid Range From: 0  
 Valid Range To: 100  
 Table Format: Default Row Numbers  
 Set Value in Input Manager:  
 Variable Sharing: Not Shared  
 Category Order: 0

#### **5.3.1.1.87      *no\_gtee\_ppn\_temp***

Description: pmt prcntg of not gteed annuity  
 Help:  
 Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
 Modified By: CLAL-INS\ninab  
 Category: Mortality  
 Variable Type: Floating Point Number  
 Default Value: 60  
 Length: 110  
 Number of Decimals: 1  
 Choice List:  
 Character Type: Not Applicable  
 Valid Range From: 0  
 Valid Range To: 100  
 Table Format: Default Row Numbers  
 Set Value in Input Manager:  
 Variable Sharing: Not Shared  
 Category Order: 0

#### **5.3.1.1.88      *qx\_sd\_comp***

Description: qx composition standard deviation  
 Help: This flat rate adjustment is added directly to the qx. Array items relate to policy years: year 1=array item 1. Array item 0 is not used.  
 Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
 Modified By: CLAL-INS\ninab  
 Category: Mortality  
 Variable Type: Floating Point Number  
 Default Value: 0

Length:	110
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	-1
Valid Range To:	10
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.1.1.89      *qx\_sd\_comp\_res***

Description:	qx composition standard deviation for reserves
Help:	This flat rate adjustment is added directly to the qx. Array items relate to policy years: year 1=array item 1. Array item 0 is not used.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Mortality
Variable Type:	Floating Point Number
Default Value:	1
Length:	110
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	-1
Valid Range To:	10
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.1.1.90      *qx\_sd\_random***

Description:	qx random standard deviation
Help:	This flat rate adjustment is added directly to the qx. Array items relate to policy years: year 1=array item 1. Array item 0 is not used.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Mortality
Variable Type:	Floating Point Number
Default Value:	0
Length:	110
Number of Decimals:	7
Choice List:	
Character Type:	Not Applicable
Valid Range From:	-1

Valid Range To:	2
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.1.1.91 ***qx\_sd\_random\_res***

Description:	qx random standard deviation for reserves
Help:	This flat rate adjustment is added directly to the qx. Array items relate to policy years: year 1=array item 1. Array item 0 is not used.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Mortality
Variable Type:	Floating Point Number
Default Value:	2
Length:	110
Number of Decimals:	7
Choice List:	
Character Type:	Not Applicable
Valid Range From:	-1
Valid Range To:	2
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.1.1.92 ***sel\_ret\_qx\_im\_dth\_1***

Description:	
Help:	Death-only rate table
Modified On:	7/23/2024 3:17:56 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Mortality
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	0
Table Format:	Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.1.1.93      *sel\_ret\_qx\_im\_dth\_2***

Description:	Death-only rate table
Help:	
Modified On:	7/23/2024 3:37:02 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Mortality
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	0
Table Format:	Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.1.1.94      *benefit\_term***

Description:	Policy benefit term (months)
Help:	The original policy (benefit) term in integral months calculated from the issue date to the date of policy expiry.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Integer Number
Default Value:	120
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	1
Valid Range To:	1200
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.1.1.95      *joint\_life\_status***

Description:	Joint life status
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details

Variable Type:	Character
Default Value:	Single
Length:	10
Number of Decimals:	0
Choice List:	Single,Joint Life
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.1.1.96**      *maturity\_period\_ann*

Description:	Period t in which annuity ends up
Help:	Calculated in startup column.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Integer Number
Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	600
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.1.1.97**      *maturity\_period\_w*

Description:	Period t in which policy matures
Help:	Calculated in startup column.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Integer Number
Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	600

Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.1.1.98 ***gteed\_term***

Description:	Guaranteed term (in months)
Help:	Original guaranteed term (in months from policy commencement)
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details Annuity
Variable Type:	Integer Number
Default Value:	240
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	240
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.1.1.99 ***age\_ann\_start\_1***

Description:	Age at annuity start- life 1
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details Details Life 1
Variable Type:	Integer Number
Default Value:	67
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.1.1.100      age\_ann\_start\_2**

Description:	Age at annuity start- life 2
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details Details Life 1
Variable Type:	Integer Number
Default Value:	65
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.1.1.101      age\_diff\_temp**

Description:	Age diff. between life 1 & 2
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details Details Life 1
Variable Type:	Integer Number
Default Value:	2
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.1.1.102      sex1**

Description:	Sex of life 1
Help:	M = Male F = Female
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details Details Life 1



Variable Type:	Character
Default Value:	M
Length:	1
Number of Decimals:	0
Choice List:	M,F
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.1.1.103 **sex2**

Description:	Sex of life 2
Help:	M = Male F = Female
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details Details Life 2
Variable Type:	Character
Default Value:	F
Length:	1
Number of Decimals:	0
Choice List:	M,F
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.1.1.104 **dump\_vars**

Description:	Output all variables to logfile?
Help:	If Y, the program will output all variables to the log stream after the startup has been executed in each model. NB Should only be set to Y for testing purposes, as will create HUGE amounts of output in the log file when run with a large policy file.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Setup
Variable Type:	Character
Default Value:	N
Length:	1

Number of Decimals:	0
Choice List:	Y,N
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.1.1.105**      ***commence\_period\_w***

Description:	Period t in which policy commences
Help:	Calculated in startup column.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Setup Initialise
Variable Type:	Integer Number
Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	-600
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.1.1.106**      ***yob\_1***

Description:	year of birth of Life 1
Help:	Highest age in mortality table for life 1. Internal logic variable calculated in matured formula.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Setup Initialise
Variable Type:	Integer Number
Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	

Variable Sharing:	Not Shared
Category Order:	0

### 5.3.1.1.107 *yob\_2*

Description:	year of birth of Life 2
Help:	Highest age in mortality table for life 1. Internal logic variable calculated in matured formula.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Setup Initialise
Variable Type:	Integer Number
Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.1.1.108 *death\_ben*

Description:	Death benefit payable?
Help:	The death benefit, if payable, would be the accumulation of premiums. Y = death benefit payable during the deferment period N = no death benefit payable
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Setup Run Control
Variable Type:	Character
Default Value:	N
Length:	1
Number of Decimals:	0
Choice List:	Y,N
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.1.1.109      *death\_ben\_curr***

Description:	current death benefit at valuation date
Help:	The death benefit, if payable, would be the accumulation of premiums. Y = death benefit payable during the deferment period N = no death benefit payable
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Setup Run Control
Variable Type:	Floating Point Number
Default Value:	0
Length:	1
Number of Decimals:	0
Choice List:	Y,N
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	2500
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.1.1.110      *year\_ann\_start***

Description:	year at annuity start
Help:	Valuation occurs at end of valn_month in valn_year
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Setup Run Control
Variable Type:	Integer Number
Default Value:	2010
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	1980
Valid Range To:	2100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.1.2 Columns**

**5.3.1.2.1 cashflow\_b\_post\_ret**

Description:	Cashflow B Post Ret
Help:	
Modified On:	8/4/2021 3:22:20 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	
Column Header:	cashflow_b_post_ret
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.1.2.2 res\_ann\_deficiency**

Description:	Res Ann Deficiency
Help:	
Modified On:	6/17/2025 11:11:59 AM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	
Column Header:	res_ann_deficiency
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.1.2.3 claims\_annuity\_nogt\_pv**

Description:	claims_annuity_nogt_pv
Help:	
Modified On:	8/4/2021 3:22:41 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Annuity
Column Header:	claims_annuity_nogt_pv
Combine Groups By:	Sum Both
Combine Periods:	Last

Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.1.2.4 *claims\_annuity\_pv***

Description:	Claims Annuity Pv
Help:	
Modified On:	8/4/2021 3:22:56 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Annuity
Column Header:	claims_annuity_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.1.2.5 *int\_rate\_annuity\_reserve***

Description:	Interest Rate for Annuity Reserve
Help:	
Modified On:	8/4/2021 3:30:26 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Balance Sheet Reserves
Column Header:	int_rate_annuity_reserve
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

### 5.3.1.2.6 res\_basic\_jl\_1

Description:	Res Basic for joint life - first life
Help:	
Modified On:	8/4/2021 3:37:24 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Balance Sheet Reserves
Column Header:	res_basic_jl_1
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

### 5.3.1.2.7 res\_payment\_1

Description:	Reserve annuity payment for main life
Help:	
Modified On:	1/11/2023 11:29:01 AM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Balance Sheet Reserves
Column Header:	res_payment_1
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

### 5.3.1.2.8 res\_payment\_2

Description:	Reserve annuity payment for secondary life
Help:	
Modified On:	1/11/2023 11:29:05 AM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Balance Sheet Reserves
Column Header:	res_payment_2
Combine Groups By:	Sum Both
Combine Periods:	Last

Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### 5.3.1.2.9 *res\_payment\_pv\_1*

Description:	Present value for reserve payments for primary life
Help:	
Modified On:	8/4/2021 3:38:13 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Balance Sheet Reserves
Column Header:	res_payment_pv_1
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### 5.3.1.2.10 *res\_payment\_pv\_2*

Description:	Present value for reserve payments for secondary life
Help:	
Modified On:	8/4/2021 3:38:21 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Balance Sheet Reserves
Column Header:	res_payment_pv_2
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False



**5.3.1.2.11      *ann\_takeup\_rate***

Description:	annuitization rate
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Balance Sheet Reserves Annuity Deficiency
Column Header:	ann_takeup_rate
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.1.2.12      *ann\_certain\_fac***

Description:	Annuity certain factor - an
Help:	
Modified On:	8/12/2021 4:17:11 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Balance Sheet Reserves Basic Reserve
Column Header:	ann_certain_fac
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.1.2.13      *ann\_defer\_fac***

Description:	Annuity factor for life 1- ax deferred of gteed
Help:	
Modified On:	8/15/2021 3:44:46 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Balance Sheet Reserves Basic Reserve
Column Header:	ann_defer_fac
Combine Groups By:	Average Both

Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.1.2.14      *annuity\_fac\_1***

Description:	Annuity factor for life 1- ax
Help:	
Modified On:	8/15/2021 3:44:25 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Balance Sheet Reserves Basic Reserve
Column Header:	annuity_fac_1
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.1.2.15      *assurance\_fac\_1***

Description:	Assurance factor for life 1- Ax
Help:	
Modified On:	8/9/2021 10:40:24 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Balance Sheet Reserves Basic Reserve
Column Header:	assurance_fac_1
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.1.2.16      *bonus\_index***

Description:	accumulate bonus rate for ann payment
Help:	
Modified On:	2/26/2025 1:59:09 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Balance Sheet Reserves Basic Reserve
Column Header:	bonus_index
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.1.2.17      *res\_basic\_dth***

Description:	Basic reserve - death benefit
Help:	
Modified On:	8/9/2021 10:46:00 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Balance Sheet Reserves Basic Reserve
Column Header:	res_basic_dth
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.1.2.18      *res\_basic\_gt***

Description:	Basic reserve - guaranteed
Help:	
Modified On:	10/3/2021 1:48:40 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Balance Sheet Reserves Basic Reserve
Column Header:	res_basic_gt
Combine Groups By:	Sum Both
Combine Periods:	Last

Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.1.2.19      *res\_basic\_gt\_su***

Description:	Basic reserve - guaranteed - for Service units
Help:	
Modified On:	5/29/2025 12:12:00 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Balance Sheet Reserves Basic Reserve
Column Header:	res_basic_gt_su
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.1.2.20      *res\_basic\_jl***

Description:	Basic reserve - jointlife
Help:	
Modified On:	8/9/2021 10:46:17 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Balance Sheet Reserves Basic Reserve
Column Header:	res_basic_jl
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.1.2.21      *res\_basic\_jl\_2***

Description:	Res basic for joint life - second life
Help:	
Modified On:	8/9/2021 10:46:27 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Balance Sheet Reserves Basic Reserve
Column Header:	res_basic_jl_2
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.1.2.22      *res\_basic\_nogt***

Description:	Basic reserve - not guaranteed
Help:	
Modified On:	8/9/2021 10:46:35 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Balance Sheet Reserves Basic Reserve
Column Header:	res_basic_nogt
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.1.2.23      *reserve\_basic***

Description:	Total basic reserve -
Help:	
Modified On:	5/16/2023 2:17:38 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Balance Sheet Reserves Basic Reserve
Column Header:	reserve_basic
Combine Groups By:	Sum Both
Combine Periods:	Last

Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.1.2.24      *bonus\_index\_dth***

Description:	Bonus Index for death benefits
Help:	
Modified On:	8/4/2021 2:47:36 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows
Column Header:	bonus_index_dth
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.1.2.25      *bonus\_index\_gteed***

Description:	Bonus Index for guaranteed annuities
Help:	
Modified On:	8/4/2021 2:47:50 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows
Column Header:	bonus_index_gteed
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.1.2.26      *bonus\_index\_jl\_1***

Description:	Bonus Index for joint life annuities - first life
Help:	
Modified On:	8/4/2021 2:48:07 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows
Column Header:	bonus_index_jl_1
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.1.2.27      *bonus\_index\_jl\_2***

Description:	Bonus Index for joint life annuities - first life
Help:	
Modified On:	8/4/2021 2:48:22 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows
Column Header:	bonus_index_jl_2
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.1.2.28      *bonus\_index\_no\_gtee***

Description:	Bonus Index for non-guaranteed annuities
Help:	
Modified On:	8/4/2021 2:48:57 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows
Column Header:	bonus_index_no_gtee
Combine Groups By:	Sum Both
Combine Periods:	Last

Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.1.2.29      *bor\_acc\_dth***

Description:	Management fees owing (bor) for death benefits
Help:	
Modified On:	8/4/2021 2:49:09 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows
Column Header:	bor_acc_dth
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.1.2.30      *bor\_acc\_gtd***

Description:	Management fees owing (bor) for guaranteed policies
Help:	
Modified On:	8/4/2021 2:53:50 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows
Column Header:	bor_acc_gtd
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False



**5.3.1.2.31      *bor\_acc\_jl1***

Description:	Management fees owing (bor) for joint-life policies (first life)
Help:	
Modified On:	8/4/2021 2:53:58 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows
Column Header:	bor_acc_jl1
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.1.2.32      *bor\_acc\_jl2***

Description:	Management fees owing (bor) for joint-life policies (second life)
Help:	
Modified On:	8/4/2021 2:54:55 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows
Column Header:	bor_acc_jl2
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.1.2.33      *bor\_acc\_notg***

Description:	Management fees owing (bor) for non-guaranteed policies
Help:	
Modified On:	8/4/2021 3:21:05 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm

Category:	Cashflows
Column Header:	bor_acc_notg
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.1.2.34      *bor\_return\_dth***

Description:	Owed management fees (bor) returned for death benefits
Help:	
Modified On:	9/25/2024 1:25:38 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Cashflows
Column Header:	bor_return_dth
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.1.2.35      *bor\_return\_gtd***

Description:	Owed management fees (bor) returned for guaranteed annuities
Help:	
Modified On:	9/25/2024 12:44:13 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Cashflows
Column Header:	bor_return_gtd
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow

Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.1.2.36      *bor\_return\_jl1***

Description:	Owed management fees (bor) returned for joint-life annuities (first life)
Help:	
Modified On:	9/25/2024 1:12:50 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Cashflows
Column Header:	bor_return_jl1
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.1.2.37      *bor\_return\_jl2***

Description:	Owed management fees (bor) returned for joint-life annuities (second life)
Help:	
Modified On:	9/25/2024 1:23:44 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Cashflows
Column Header:	bor_return_jl2
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.1.2.38      *bor\_return\_nogt***

Description:	Owed management fees (bor) returned for non-guaranteed annuities
Help:	
Modified On:	9/25/2024 1:02:06 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Cashflows
Column Header:	bor_return_nogt
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.1.2.39      *cashflow\_pv***

Description:	Cashflow Pv
Help:	
Modified On:	8/4/2021 3:22:30 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows
Column Header:	cashflow_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.1.2.40      *cashflow\_pv\_chetz***

Description:	Cashflow Pv - chetz ribit
Help:	
Modified On:	7/14/2024 2:11:41 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Cashflows
Column Header:	cashflow_pv_chetz
Combine Groups By:	Sum Both

Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.1.2.41**      ***cashflow\_pv\_e***

Description:	Cashflow Pv discountned EOP
Help:	
Modified On:	7/19/2022 3:19:28 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Cashflows
Column Header:	cashflow_pv_e
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.1.2.42**      ***cashflow\_pv\_ifrs***

Description:	Cashflow Pv - IFRS
Help:	
Modified On:	7/14/2024 2:10:25 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Cashflows
Column Header:	cashflow_pv_ifrs
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.1.2.43**      ***cashflow\_pv\_res***

Description:	Cashflow Pv - res
Help:	
Modified On:	7/14/2024 2:09:10 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Cashflows
Column Header:	cashflow_pv_res
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.1.2.44**      ***har\_acc\_dth***

Description:	Variable management fees paid to that point in year (har) for death benefits
Help:	
Modified On:	8/4/2021 3:28:27 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows
Column Header:	har_acc_dth
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.1.2.45**      ***har\_acc\_jl1***

Description:	Variable management fees paid to that point in year (har) for joint-life annuities (first life)
Help:	
Modified On:	8/4/2021 3:28:37 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows
Column Header:	har_acc_jl1
Combine Groups By:	Sum Both

Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.1.2.46**      *har\_acc\_jl2*

Description:	Variable management fees paid to that point in year (har) for joint-life annuities (second life)
Help:	
Modified On:	8/4/2021 3:28:44 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows
Column Header:	har_acc_jl2
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.1.2.47**      *har\_acc\_nogt*

Description:	Variable management fees paid to that point in year (har) for non-guaranteed annuities
Help:	
Modified On:	8/4/2021 3:28:51 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows
Column Header:	har_acc_nogt
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False

Virtual: False

#### **5.3.1.2.48      *har\_accum\_gtd***

Description: Variable management fees paid to that point in year (har) for guaranteed annuities

Help:

Modified On: 8/4/2021 3:28:59 PM (UTC+03:00)

Modified By: CLAL-INS\joshm

Category: Cashflows

Column Header: har\_accum\_gtd

Combine Groups By: Sum Both

Combine Periods: Last

Default sliding Size: -1

Discount Timing: End

Discount Use: Yes

Rate Use: Cash Flow

Rebase Type: Previous

Retain Value: Yes

Override: False

Virtual: False

#### **5.3.1.2.49      *har\_return\_dth***

Description: Management Fees Variable Repaid for death benefits

Help:

Modified On: 8/4/2021 3:29:08 PM (UTC+03:00)

Modified By: CLAL-INS\joshm

Category: Cashflows

Column Header: har\_return\_dth

Combine Groups By: Sum Both

Combine Periods: Last

Default sliding Size: -1

Discount Timing: End

Discount Use: Yes

Rate Use: Cash Flow

Rebase Type: Previous

Retain Value: Yes

Override: False

Virtual: False

#### **5.3.1.2.50      *har\_return\_gtd***

Description: Management Fees Variable Repaid for guaranteed

Help:

Modified On: 8/4/2021 3:29:14 PM (UTC+03:00)



Modified By:	CLAL-INS\joshm
Category:	Cashflows
Column Header:	har_return_gtd
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.1.2.51      *har\_return\_jl1***

Description:	Management Fees Variable Repaid for joint life (first life)
Help:	
Modified On:	8/4/2021 3:29:20 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows
Column Header:	har_return_jl1
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.1.2.52      *har\_return\_jl2***

Description:	Management Fees Variable Repaid for joint life (second life)
Help:	
Modified On:	8/4/2021 3:29:30 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows
Column Header:	har_return_jl2
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes

Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.1.2.53**      *har\_return\_nogt*

Description:	Management Fees Variable Repaid for non-guaranteed
Help:	
Modified On:	8/4/2021 3:29:37 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows
Column Header:	har_return_nogt
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.1.2.54**      *mgt\_var\_no\_bor\_dth*

Description:	calculate management fees on side for same results in management_fees_variable and bor_return
Help:	
Modified On:	9/25/2024 1:25:01 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Cashflows
Column Header:	mgt_var_no_bor_dth
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.1.2.55**      ***mgt\_var\_no\_bor\_gtd***

Description:	calculate management fees on side for same results in management_fees_variable and bor_return
Help:	
Modified On:	9/25/2024 12:44:13 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Cashflows
Column Header:	mgt_var_no_bor_gtd
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.1.2.56**      ***mgt\_var\_no\_bor\_jl1***

Description:	calculate management fees on side for same results in management_fees_variable and bor_return
Help:	
Modified On:	9/25/2024 1:12:10 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Cashflows
Column Header:	mgt_var_no_bor_jl1
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.1.2.57**      ***mgt\_var\_no\_bor\_jl2***

Description:	calculate management fees on side for same results in management_fees_variable and bor_return
Help:	
Modified On:	9/25/2024 1:22:34 PM (UTC+03:00)

Modified By:	CLAL-INS\arikt
Category:	Cashflows
Column Header:	mgt_var_no_bor_jl2
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

### 5.3.1.2.58 *mgt\_var\_no\_bor\_nogt*

Description:	calculate management fees on side for same results in management_fees_variable and bor_return
Help:	
Modified On:	9/25/2024 1:00:22 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Cashflows
Column Header:	mgt_var_no_bor_nogt
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

### 5.3.1.2.59 *net\_interest\_rate*

Description:	Interest rate net of fixed management fees
Help:	
Modified On:	2/26/2025 1:59:36 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Cashflows
Column Header:	net_interest_rate
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes

Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.1.2.60      *mgt\_fee\_fixed\_pv***

Description:	Management Fees Fixed Pv
Help:	
Modified On:	8/4/2021 3:33:44 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows Income
Column Header:	mgt_fee_fixed_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.1.2.61      *mgt\_fee\_var\_pv***

Description:	Management Fees Variable Pv
Help:	
Modified On:	8/4/2021 3:34:55 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows Income
Column Header:	mgt_fee_var_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.1.2.62      *investment\_income***

Description:	Investment Income
--------------	-------------------

Help:  
 Modified On: 7/19/2024 9:45:58 AM (UTC+03:00)  
 Modified By: CLAL-INS\arikt  
 Category: Cashflows|Income|Investment Income  
 Column Header: investment\_income  
 Combine Groups By: Sum Both  
 Combine Periods: Sum  
 Default sliding Size: -1  
 Discount Timing: End  
 Discount Use: Yes  
 Rate Use: Cash Flow  
 Rebase Type: Previous  
 Retain Value: Yes  
 Override: False  
 Virtual: False

#### **5.3.1.2.63**      *investment\_income\_chetz*

Description: Investment Income form chetz  
 Help:  
 Modified On: 7/8/2024 4:54:31 PM (UTC+03:00)  
 Modified By: CLAL-INS\arikt  
 Category: Cashflows|Income|Investment Income  
 Column Header: investment\_income\_chetz  
 Combine Groups By: Sum Both  
 Combine Periods: Sum  
 Default sliding Size: -1  
 Discount Timing: End  
 Discount Use: Yes  
 Rate Use: Cash Flow  
 Rebase Type: Previous  
 Retain Value: Yes  
 Override: False  
 Virtual: False

#### **5.3.1.2.64**      *reserve\_bonus\_units\_e\_0*

Description: Reserve Bonus Units E 0  
 Help:  
 Modified On: 10/26/2021 10:29:53 AM (UTC+03:00)  
 Modified By: CLAL-INS\joshm  
 Category: Cashflows|Income|Investment Income  
 Column Header: reserve\_bonus\_units\_e\_0  
 Combine Groups By: Sum Both  
 Combine Periods: Last  
 Default sliding Size: -1  
 Discount Timing: End  
 Discount Use: Yes

Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.1.2.65      *reserve\_bonus\_units\_e\_t***

Description:	Reserve Bonus Units E T
Help:	
Modified On:	10/26/2021 10:29:47 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows Income Investment Income
Column Header:	reserve_bonus_units_e_t
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.1.2.66      *investment\_income\_chetz\_pv***

Description:	PV of investment income from chetz
Help:	PV of investment income at the "discount rate", at the beginning of the period. The discount rate in the experience model is the embedded value discount rate
Modified On:	6/19/2023 1:33:17 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Cashflows Income Premium
Column Header:	investment_income_chetz_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	0
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.1.2.67 investment\_income\_pv**

Description:	PV of investment income
Help:	PV of investment income at the "discount rate", at the beginning of the period. The discount rate in the experience model is the embedded value discount rate
Modified On:	8/9/2021 10:44:35 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows Income Premium
Column Header:	investment_income_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	0
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.1.2.68 outgo\_b\_post\_ret\_pv**

Description:	Outgo B Post Ret Pv
Help:	
Modified On:	8/4/2021 3:35:14 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows Outgo
Column Header:	outgo_b_post_ret_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.1.2.69 expense\_investment\_pv**

Description:	Expense Investment Post Ret Pv
Help:	
Modified On:	8/4/2021 3:27:56 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows Outgo Expenses
Column Header:	expense_investment_pv



Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.1.2.70**      ***expense\_ren\_perc\_post\_ret\_pv***

Description:	Expense Ren Perc Post Ret Pv
Help:	
Modified On:	8/4/2021 3:28:14 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows Outgo Expenses
Column Header:	expense_ren_perc_post_ret_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.1.2.71**      ***expense\_investment\_post\_ret***

Description:	Investment expenses post retirement
Help:	
Modified On:	4/4/2023 9:51:22 AM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Cashflows Outgo Expenses Renewal
Column Header:	expense_investment_post_ret
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.1.2.72**      ***expense\_ren\_perc\_post\_ret***

Description:	Variable renewal expenses post retirement
Help:	
Modified On:	9/18/2022 8:51:05 AM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Cashflows Outgo Expenses Renewal
Column Header:	expense_ren_perc_post_ret
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.1.2.73**      ***reserve\_increase***

Description:	Increase in total reserve
Help:	
Modified On:	8/9/2021 10:47:01 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows Outgo Reserve Increase
Column Header:	reserve_increase
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.1.2.74**      ***sel\_death\_rate\_1***

Description:	Select death rate
Help:	
Modified On:	7/29/2024 2:10:48 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Cashflows Outgo Reserve Increase
Column Header:	sel_death_rate_1
Combine Groups By:	Sum Both

Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### 5.3.1.2.75 ***sel\_death\_rate\_2***

Description:	Select death rate
Help:	
Modified On:	7/29/2024 2:10:48 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Cashflows Outgo Reserve Increase
Column Header:	sel_death_rate_2
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### 5.3.1.2.76 ***profit\_book\_pv***

Description:	Profit Book Pv
Help:	
Modified On:	8/4/2021 3:35:51 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows Profit
Column Header:	profit_book_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.1.2.77**      ***profit\_book\_vif\_post\_ret***

Description:	Before Tax Profit (cashflow basis for MCEV) without DAC (VIF Profit) after retirement
Help:	After Tax Profit (cashflow basis for MCEV) without DAC (VIF Profit)
Modified On:	7/19/2024 9:43:00 AM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Cashflows Profit
Column Header:	profit_book_vif_post_ret
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.1.2.78**      ***profit\_net\_vif\_post\_ret\_pv***

Description:	Profit Net Vif Post Ret Pv
Help:	
Modified On:	8/28/2022 4:41:50 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Cashflows Profit
Column Header:	profit_net_vif_post_ret_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.1.2.79**      ***surv\_gteed***

Description:	Propn gteed surv'ng at end t
Help:	Probability of annuity with gteed term surviving from outset.
Modified On:	8/4/2021 3:40:48 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Decrements Death Joint Life
Column Header:	surv_gteed

Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.1.2.80**      ***surv\_jl\_lastsurv***

Description:	Propn JL last surv'ng at end t
Help:	Probability of joint life last survivor surviving from outset.
Modified On:	8/9/2021 10:47:20 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Decrements Death Joint Life
Column Header:	surv_jl_lastsurv
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.1.2.81**      ***surv\_per\_gteed***

Description:	Marginal propn gteed surviving at end t
Help:	Probability of annuity with gteed term surviving from outset.
Modified On:	8/4/2021 3:41:23 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Decrements Death Joint Life
Column Header:	surv_per_gteed
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes

Override: False  
Virtual: False

#### **5.3.1.2.82      *death\_rate\_1***

Description: monthly death rate for life 1  
Help: Annual mortality rate for life 1 adjusted for selection, sex, smoking status and mortality improvement.  
  
Modified On: 7/22/2024 12:14:22 PM (UTC+03:00)  
Modified By: CLAL-INS\arikt  
Category: Decrements|Death|Life 1  
Column Header: death\_rate\_1  
Combine Groups By: Average Both  
Combine Periods: Last  
Default sliding Size: -1  
Discount Timing: Middle  
Discount Use: No  
Rate Use: Cash Flow  
Rebase Type: Current  
Retain Value: Yes  
Override: False  
Virtual: False

#### **5.3.1.2.83      *death\_rate\_res\_1***

Description: monthly death rate for life 1 for reserve  
Help: Annual mortality rate for life 1 adjusted for selection, sex, smoking status and mortality improvement.  
  
Modified On: 7/22/2024 12:12:27 PM (UTC+03:00)  
Modified By: CLAL-INS\arikt  
Category: Decrements|Death|Life 1  
Column Header: death\_rate\_res\_1  
Combine Groups By: Average Both  
Combine Periods: Last  
Default sliding Size: -1  
Discount Timing: Middle  
Discount Use: No  
Rate Use: Cash Flow  
Rebase Type: Current  
Retain Value: Yes  
Override: False  
Virtual: False

#### **5.3.1.2.84      *surv\_1***

Description: Propn life 1's surv'ng at end time t

Help:	Proportion of life 1 surviving from outset. This is based on the death rate for life 1, and is adjusted.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Decrements Death Life 1
Column Header:	surv_1
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.1.2.85      *surv\_1\_res***

Description:	Propn life 1's surv'ng at end time t for reserve
Help:	Proportion of life 1 surviving from outset. This is based on the death rate for life 1, and is adjusted.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Decrements Death Life 1
Column Header:	surv_1_res
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.1.2.86      *surv\_per\_1***

Description:	Prob of life 1 surv'n for period
Help:	Probability of survival of period for life 1. Uses adjusted survival of life 1 (surv_1).
Modified On:	8/9/2021 10:47:27 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Decrements Death Life 1
Column Header:	surv_per_1
Combine Groups By:	Average Both

Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.1.2.87      *death\_rate\_2***

Description:	monthly death rate for life 2
Help:	Annual mortality for life 2 adjusted for selection, sex, smoking status and mortality improvement.
Modified On:	7/23/2024 3:44:21 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Decrement Death Life 2
Column Header:	death_rate_2
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	Middle
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.1.2.88      *death\_rate\_2\_b3***

Description:	monthly death rate for life 2 - B3 mortality
Help:	Annual mortality for life 2 adjusted for selection, sex, smoking status and mortality improvement.
Modified On:	1/11/2023 11:28:55 AM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Decrement Death Life 2
Column Header:	death_rate_2_b3
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	Middle
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes



Override: False  
Virtual: False

#### **5.3.1.2.89      *death\_rate\_res\_2***

Description: monthly death rate for life 2 for reserve  
Help: Annual mortality for life 2 adjusted for selection, sex, smoking status and mortality improvement.

Modified On: 7/23/2024 3:45:08 PM (UTC+03:00)  
Modified By: CLAL-INS\arikt  
Category: Decrements|Death|Life 2  
Column Header: death\_rate\_res\_2  
Combine Groups By: Average Both  
Combine Periods: Last  
Default sliding Size: -1  
Discount Timing: Middle  
Discount Use: No  
Rate Use: Cash Flow  
Rebase Type: Current  
Retain Value: Yes  
Override: False  
Virtual: False

#### **5.3.1.2.90      *death\_rate\_res\_2\_b3***

Description: monthly death rate for life 2 for reserve - B3 mortality  
Help: Annual mortality for life 2 adjusted for selection, sex, smoking status and mortality improvement.

Modified On: 1/11/2023 11:29:18 AM (UTC+02:00)  
Modified By: CLAL-INS\ahuvaa  
Category: Decrements|Death|Life 2  
Column Header: death\_rate\_res\_3\_b3  
Combine Groups By: Average Both  
Combine Periods: Last  
Default sliding Size: -1  
Discount Timing: Middle  
Discount Use: No  
Rate Use: Cash Flow  
Rebase Type: Current  
Retain Value: Yes  
Override: False  
Virtual: False

**5.3.1.2.91      *surv\_2***

Description:	Propn life 2's surv'ng at end t
Help:	Proportion of life 2's surviving from outset. Only calculated when joint life status is set to "Joint Life".
Modified On:	8/4/2021 3:40:10 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Decrements Death Life 2
Column Header:	surv_2
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.1.2.92      *surv\_2\_joint\_life***

Description:	Propn life 2's receiving joint life payment at end t
Help:	Proportion of life 2's surviving from outset. Only calculated when joint life status is set to "Joint Life".
Modified On:	8/4/2021 3:40:22 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Decrements Death Life 2
Column Header:	surv_2_joint_life
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.1.2.93      *surv\_2\_joint\_life\_res***

Description:	Propn life 2's receiving joint life payment at end t - reserving mortality
--------------	--

Help:	Proportion of life 2's surviving from outset. Only calculated when joint life status is set to "Joint Life".
Modified On:	8/4/2021 3:40:33 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Decrement Death Life 2
Column Header:	surv_2_joint_life_res
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### 5.3.1.2.94 **surv\_2\_res**

Description:	Propn life 2's surv'ng at end t - reserving mortality
Help:	Proportion of life 2's surviving from outset. Only calculated when joint life status is set to "Joint Life".
Modified On:	8/4/2021 3:40:44 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Decrement Death Life 2
Column Header:	surv_2_res
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### 5.3.1.2.95 **surv\_per\_2**

Description:	Prob of life 2 surv'ng for period
Help:	Probability of survival of period for life 2 .
Modified On:	8/9/2021 10:47:34 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Decrement Death Life 2
Column Header:	surv_per_2
Combine Groups By:	Average Both

Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.1.2.96**      ***dth\_ben\_if\_b***

Description:	death benefit inforce in the beginning of the month
Help:	
Modified On:	8/9/2021 10:43:20 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Embedded Value Outgo Claims
Column Header:	dth_ben_if_b
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.1.2.97**      ***dth\_ben\_if\_b\_final***

Description:	death benefit inforce in the beginning of the month
Help:	
Modified On:	5/16/2023 2:43:19 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Embedded Value Outgo Claims
Column Header:	dth_ben_if_b_final
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False

Virtual: False

#### **5.3.1.2.98**      *pmt\_total*

Description: Total payments in force in the beginning of the month

Help:

Modified On: 5/16/2023 2:19:13 PM (UTC+03:00)

Modified By: CLAL-INS\ahuvaa

Category: Embedded Value|Outgo|Claims

Column Header: pmt\_total

Combine Groups By: Sum Both

Combine Periods: Last

Default sliding Size: -1

Discount Timing: End

Discount Use: Yes

Rate Use: Cash Flow

Rebase Type: Previous

Retain Value: Yes

Override: False

Virtual: False

#### **5.3.1.2.99**      *pmt\_total\_nogt*

Description: Total payments in force in the beginning of the month, excluding guaranteed payments.

Help:

Modified On: 5/16/2023 2:24:41 PM (UTC+03:00)

Modified By: CLAL-INS\ahuvaa

Category: Embedded Value|Outgo|Claims

Column Header: pmt\_total\_nogt

Combine Groups By: Sum Both

Combine Periods: Last

Default sliding Size: -1

Discount Timing: End

Discount Use: Yes

Rate Use: Cash Flow

Rebase Type: Previous

Retain Value: Yes

Override: False

Virtual: False

#### **5.3.1.2.100**      *int\_cred\_dth*

Description: Interest credited to death benefits

Help:

Modified On: 2/26/2025 1:59:45 PM (UTC+02:00)

Modified By: CLAL-INS\ahuvaa

Category:	Embedded Value Outgo Claims Guaranteed Claims
Column Header:	int_cred_dth
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.1.2.101     *int\_cred\_gteed***

Description:	Interest credited to guaranteed annuities
Help:	
Modified On:	2/26/2025 1:59:20 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Embedded Value Outgo Claims Guaranteed Claims
Column Header:	int_cred_gteed
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.1.2.102     *int\_cred\_jl1***

Description:	Interest credited to joint-life annuities (first life)
Help:	
Modified On:	2/26/2025 1:59:31 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Embedded Value Outgo Claims Guaranteed Claims
Column Header:	int_cred_jl1
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes

Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.1.2.103**     *int\_cred\_jl2*

Description:	Interest credited to joint-life annuities (first life)
Help:	
Modified On:	2/26/2025 1:59:40 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Embedded Value Outgo Claims Guaranteed Claims
Column Header:	int_cred_jl2
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.1.2.104**     *int\_cred\_no\_gteed*

Description:	Interest credited to non-guaranteed annuities
Help:	
Modified On:	2/26/2025 1:59:25 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Embedded Value Outgo Claims Guaranteed Claims
Column Header:	int_cred_no_gteed
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.1.2.105**     ***mgt\_fee\_fixed\_dth***

Description:	Fixed management fees for death benefits
Help:	
Modified On:	8/30/2021 8:43:56 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Embedded Value Outgo Claims Guaranteed Claims
Column Header:	mgt_fee_fixed_dth
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.1.2.106**     ***mgt\_fee\_fixed\_gtd***

Description:	Fixed management fees for guaranteed annuities
Help:	
Modified On:	8/30/2021 8:43:32 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Embedded Value Outgo Claims Guaranteed Claims
Column Header:	mgt_fee_fixed_gtd
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.1.2.107**     ***mgt\_fee\_fixed\_jl1***

Description:	Fixed management fees for joint-life annuities (first life)
Help:	
Modified On:	8/4/2021 3:33:23 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm



Category:	Embedded Value Outgo Claims Guaranteed Claims
Column Header:	mgt_fee_fixed_jl1
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.1.2.108     *mgt\_fee\_fixed\_jl2***

Description:	Fixed management fees for joint-life annuities (first life)
Help:	
Modified On:	8/4/2021 3:33:27 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Embedded Value Outgo Claims Guaranteed Claims
Column Header:	mgt_fee_fixed_jl2
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.1.2.109     *mgt\_fee\_fixed\_nogt***

Description:	Fixed management fees for non-guaranteed annuities
Help:	
Modified On:	8/4/2021 3:33:31 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Embedded Value Outgo Claims Guaranteed Claims
Column Header:	mgt_fee_fixed_nogt
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End

Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.1.2.110     *mgt\_fee\_var\_dth***

Description:	Variable management fees for death benefits
Help:	
Modified On:	9/25/2024 1:25:20 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Embedded Value Outgo Claims Guaranteed Claims
Column Header:	mgt_fee_var_dth
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.1.2.111     *mgt\_fee\_var\_gtd***

Description:	Variable management fees for guaranteed annuities
Help:	
Modified On:	9/25/2024 1:14:05 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Embedded Value Outgo Claims Guaranteed Claims
Column Header:	mgt_fee_var_gtd
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.1.2.112      *mgt\_fee\_var\_jl1***

Description:	Variable management fees for joint-life annuities (first life)
Help:	
Modified On:	9/25/2024 1:13:09 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Embedded Value Outgo Claims Guaranteed Claims
Column Header:	mgt_fee_var_jl1
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.1.2.113      *mgt\_fee\_var\_jl2***

Description:	Variable management fees for joint-life annuities (second life)
Help:	
Modified On:	9/25/2024 1:23:13 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Embedded Value Outgo Claims Guaranteed Claims
Column Header:	mgt_fee_var_jl2
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.1.2.114      *mgt\_fee\_var\_nogt***

Description:	Variable management fees for non-guaranteed annuities
Help:	
Modified On:	9/25/2024 1:02:21 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt

Category:	Embedded Value Outgo Claims Guaranteed Claims
Column Header:	mgt_fee_var_nogt
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.1.2.115      *tarif\_deduction\_dth***

Description:	Pricing rate deduction for death benefits
Help:	
Modified On:	8/24/2021 4:51:46 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Embedded Value Outgo Claims Guaranteed Claims
Column Header:	tarif_deduction_dth
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.1.2.116      *tarif\_deduction\_gteed***

Description:	Pricing rate deduction for guaranteed annuities
Help:	
Modified On:	8/4/2021 3:42:34 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Embedded Value Outgo Claims Guaranteed Claims
Column Header:	tarif_deduction_gteed
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes

Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.1.2.117      *tarif\_deduction\_jl1***

Description:	Pricing rate deduction for joint-life annuities (first life)
Help:	
Modified On:	8/4/2021 3:42:32 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Embedded Value Outgo Claims Guaranteed Claims
Column Header:	tarif_deduction_jl1
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.1.2.118      *tarif\_deduction\_jl2***

Description:	Pricing rate deduction for joint-life annuities (first life)
Help:	
Modified On:	8/4/2021 3:42:29 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Embedded Value Outgo Claims Guaranteed Claims
Column Header:	tarif_deduction_jl2
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.1.2.119      *tarif\_deduction\_no\_gteed***

Description:	Pricing rate deduction for non-guaranteed annuities
Help:	
Modified On:	8/4/2021 3:42:27 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Embedded Value Outgo Claims Guaranteed Claims
Column Header:	tarif_deduction_no_gteed
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.1.2.120      *ann\_pay\_gteed\_if***

Description:	guaranteed Ann payments to life 1
Help:	guaranteed annuity payments inforce in the beginning of the month for life 1.
Modified On:	8/9/2021 10:38:32 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Embedded Value Outgo Claims Life 1 Claims
Column Header:	ann_pay_gteed_if
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.1.2.121      *ann\_pay\_gteed\_if\_final***

Description:	guaranteed Ann payments to life 1
Help:	guaranteed annuity payments inforce in the beginning of the month for life 1.
Modified On:	5/16/2023 1:51:37 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Embedded Value Outgo Claims Life 1 Claims

Column Header:	ann_pay_gteed_if_final
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### 5.3.1.2.122 *ann\_pay\_jl\_if\_1*

Description:	joint life last survivor Ann payments inforce in the begining of the month - main life
Help:	guaranteed annuity payments inforce in the begining of the month for life 1.
Modified On:	8/9/2021 10:38:39 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Embedded Value Outgo Claims Life 1 Claims
Column Header:	ann_pay_jl_if_1
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### 5.3.1.2.123 *ann\_pay\_jl\_if\_1\_final*

Description:	joint life last survivor Ann payments inforce in the begining of the month - main life
Help:	guaranteed annuity payments inforce in the begining of the month for life 1.
Modified On:	5/16/2023 1:52:08 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Embedded Value Outgo Claims Life 1 Claims
Column Header:	ann_pay_jl_if_1_final
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow

Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

### **5.3.1.2.124     *ann\_pay\_jl\_if\_2***

Description:	joint life last survivor Ann payments inforce in the beginning of the month - secondary life
Help:	guaranteed annuity payments inforce in the beginning of the month for life 1.
Modified On:	8/9/2021 10:38:46 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Embedded Value Outgo Claims Life 1 Claims
Column Header:	ann_pay_jl_if_2
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

### **5.3.1.2.125     *ann\_pay\_jl\_if\_2\_final***

Description:	joint life last survivor Ann payments inforce in the beginning of the month - secondary life
Help:	guaranteed annuity payments inforce in the beginning of the month for life 1.
Modified On:	5/16/2023 1:53:14 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Embedded Value Outgo Claims Life 1 Claims
Column Header:	ann_pay_jl_if_2_final
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False



**5.3.1.2.126      *ann\_pay\_no\_gteed***

Description:	Ann payments to life 1 in force in the beginning of the month
Help:	Annuity payments due to life 1, whether in premium paying or paid up status.
Modified On:	8/9/2021 10:38:56 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Embedded Value Outgo Claims Life 2 Claims
Column Header:	ann_pay_no_gteed
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.1.2.127      *ann\_pay\_no\_gteed\_final***

Description:	Ann payments to life 1 in force in the beginning of the month
Help:	Annuity payments due to life 1, whether in premium paying or paid up status.
Modified On:	5/16/2023 1:40:27 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Embedded Value Outgo Claims Life 2 Claims
Column Header:	ann_pay_no_gteed_final
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.1.2.128      *death\_rates\_row\_1***

Description:	Death Rates Row 1
Help:	
Modified On:	8/12/2021 2:55:10 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Mortality

Column Header:	death_rates_row_1
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.1.2.129**      *death\_rates\_row\_2*

Description:	Death Rates Row 2
Help:	
Modified On:	8/12/2021 2:55:16 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Mortality
Column Header:	death_rates_row_2
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.1.2.130**      *units\_for\_takeup*

Description:	Units For Takeup
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Policy Details Annuity
Column Header:	units_for_takeup
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False

Virtual: False

### **5.3.1.2.131**     *age\_pol\_1*

Description: Age last by policy year for life 1  
 Help: Age last by policy year for life 1 at start of period t.  
 Modified On: 8/4/2021 2:44:38 PM (UTC+03:00)  
 Modified By: CLAL-INS\joshm  
 Category: Policy Details|Details Life 1  
 Column Header: age\_pol\_1  
 Combine Groups By: Average Both  
 Combine Periods: Last  
 Default sliding Size: -1  
 Discount Timing: Beginning  
 Discount Use: No  
 Rate Use: Cash Flow  
 Rebase Type: Current  
 Retain Value: Yes  
 Override: False  
 Virtual: False

### **5.3.1.2.132**     *age\_pol\_2*

Description: Age last by policy year for life 2  
 Help: Age last by policy year for life 2 at start of period t.  
 Modified On: 8/4/2021 2:44:48 PM (UTC+03:00)  
 Modified By: CLAL-INS\joshm  
 Category: Policy Details|Details Life 2  
 Column Header: age\_pol\_2  
 Combine Groups By: Average Both  
 Combine Periods: Last  
 Default sliding Size: -1  
 Discount Timing: Beginning  
 Discount Use: No  
 Rate Use: Cash Flow  
 Rebase Type: Current  
 Retain Value: Yes  
 Override: False  
 Virtual: False

### **5.3.1.2.133**     *profit\_book\_vif\_post\_ret\_pv*

Description: PV of Book Profit (end of month) VIF basis  
 Help:  
 Modified On: 8/9/2021 10:45:47 AM (UTC+03:00)  
 Modified By: CLAL-INS\joshm

Category:	Profitability Measures
Column Header:	profit_book_vif_post_ret_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

### 5.3.1.2.134 *reserve\_increase\_pv*

Description:	Reserve Increase Pv
Help:	
Modified On:	8/4/2021 3:38:55 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Reserve
Column Header:	reserve_increase_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

### 5.3.1.2.135 *startup*

Description:	Startup
Help:	This column is always called first when running a projection.
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Setup Initialise
Column Header:	startup
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous

Retain Value:	Yes
Override:	False
Virtual:	False

### 5.3.1.3 External Functions

#### 5.3.1.3.1 *initialise\_variables*

Description:	Initialise Variables
Help:	
Modified On:	11/13/2024 10:50:55 PM (UTC+02:00)
Modified By:	CLAL-INS\arikt
Category:	

#### 5.3.1.3.2 *set\_work\_variables*

Description:	Set Working variables
Help:	Variables that are calculated within MoSes.
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\arikt
Category:	Setup Initialise

### 5.3.1.4 Temporary Tables

#### 5.3.1.4.1 *qx\_final\_res*

Description:	final qx tab for reserves
Help:	
Modified On:	7/26/2021 3:06:55 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	
Number of Rows:	130
Number of Columns:	4
Calculation Methods:	Cell
Reset Variables:	
Resetting Type:	Reset Temporary Table for every model point and/or projection loop
Override:	False
Virtual:	False

#### 5.3.1.4.2 *res\_cx\_ann*

Description:	Reserve Commutation Cx
Help:	

Modified On:	8/15/2021 3:46:28 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Balance Sheet Reserves Basic Reserve
Number of Rows:	125
Number of Columns:	1
Calculation Methods:	Cell
Reset Variables:	
Resetting Type:	Reset Temporary Table for every model point and/or projection loop
Override:	False
Virtual:	False

#### **5.3.1.4.3 res\_dx\_ann**

Description:	Reserve Commutation Dx
Help:	
Modified On:	8/15/2021 3:49:15 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Balance Sheet Reserves Basic Reserve
Number of Rows:	125
Number of Columns:	3
Calculation Methods:	Cell
Reset Variables:	
Resetting Type:	Reset Temporary Table for every model point and/or projection loop
Override:	False
Virtual:	False

#### **5.3.1.4.4 res\_lx\_ann**

Description:	Reserve Commutation lx
Help:	
Modified On:	1/11/2023 11:29:27 AM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Balance Sheet Reserves Basic Reserve
Number of Rows:	125
Number of Columns:	3
Calculation Methods:	Cell
Reset Variables:	
Resetting Type:	Reset Temporary Table for every model point and/or projection loop
Override:	False
Virtual:	False

#### **5.3.1.4.5 res\_mx\_ann**

Description:	Reserve Commutation Mx
Help:	

Modified On:	8/15/2021 3:46:55 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Balance Sheet Reserves Basic Reserve
Number of Rows:	125
Number of Columns:	1
Calculation Methods:	Cell
Reset Variables:	
Resetting Type:	Reset Temporary Table for every model point and/or projection loop
Override:	False
Virtual:	False

#### **5.3.1.4.6 res\_nx\_ann**

Description:	Reserve Commutation Nx
Help:	
Modified On:	8/15/2021 3:46:36 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Balance Sheet Reserves Basic Reserve
Number of Rows:	125
Number of Columns:	3
Calculation Methods:	Cell
Reset Variables:	
Resetting Type:	Reset Temporary Table for every model point and/or projection loop
Override:	False
Virtual:	False

#### **5.3.1.4.7 res\_vx\_ann**

Description:	Reserve Interest vx
Help:	
Modified On:	6/15/2021 12:02:43 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Balance Sheet Reserves Basic Reserve
Number of Rows:	131
Number of Columns:	3
Calculation Methods:	Cell
Reset Variables:	
Resetting Type:	Reset Temporary Table for every model point and/or projection loop
Override:	False
Virtual:	False

#### **5.3.1.5 Scalars**

**5.3.1.5.1 ann\_pmt\_curr**

Description: current payment of not guaranteed annuity.  
Help:  
Modified On: 7/22/2021 4:20:50 PM (UTC+03:00)  
Modified By: CLAL-INS\joshm  
Category:  
Rebase: N/A  
Type: Double  
Override: False  
Virtual: False

**5.3.1.5.2 ann\_pmt\_curr\_gteed**

Description: current payment of guaranteed annuity.  
Help:  
Modified On: 10/3/2021 1:49:09 PM (UTC+03:00)  
Modified By: CLAL-INS\joshm  
Category:  
Rebase: N/A  
Type: Double  
Override: False  
Virtual: False

**5.3.1.5.3 ann\_pmt\_curr\_jl**

Description: current payment of jointlife annuity.  
Help:  
Modified On: 7/22/2021 4:21:53 PM (UTC+03:00)  
Modified By: CLAL-INS\joshm  
Category:  
Rebase: N/A  
Type: Double  
Override: False  
Virtual: False

**5.3.1.5.4 antisel\_weight**

Description: antisel\_weight  
Help:  
Modified On: 5/16/2023 2:25:17 PM (UTC+03:00)  
Modified By: CLAL-INS\ahuvaa  
Category:  
Rebase: N/A  
Type: Double  
Override: False  
Virtual: False



**5.3.1.5.5 antisel\_weight\_res**

Description:	antisel_weight
Help:	
Modified On:	5/16/2023 2:25:19 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	
Rebase:	N/A
Type:	Double
Override:	False
Virtual:	False

**5.3.1.5.6 fund\_type**

Description:	Fund Type
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	
Rebase:	N/A
Type:	Character
Override:	False
Virtual:	False

**5.3.1.5.7 mgt\_fixed\_max\_mth**

Description:	maximum monthly fixed mgt fee rate.
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	
Rebase:	N/A
Type:	Double
Override:	False
Virtual:	False

**5.3.1.5.8 mgt\_fixed\_mth**

Description:	monthly fixed mgt fee rate.
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	
Rebase:	N/A
Type:	Double
Override:	False
Virtual:	False

**5.3.1.5.9 rate\_tarif\_mth**

Description: monthly tarif rate.  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category:  
Rebase: N/A  
Type: Double  
Override: False  
Virtual: False

**5.3.1.5.10 sexcode\_1**

Description: Code for sex of insured 1. 0=Male, 1=Female  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category:  
Rebase: N/A  
Type: Integer  
Override: False  
Virtual: False

**5.3.1.5.11 sexcode\_2**

Description: Code for sex of insured 2. 0=Male, 1=Female  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category:  
Rebase: N/A  
Type: Integer  
Override: False  
Virtual: False

**5.3.1.5.12 annuity\_pmt\_curr\_tot**

Description: Total current annuity payment  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Annuity  
Rebase: N/A  
Type: Double  
Override: False  
Virtual: False

**5.3.1.5.13      *ann\_ratio\_res***

Description: Reserve Annuity ratio at retirement. Reserve Factors vs Policy Factors

Help:

Modified On: 5/22/2023 1:18:41 PM (UTC+03:00)

Modified By: CLAL-INS\ahuvaa

Category: Balance Sheet|Reserves

Rebase: N/A

Type: Double

Override: False

Virtual: False

**5.3.1.5.14      *antisel\_ppn***

Description: Old money prop. for anti select in annuity payment

Help:

Modified On: 12/11/2023 2:43:58 PM (UTC+02:00)

Modified By: CLAL-INS\Arikt

Category: Balance Sheet|Reserves|Annuity Deficiency

Rebase: N/A

Type: Double

Override: False

Virtual: False

**5.3.1.5.15      *antisel\_ppn\_res***

Description: Old money prop. for anti select in annuity payment

Help:

Modified On: 12/11/2023 2:43:58 PM (UTC+02:00)

Modified By: CLAL-INS\Arikt

Category: Balance Sheet|Reserves|Annuity Deficiency

Rebase: N/A

Type: Double

Override: False

Virtual: False

**5.3.1.5.16      *no\_antisel\_at\_ann***

Description: Annuity money not receiving anti-selection

Help:

Modified On: 12/11/2023 2:43:58 PM (UTC+02:00)

Modified By: CLAL-INS\Arikt

Category: Balance Sheet|Reserves|Annuity Deficiency

Rebase: N/A

Type: Double

Override: False  
Virtual: False

#### **5.3.1.5.17**      *no\_antisel\_at\_ann\_for\_res*

Description: Annuity money not receiving anti-selection (for reserve calc)  
Help:  
Modified On: 12/11/2023 2:43:58 PM (UTC+02:00)  
Modified By: CLAL-INS\Arikt  
Category: Balance Sheet|Reserves|Annuity Deficiency  
Rebase: N/A  
Type: Double  
Override: False  
Virtual: False

#### **5.3.1.5.18**      *init\_bor\_har*

Description: Initial bor/har passed to annuity  
Help:  
Modified On: 7/4/2024 3:36:22 PM (UTC+03:00)  
Modified By: CLAL-INS\arikt  
Category: Cashflows  
Rebase: N/A  
Type: Double  
Override: False  
Virtual: False

#### **5.3.1.5.19**      *retirement\_prop*

Description: Retirement Prop  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Economic assumptions  
Rebase: N/A  
Type: Double  
Override: False  
Virtual: False

#### **5.3.1.5.20**      *temp\_fund\_scalar*

Description: Temp Fund  
Help:  
Modified On: 11/18/2020 4:47:42 PM (UTC+02:00)  
Modified By: CLAL-INS\NinaB  
Category: Economic assumptions

Rebase:	N/A
Type:	Character
Override:	False
Virtual:	False

### 5.3.1.5.21 *initial\_annuity\_purchase*

Description:	initial sum for annuity purchase
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Policy Details Annuity
Rebase:	N/A
Type:	Double
Override:	False
Virtual:	False

### 5.3.2 *fund\_cflow*

Description:	
Help:	
Base Model Class:	none
Model References	All
Read File:	Before Start Up
Modified On:	4/27/2023 9:57:35 AM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa

### 5.3.2.1 Variables

#### 5.3.2.1.1 *prem\_freq*

Description:	Premium frequency
Help:	Frequency of premium payment: 1 - Annually 2 - Half-Yearly 4 - Quarterly 12 - Monthly
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Embedded Value Income Premium
Variable Type:	Integer Number
Default Value:	12
Length:	0
Number of Decimals:	0
Choice List:	

Character Type:	Not Applicable
Valid Range From:	1
Valid Range To:	12
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.2.1.2 *surr\_chg\_perc\_units*

Description:	Surrender charges (% units) for basic savings
Help:	Surrender charges as a % of units for each policy month.
Modified On:	6/27/2022 5:15:18 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Embedded Value Outgo Claims Surrender Claims
Variable Type:	Floating Point Array
Default Value:	0
Length:	1400
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.2.1.3 *comm\_perc\_res*

Description:	Commissions as % of reserves
Help:	Commissions as a % of reserves held. Loaded from table.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Embedded Value Outgo Commission
Variable Type:	Floating Point Array
Default Value:	0.5
Length:	116
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1000
Table Format:	Default Row Numbers
Set Value in Input Manager:	

Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.2.1.4 comm\_regular\_pc**

Description:	Annual regular commission (%)
Help:	Annual regular comm %. Read from table in set_comm_variables() .
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Embedded Value Outgo Commission
Variable Type:	Floating Point Array
Default Value:	0 10
Length:	116
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.2.1.5 comm\_ren\_perc\_prem**

Description:	Renewal commission (%)
Help:	Renewal commission expressed as a % of premium income.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Embedded Value Outgo Commission
Variable Type:	Floating Point Number
Default Value:	0
Length:	116
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.2.1.6 comm\_renewal\_year**

Description:	First year from when renewal commission is paid
Help:	policy year from when renewal commission is paid.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Embedded Value Outgo Commission
Variable Type:	Integer Number
Default Value:	1
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	1
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.2.1.7 unit\_value\_if**

Description:	Unit balance at valn date
Help:	Nominal amount before allowing for actuarial funding.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Embedded Value Reserves Unit Reserves
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	-99999999
Valid Range To:	999999999
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.2.1.8 commence\_period\_w**

Description:	Period t in which policy commences
Help:	Calculated in startup column.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab



Category:	Policy Details
Variable Type:	Integer Number
Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1000
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.2.1.9 *elapsed\_months***

Description:	Elapsed months at valn date
Help:	The number of months, rounded up, from policy inception to the valuation date.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Integer Number
Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1200
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.2.1.10 *maturity\_period\_w***

Description:	Period t in which policy matures
Help:	Calculated in startup column.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Integer Number
Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable

Valid Range From:	0
Valid Range To:	1000
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.2.1.11 *paid\_up*

Description:	Paid up at valuation date?
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Character
Default Value:	N
Length:	1
Number of Decimals:	0
Choice List:	N,Y
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.2.1.12 *par\_nonpar*

Description:	Participating or Non-Participating?
Help:	P = Participating business (i.e. with-profit) N = Non participating business (i.e. non-profit)
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details Bonus
Variable Type:	Character
Default Value:	P
Length:	1
Number of Decimals:	0
Choice List:	P,N
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.2.1.13**      ***benefits\_curr***

Description:	Number of covers at valuation date
Help:	Current number of in force benefits at the valuation date.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details Policies & Benefits
Variable Type:	Floating Point Number
Default Value:	1
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1000000
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.2.1.14**      ***policies\_curr***

Description:	Number of policies inforce at valn date
Help:	Current number of in force policies at the valuation date.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details Policies & Benefits
Variable Type:	Floating Point Number
Default Value:	1
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1000000
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.2.1.15**      ***decrements\_apply***

Description:	Apply decrements?
Help:	Do you want in force columns decremented by survivorship?
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab

Category:	Setup Options
Variable Type:	Character
Default Value:	Y
Length:	1
Number of Decimals:	0
Choice List:	Y,N
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.2.1.16 *projection\_type*

Description:	Projection type
Help:	Purpose of the projection run: Valn = perform a valuation for an in force policy New_Bus = used to project future new business layers Pricing = project one new business policy
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Setup Values
Variable Type:	Character
Default Value:	Pricing
Length:	7
Number of Decimals:	0
Choice List:	Valn,New_Bus,Pricing
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.2.1.17 *unit\_type*

Description:	Unit type
Help:	Init_prem = Initial unit premium paying policy Accum_prem = Accumulation unit premium paying policy Init_pup = Initial unit paid up policy Accum_pup = Accumulation unit paid up policy
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Unit Fund

Variable Type:	Character
Default Value:	Accum_prem
Length:	10
Number of Decimals:	0
Choice List:	Accum_prem,Accum_pup,Saving,Saving_pup
Character Type:	Standard
Valid Range From:	1
Valid Range To:	2
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.2.1.18 *alloc\_rate*

Description:	Percentage of Premium allocated to units
Help:	Percentage of premium allocated to units. Each element in the array applies for the number of months specified in the alloc_rate_period array. The first element in the array should be entered for period 1.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Unit Fund Charges
Variable Type:	Floating Point Array
Default Value:	0 100
Length:	16
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	150
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.2.1.19 *alloc\_rate\_period*

Description:	Months applying to alloc rate
Help:	Each element in this array relates to a corresponding element in the alloc_rate array. The elements in this array specify the number of months for which the allocation rates in alloc_rate apply. The first element in the array should be entered for period 1.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Unit Fund Charges

Variable Type:	Floating Point Array
Default Value:	6 12
Length:	16
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	600
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### 5.3.2.1.20 *mgt\_fee\_fixed*

Description:	Fixed Management fee %
Help:	Management fee as an annual percentage of the unit value. Loaded from table.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Unit Fund Fees
Variable Type:	Floating Point Number
Default Value:	1.3
Length:	0
Number of Decimals:	4
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### 5.3.2.1.21 *mgt\_fee\_variable*

Description:	variable management fee proportion (%)
Help:	Only for used for Adif. Percentage of investment return taken as management fee, in addition to fixed management fee which is a percentage of units.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Unit Fund Fees
Variable Type:	Floating Point Number
Default Value:	15
Length:	0
Number of Decimals:	4
Choice List:	

Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.2.2 Columns

#### 5.3.2.2.1 *decrement\_rate*

Description:	Total marginal decrement rate from units
Help:	
Modified On:	1/11/2023 6:45:46 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Decrements
Column Header:	decrement_rate
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### 5.3.2.2.2 *lapse\_rate\_bal*

Description:	lapse rate - on balance exposure
Help:	
Modified On:	1/11/2023 6:46:39 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Decrements Lapse
Column Header:	lapse_rate_bal
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False

Virtual: False

#### **5.3.2.2.3 premium**

Description: premium (excluding policy fee)  
 Help:  
 Modified On: 7/20/2021 5:19:38 PM (UTC+03:00)  
 Modified By: CLAL-INS\joshm  
 Category: Embedded Value|Income|Premium  
 Column Header: premium  
 Combine Groups By: Sum Both  
 Combine Periods: Last  
 Default sliding Size: -1  
 Discount Timing: Beginning  
 Discount Use: Yes  
 Rate Use: Cash Flow  
 Rebase Type: Previous  
 Retain Value: Yes  
 Override: False  
 Virtual: False

#### **5.3.2.2.4 death\_claims\_units**

Description: Death claims on units  
 Help:  
 Modified On: 8/12/2021 1:14:03 PM (UTC+03:00)  
 Modified By: CLAL-INS\joshm  
 Category: Embedded Value|Outgo|Claims|Death Claims  
 Column Header: death\_claims\_units  
 Combine Groups By: Sum Both  
 Combine Periods: Last  
 Default sliding Size: -1  
 Discount Timing: End  
 Discount Use: Yes  
 Rate Use: Cash Flow  
 Rebase Type: Previous  
 Retain Value: Yes  
 Override: False  
 Virtual: False

#### **5.3.2.2.5 claims\_surrender**

Description: Surrender claims at middle of period  
 Help:  
 Modified On: 1/11/2023 7:36:19 PM (UTC+02:00)  
 Modified By: CLAL-INS\joshm  
 Category: Embedded Value|Outgo|Claims|Surrender Claims



Column Header:	claims_surrender
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	Middle
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### 5.3.2.2.6 *surr\_charge*

Description:	Surrender charge received in period
Help:	
Modified On:	1/11/2023 8:13:16 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Embedded Value Outgo Claims Surrender Claims
Column Header:	surr_charge
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### 5.3.2.2.7 *surr\_penalty\_e\_bef*

Description:	Surrender penalty inforce, end of period
Help:	
Modified On:	10/20/2021 1:28:28 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Embedded Value Outgo Claims Surrender Claims
Column Header:	surr_penalty_e_bef
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current

Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.2.2.8 *surr\_value***

Description:	Surrender value inforce EOM
Help:	
Modified On:	11/17/2022 4:52:43 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Embedded Value Outgo Claims Surrender Claims
Column Header:	surr_value
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.2.2.9 *comm\_regular***

Description:	Regular commission
Help:	
Modified On:	8/9/2021 10:35:23 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Embedded Value Outgo Commission
Column Header:	comm_regular
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.2.2.10 *comm\_renewal***

Description:	Renewal commission
Help:	
Modified On:	8/9/2021 10:35:30 AM (UTC+03:00)

Modified By:	CLAL-INS\joshm
Category:	Embedded Value Outgo Commission
Column Header:	comm_renewal
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### 5.3.2.2.11 *comm\_reserve*

Description:	commissions on reserve in the peri
Help:	
Modified On:	11/14/2024 4:58:23 PM (UTC+02:00)
Modified By:	CLAL-INS\arikt
Category:	Embedded Value Outgo Commission
Column Header:	comm_reserve
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### 5.3.2.2.12 *int\_cred\_units\_e*

Description:	Interest credited to units
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\arikt
Category:	Embedded Value Reserves Unit Reserves
Column Header:	int_cred_units_e
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous

Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.2.2.13**      ***pol\_year***

Description:	Policy year
Help:	
Modified On:	7/20/2021 5:18:37 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Timing
Column Header:	pol_year
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.2.2.14**      ***units\_b***

Description:	Unit account after allocation (beg)
Help:	
Modified On:	6/2/2022 9:29:28 AM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Unit Fund
Column Header:	units_b
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.2.2.15**      ***units\_b\_bef***

Description:	Unit account before allocation (beg)
Help:	
Modified On:	1/11/2023 8:14:44 PM (UTC+02:00)

Modified By:	CLAL-INS\joshm
Category:	Unit Fund
Column Header:	units_b_bef
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.2.2.16**      ***units\_b\_bef\_pup***

Description:	Unit value for new PUPs
Help:	
Modified On:	8/9/2021 10:37:15 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Unit Fund
Column Header:	units_b_bef_pup
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.2.2.17**      ***units\_e***

Description:	Unit account EOM after claims
Help:	Unit account EOM after interest, mgmt fees and claims
Modified On:	1/11/2023 6:45:46 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Unit Fund
Column Header:	units_e
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow

Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.2.2.18**      *units\_e\_bef*

Description:	Unit account EOM after int before clms
Help:	Unit account EOM after interest and mgmt fees but before claims
Modified On:	8/9/2021 10:37:21 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Unit Fund
Column Header:	units_e_bef
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.2.2.19**      *alloc\_units*

Description:	Premium allocation to units
Help:	
Modified On:	8/9/2021 10:34:59 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Unit Fund Charges
Column Header:	alloc_units
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.2.2.20**      *allocation\_rate*

Description:	Allocation rate
Help:	

Modified On:	8/9/2021 10:35:07 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Unit Fund Charges
Column Header:	allocation_rate
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.2.2.21      *cover\_charge***

Description:	Life cover charge (beg)
Help:	
Modified On:	8/9/2021 10:35:42 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Unit Fund Charges
Column Header:	cover_charge
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.2.2.22      *int\_rate\_net\_cumm***

Description:	Cummulative interest rate for year
Help:	
Modified On:	12/17/2019 11:36:03 AM (UTC+02:00)
Modified By:	CLAL-INS\NinaB
Category:	Unit Fund Charges
Column Header:	int_rate_net_cumm
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow

Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.2.2.23      *management\_fee\_fixed***

Description:	Fixed Management Fees
Help:	
Modified On:	6/29/2021 9:06:00 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Unit Fund Charges
Column Header:	management_fee_fixed
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.2.2.24      *management\_fee\_variable***

Description:	Variable Management Fee
Help:	
Modified On:	4/30/2020 3:47:08 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Category:	Unit Fund Charges
Column Header:	management_fee_variable
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.2.2.25      *net\_interest\_rate***

Description:	Interest rate, net of fixed management fees
Help:	



Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Unit Fund Charges
Column Header:	net_interest_rate
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.2.2.26      *management\_fee***

Description:	Management fee
Help:	
Modified On:	8/9/2021 10:36:18 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Unit Fund Fees
Column Header:	management_fee
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.2.2.27      *management\_fee\_rate***

Description:	Monthly management fee rate for the unit sub fund
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Unit Fund Fees
Column Header:	management_fee_rate
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No

Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### 5.3.2.2.28 *management\_fee\_rate\_annual*

Description:	Annual management fee rate for the unit sub fund - only for profil non profit participating
Help:	
Modified On:	5/1/2023 1:52:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Unit Fund Fees
Column Header:	management_fee_rate_annual
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### 5.3.2.3 External Functions

<No External Functions Exist>

#### 5.3.2.4 Temporary Tables

##### 5.3.2.4.1 *pup\_units\_tt*

Description:	PUP units by prem. cessation period (r) and duration since (c)
Help:	
Modified On:	1/11/2023 6:45:46 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Unit Fund
Number of Rows:	1200
Number of Columns:	1200
Calculation Methods:	Cell
Reset Variables:	
Resetting Type:	Reset Temporary Table for every model point and/or projection loop

Override: False  
Virtual: False

### 5.3.2.5 Scalars

#### 5.3.2.5.1 *premium\_nb\_sp*

Description: Extra Single Premium for NB to get opening account balance  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Embedded Value|Income|Premium  
Rebase: N/A  
Type: Double  
Override: False  
Virtual: False

### 5.3.3 *life\_cflow*

Description:  
Help:  
Base Model Class: none  
Model References: All  
Read File: Before Start Up  
Modified On: 5/29/2025 12:13:29 PM (UTC+03:00)  
Modified By: CLAL-INS\arikt

### 5.3.3.1 Variables

#### 5.3.3.1.1 *ben\_period\_min*

Description: Ben Period Min  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category:  
Variable Type: Integer Number  
Default Value: 0  
Length: 1  
Number of Decimals: 0  
Choice List:  
Character Type: Not Applicable

Valid Range From:  
Valid Range To:  
Table Format: Default Row Numbers  
Set Value in Input Manager:  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.2 *ben\_term\_max***

Description: max benefit term  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category:  
Variable Type: Integer Number  
Default Value: 0  
Length: 0  
Number of Decimals: 0  
Choice List:  
Character Type: Not Applicable  
Valid Range From:  
Valid Range To:  
Table Format: Default Row Numbers  
Set Value in Input Manager:  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.3 *benefit\_term\_original***

Description: Benefit Term Original  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category:  
Variable Type: Integer Number  
Default Value: 0  
Length: 0  
Number of Decimals: 0  
Choice List:  
Character Type: Not Applicable  
Valid Range From:  
Valid Range To:  
Table Format: Default Row Numbers  
Set Value in Input Manager:  
Variable Sharing: Not Shared  
Category Order: 0

**5.3.3.1.4 col\_char**

Description:	Col Char
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	
Variable Type:	Character
Default Value:	0
Length:	20
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.5 dactype**

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.6 decrem\_mult\_col\_key**

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	
Variable Type:	Character

Default Value:	0
Length:	20
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.7 *decrem\_mult\_row\_key***

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	
Variable Type:	Character
Default Value:	0
Length:	20
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.8 *fund\_name***

Description:	Lookup value code variable wildcard
Help:	Model fund name/number, based on actual fund number from data file
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	
Variable Type:	Floating Point Number
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	

Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.9 *mat\_period\_min***

Description:	Mat Period Min
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	
Variable Type:	Integer Number
Default Value:	0
Length:	10
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.10 *mat\_period\_original***

Description:	Original maturity period for trad polices using multi age retirement
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	
Variable Type:	Integer Number
Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.11**      ***mgtfee\_acc\_after***

Description:	Lookup value code variable
Help:	Fixed Management fee % after accumulation amount x
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.12**      ***mgtfee\_age\_after***

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.13**      ***mgtfee\_dthben***

Description:	Lookup value code variable wildcard
Help:	Management fee reduction % by Death benefit
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	



Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.14**      ***par\_npar***

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.15**      ***par\_npar\_yesodi***

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	

Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.16**      ***prate\_level***

Description:	Prate Level
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	2
Choice List:	
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.17**      ***retirement\_rate***

Description:	Percentage retiring at current age
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.18**      ***row\_char***

Description:	Row Char
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	
Variable Type:	Character
Default Value:	0
Length:	20
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.19**      ***row\_num***

Description:	Row Num
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	3
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.20**      ***sal\_rider\_tbl***

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	
Variable Type:	Floating Point Number

Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.21      *tarif\_spec\_lookup\_freq***

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.22      *temp\_col\_fund***

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	
Variable Type:	Character
Default Value:	0
Length:	20
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers

Set Value in Input Manager:  
 Variable Sharing: Not Shared  
 Category Order: 0

### 5.3.3.1.23 *ann\_maslul*

Description: Annuity track of annuities currently in payment  
 Help: policy annuity factor of not guaranteed period  
 Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
 Modified By: CLAL-INS\ninab  
 Category: Annuity  
 Variable Type: Integer Number  
 Default Value: 0  
 Length: 6  
 Number of Decimals: 0  
 Choice List:  
 Character Type: Not Applicable  
 Valid Range From: 0  
 Valid Range To: 9000  
 Table Format: Default Row Numbers  
 Set Value in Input Manager:  
 Variable Sharing: Not Shared  
 Category Order: 0

### 5.3.3.1.24 *annuitization\_rate*

Description: % of maturing policies taking annuity  
 Help: Percentage of maturing units that are converted to an annuity.  
 Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
 Modified By: CLAL-INS\ninab  
 Category: Annuity  
 Variable Type: Floating Point Number  
 Default Value: 0.1  
 Length: 0  
 Number of Decimals: 1  
 Choice List:  
 Character Type: Not Applicable  
 Valid Range From: 0  
 Valid Range To: 100  
 Table Format: Default Row Numbers  
 Set Value in Input Manager:  
 Variable Sharing: Not Shared  
 Category Order: 0

### 5.3.3.1.25 *annuity\_takeup\_max*

Description: Maximum annuity take up rate at maturity

Help:	This rate should be net of annuity payment expenses, and should be rounded to the nearest basis point. It is used to lookup the annuity value.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Annuity
Variable Type:	Floating Point Number
Default Value:	1
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	6
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.26      *annuity\_takeup\_new\_tag***

Description:	New tagmulim new annuity take up rate at maturity
Help:	This rate should be net of annuity payment expenses, and should be rounded to the nearest basis point. It is used to lookup the annuity value.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Annuity
Variable Type:	Floating Point Number
Default Value:	87.5
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.27      *annuity\_takeup\_old***

Description:	Old money annuity take up rate at maturity
--------------	--

Help:	This rate should be net of annuity payment expenses, and should be rounded to the nearest basis point. It is used to lookup the annuity value.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Annuity
Variable Type:	Floating Point Number
Default Value:	16.3
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.28      *annuity\_takeup\_piz***

Description:	Pitzuim annuity take up rate at maturity
Help:	This rate should be net of annuity payment expenses, and should be rounded to the nearest basis point. It is used to lookup the annuity value.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Annuity
Variable Type:	Floating Point Number
Default Value:	17.9
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.29      *annuity\_takeup\_prat***

Description:	Prat money annuity take up rate at maturity
--------------	---

Help:	This rate should be net of annuity payment expenses, and should be rounded to the nearest basis point. It is used to lookup the annuity value.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Annuity
Variable Type:	Floating Point Number
Default Value:	6
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.30      *bonus\_tbl\_row***

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Annuity
Variable Type:	Integer Number
Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.31      *gimla\_col\_key***

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Annuity
Variable Type:	Character



Default Value:	0
Length:	10
Number of Decimals:	0
Choice List:	
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### 5.3.3.1.32 ***gimla\_db\_period\_w***

Description:	Term during which Gimla death benefit exists (in months)
Help:	Working Variable - calculated in set other variable. For Klassi Gimla product (ben_class="gimla"). Death benefit term (in months) during which a death lump sum is paid out.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Annuity
Variable Type:	Integer Number
Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	10000
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### 5.3.3.1.33 ***gimla\_row\_key***

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Annuity
Variable Type:	Integer Number
Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	

Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.34      *gimla\_table***

Description:	Annuity value at maturity (SP)
Help:	PV of annuities at maturity (or at 'vesting date'). Deducted from single premium table.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Annuity
Variable Type:	Integer Number
Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	0
Table Format:	Row Name (numeric)
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.35      *old\_antiselection\_adj***

Description:	Adjustment to anti-selection i.r.o. old tagmulim
Help:	
Modified On:	2/18/2021 10:36:16 AM (UTC+02:00)
Modified By:	CLAL-INS\NinaB
Category:	Annuity
Variable Type:	Floating Point Number
Default Value:	100
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.36      *piz\_antiselection\_adj***

Description:	Adjustment to anti-selection i.r.o. pizuim due to tax-exempt limit
Help:	
Modified On:	2/14/2021 3:19:28 PM (UTC+02:00)
Modified By:	CLAL-INS\NinaB
Category:	Annuity
Variable Type:	Floating Point Number
Default Value:	100
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.37      *prat\_antiselection\_adj***

Description:	Adjustment to anti-selection i.r.o. private funds
Help:	
Modified On:	2/18/2021 10:35:59 AM (UTC+02:00)
Modified By:	CLAL-INS\NinaB
Category:	Annuity
Variable Type:	Floating Point Number
Default Value:	100
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.38      *res\_prop\_key***

Description:	res prop column lookup
Help:	res_prop type to lookup in tk prop data: old_tag, new_tag, piz or prat
Modified On:	12/11/2022 1:17:24 PM (UTC+02:00)

Modified By:	CLAL-INS\ahuvaa
Category:	Annuity
Variable Type:	Character
Default Value:	0
Length:	22
Number of Decimals:	0
Choice List:	
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.39      *res\_prop\_newtag\_data***

Description:	Proportion of new tagmulim money from data
Help:	
Modified On:	9/14/2020 9:59:47 AM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Category:	Annuity
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	4
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.40      *res\_prop\_old\_data***

Description:	Proportion of old money from data
Help:	
Modified On:	9/14/2020 9:52:32 AM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Category:	Annuity
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	4
Choice List:	
Character Type:	Not Applicable

Valid Range From:	0
Valid Range To:	1
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.41      *res\_prop\_piz\_data***

Description:	Proportion of pizuim money from data
Help:	
Modified On:	9/14/2020 10:02:44 AM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Category:	Annuity
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	4
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.42      *res\_prop\_prat\_data***

Description:	Proportion of prat money from data
Help:	
Modified On:	9/14/2020 10:53:16 AM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Category:	Annuity
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	4
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.43      *takeup\_age***

Description:	Take-up age for annuities
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Annuity
Variable Type:	Integer Number
Default Value:	67
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.44      *col***

Description:	Col
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Balance Sheet Reserves
Variable Type:	Character
Default Value:	F
Length:	10
Number of Decimals:	1
Choice List:	F
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.45      *cap\_req\_perc\_premium\_temp***

Description:	Capital required as % of annual premium
Help:	The percentage of the annual premium for the capital required (Set in the product specifications table)
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Capital

Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.46**      ***cap\_req\_perc\_reserve\_temp***

Description:	Capital required as % of reserve
Help:	The percentage of the annual premium for the capital required (Set in the product specifications table)
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Capital
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.47**      ***profit\_weighting***

Description:	profit weighting for IFRS - gross
Help:	Maximum cumulative claim inflation allowed (for health products). As a percentage. For example, 200 means that the claims inflation will stop if and when the claims reach double the base assumption.
Modified On:	5/8/2023 9:52:48 AM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Cashflows
Variable Type:	Floating Point Number
Default Value:	0

Length:	1
Number of Decimals:	5
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### 5.3.3.1.48 *profit\_weighting\_re*

Description:	profit weighting reinsurance - for IFRS
Help:	Maximum cumulative claim inflation allowed (for health products). As a percentage. For example, 200 means that the claims inflation will stop if and when the claims reach double the base assumption.
Modified On:	5/8/2023 9:44:05 AM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Cashflows
Variable Type:	Floating Point Number
Default Value:	0
Length:	1
Number of Decimals:	5
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### 5.3.3.1.49 *charge\_addition\_absolute*

Description:	charge_addition_absolute
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Charges
Variable Type:	Floating Point Array
Default Value:	0
Length:	26
Number of Decimals:	8
Choice List:	
Character Type:	Not Applicable



Valid Range From:	-1
Valid Range To:	1
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.50**      *charge\_addition\_perc*

Description:	charge_addition_perc
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Charges
Variable Type:	Floating Point Array
Default Value:	0
Length:	26
Number of Decimals:	3
Choice List:	
Character Type:	Not Applicable
Valid Range From:	-100
Valid Range To:	500
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.51**      *comm\_by\_cal*

Description:	Calculate Comm Re by Calendar duration or Policy duration
Help:	
Modified On:	1/5/2025 3:01:34 PM (UTC+02:00)
Modified By:	CLAL-INS\arikt
Category:	Charges
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	0
Choice List:	
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.52      *gorem\_mult***

Description:	Percentage of police fee from assumption file
Help:	
Modified On:	12/10/2023 11:48:02 AM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Charges
Variable Type:	Floating Point Number
Default Value:	1
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.53      *interest\_re\_calculate***

Description:	Calculate interest re (Y) or make zero (N)
Help:	
Modified On:	9/12/2024 11:23:11 AM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Charges
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	0
Choice List:	
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.54      *margin\_disc\_col\_key***

Description:	Column lookup key for retention scenario
Help:	
Modified On:	11/30/2020 10:15:12 AM (UTC+02:00)
Modified By:	CLAL-INS\NinaB
Category:	Charges
Variable Type:	Character

Default Value:	DN_prem_adif
Length:	15
Number of Decimals:	1
Choice List:	DN_prem_adif
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### 5.3.3.1.55 *mgt\_deficit\_perc*

Description:	Management fee surplus/deficit as % of accumulation
Help:	
Modified On:	12/12/2019 2:36:29 PM (UTC+02:00)
Modified By:	CLAL-INS\NinaB
Category:	Charges
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	5
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### 5.3.3.1.56 *mgt\_fee\_disc*

Description:	Discount on management fees
Help:	
Modified On:	11/29/2020 2:51:48 PM (UTC+02:00)
Modified By:	CLAL-INS\NinaB
Category:	Charges
Variable Type:	Floating Point Number
Default Value:	2
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100

Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.57**      ***mgt\_fee\_fixed***

Description:	Fixed Management fee %
Help:	Management fee as a fixed annual percentage . Loaded from table.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Charges
Variable Type:	Floating Point Number
Default Value:	0.6
Length:	0
Number of Decimals:	4
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.58**      ***mgt\_fee\_fixed\_input***

Description:	Fixed Management fee %
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Charges
Variable Type:	Floating Point Number
Default Value:	0.6
Length:	0
Number of Decimals:	4
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.59**      ***mgt\_fee\_fixed\_puresav***

Description:	Fixed Management fee % for pure saving component
Help:	Management fee as a fixed annual percentage . Loaded from table.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Charges
Variable Type:	Floating Point Number
Default Value:	0.6
Length:	0
Number of Decimals:	4
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.60**      ***mgt\_fee\_variable***

Description:	variable management fee proportion (%)
Help:	Loaded from table as a percentage
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Charges
Variable Type:	Floating Point Number
Default Value:	15
Length:	0
Number of Decimals:	4
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.61**      ***mgt\_fee\_variable\_input***

Description:	variable management fee proportion (%)
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Charges

Variable Type:	Floating Point Number
Default Value:	0.6
Length:	0
Number of Decimals:	4
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.62**      ***mgtfee\_acc***

Description:	Accumulation amount for Fixed Management fee % to apply after amount x
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Charges
Variable Type:	Integer Number
Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1000000
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.63**      ***mgtfee\_age***

Description:	Age for Fixed Management fee % to apply after age x
Help:	Management fee as a fixed annual percentage . Loaded from table.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Charges
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	

Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.64**      ***mgtfee\_disc\_after***

Description:	Lookup value code variable wildcard
Help:	Management fee rate after discount
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Charges
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.65**      ***mgtfee\_disc\_mth***

Description:	Lookup value code variable wildcard
Help:	Management fee discount month from t=0
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Charges
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.66**      ***mgtfee\_floor***

Description:	Minimum Fixed Management fee %
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Charges
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	4
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.67**      ***mgtfee\_format***

Description:	Management fee format number
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Charges
Variable Type:	Integer Number
Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.68**      ***mgtfee\_from\_dthben***

Description:	Amount of dath benefit from which discount starts
Help:	Management fee as a fixed annual percentage . Loaded from table.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Charges



Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1000000
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.69**      ***mgtfee\_from\_senior***

Description:	The number of premium paying month from which start Management fee discount
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Charges
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	240
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.70**      ***mgtfee\_max\_dthben***

Description:	Maximum dath benefit from which minimum management fee starts
Help:	Management fee as a fixed annual percentage . Loaded from table.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Charges
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	

Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	2000000
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.71 *mgtfee\_orig*

Description:	original Fixed Management fee % before discount
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Charges
Variable Type:	Floating Point Number
Default Value:	0.6
Length:	0
Number of Decimals:	4
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.72 *mgtfee\_senior*

Description:	Lookup value code variable wildcard
Help:	Management fee reduction % by premium paying months
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Charges
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared

Category Order: 0

#### **5.3.3.1.73      *reserve\_re\_increase\_calculate***

Description: Calculate reserve re increase (Y) or make zero (N)  
Help:  
Modified On: 9/12/2024 11:23:53 AM (UTC+03:00)  
Modified By: CLAL-INS\arikt  
Category: Charges  
Variable Type: Character  
Default Value: 0  
Length: 10  
Number of Decimals: 0  
Choice List:  
Character Type: Standard  
Valid Range From:  
Valid Range To:  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.74      *surr\_charge\_set\_temp***

Description: Surrender charge assumption set  
Help: Surrender charge assumption set. If read\_from\_table = "Y", then this variable will be set in set\_common\_variables.  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Charges  
Variable Type: Character  
Default Value: default  
Length: 7  
Number of Decimals: 0  
Choice List:  
Character Type: Standard  
Valid Range From: 0  
Valid Range To: 0  
Table Format: Default Row Numbers  
Set Value in Input Manager:  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.75      *surr\_chg\_perc\_sav***

Description: Surrender charges (% units) for extra savings

Help:	Surrender charges as a % of units for each policy month, for the pure saving portion of the policy (Adif or Profil).
Modified On:	Read from <i>surr_chg_tbl</i> in <i>set_by_prodcode</i> . 8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Charges
Variable Type:	Floating Point Array
Default Value:	0
Length:	1001
Number of Decimals:	4
Choice List:	
Character Type:	Not Applicable
Valid Range From:	-100
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.76      *surr\_chg\_perc\_units***

Description:	Surrender charges (% units) for basic savings
Help:	Surrender charges as a % of units for each policy month, for the basic portion of the policy (Adif or Profil).
Modified On:	Read from <i>surr_chg_tbl</i> in <i>set_by_prodcode</i> . 8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Charges
Variable Type:	Floating Point Array
Default Value:	0
Length:	1001
Number of Decimals:	4
Choice List:	
Character Type:	Not Applicable
Valid Range From:	-100
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.77      *tat\_shnatiut\_assum***

Description:	Tat Shnatiut rate from assumption table FundRate
--------------	---

Help:  
 Modified On: 12/7/2023 12:31:25 PM (UTC+02:00)  
 Modified By: CLAL-INS\Arikt  
 Category: Charges  
 Variable Type: Floating Point Number  
 Default Value: 0  
 Length: 0  
 Number of Decimals: 1  
 Choice List:  
 Character Type: Not Applicable  
 Valid Range From:  
 Valid Range To:  
 Table Format: Default Row Numbers  
 Set Value in Input Manager: All  
 Variable Sharing: Not Shared  
 Category Order: 0

#### **5.3.3.1.78      *tat\_shnatiut\_input***

Description: percentage of modal loading  
 Help: modal loading percentage. Set from fund\_rate\_tbl in set exp variables if "read\_from\_table" = Y;  
 Modified On: 12/7/2023 12:04:16 PM (UTC+02:00)  
 Modified By: CLAL-INS\Arikt  
 Category: Charges  
 Variable Type: Floating Point Number  
 Default Value: 0  
 Length: 0  
 Number of Decimals: 2  
 Choice List:  
 Character Type: Not Applicable  
 Valid Range From: -100  
 Valid Range To: 100  
 Table Format: Default Row Numbers  
 Set Value in Input Manager: All  
 Variable Sharing: Not Shared  
 Category Order: 0

#### **5.3.3.1.79      *tat\_shnatiut\_rate***

Description: Percentage of modal loading  
 Help:  
 Modified On: 12/7/2023 12:32:59 PM (UTC+02:00)  
 Modified By: CLAL-INS\Arikt  
 Category: Charges  
 Variable Type: Floating Point Number  
 Default Value: 0

Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.80      *use\_tat\_shnatiut\_assum***

Description:	Use Tat Shnatiut from assumption file (Y) or data (N)
Help:	
Modified On:	12/20/2023 6:04:35 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Charges
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	0
Choice List:	
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.81      *age\_adj***

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Claims
Variable Type:	Integer Number
Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers

Set Value in Input Manager:  
 Variable Sharing: Not Shared  
 Category Order: 0

### 5.3.3.1.82 *claim\_cost\_factors\_tbl*

Description: Claims Cost Table for Profil Riders  
 Help: Annuity Factors applied to claims of type PHI/FIB/LTC etc.  
 Factors are in respect of 1 shekel monthly benefit.

Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
 Modified By: CLAL-INS\ninab  
 Category: Claims  
 Variable Type: Floating Point Number  
 Default Value: 0  
 Length: 1  
 Number of Decimals: 1  
 Choice List:  
 Character Type: Not Applicable  
 Valid Range From: 0  
 Valid Range To: 0  
 Table Format: Row Numbers  
 Set Value in Input Manager: All  
 Variable Sharing: Not Shared  
 Category Order: 0

### 5.3.3.1.83 *claim\_inflation\_max*

Description: claim\_inflation\_max  
 Help: Maximum cumulative claim inflation allowed (for health products).  
 As a percentage.  
 For example, 200 means that the claims inflation will stop if and when the claims reach double the base assumption.

Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
 Modified By: CLAL-INS\ninab  
 Category: Claims  
 Variable Type: Floating Point Number  
 Default Value: 200  
 Length: 1  
 Number of Decimals: 2  
 Choice List:  
 Character Type: Not Applicable  
 Valid Range From: 100  
 Valid Range To: 1000  
 Table Format: Default Row Numbers  
 Set Value in Input Manager: All

Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.84**      ***claim\_inflation\_max\_re***

Description: reinsurance claim\_inflation\_max  
Help: Maximum cumulative claim inflation allowed (for health products).  
As a percentage.  
For example, 200 means that the claims inflation will stop if and when the claims reach double the base assumption.

Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Claims  
Variable Type: Floating Point Number  
Default Value: 200  
Length: 1  
Number of Decimals: 2  
Choice List:  
Character Type: Not Applicable  
Valid Range From: 100  
Valid Range To: 1000  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.85**      ***claim\_inflation\_perc***

Description: claim\_inflation\_perc  
Help: Annual rate of claim increase for health covers (as a percentage).  
Limited by claim\_inflation\_max.  
Applied from the valuation date.  
If (Read\_from\_Tables="Y") then read from the claims multiple table (duration 0)

Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Claims  
Variable Type: Floating Point Number  
Default Value: 0  
Length: 1  
Number of Decimals: 2  
Choice List:  
Character Type: Not Applicable  
Valid Range From: 0  
Valid Range To: 25  
Table Format: Default Row Numbers



Set Value in Input Manager:  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.86      *claim\_rates\_tbl***

Description: claim rates table for Profil riders  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Claims  
Variable Type: Floating Point Number  
Default Value: 0  
Length: 1  
Number of Decimals: 1  
Choice List:  
Character Type: Not Applicable  
Valid Range From: 0  
Valid Range To: 0  
Table Format: Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.87      *claims\_cost\_factors\_tbl***

Description: claim cost factors table (annuity factors)  
Help: PV of future claim payments (here: as multiplier of the sum assured).  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Claims  
Variable Type: Floating Point Number  
Default Value: 0  
Length: 0  
Number of Decimals: 1  
Choice List:  
Character Type: Not Applicable  
Valid Range From: 0  
Valid Range To: 0  
Table Format: Row Name (numeric)  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.88      *claims\_cost\_key***

Description: claims cost lookup code

Help:	This is the index value to lookup claims cost rates in the claim_cost.tbl. It is looked up initially from the tarif_spec.tbl but may be modified by policy data from the file.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Claims
Variable Type:	Character
Default Value:	0
Length:	22
Number of Decimals:	0
Choice List:	
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.89      *claims\_cost\_key\_rider***

Description:	claims cost lookup code for profil riders
Help:	This is the index value to lookup claims cost rates in the claim_cost.tbl. It is looked up initially from the tarif_spec.tbl but may be modified by policy data from the file.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Claims
Variable Type:	Character
Default Value:	0
Length:	550
Number of Decimals:	0
Choice List:	
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.90      *claims\_cost\_multiplier***

Description:	percentage; modifies basic claims cost
--------------	--

Help:	Applied to basic claims cost rate to account for special covers such as muchav and franchisa whose claims costs are assumed to be a straight percentage higher than the basic claims cost assumption. Also accounts for reduced claims costs of shichrur covers (hachnasa btucha).
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Claims
Variable Type:	Floating Point Number
Default Value:	1
Length:	0
Number of Decimals:	3
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	10
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.91 *claims\_multiplier*

Description:	Claim multiplier by policy year (for non-death)
Help:	Rate multiplier as a percent and is used to adjust rates for expected future improvements/worsening in experience. Array items relate to policy years: year 1=array item 1. Array item 0 is not used. The array contains factors (%) that are multiplied by the basic decrement rates (in addition to the global/constant multiplier).
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Claims
Variable Type:	Floating Point Array
Default Value:	0 100 100
Length:	120
Number of Decimals:	7
Choice List:	
Character Type:	Not Applicable
Valid Range From:	-100
Valid Range To:	1000
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.92**      ***clms\_mult\_col\_key***

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Claims
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	0
Choice List:	
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.93**      ***clms\_mult\_i***

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Claims
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.94**      ***clms\_mult\_i\_col***

Description:	Lookup value code variable
Help:	
Modified On:	7/13/2021 1:45:35 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Claims
Variable Type:	Integer Number

Default Value:	0
Length:	10
Number of Decimals:	0
Choice List:	0
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.95**      *clms\_mult\_infl*

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Claims
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.96**      *clms\_mult\_pol\_yr\_key*

Description:	Lookup value code variable
Help:	
Modified On:	7/13/2021 1:45:43 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Claims
Variable Type:	Integer Number
Default Value:	0
Length:	10
Number of Decimals:	0
Choice List:	0
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers

Set Value in Input Manager:  
 Variable Sharing: Not Shared  
 Category Order: 0

#### **5.3.3.1.97**      ***clms\_mult\_row\_key***

Description: Lookup value code variable  
 Help:  
 Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
 Modified By: CLAL-INS\ninab  
 Category: Claims  
 Variable Type: Character  
 Default Value: 0  
 Length: 10  
 Number of Decimals: 0  
 Choice List:  
 Character Type: Standard  
 Valid Range From:  
 Valid Range To:  
 Table Format: Default Row Numbers  
 Set Value in Input Manager:  
 Variable Sharing: Not Shared  
 Category Order: 0

#### **5.3.3.1.98**      ***clms\_mult\_set\_temp***

Description: Claims multipliers assumption set  
 Help: Claims multipliers assumption set.  
 If read\_from\_table = "Y", then this variable is used, in set\_from\_tables, to lookup the appropriate values for the claims\_mult variable, from the clms\_mult\_tbl table.  
 If lookup\_by\_prodcode = "Y" then this variable is set, in set\_by\_prodcode, to a value from the prodspec\_tbl table.  
 Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
 Modified By: CLAL-INS\ninab  
 Category: Claims  
 Variable Type: Character  
 Default Value: default  
 Length: 7  
 Number of Decimals: 0  
 Choice List:  
 Character Type: Standard  
 Valid Range From: 0  
 Valid Range To: 0  
 Table Format: Default Row Numbers  
 Set Value in Input Manager:  
 Variable Sharing: Not Shared

Category Order: 0

#### **5.3.3.1.99**      ***clms\_mult\_tt***

Description: Lookup value code variable  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Claims  
Variable Type: Floating Point Number  
Default Value: 0  
Length: 0  
Number of Decimals: 1  
Choice List:  
Character Type: Not Applicable  
Valid Range From:  
Valid Range To:  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.100**      ***exp\_col\_key***

Description: Lookup value code variable  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Claims  
Variable Type: Character  
Default Value: 0  
Length: 20  
Number of Decimals: 1  
Choice List: 0  
Character Type: Standard  
Valid Range From:  
Valid Range To:  
Table Format: Default Row Numbers  
Set Value in Input Manager:  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.101**      ***exp\_madad***

Description: Lookup value code variable wildcard  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)

Modified By:	CLAL-INS\ninab
Category:	Claims
Variable Type:	Floating Point Number
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### **5.3.3.1.102      *exp\_row\_key***

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Claims
Variable Type:	Character
Default Value:	0
Length:	20
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### **5.3.3.1.103      *interest***

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Claims
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard



Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.104 *madad*

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Claims
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.105 *min\_ytron\_perc*

Description:	Min ytron perc of SI
Help:	Percentage of SI of minimum DB. Applicable for benefit class "Ytron" only.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Claims
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.106      *prem\_extra***

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Claims
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.107      *prem\_re\_mult***

Description:	Premium re factor by contract
Help:	
Modified On:	1/22/2024 3:53:35 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Claims
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.108      *prof\_comm***

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Claims
Variable Type:	Character

Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.109**      *quota\_share\_ppn*

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Claims
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.110**      *rein\_series\_end\_key\_temp*

Description:	Reinsurance series end data for the temp reins set key
Help:	
Modified On:	10/25/2021 3:08:07 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Claims
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	0
Choice List:	
Character Type:	Standard
Valid Range From:	
Valid Range To:	

Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.111 *retention*

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Claims
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.112 *risk\_rate\_freq\_w*

Description:	frequency of Risk rates in table
Help:	Frequency of risk rate in risk rates table i.e per month (12) or per annum (1)
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Claims
Variable Type:	Integer Number
Default Value:	1
Length:	1
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	1
Valid Range To:	12
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.113 row**

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Claims
Variable Type:	Character
Default Value:	0
Length:	20
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.114 series\_col\_key**

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Claims
Variable Type:	Character
Default Value:	0
Length:	20
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.115 sum\_ins\_incr\_rider**

Description:	Sum Insured increase rate pa (%) for profil riders
Help:	Annual rate of increase in the sum assured
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Claims

Variable Type:	Floating Point Array
Default Value:	0
Length:	25
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	10
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.116 *type*

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Claims
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.117 *use\_phi\_claims\_cf*

Description:	Creat CF of phi claims in payment?
Help:	If "N" claims paid as lump sum by using claims cost factor. If "Y" creat cash flow of claims in payment
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Claims
Variable Type:	Character
Default Value:	Y
Length:	1
Number of Decimals:	0
Choice List:	Y,N
Character Type:	Standard

Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.118 *clwback\_set\_temp*

Description:	Clawback assumption set
Help:	Clawback assumption set. If read_from_tables = "Y", then this variable will be set in set_common_variables.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Clawback
Variable Type:	Character
Default Value:	default
Length:	7
Number of Decimals:	0
Choice List:	
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.119 *comm\_claw\_prpn\_hekef*

Description:	Clawback proportion for nihul com
Help:	Clawback in each policy month as a proportion of the extra initial commission paid. Set from comm_claw_tbl in set_comm_variables
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Clawback
Variable Type:	Floating Point Array
Default Value:	0 0 0
Length:	600
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared

Category Order: 0

### 5.3.3.1.120 *comm\_claw\_prpn\_spv*

Description: Clawback proportion for supervisor com  
 Help: Clawback in each policy month as a proportion of the supervisor commission paid.

When read\_from\_tables = "Y" set based on comm\_claw\_prpn\_tbl and clwback\_set and product\_type in the set\_from\_tables formula

Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
 Modified By: CLAL-INS\ninab  
 Category: Clawback  
 Variable Type: Floating Point Array  
 Default Value: 0 0 0  
 Length: 600  
 Number of Decimals: 0  
 Choice List:  
 Character Type: Not Applicable  
 Valid Range From: 0  
 Valid Range To: 100  
 Table Format: Default Row Numbers  
 Set Value in Input Manager:  
 Variable Sharing: Not Shared  
 Category Order: 0

### 5.3.3.1.121 *comm\_claw\_prpn\_tbl*

Description: Clawback proportion table  
 Help: Table of clawback in each policy month as a proportion of the initial commission paid.

Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
 Modified By: CLAL-INS\ninab  
 Category: Clawback  
 Variable Type: Floating Point Number  
 Default Value: 0  
 Length: 0  
 Number of Decimals: 1  
 Choice List:  
 Character Type: Not Applicable  
 Valid Range From: 0  
 Valid Range To: 0  
 Table Format: Row Name (numeric)  
 Set Value in Input Manager: All  
 Variable Sharing: Not Shared  
 Category Order: 0



**5.3.3.1.122 alloc\_rate\_set**

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Commission
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.123 amala\_0**

Description:	commission field from data file
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Commission
Variable Type:	Floating Point Number
Default Value:	0
Length:	1
Number of Decimals:	3
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	200
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.124 amala\_1**

Description:	commission field from data file
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Commission
Variable Type:	Floating Point Number

Default Value:	0
Length:	1
Number of Decimals:	3
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	200
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.125      *amala\_10***

Description:	commission field from data file
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Commission
Variable Type:	Floating Point Number
Default Value:	0
Length:	1
Number of Decimals:	3
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	200
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.126      *amala\_11***

Description:	commission field from data file
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Commission
Variable Type:	Floating Point Number
Default Value:	0
Length:	1
Number of Decimals:	3
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	200
Table Format:	Default Row Numbers

Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.127      *amala\_12***

Description: commission field from data file  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Commission  
Variable Type: Floating Point Number  
Default Value: 0  
Length: 1  
Number of Decimals: 3  
Choice List:  
Character Type: Not Applicable  
Valid Range From: 0  
Valid Range To: 200  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.128      *amala\_13***

Description: commission field from data file  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Commission  
Variable Type: Floating Point Number  
Default Value: 0  
Length: 1  
Number of Decimals: 3  
Choice List:  
Character Type: Not Applicable  
Valid Range From: 0  
Valid Range To: 200  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.129      *amala\_14***

Description: commission field from data file

Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Commission  
Variable Type: Floating Point Number  
Default Value: 0  
Length: 1  
Number of Decimals: 3  
Choice List:  
Character Type: Not Applicable  
Valid Range From: 0  
Valid Range To: 200  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.130      *amala\_15***

Description: commission field from data file  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Commission  
Variable Type: Floating Point Number  
Default Value: 0  
Length: 1  
Number of Decimals: 3  
Choice List:  
Character Type: Not Applicable  
Valid Range From: 0  
Valid Range To: 200  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.131      *amala\_16***

Description: commission field from data file  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Commission  
Variable Type: Floating Point Number  
Default Value: 0  
Length: 1  
Number of Decimals: 3

Choice List:  
Character Type: Not Applicable  
Valid Range From: 0  
Valid Range To: 200  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.132      *amala\_2***

Description: commission field from data file  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Commission  
Variable Type: Floating Point Number  
Default Value: 0  
Length: 1  
Number of Decimals: 3  
Choice List:  
Character Type: Not Applicable  
Valid Range From: 0  
Valid Range To: 200  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.133      *amala\_3***

Description: commission field from data file  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Commission  
Variable Type: Floating Point Number  
Default Value: 0  
Length: 1  
Number of Decimals: 3  
Choice List:  
Character Type: Not Applicable  
Valid Range From: 0  
Valid Range To: 200  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

**5.3.3.1.134      amala\_4**

Description:	commission field from data file
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Commission
Variable Type:	Floating Point Number
Default Value:	0
Length:	1
Number of Decimals:	3
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	200
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.135      amala\_5**

Description:	commission field from data file
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Commission
Variable Type:	Floating Point Number
Default Value:	0
Length:	1
Number of Decimals:	3
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	200
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.136      amala\_6**

Description:	commission field from data file
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Commission

Variable Type:	Floating Point Number
Default Value:	0
Length:	1
Number of Decimals:	3
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	200
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.137**     ***amala\_7***

Description:	commission field from data file
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Commission
Variable Type:	Floating Point Number
Default Value:	0
Length:	1
Number of Decimals:	3
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	200
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.138**     ***amala\_8***

Description:	commission field from data file
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Commission
Variable Type:	Floating Point Number
Default Value:	0
Length:	1
Number of Decimals:	3
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	200

Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.139**     *amala\_9*

Description:	commission field from data file
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Commission
Variable Type:	Floating Point Number
Default Value:	0
Length:	1
Number of Decimals:	3
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	200
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.140**     *amala\_nihul\_0*

Description:	commission field from data file
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Commission
Variable Type:	Floating Point Number
Default Value:	0
Length:	1
Number of Decimals:	3
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	200
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0



**5.3.3.1.141      *amala\_nihul\_1***

Description:	commission field from data file
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Commission
Variable Type:	Floating Point Number
Default Value:	0
Length:	1
Number of Decimals:	3
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	200
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.142      *amala\_nihul\_10***

Description:	commission field from data file
Help:	
Modified On:	4/9/2024 5:40:26 PM (UTC+03:00)
Modified By:	CLAL-INS\Arikt
Category:	Commission
Variable Type:	Floating Point Number
Default Value:	0
Length:	1
Number of Decimals:	3
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	200
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.143      *amala\_nihul\_11***

Description:	commission field from data file
Help:	
Modified On:	4/9/2024 5:40:33 PM (UTC+03:00)
Modified By:	CLAL-INS\Arikt
Category:	Commission
Variable Type:	Floating Point Number

Default Value:	0
Length:	1
Number of Decimals:	3
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	200
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.144**     *amala\_nihul\_12*

Description:	commission field from data file
Help:	
Modified On:	4/9/2024 5:40:34 PM (UTC+03:00)
Modified By:	CLAL-INSArict
Category:	Commission
Variable Type:	Floating Point Number
Default Value:	0
Length:	1
Number of Decimals:	3
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	200
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.145**     *amala\_nihul\_13*

Description:	commission field from data file
Help:	
Modified On:	4/9/2024 5:40:35 PM (UTC+03:00)
Modified By:	CLAL-INSArict
Category:	Commission
Variable Type:	Floating Point Number
Default Value:	0
Length:	1
Number of Decimals:	3
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	200
Table Format:	Default Row Numbers

Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.146      *amala\_nihul\_14***

Description: commission field from data file  
Help:  
Modified On: 4/9/2024 5:40:36 PM (UTC+03:00)  
Modified By: CLAL-INS\Arikt  
Category: Commission  
Variable Type: Floating Point Number  
Default Value: 0  
Length: 1  
Number of Decimals: 3  
Choice List:  
Character Type: Not Applicable  
Valid Range From: 0  
Valid Range To: 200  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.147      *amala\_nihul\_15***

Description: commission field from data file  
Help:  
Modified On: 4/9/2024 5:40:37 PM (UTC+03:00)  
Modified By: CLAL-INS\Arikt  
Category: Commission  
Variable Type: Floating Point Number  
Default Value: 0  
Length: 1  
Number of Decimals: 3  
Choice List:  
Character Type: Not Applicable  
Valid Range From: 0  
Valid Range To: 200  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.148      *amala\_nihul\_16***

Description: commission field from data file

Help:  
Modified On: 4/9/2024 5:40:38 PM (UTC+03:00)  
Modified By: CLAL-INS\Arikt  
Category: Commission  
Variable Type: Floating Point Number  
Default Value: 0  
Length: 1  
Number of Decimals: 3  
Choice List:  
Character Type: Not Applicable  
Valid Range From: 0  
Valid Range To: 200  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.149      *amala\_nihul\_2***

Description: commission field from data file  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Commission  
Variable Type: Floating Point Number  
Default Value: 0  
Length: 1  
Number of Decimals: 3  
Choice List:  
Character Type: Not Applicable  
Valid Range From: 0  
Valid Range To: 200  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.150      *amala\_nihul\_3***

Description: commission field from data file  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Commission  
Variable Type: Floating Point Number  
Default Value: 0  
Length: 1  
Number of Decimals: 3

Choice List:  
Character Type: Not Applicable  
Valid Range From: 0  
Valid Range To: 200  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.151      *amala\_nihul\_4***

Description: commission field from data file  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Commission  
Variable Type: Floating Point Number  
Default Value: 0  
Length: 1  
Number of Decimals: 3  
Choice List:  
Character Type: Not Applicable  
Valid Range From: 0  
Valid Range To: 200  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.152      *amala\_nihul\_5***

Description: commission field from data file  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Commission  
Variable Type: Floating Point Number  
Default Value: 0  
Length: 1  
Number of Decimals: 3  
Choice List:  
Character Type: Not Applicable  
Valid Range From: 0  
Valid Range To: 200  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

**5.3.3.1.153      *amala\_nihul\_6***

Description:	commission field from data file
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Commission
Variable Type:	Floating Point Number
Default Value:	0
Length:	1
Number of Decimals:	3
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	200
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.154      *amala\_nihul\_7***

Description:	commission field from data file
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Commission
Variable Type:	Floating Point Number
Default Value:	0
Length:	1
Number of Decimals:	3
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	200
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.155      *amala\_nihul\_8***

Description:	commission field from data file
Help:	
Modified On:	4/9/2024 5:40:18 PM (UTC+03:00)
Modified By:	CLAL-INS\Arikt
Category:	Commission

Variable Type:	Floating Point Number
Default Value:	0
Length:	1
Number of Decimals:	3
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	200
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.156**     *amala\_nihul\_9*

Description:	commission field from data file
Help:	
Modified On:	4/9/2024 5:40:24 PM (UTC+03:00)
Modified By:	CLAL-INS\Arikt
Category:	Commission
Variable Type:	Floating Point Number
Default Value:	0
Length:	1
Number of Decimals:	3
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	200
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.157**     *amala\_pikuach\_0*

Description:	supervisor commission field from data file
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Commission
Variable Type:	Floating Point Number
Default Value:	0
Length:	1
Number of Decimals:	3
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	200

Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.158**     *amala\_pikuach\_1*

Description:	supervisor commission field from data file
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Commission
Variable Type:	Floating Point Number
Default Value:	0
Length:	1
Number of Decimals:	3
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	200
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.159**     *clms\_mult\_set*

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Commission
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0



**5.3.3.1.160      *clwback\_set***

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Commission
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.161      *comm\_claw\_row\_key***

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Commission
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	0
Choice List:	
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.162      *comm\_ext\_col\_key***

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Commission
Variable Type:	Character

Default Value:	0
Length:	10
Number of Decimals:	0
Choice List:	
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.163      *comm\_extra\_agent\_tbl***

Description:	Extra commission table (%) by Osek Mureshe
Help:	Super commission expressed as a % of initial regular commissions or % of premium.
Modified On:	3/15/2020 3:45:54 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Commission
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	9999999999
Table Format:	Row Name (numeric)
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.164      *comm\_extra\_agent\_use***

Description:	Read extra comms from agent table ?
Help:	Y= overwrite defaults variable values and assumption screen inputs of data with values read from external tables.  N = do not overwrite defaults variable values and assumption screen inputs of data with values read from external tables.  Set to N for the profit testing user group since this group will use the assumptions screen inputs.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab

Category:	Commission
Variable Type:	Character
Default Value:	N
Length:	1
Number of Decimals:	0
Choice List:	Y,N
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.165 *comm\_extra\_tbl*

Description:	Extra commission table (%)
Help:	Extra and Shimur Tik commissions (manual commns) expressed as a % of New Premium by Production Year or all years (if year is omitted).
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Commission
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	0
Table Format:	Row Name (numeric)
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.166 *comm\_extra1*

Description:	Initial commission (%) paid at start
Help:	Initial commission as a percentage of 1st annual premium . Paid at start
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Commission
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	2

Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	200
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.167      *comm\_extra1\_sav***

Description:	Extra commission (%) paid at start on savings
Help:	Extra %. Commision expressed as a percentage of initial annualized SAVINGS gross premium - WITH clawback.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Commission
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	200
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.168      *comm\_extra2***

Description:	Initial comm (%) paid at end cal. yr 1.
Help:	Initial commission as a percentage of 1st annual premium . Paid at the end of the 1st calendar year.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Commission
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	200
Table Format:	Default Row Numbers

Set Value in Input Manager:  
 Variable Sharing: Not Shared  
 Category Order: 0

### 5.3.3.1.169 *comm\_fix*

Description: Monthly fixed commission (shk)  
 Help: Monthly fixed commission set by policy year.  
 "dimay tipul" is under consideration at clal but not yet used.

If read\_from\_tables=Y then value set in set\_from\_tables formula.  
 Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
 Modified By: CLAL-INS\ninab  
 Category: Commission  
 Variable Type: Floating Point Array  
 Default Value: 0  
 Length: 116  
 Number of Decimals: 2  
 Choice List:  
 Character Type: Not Applicable  
 Valid Range From: 0  
 Valid Range To: 100  
 Table Format: Default Row Numbers  
 Set Value in Input Manager:  
 Variable Sharing: Not Shared  
 Category Order: 0

### 5.3.3.1.170 *comm\_hekef\_pc*

Description: Hekef commission (%) paid at start  
 Help: Prizes % commision expressed as a percentage of initial annualized gross premium with no clawback.

Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
 Modified By: CLAL-INS\ninab  
 Category: Commission  
 Variable Type: Floating Point Number  
 Default Value: 0  
 Length: 0  
 Number of Decimals: 2  
 Choice List:  
 Character Type: Not Applicable  
 Valid Range From: 0  
 Valid Range To: 100  
 Table Format: Default Row Numbers  
 Set Value in Input Manager:  
 Variable Sharing: Not Shared

Category Order: 0

#### **5.3.3.1.171      *comm\_hekef\_pc\_res***

Description: Hekef commission (%) paid at start on reserve  
Help:  
Modified On: 2/20/2020 10:31:44 AM (UTC+02:00)  
Modified By: CLAL-INS\NinaB  
Category: Commission  
Variable Type: Floating Point Number  
Default Value: 0  
Length: 0  
Number of Decimals: 2  
Choice List:  
Character Type: Not Applicable  
Valid Range From: 0  
Valid Range To: 100  
Table Format: Default Row Numbers  
Set Value in Input Manager:  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.172      *comm\_hekef\_pc\_rider***

Description: Profil rider hekef commission (%) paid at start  
Help: Prizes % commision expressed as a percentage of initial annualized gross premium with no clawback.  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Commission  
Variable Type: Floating Point Number  
Default Value: 0  
Length: 0  
Number of Decimals: 2  
Choice List:  
Character Type: Not Applicable  
Valid Range From: 0  
Valid Range To: 100  
Table Format: Default Row Numbers  
Set Value in Input Manager:  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.173      *comm\_hekef\_pc\_sav***

Description: Hekef commission (%) paid for saving at start

Help:	Prizes % commision expressed as a percentage of initial annualized gross premium with no clawback.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Commission
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.174      *comm\_min\_prem\_term***

Description:	Minimum term (in months) for full commission
Help:	Minimum premium term for agent to receive 100% commission. If there is no such minimum, then this should be set to 0.  This is applied to regular initial commission only.  For LTC (ben-class = "LTC") this is ignored, and instead a special formula based on age at issue is applied.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Commission
Variable Type:	Floating Point Number
Default Value:	72
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1000000
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.175 comm\_nihul\_rate**

Description:	Annual Nihul commission (%)
Help:	Annual regular comm %. Read from table in set_comm_variables() .
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Commission
Variable Type:	Floating Point Array
Default Value:	0
Length:	116
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.176 comm\_perc\_res\_a**

Description:	Commissions as % of reserves (basic)
Help:	Commission as a % of units, or as a % of the management fee. If it is <= 1.0 then it is interpreted as a % of units (used for Adif and Profil normally) and if it is > 1.0 then it is interpreted as a % of management fees ("mgt_fee_fixed") but limited so that the minimum management fee will be "mgt_fee_min". For Adif this variable is for the basic part only, for Profi & Kav Chadashl it is for the total including pure savings.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Commission
Variable Type:	Floating Point Array
Default Value:	0
Length:	116
Number of Decimals:	3
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0



**5.3.3.1.177 comm\_perc\_res\_a\_input**

Description:	Commissions as % of reserves (basic)
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Commission
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	3
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.178 comm\_perc\_res\_b**

Description:	Commission. as % of reserves (Pure savin
Help:	Commissions as a % of pure saving part of reserves held. Loaded from table. Only if Adif (For Profil pure savings and basic parts the comm_perc_res_a varianble is used)
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Commission
Variable Type:	Floating Point Array
Default Value:	0
Length:	116
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1000
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.179 comm\_perc\_res\_b\_input**

Description:	Commissions as % of reserves (Pure savings)
Help:	

Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Commission
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1000
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### **5.3.3.1.180      *comm\_prizes\_pc***

Description:	Prizes commission (%) paid at start
Help:	Prizes % commision expressed as a percentage of initial annualized gross premium with no clawback.
Modified On:	3/15/2020 11:06:28 AM (UTC+02:00)
Modified By:	CLAL-INS\NinaB
Category:	Commission
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### **5.3.3.1.181      *comm\_prizes\_pc\_res***

Description:	Prize commission (%) paid at start on reserve
Help:	
Modified On:	11/25/2020 8:37:44 AM (UTC+02:00)
Modified By:	CLAL-INS\NinaB
Category:	Commission
Variable Type:	Floating Point Number
Default Value:	0
Length:	0

Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.182 *comm\_prizes\_pc\_sav*

Description:	Prizes commission (%) paid at start on savings
Help:	Prizes % commission expressed as a percentage of initial annualized savings gross premium with no clawback.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Commission
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.183 *comm\_prof*

Description:	Profit commission (%) - shareholder
Help:	Renewal commission expressed as a % of premium income.
Modified On:	1/11/2023 9:13:17 AM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Commission
Variable Type:	Floating Point Number
Default Value:	0
Length:	116
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100

Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.184 *comm\_reg\_tot\_w*

Description:	total annual regular commission (%)
Help:	total Annual regular comm %. Calculated variable : in set_other_variables
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Commission
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1000
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.185 *comm\_regular\_pc*

Description:	Annual regular commission (%)
Help:	Annual regular comm %. Read from table in set_comm_variables() .
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Commission
Variable Type:	Floating Point Array
Default Value:	0
Length:	116
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.186      *comm\_ren\_perc\_prem***

Description:	Renewal commission (%)
Help:	Renewal commission expressed as a % of premium income.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Commission
Variable Type:	Floating Point Number
Default Value:	0
Length:	116
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.187      *comm\_ren\_perc\_prem\_mrtg***

Description:	Renewal Commission (%) for mortgage policies sold after 04/2007 and after 16 yrs vetek
Help:	
Modified On:	1/25/2022 12:07:01 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Commission
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.188      *comm\_ren\_perc\_sav***

Description:	Renewal commission (%) for pure saving
Help:	Renewal commission expressed as a % of pure saving premium income. Used when benef class is adif
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)

Modified By:	CLAL-INS\ninab
Category:	Commission
Variable Type:	Floating Point Number
Default Value:	1
Length:	116
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.189 *comm\_ren\_shimur*

Description:	Shimur Tik Renewal commission (%)
Help:	Shimur Tik Renewal commission expressed as a % of premium income for portfolio IF as at 12/2003
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Commission
Variable Type:	Floating Point Number
Default Value:	0
Length:	116
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.190 *comm\_renewal\_year*

Description:	First year from when renewal commission is paid
Help:	policy year from when renewal commission is paid.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Commission
Variable Type:	Integer Number
Default Value:	1
Length:	0

Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	1
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.191**      ***comm\_renewal\_year\_input***

Description:	First year from when renewal commission is paid
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Commission
Variable Type:	Floating Point Number
Default Value:	1
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	1
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.192**      ***comm\_set***

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Commission
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All

Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.193 *comm\_set\_temp*

Description:	Commission assumption set
Help:	Commission assumption set. If read_from_table = "Y", then this variable will be set in set_common_variables.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Commission
Variable Type:	Character
Default Value:	default
Length:	7
Number of Decimals:	0
Choice List:	
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.194 *comm\_spvisor*

Description:	Annual supervisor commission (%)
Help:	Annual supervisor comm %. Read from table in set_comm_variables() .
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Commission
Variable Type:	Floating Point Array
Default Value:	0
Length:	116
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0



**5.3.3.1.195      *decrem\_mult\_set***

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Commission
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.196      *exp\_mult\_set***

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Commission
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.197      *exp\_set\_cvr***

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Commission
Variable Type:	Character

Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.198**     *exp\_set\_pol*

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Commission
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.199**     *key\_temp*

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Commission
Variable Type:	Character
Default Value:	0
Length:	21
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers

Set Value in Input Manager:  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.200**     ***lapse\_set***

Description: Lookup value code variable wildcard  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Commission  
Variable Type: Character  
Default Value: 0  
Length: 10  
Number of Decimals: 1  
Choice List: 0  
Character Type: Standard  
Valid Range From:  
Valid Range To:  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.201**     ***lapse\_set\_riders***

Description: Lookup value code variable wildcard  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Commission  
Variable Type: Character  
Default Value: lapse\_set\_riders  
Length: 20  
Number of Decimals: 1  
Choice List: lapse\_set\_riders  
Character Type: Standard  
Valid Range From:  
Valid Range To:  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.202**     ***mort\_mult\_set***

Description: Lookup value code variable wildcard

Help:  
 Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
 Modified By: CLAL-INS\ninab  
 Category: Commission  
 Variable Type: Character  
 Default Value: 0  
 Length: 10  
 Number of Decimals: 1  
 Choice List: 0  
 Character Type: Standard  
 Valid Range From:  
 Valid Range To:  
 Table Format: Default Row Numbers  
 Set Value in Input Manager: All  
 Variable Sharing: Not Shared  
 Category Order: 0

### 5.3.3.1.203 *om*

Description: Lookup value code variable  
 Help:  
 Modified On: 3/15/2020 11:09:26 AM (UTC+02:00)  
 Modified By: CLAL-INS\NinaB  
 Category: Commission  
 Variable Type: Floating Point Number  
 Default Value: 0  
 Length: 0  
 Number of Decimals: 1  
 Choice List:  
 Character Type: Not Applicable  
 Valid Range From: 0  
 Valid Range To:  
 Table Format: Default Row Numbers  
 Set Value in Input Manager:  
 Variable Sharing: Not Shared  
 Category Order: 0

### 5.3.3.1.204 *prod\_assumpt\_rider\_clms\_tbl*

Description: Lookup value for clms table  
 Help:  
 Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
 Modified By: CLAL-INS\ninab  
 Category: Commission  
 Variable Type: Character  
 Default Value: 0  
 Length: 10  
 Number of Decimals: 1

Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.205      *profil\_rider\_comm1\_6***

Description:	average profil rider comm rate for years 1 to 6
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Commission
Variable Type:	Floating Point Array
Default Value:	0
Length:	25
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.206      *profil\_rider\_comm7***

Description:	average profil rider comm rate for years 7+
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Commission
Variable Type:	Floating Point Array
Default Value:	0
Length:	25
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.207      *surr\_chg\_set***

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Commission
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.208      *temp\_comm\_set***

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Commission
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.209      *dac\_amort\_per***

Description:	DAC amortisation period(months)
Help:	DAC amortisation period (in months). Set to benefit term in startup if DAC_amort_type is "lifetime".
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)

Modified By:	CLAL-INS\ninab
Category:	DAC
Variable Type:	Integer Number
Default Value:	180
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1000
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.210 ***dac\_amort\_per\_tax***

Description:	dac_tax amort period(months)
Help:	DAC_tax amortisation period (in months). This period starts from expense payment date.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	DAC
Variable Type:	Integer Number
Default Value:	48
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	1
Valid Range To:	1000
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.211 ***dac\_amort\_type***

Description:	DAC amortisation type
Help:	DAC amortisation type
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	DAC
Variable Type:	Character
Default Value:	Fixed
Length:	8
Number of Decimals:	0
Choice List:	Fixed,Lifetime

Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.212 ***dac\_book\_adj\_factor***

Description:	Adjustment factor for Dac book (to scale up DAC to actuals)
Help:	The percentage of the DAC for books that has to be retained in the company's capital (0%/30%/100% according to current regulations.) Set in set_other_variables.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	DAC
Variable Type:	Floating Point Number
Default Value:	100
Length:	0
Number of Decimals:	5
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1000
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.213 ***dac\_book\_adj\_factor\_input***

Description:	Adjustment factor for Dac book (to scale up DAC to actuals)
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	DAC
Variable Type:	Floating Point Number
Default Value:	100
Length:	0
Number of Decimals:	5
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1000
Table Format:	Default Row Numbers



Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.214      *dac\_book\_inforce***

Description:	DAC books from inforce
Help:	variable linked with the reserve field from the inforce file
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	DAC
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100000000
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.215      *dac\_book\_inforce\_input***

Description:	DAC books from inforce
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	DAC
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100000000
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.216      *dac\_cap\_apply***

Description:	Apply capital requirement of x% of DAC
--------------	--

Help:	The percentage of the DAC for books that has to be retained in the company's capital
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	DAC
Variable Type:	Character
Default Value:	Y
Length:	1
Number of Decimals:	0
Choice List:	Y,N
Character Type:	Standard
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.217      *dac\_cap\_perc\_w***

Description:	Perc. of DAC held as capital
Help:	The percentage of the DAC for books that has to be retained in the company's capital (0%/30%/100% according to current regulations.) Set in set_other_variables.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	DAC
Variable Type:	Floating Point Number
Default Value:	30
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.218      *dac\_tax\_adj\_factor***

Description:	Adjustment factor for Dac tax (to scale up DAC to actuals)
Help:	The percentage applied to the DAC tax value taken from the data file to adjust it according to the actual DAC held.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)

Modified By:	CLAL-INS\ninab
Category:	DAC
Variable Type:	Floating Point Number
Default Value:	100
Length:	0
Number of Decimals:	5
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1000
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.219      *dac\_tax\_adj\_factor\_input***

Description:	Adjustment factor for Dac tax (to scale up DAC to actuals)
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	DAC
Variable Type:	Floating Point Number
Default Value:	100
Length:	0
Number of Decimals:	5
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1000
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.220      *dac\_tax\_inforce***

Description:	DAC tax or Zillmer from inforce
Help:	variable linked with the reserve field from the inforce file
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	DAC
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	2

Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100000000
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.221      *dac\_tax\_inforce\_input***

Description:	DAC tax or Zillmer from inforce
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	DAC
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100000000
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.222      *dac\_type\_temp***

Description:	deferred acquisitions cost type
Help:	deferred acquisitions cost type/method used in dac or zillmer calculations.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	DAC
Variable Type:	Character
Default Value:	il_dac
Length:	6
Number of Decimals:	0
Choice List:	il_dac,none
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared

Category Order: 0

#### **5.3.3.1.223     *ann\_def\_res\_inv\_margin***

Description: investment margin % for annuity reserve  
Help: Investment margin used as discount rate for annuity deficiency reserve  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Economic assumptions  
Variable Type: Floating Point Number  
Default Value: 0.3  
Length: 0  
Number of Decimals: 3  
Choice List:  
Character Type: Not Applicable  
Valid Range From: 0  
Valid Range To: 10  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.224     *ann\_def\_res\_inv\_margin\_par***

Description: investment margin % for annuity reserve for Participating funds  
Help: Investment margin used as discount rate for annuity deficiency reserve  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Economic assumptions  
Variable Type: Floating Point Number  
Default Value: 0.7  
Length: 0  
Number of Decimals: 3  
Choice List:  
Character Type: Not Applicable  
Valid Range From: 0  
Valid Range To: 10  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.225     *ann\_disc\_rate\_m***

Description: Lookup value code variable wildcard

Help:  
Modified On: 8/25/2022 2:03:37 PM (UTC+03:00)  
Modified By: CLAL-INS\ahuvaa  
Category: Economic assumptions  
Variable Type: Floating Point Number  
Default Value: 0  
Length: 0  
Number of Decimals: 1  
Choice List:  
Character Type: Not Applicable  
Valid Range From:  
Valid Range To:  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.226     *ann\_inv\_rate\_m***

Description: Lookup value code variable wildcard  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Economic assumptions  
Variable Type: Floating Point Number  
Default Value: 0  
Length: 0  
Number of Decimals: 1  
Choice List:  
Character Type: Not Applicable  
Valid Range From:  
Valid Range To:  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.227     *ann\_inv\_rate\_m\_ifrs***

Description: Lookup value code variable wildcard  
Help:  
Modified On: 6/27/2024 12:42:42 PM (UTC+03:00)  
Modified By: CLAL-INS\arikt  
Category: Economic assumptions  
Variable Type: Floating Point Number  
Default Value: 0  
Length: 0  
Number of Decimals: 1

Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.228 *ann\_inv\_rate\_mth\_t*

Description:	Mthly ann investment income rate by projn month t
Help:	Monthly investment income rate. Calculated from inv_rate in set_exp_variables. This varies by calender year, in the same way as inv_rate
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Economic assumptions
Variable Type:	Floating Point Array
Default Value:	0
Length:	120
Number of Decimals:	9
Choice List:	
Character Type:	Not Applicable
Valid Range From:	-1
Valid Range To:	1
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.229 *ann\_inv\_rate\_mth\_t\_ifrs*

Description:	Mthly ann investment income rate by projn month t - IFRS
Help:	Monthly investment income rate. Calculated from inv_rate in set_exp_variables. This varies by calender year, in the same way as inv_rate
Modified On:	6/27/2024 2:09:53 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Economic assumptions
Variable Type:	Floating Point Array
Default Value:	0
Length:	120
Number of Decimals:	9
Choice List:	
Character Type:	Not Applicable
Valid Range From:	-1

Valid Range To:	1
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.230 *ann\_inv\_rate\_rf\_mth\_t*

Description:	Mthly free investment income rate by projn month t
Help:	Monthly investment income rate. Calculated from inv_rate in set_exp_variables. This varies by calender year, in the same way as inv_rate
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Economic assumptions
Variable Type:	Floating Point Array
Default Value:	0
Length:	120
Number of Decimals:	9
Choice List:	
Character Type:	Not Applicable
Valid Range From:	-1
Valid Range To:	1
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.231 *ann\_series*

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Economic assumptions
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0



**5.3.3.1.232      *ann\_v\_month\_t***

Description:	Ann Discount rate (v) monthly by projection month t
Help:	Monthly v factor by month t for MCEV calculations in the Experience model and reserve calculations in the Net_Prem model. Input in assumption set
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Economic assumptions
Variable Type:	Floating Point Array
Default Value:	0.0025
Length:	120
Number of Decimals:	8
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.233      *ann\_v\_month\_t\_ifrs***

Description:	Ann Discount rate (v) monthly by projection month t
Help:	Monthly v factor by month t for MCEV calculations in the Experience model and reserve calculations in the Net_Prem model. Input in assumption set
Modified On:	6/27/2024 12:50:49 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Economic assumptions
Variable Type:	Floating Point Array
Default Value:	0.0025
Length:	120
Number of Decimals:	8
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.234      *cap\_req\_perc\_premium***

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Economic assumptions
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.235      *cap\_req\_perc\_reserve***

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Economic assumptions
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.236      *commres\_addvat***

Description:	add vat to comm reserves
Help:	
Modified On:	11/14/2024 4:49:37 PM (UTC+02:00)
Modified By:	CLAL-INS\arikt
Category:	Economic assumptions
Variable Type:	Character

Default Value:	N
Length:	1
Number of Decimals:	0
Choice List:	Y,N
Character Type:	Standard
Valid Range From:	-100
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.237 *cu\_discounted*

Description:	discount coverage-units
Help:	
Modified On:	8/1/2024 11:50:20 AM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Economic assumptions
Variable Type:	Character
Default Value:	N
Length:	1
Number of Decimals:	0
Choice List:	Y,N
Character Type:	Standard
Valid Range From:	-100
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.238 *dac\_book\_fac*

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Economic assumptions
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers

Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.239      *dac\_tax\_fac***

Description: Lookup value code variable wildcard  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Economic assumptions  
Variable Type: Floating Point Number  
Default Value: 0  
Length: 0  
Number of Decimals: 1  
Choice List:  
Character Type: Not Applicable  
Valid Range From:  
Valid Range To:  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.240      *disc\_rate\_m***

Description: Lookup value code variable wildcard  
Help:  
Modified On: 8/25/2022 2:03:19 PM (UTC+03:00)  
Modified By: CLAL-INS\ahuvaa  
Category: Economic assumptions  
Variable Type: Floating Point Number  
Default Value: 0  
Length: 0  
Number of Decimals: 1  
Choice List:  
Character Type: Not Applicable  
Valid Range From:  
Valid Range To:  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.241      *ev\_disc\_rate***

Description: EV discount rate (%)

Help:	Annual discount rate for embedded value valuations.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Economic assumptions
Variable Type:	Floating Point Number
Default Value:	12
Length:	30
Number of Decimals:	3
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.242      *ev\_discount\_rate\_type***

Description:	Use Vector or Single rate
Help:	EV Dscount rate to use for the run: Vector= uses the input discount vector (v-month_t) Single = replaces discount vector based on the single input rate
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Economic assumptions
Variable Type:	Character
Default Value:	Vector
Length:	7
Number of Decimals:	0
Choice List:	Single,Vector,Earned
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.243      *fixed\_mgt\_fee***

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Economic assumptions

Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.244**      ***fixed\_mgt\_fee\_term***

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Economic assumptions
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.245**      ***free\_inv\_prop\_t***

Description:	Free investment proportion by projn month t
Help:	Monthly investment income rate. Calculated from inv_rate in set_exp_variables. This varies by calender year, in the same way as inv_rate
Modified On:	4/2/2023 1:55:08 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Economic assumptions
Variable Type:	Floating Point Array
Default Value:	1
Length:	120
Number of Decimals:	9
Choice List:	
Character Type:	Not Applicable

Valid Range From:	-1
Valid Range To:	1
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.246**     *free\_inv\_ratio\_tbl*

Description:	Free investment ratio by fund
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Economic assumptions
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	0
Table Format:	Row Name (numeric)
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.247**     *fund\_rates\_code\_tbl*

Description:	Table of parameters by fund
Help:	Read in set_exp_variables.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Economic assumptions
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	0
Table Format:	Row Name (numeric)
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.248**     *int\_rate\_res*

Description:	interest rate (%) for reserves
Help:	Annual rate.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Economic assumptions
Variable Type:	Floating Point Number
Default Value:	4
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	20
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.249**     *intres*

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Economic assumptions
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.250**     *intres\_puresav*

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Economic assumptions
Variable Type:	Floating Point Number



Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.251**     *inv\_rate\_m*

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Economic assumptions
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.252**     *inv\_rate\_m\_ifrs*

Description:	Lookup value code variable wildcard
Help:	
Modified On:	6/27/2024 12:43:35 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Economic assumptions
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers

Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.253 *inv\_rate\_mth\_t*

Description:	Mthly investment income rate by projn month t
Help:	Monthly investment income rate. Calculated from inv_rate in set_exp_variables. This varies by calender year, in the same way as inv_rate
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Economic assumptions
Variable Type:	Floating Point Array
Default Value:	0
Length:	120
Number of Decimals:	9
Choice List:	
Character Type:	Not Applicable
Valid Range From:	-1
Valid Range To:	1
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.254 *inv\_rate\_mth\_t\_ifrs*

Description:	Mthly investment income rate by projn month t - IFRS
Help:	Monthly investment income rate. Calculated from inv_rate in set_exp_variables. This varies by calender year, in the same way as inv_rate
Modified On:	6/27/2024 2:09:05 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Economic assumptions
Variable Type:	Floating Point Array
Default Value:	0
Length:	120
Number of Decimals:	9
Choice List:	
Character Type:	Not Applicable
Valid Range From:	-1
Valid Range To:	1
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.255**     *inv\_rate\_rf\_mth\_t*

Description:	Mthly free investment income rate by projn month t
Help:	Monthly investment income rate. Calculated from inv_rate in set_exp_variables. This varies by calender year, in the same way as inv_rate
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Economic assumptions
Variable Type:	Floating Point Array
Default Value:	0
Length:	120
Number of Decimals:	9
Choice List:	
Character Type:	Not Applicable
Valid Range From:	-1
Valid Range To:	1
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.256**     *inv\_rate\_rm\_m*

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/25/2022 9:44:50 AM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Economic assumptions
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.257**     *inv\_rate\_rollup*

Description:	Investment income rate (%) for the rollup period
--------------	--

Help:	Annual investment income rate on free assets. This is average with the fund-specific rate on special bonds, to obtain inv_rate_mth_w which is used in the projection. This may vary by calendar year. The array index = calendar year - valn_year + 1. In the fund rate table inv_free may be entered for a specific calendar year.
Modified On:	9/12/2019 11:23:33 AM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Category:	Economic assumptions
Variable Type:	Floating Point Number
Default Value:	0
Length:	30
Number of Decimals:	9
Choice List:	
Character Type:	Not Applicable
Valid Range From:	-1
Valid Range To:	1
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.258 *invinc*

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Economic assumptions
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.259 *max\_chetz*

Description:	Lookup value code variable wildcard
Help:	
Modified On:	6/27/2024 12:41:21 PM (UTC+03:00)

Modified By:	CLAL-INS\arikt
Category:	Economic assumptions
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.260**     ***mort\_addn***

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Economic assumptions
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.261**     ***mort\_res***

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Economic assumptions
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable

Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.262 *mortg\_int*

Description:	Annual interest (%) on mortgage
Help:	Annual interest for benefit class "mortg" (mortgage).
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Economic assumptions
Variable Type:	Floating Point Number
Default Value:	4
Length:	0
Number of Decimals:	3
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.263 *mortg\_int\_mth\_w*

Description:	Monthly interest (%) on mortgage
Help:	Monthly interest for benefit class "mortg" (mortgage). Calculated in startup
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Economic assumptions
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	3
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.264      *phi\_res\_discount\_rate\_type***

Description:	Use RF or Single rate
Help:	EV Dscount rate to use for the run: Vector= uses the input discount vector (v-month_t) Single = replaces discount vector based on the single input rate
Modified On:	1/18/2024 2:20:52 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Economic assumptions
Variable Type:	Character
Default Value:	Vector
Length:	7
Number of Decimals:	0
Choice List:	Single,Vector,Earned
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.265      *res\_fac\_row\_key***

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Economic assumptions
Variable Type:	Integer Number
Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.266      *risk\_free\_row\_key***

Description:	Lookup value code variable
--------------	----------------------------

Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Economic assumptions  
Variable Type: Integer Number  
Default Value: 0  
Length: 0  
Number of Decimals: 0  
Choice List:  
Character Type: Not Applicable  
Valid Range From:  
Valid Range To:  
Table Format: Default Row Numbers  
Set Value in Input Manager:  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.267      *tax\_rate***

Description: Lookup value coode variable wildcard  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Economic assumptions  
Variable Type: Floating Point Number  
Default Value: 0  
Length: 0  
Number of Decimals: 1  
Choice List:  
Character Type: Not Applicable  
Valid Range From:  
Valid Range To:  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.268      *temp\_fund***

Description: Lookup value code variable  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Economic assumptions  
Variable Type: Character  
Default Value: 0  
Length: 10  
Number of Decimals: 1



Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.269 ***v\_month\_t***

Description:	Discount rate (v) monthly by projection month t
Help:	Monthly v factor by month t for MCEV calculations in the Experience model and reserve calculations in the Net_Prem model. Input in assumption set
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Economic assumptions
Variable Type:	Floating Point Array
Default Value:	0.0025
Length:	120
Number of Decimals:	8
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.270 ***v\_month\_t\_ifrs***

Description:	Discount rate (v) monthly by projection month t - IFRS
Help:	Monthly v factor by month t for MCEV calculations in the Experience model and reserve calculations in the Net_Prem model. Input in assumption set
Modified On:	6/27/2024 12:51:18 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Economic assumptions
Variable Type:	Floating Point Array
Default Value:	0.0025
Length:	120
Number of Decimals:	8
Choice List:	
Character Type:	Not Applicable

Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.271 **v\_month\_t\_int\_res**

Description:	
Help:	Annual rate.
Modified On:	6/25/2024 7:06:08 PM (UTC+03:00)
Modified By:	CLAL-INS\Arikt
Category:	Economic assumptions
Variable Type:	Floating Point Number
Default Value:	4
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	20
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.272 **v\_month\_t\_rm**

Description:	Discount rate (v) monthly by projection month t - for risk margin
Help:	Monthly v factor by month t for MCEV calculations in the Experience model and reserve calculations in the Net_Prem model. Input in assumption set
Modified On:	12/3/2020 11:52:43 AM (UTC+02:00)
Modified By:	CLAL-INS\NinaB
Category:	Economic assumptions
Variable Type:	Floating Point Array
Default Value:	0.0025
Length:	120
Number of Decimals:	8
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared

Category Order: 0

### 5.3.3.1.273 **v\_month\_w**

Description: Discount rate (v) monthly  
 Help: Equals  $1 / (1 + \text{disc\_rate\_mth})$ .  
 Monthly v factor for embedded value calculations in the Experience model and reserve calculations in the Net\_Prem model. Calculated in function calls from startup.  
 Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
 Modified By: CLAL-INS\ninab  
 Category: Economic assumptions  
 Variable Type: Floating Point Number  
 Default Value: 0.99  
 Length: 0  
 Number of Decimals: 6  
 Choice List:  
 Character Type: Not Applicable  
 Valid Range From: 0  
 Valid Range To: 100  
 Table Format: Default Row Numbers  
 Set Value in Input Manager:  
 Variable Sharing: Not Shared  
 Category Order: 0

### 5.3.3.1.274 **var\_mgt\_fee**

Description: Lookup value code variable  
 Help:  
 Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
 Modified By: CLAL-INS\ninab  
 Category: Economic assumptions  
 Variable Type: Floating Point Number  
 Default Value: 0  
 Length: 0  
 Number of Decimals: 1  
 Choice List:  
 Character Type: Not Applicable  
 Valid Range From:  
 Valid Range To:  
 Table Format: Default Row Numbers  
 Set Value in Input Manager: All  
 Variable Sharing: Not Shared  
 Category Order: 0

**5.3.3.1.275      vat**

Description:	V.A.T
Help:	V.A.T as a percentage
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Economic assumptions
Variable Type:	Floating Point Number
Default Value:	17
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.276      exp\_claim\_fix**

Description:	Fixed claim expense per policy
Help:	Exit expenses per policy. Loaded from table.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Expenses
Variable Type:	Floating Point Number
Default Value:	10
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	10000
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.277      exp\_claim\_max**

Description:	Maximum claim expense
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Expenses
Variable Type:	Floating Point Number

Default Value:	10
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100000
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.278 *exp\_claim\_perc*

Description:	Claim expense as % of claim payment
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Expenses
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1000
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.279 *exp\_dac\_perc*

Description:	Proportion of initial expenses deferred (in DAC)
Help:	Proportion (%) of initial expenses that are deferrable in the DAC.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Expenses
Variable Type:	Floating Point Number
Default Value:	80
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100

Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.280 *exp\_extra\_perc\_charge*

Description:	extra expenses as % of risk rider charges
Help:	An array of expense percentage applied to charges on risk riders (for Profil riders). This is the "extra" amount, because there is already an expense on the total premium received.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Expenses
Variable Type:	Floating Point Array
Default Value:	10
Length:	26
Number of Decimals:	3
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1000
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.281 *exp\_init\_fix*

Description:	Initial expense per policy (sheckels)
Help:	Initial expenses per policy. Loaded from table.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Expenses
Variable Type:	Floating Point Number
Default Value:	10
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	2000
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.282 exp\_init\_fix\_cov**

Description:	Initial expense per cover (sheckels)
Help:	Initial expenses per policy. Loaded from table.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Expenses
Variable Type:	Floating Point Number
Default Value:	10
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	2000
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.283 exp\_init\_perc\_prem**

Description:	Initial expenses as % of premium
Help:	Initial expenses expressed as a % of the first year premium. Loaded from table.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Expenses
Variable Type:	Floating Point Number
Default Value:	5
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1000
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.284 exp\_initial\_extra\_perc\_charge**

Description:	extra initial expenses as % of risk rider charges
Help:	An array of expense percentage applied to charges on risk riders (for Profil riders). This is the "extra" amount, because there is already an expense on the total premium received.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)

Modified By:	CLAL-INS\ninab
Category:	Expenses
Variable Type:	Floating Point Array
Default Value:	10
Length:	26
Number of Decimals:	3
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1000
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.285      *exp\_initial\_fix\_rider***

Description:	extra initial expenses in shekels for rider
Help:	An array of expense percentage applied to charges on risk riders (for Profil riders). This is the "extra" amount, because there is already an expense on the total premium received.
Modified On:	11/17/2021 9:42:27 AM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Expenses
Variable Type:	Floating Point Array
Default Value:	10
Length:	26
Number of Decimals:	3
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1000
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.286      *exp\_mult\_set\_temp***

Description:	Expense multiplier assumption set
Help:	Expense multiplier assumption set. If read_from_table = "Y", then this variable will be set in set_common_variables.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Expenses
Variable Type:	Character
Default Value:	default



Length:	7
Number of Decimals:	0
Choice List:	
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.287 *exp\_mult\_tbl*

Description:	Expense multipliers (%)
Help:	Expense multipliers table for initial and renewal expenses.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Expenses
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Row Name (numeric)
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.288 *exp\_pup\_fix*

Description:	Renewal expense for paid-up policies (shekels)
Help:	Fixed, annual, per policy renewal expense for paid up policies. Read in from expense_tbl in set categ_variables.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Expenses
Variable Type:	Floating Point Number
Default Value:	50
Length:	0
Number of Decimals:	4
Choice List:	
Character Type:	Not Applicable

Valid Range From:	0
Valid Range To:	1000
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.289 *exp\_ren\_fix*

Description:	Renewal expense per policy (sheckels)
Help:	Annual renewal expenses per policy. Loaded from table.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Expenses
Variable Type:	Floating Point Number
Default Value:	4
Length:	110
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	10000
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.290 *exp\_ren\_fix\_cov*

Description:	Renewal expense per cover (sheckels)
Help:	Annual renewal expenses per policy. Loaded from table.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Expenses
Variable Type:	Floating Point Number
Default Value:	4
Length:	110
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	10000
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.291 exp\_ren\_fix\_rider**

Description:	extra renewal expenses in shekels for rider
Help:	An array of expense percentage applied to charges on risk riders (for Profil riders). This is the "extra" amount, because there is already an expense on the total premium received.
Modified On:	11/17/2021 9:44:28 AM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Expenses
Variable Type:	Floating Point Array
Default Value:	10
Length:	26
Number of Decimals:	3
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1000
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.292 exp\_ren\_perc\_annuity**

Description:	Renewal expense as % of annuity
Help:	Renewal expenses as a % of annuity payment. Loaded from table.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Expenses
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1000
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.293 exp\_ren\_perc\_prem**

Description:	Renewal expense as % of premium
--------------	---------------------------------

Help:	Renewal expenses as a % of premium income. Loaded from table.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Expenses
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1000
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.294      *exp\_ren\_res***

Description:	expense as percentage of reserves
Help:	Renewal expenses (annual percentage but paid monthly) as a percentage of reserve. Loaded from table.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Expenses
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	2000
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.295      *exp\_row\_lookup***

Description:	Expenses table row lookup
Help:	Expenses assumption set. If read_from_table = "Y", then this variable will be used in set_common_variables.
Modified On:	10/26/2021 1:29:18 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Expenses

Variable Type:	Character
Default Value:	default
Length:	7
Number of Decimals:	0
Choice List:	
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.296**     *i\_percov*

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Expenses
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.297**     *i\_percov\_sp*

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Expenses
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	

Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### **5.3.3.1.298**     *i\_perpol*

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Expenses
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### **5.3.3.1.299**     *i\_perpol\_sp*

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Expenses
Variable Type:	Floating Point Number
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.300**     *i\_prem*

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Expenses
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.301**     *i\_single*

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Expenses
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.302**     *infl\_rate\_expenses*

Description:	Expense inflation rate (%)
Help:	Per policy expenses are subject to be adjusted every month to an inflation index. This is the annual inflation rate and is the experience projection assumption.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab

Category:	Expenses
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	50
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.303     *m\_ann\_pmt***

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Expenses
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.304     *m\_clms***

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Expenses
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	



Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.305**     ***m\_clms\_cov***

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Expenses
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.306**     ***m\_percov***

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Expenses
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.307**     *m\_percov\_sp*

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Expenses
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.308**     *m\_perpol*

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Expenses
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.309**     *m\_prem*

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Expenses
Variable Type:	Floating Point Number

Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### **5.3.3.1.310**     *m\_pup*

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Expenses
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### **5.3.3.1.311**     *m\_res*

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Expenses
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers

Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.312     *m\_res\_nonpar***

Description: Lookup value code variable wildcard  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Expenses  
Variable Type: Floating Point Number  
Default Value: 0  
Length: 0  
Number of Decimals: 1  
Choice List:  
Character Type: Not Applicable  
Valid Range From:  
Valid Range To:  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.313     *m\_res\_par***

Description: Lookup value code variable wildcard  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Expenses  
Variable Type: Floating Point Number  
Default Value: 0  
Length: 0  
Number of Decimals: 1  
Choice List:  
Character Type: Not Applicable  
Valid Range From:  
Valid Range To:  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.314     *agency\_no\_lookup***

Description: Lookup value code variable

Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Lapses  
Variable Type: Character  
Default Value: 0  
Length: 10  
Number of Decimals: 1  
Choice List: 0  
Character Type: Standard  
Valid Range From:  
Valid Range To:  
Table Format: Default Row Numbers  
Set Value in Input Manager:  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.315**      ***claw\_fact\_set***

Description: Lookup set for clawback factor  
Help:  
Modified On: 3/9/2020 10:07:07 AM (UTC+02:00)  
Modified By: CLAL-INS\NinaB  
Category: Lapses  
Variable Type: Character  
Default Value: clawback\_factor\_default  
Length: 25  
Number of Decimals: 1  
Choice List: clawback\_factor\_default  
Character Type: Standard  
Valid Range From:  
Valid Range To:  
Table Format: Default Row Numbers  
Set Value in Input Manager:  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.316**      ***factor\_key***

Description: Lookup value code variable  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Lapses  
Variable Type: Character  
Default Value: 0  
Length: 10  
Number of Decimals: 1

Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.317**      ***lapse\_clawback\_factor***

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Lapses
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.318**      ***lapse\_expos\_col\_key***

Description:	Lookup value code variable
Help:	
Modified On:	12/14/2022 2:49:41 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Lapses
Variable Type:	Character
Default Value:	0
Length:	20
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.319**      ***lapse\_factor\_profil\_rider***

Description:	Lookup value code variable
Help:	
Modified On:	12/23/2024 8:57:50 PM (UTC+02:00)
Modified By:	CLAL-INS\arikt
Category:	Lapses
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.320**      ***lapse\_factor\_proj***

Description:	Lookup value code variable
Help:	
Modified On:	12/19/2024 3:33:26 PM (UTC+02:00)
Modified By:	CLAL-INS\arikt
Category:	Lapses
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.321**      ***lapse\_factor\_proj\_rider***

Description:	Lookup value code variable
Help:	
Modified On:	12/19/2024 3:34:40 PM (UTC+02:00)
Modified By:	CLAL-INS\arikt
Category:	Lapses

Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.322**     ***lapse\_factor\_y\_col***

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Lapses
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.323**     ***lapse\_factor\_y1***

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Lapses
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	



Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.324**      ***lapse\_factor\_y1\_row***

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Lapses
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.325**      ***lapse\_factor\_yplus***

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Lapses
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.326**      ***lapse\_force\_month***

Description:	month in which mass lapse occurs
Help:	This can be used to override lapse rates, and make the policy remain in force until this month, and then lapse in this month. Set value to 0 to ignore this and use normal lapse assumptions.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Lapses
Variable Type:	Integer Number
Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	600
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.327**      ***lapse\_force\_rate***

Description:	Forced lapse rate
Help:	This can be used to override lapse rates, and make the policy remain in force until this month, and then lapse in this month. Set value to 0 to ignore this and use normal lapse assumptions.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Lapses
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.328 lapse\_force\_rate\_input**

Description:	Forced lapse rate for input variable
Help:	This can be used to override lapse rates, and make the policy remain in force until this month, and then lapse in this month. Set value to 0 to ignore this and use normal lapse assumptions.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Lapses
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.329 lapse\_rate\_im**

Description:	Lookup value code variable
Help:	
Modified On:	1/15/2023 2:57:55 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Lapses
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.330 lapse\_rate\_pup\_im**

Description:	Lookup value code variable
Help:	
Modified On:	1/15/2023 2:58:12 PM (UTC+02:00)

Modified By:	CLAL-INS\joshm
Category:	Lapses
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.331 ***lapse\_rates***

Description:	Lapse rates (%)
Help:	Lapse rates (as %) for each policy year. Item zero is not used. Read from table in set_exp/res_variables() if read_from_tables = Y.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Lapses
Variable Type:	Floating Point Array
Default Value:	0 0 0
Length:	120
Number of Decimals:	3
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.332 ***lapse\_rider\_other***

Description:	lapse rate for other riders
Help:	
Modified On:	1/15/2023 3:01:49 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Lapses
Variable Type:	Floating Point Number
Default Value:	0
Length:	0

Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.333**     ***lapse\_rider\_profil\_dth***

Description:	lapse rate for profil dth rider
Help:	
Modified On:	1/15/2023 2:55:58 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Lapses
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.334**     ***lapse\_rider\_profil\_dth\_array***

Description:	Additional profil rider lapse rates (%)
Help:	Additional rider lapse rates (as %) forprofil riders
Modified On:	1/15/2023 2:57:24 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Lapses
Variable Type:	Floating Point Array
Default Value:	0 0 0
Length:	25
Number of Decimals:	4
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	

Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.335**     ***lapse\_set\_pup***

Description: Lookup value code variable  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Lapses  
Variable Type: Character  
Default Value: 0  
Length: 10  
Number of Decimals: 1  
Choice List: 0  
Character Type: Standard  
Valid Range From:  
Valid Range To:  
Table Format: Default Row Numbers  
Set Value in Input Manager:  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.336**     ***lapse\_type\_col\_key***

Description: Lookup value code variable  
Help:  
Modified On: 12/14/2022 1:15:59 PM (UTC+02:00)  
Modified By: CLAL-INS\ahuvaa  
Category: Lapses  
Variable Type: Character  
Default Value: 0  
Length: 20  
Number of Decimals: 1  
Choice List: 0  
Character Type: Standard  
Valid Range From:  
Valid Range To:  
Table Format: Default Row Numbers  
Set Value in Input Manager:  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.337**     ***masslaps\_tbl***

Description: Lapse rate table by flag\_code for solvency scenario  
Help:

Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Lapses
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	0
Table Format:	Row Name (numeric)
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.338 ***prem\_termination\_rate***

Description:	Prem Termination Rate
Help:	
Modified On:	9/5/2019 11:01:48 AM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Category:	Lapses
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	3
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.339 ***pup\_sv\_charge\_rebate\_temp***

Description:	Annual rebate percentage for PUP's SV penalty
Help:	For Profil there is an annual reduction to the initial surrender charge that was applied to policies made paid-up in a given policy year. The reduction is applied for each year the PUP does not surrender.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Lapses
Variable Type:	Floating Point Number
Default Value:	1

Length:	1
Number of Decimals:	3
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	10
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.340 *secondary\_lapse\_mult*

Description:	secondary lapse multier
Help:	For "achrayut le'chaim" product. The percentage of normal lapses experienced by polices that continue after a claim. (the first claim is from the group that allows the policy to continue at 50%).
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Lapses
Variable Type:	Floating Point Number
Default Value:	100
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1000
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.341 *sur\_val\_method*

Description:	Method to calculate SV
Help:	Different ways to calculate surrender values: "sv_table" = SV looked up from table. "perc_res" = SV is a percentage of reserve. Percentages come from array variable "sur_val_perc" by policy year. This method is useful for old products that do not have tables available.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Lapses
Variable Type:	Character



Default Value:	sv_table
Length:	8
Number of Decimals:	0
Choice List:	perc_res,sv_table
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.342 *sur\_val\_perc*

Description:	SV percentages (%) - of RESERVE
Help:	This is the array of surrender value factors as a percentage of reserve. Array is by policy year. Used when the surrender value method (sur_val_method) is set to "perc_res". Useful for old products when SV table not available.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Lapses
Variable Type:	Floating Point Array
Default Value:	100
Length:	116
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.343 *cat\_risk*

Description:	Margin for catastrophe scenario for risk (death)
Help:	
Modified On:	12/6/2020 4:36:44 PM (UTC+02:00)
Modified By:	CLAL-INS\NinaB
Category:	Margins
Variable Type:	Floating Point Number
Default Value:	0.0015
Length:	0
Number of Decimals:	5

Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.344**     *margin\_1styr\_clms\_add*

Description:	Additional 1st year claims margin percent
Help:	only added if margin_add = Y
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Margins
Variable Type:	Floating Point Number
Default Value:	0
Length:	1
Number of Decimals:	2
Choice List:	Y,N
Character Type:	Not Applicable
Valid Range From:	-100
Valid Range To:	1000
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.345**     *margin\_add*

Description:	Add margins to assumptions ?
Help:	Add margins to assumptions Y/N ? (expenses, mortality and lapses) eg for DAC Recoverability test.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Margins
Variable Type:	Character
Default Value:	<*tables_path*>\tables\mortality\profil_decreme nt_rates.tbl
Length:	59
Number of Decimals:	0
Choice List:	Y,N,<*tables_path*>\tables\mortality\profil_decr ement_rates.tbl
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0

Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.346 *margin\_add\_asset*

Description:	Add asset shock?
Help:	Add margins to assumptions Y/N ? (expenses, mortality and lapses) eg for DAC Recoverability test.
Modified On:	7/4/2021 4:45:45 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Margins
Variable Type:	Character
Default Value:	N
Length:	59
Number of Decimals:	0
Choice List:	Y,N
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.347 *margin\_add\_cat*

Description:	Apply catastrophe scenario?
Help:	Add margins to assumptions Y/N ? (expenses, mortality and lapses) eg for DAC Recoverability test.
Modified On:	12/8/2020 10:01:57 AM (UTC+02:00)
Modified By:	CLAL-INS\NinaB
Category:	Margins
Variable Type:	Character
Default Value:	N
Length:	1
Number of Decimals:	0
Choice List:	Y,N
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.348      *margin\_add\_discount***

Description:	Add margins to discounts and management fees?
Help:	Add margins to assumptions Y/N ? (expenses, mortality and lapses) eg for DAC Recoverability test.
Modified On:	12/7/2020 2:13:15 PM (UTC+02:00)
Modified By:	CLAL-INS\NinaB
Category:	Margins
Variable Type:	Character
Default Value:	N
Length:	1
Number of Decimals:	0
Choice List:	Y,N
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.349      *margin\_ann\_mort\_pc***

Description:	mortality for annuity margin percent
Help:	only added if margin_add = Y
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Margins
Variable Type:	Floating Point Number
Default Value:	0
Length:	1
Number of Decimals:	2
Choice List:	Y,N
Character Type:	Not Applicable
Valid Range From:	-100
Valid Range To:	1000
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.350      *margin\_annuity\_nottakeup***

Description:	not take up annuity margin
--------------	----------------------------

Help: only added if dacrec\_add\_margins = Y  
 add this margin / e\_x per mil to mortality rate  
 Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
 Modified By: CLAL-INS\ninab  
 Category: Margins  
 Variable Type: Floating Point Number  
 Default Value: 0  
 Length: 1  
 Number of Decimals: 2  
 Choice List: Y,N  
 Character Type: Not Applicable  
 Valid Range From: -50  
 Valid Range To: 100  
 Table Format: Default Row Numbers  
 Set Value in Input Manager:  
 Variable Sharing: Not Shared  
 Category Order: 0

#### **5.3.3.1.351      *margin\_annuity\_takeup***

Description: take up annuity margin  
 Help: only added if dacrec\_add\_margins = Y  
 add this margin / e\_x per mil to mortality rate  
 Modified On: 6/3/2020 10:03:58 AM (UTC+03:00)  
 Modified By: CLAL-INS\NinaB  
 Category: Margins  
 Variable Type: Floating Point Number  
 Default Value: 0  
 Length: 1  
 Number of Decimals: 2  
 Choice List: Y,N  
 Character Type: Not Applicable  
 Valid Range From: -100  
 Valid Range To: 100  
 Table Format: Default Row Numbers  
 Set Value in Input Manager: All  
 Variable Sharing: Not Shared  
 Category Order: 0

#### **5.3.3.1.352      *margin\_claim\_cost\_mitriya***

Description: claims margin percent for claims cost for phi mitriya  
 Help: only added if margin\_add = Y  
 Modified On: 11/30/2022 5:25:45 PM (UTC+02:00)  
 Modified By: CLAL-INS\joshm  
 Category: Margins  
 Variable Type: Floating Point Number

Default Value:	0
Length:	1
Number of Decimals:	2
Choice List:	Y,N
Character Type:	Not Applicable
Valid Range From:	-100
Valid Range To:	1000
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.353 *margin\_claims*

Description:	claims margin percent
Help:	only added if margin_add = Y
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Margins
Variable Type:	Floating Point Number
Default Value:	0
Length:	1
Number of Decimals:	2
Choice List:	Y,N
Character Type:	Not Applicable
Valid Range From:	-100
Valid Range To:	1000
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.354 *margin\_dnp*

Description:	Premium Charge margin
Help:	Percentage of premium allocated to units. Each element in the array applies for the number of months specified in the alloc_rate_period array. The first element in the array should be entered for period 1. Read in from alloc_rate_tbl in set_common_variables.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Margins
Variable Type:	Floating Point Number
Default Value:	0
Length:	16
Number of Decimals:	2
Choice List:	

Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	150
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.355 *margin\_dnz*

Description:	management fee replacement
Help:	Percentage of premium allocated to units. Each element in the array applies for the number of months specified in the alloc_rate_period array. The first element in the array should be entered for period 1. Read in from alloc_rate_tbl in set_common_variables.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Margins
Variable Type:	Floating Point Number
Default Value:	0
Length:	16
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	150
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.356 *margin\_exp\_ini\_fix*

Description:	fixed initial expense margin percent
Help:	only added if margin_add = Y
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Margins
Variable Type:	Floating Point Number
Default Value:	0
Length:	1
Number of Decimals:	2
Choice List:	Y,N
Character Type:	Not Applicable
Valid Range From:	-100
Valid Range To:	200
Table Format:	Default Row Numbers

Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.357**     *margin\_exp\_ini\_pc*

Description:	premium initial expense margin percent
Help:	only added if margin_add = Y
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Margins
Variable Type:	Floating Point Number
Default Value:	0
Length:	1
Number of Decimals:	2
Choice List:	Y,N
Character Type:	Not Applicable
Valid Range From:	-100
Valid Range To:	200
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.358**     *margin\_exp\_ren\_fix*

Description:	fixed renewal expense margin percent
Help:	only added if margin_add = Y
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Margins
Variable Type:	Floating Point Number
Default Value:	0
Length:	1
Number of Decimals:	2
Choice List:	Y,N
Character Type:	Not Applicable
Valid Range From:	-100
Valid Range To:	200
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.359**     *margin\_exp\_ren\_pc*

Description:	premium renewal expense margin percent
--------------	--



Help:	only added if margin_add = Y
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Margins
Variable Type:	Floating Point Number
Default Value:	0
Length:	1
Number of Decimals:	2
Choice List:	Y,N
Character Type:	Not Applicable
Valid Range From:	-100
Valid Range To:	200
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.360**      *margin\_lapses*

Description:	lapse margin percent
Help:	only added if margin_add = Y
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Margins
Variable Type:	Floating Point Number
Default Value:	0
Length:	1
Number of Decimals:	2
Choice List:	Y,N
Character Type:	Not Applicable
Valid Range From:	-100
Valid Range To:	1000
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.361**      *margin\_mort\_pc*

Description:	mortality margin percent
Help:	only added if margin_add = Y
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Margins
Variable Type:	Floating Point Number
Default Value:	0
Length:	1
Number of Decimals:	2

Choice List:	Y,N
Character Type:	Not Applicable
Valid Range From:	-100
Valid Range To:	1000
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.362**     *margin\_recover*

Description:	recovery rate margin percent for phi & ltc
Help:	only added if margin_add = Y
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Margins
Variable Type:	Floating Point Number
Default Value:	0
Length:	1
Number of Decimals:	2
Choice List:	Y,N
Character Type:	Not Applicable
Valid Range From:	-100
Valid Range To:	1000
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.363**     *margin\_res\_ann\_mort\_fac*

Description:	mortality factor for reserves - margin percent
Help:	only added if margin_add = Y
Modified On:	5/16/2023 4:01:46 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Margins
Variable Type:	Floating Point Number
Default Value:	0
Length:	1
Number of Decimals:	2
Choice List:	Y,N
Character Type:	Not Applicable
Valid Range From:	-100
Valid Range To:	1000
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.364      *ann\_death***

Description:	Status death of insured ( 1=death, 0=live)
Help:	policy annuity factor of not guaranteed period
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Mortality
Variable Type:	Integer Number
Default Value:	0
Length:	6
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	9000
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.365      *col\_dth***

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Mortality
Variable Type:	Integer Number
Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.366      *death\_rate\_row\_key***

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Mortality

Variable Type:	Integer Number
Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### **5.3.3.1.367**      *death\_rates*

Description:	Death rate table
Help:	Death-only rate table
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Mortality
Variable Type:	Character
Default Value:	
Length:	61
Number of Decimals:	6
Choice List:	
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### **5.3.3.1.368**      *death\_rates\_res*

Description:	Death rate table for reserves
Help:	Death-only rate table
Modified On:	3/13/2023 1:42:39 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Mortality
Variable Type:	Character
Default Value:	AMF80
Length:	45
Number of Decimals:	0
Choice List:	AMF80
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0

Table Format:	Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.369**      ***death\_rates\_res\_tbl***

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Mortality
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.370**      ***death\_rates\_tbl***

Description:	Lookup code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Mortality
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.371      *decrem\_mult***

Description:	decrement multiplier (%)
Help:	Percentage of basic decrement table (experience basis). Always read from table in set_categ_variables() according to sex/smoker status.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Mortality
Variable Type:	Floating Point Number
Default Value:	100
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1000
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.372      *decrem\_mult\_res***

Description:	decrement multiplier (%) for reserves
Help:	Percentage of basic decrement table (experience basis). Always read from table in set_categ_variables() according to sex/smoker status.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Mortality
Variable Type:	Floating Point Number
Default Value:	100
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1000
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.373      *decrem\_mult\_set\_temp***

Description:	Decrement multipliers assumption set
--------------	--------------------------------------

Help:	Decrement multipliers assumption set. If read_from_table = "Y", then this variable will be set in set_common_variables.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Mortality
Variable Type:	Character
Default Value:	default
Length:	7
Number of Decimals:	0
Choice List:	
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### 5.3.3.1.374 *decrem\_mult\_tbl*

Description:	decrement multiplier table (%)
Help:	Percentage of basic decrement table used for all lives, split by M/F, S/NS/A, occupation class and product type.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Mortality
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	0
Table Format:	Row Name (numeric)
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### 5.3.3.1.375 *decrem\_rate\_key*

Description:	decrement rate lookup code
Help:	This is the index value to lookup claims cost rates in the claim_cost.tbl. It is looked up initially from the tarif_spec.tbl but may be modified by policy data from the file.
Modified On:	7/28/2021 12:09:58 PM (UTC+03:00)

Modified By:	CLAL-INS\joshm
Category:	Mortality
Variable Type:	Character
Default Value:	MN
Length:	22
Number of Decimals:	0
Choice List:	MN
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.376 *decrem\_rate\_key\_rider*

Description:	Decrement rate lookup code for profil riders
Help:	This is the index value to lookup claims cost rates in the claim_cost.tbl. It is looked up initially from the tarif_spec.tbl but may be modified by policy data from the file.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Mortality
Variable Type:	Character
Default Value:	0
Length:	550
Number of Decimals:	0
Choice List:	
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.377 *decrem\_rates*

Description:	Decrement rate table
Help:	Decrement table by sex and smoker status. If set_by_prodcod = "Y" then table is set based on "risk_rates" in prod assumptions table.
	Not relevant for health and death benefits.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab



Category:	Mortality
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	0
Table Format:	Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.378 ***decrem\_rates\_check***

Description:	Decrement rate table
Help:	Decrement table by sex and smoker status. If set_by_prodcod = "Y" then table is set based on "risk_rates" in prod assumptions table.

Modified On:	Not relevant for health and death benefits. 7/27/2021 11:55:54 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Mortality
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	0
Table Format:	Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.379 ***decrem\_rates\_res***

Description:	Decrement rate table for reserves
Help:	Decrement table used for net-premium reserve calculation. If set_by_prodcod = "Y" then this is set to the same table as the experience decrements (decrem_rates).
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab

Category:	Mortality
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	0
Table Format:	Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### 5.3.3.1.380 ***decrem\_rates\_uw***

Description:	Decrement rate table by UWYear
Help:	Decrement table by sex and smoker status. If set_by_prodcod = "Y" then table is set based on "risk_rates" in prod assumptions table.

	Not relevant for health and death benefits.
Modified On:	3/19/2024 5:29:29 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Mortality
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	0
Table Format:	Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### 5.3.3.1.381 ***decrem\_rates\_uw\_res***

Description:	Decrement rate table by UWYear for reserves
Help:	Decrement table by sex and smoker status. If set_by_prodcod = "Y" then table is set based on "risk_rates" in prod assumptions table.

	Not relevant for health and death benefits.
Modified On:	3/19/2024 7:29:12 PM (UTC+02:00)

Modified By:	CLAL-INS\Arikt
Category:	Mortality
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	0
Table Format:	Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.382 *decrements\_apply*

Description:	Apply decrements?
Help:	Do you want in force columns decremented by survivorship? Only set to N for testing/debugging model.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Mortality
Variable Type:	Character
Default Value:	Y
Length:	1
Number of Decimals:	0
Choice List:	Y,N
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.383 *dth\_res\_row*

Description:	Lookup value code vatiable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Mortality
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1

Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.384      *free\_inv\_row\_key***

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Mortality
Variable Type:	Integer Number
Default Value:	0
Length:	10
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.385      *mort\_addn\_res***

Description:	Mortality rate addition (per mille) for reserves
Help:	Percentage of basic mortality table used for reserve calculation. Always read from table in set_categ_variables() according to sex/smoker status .
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Mortality
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1
Table Format:	Default Row Numbers

Set Value in Input Manager:  
Variable Sharing: Not Shared  
Category Order: 0

### 5.3.3.1.386 *mort\_mult*

Description: Mortality multiplier (%)  
Help: Percentage of basic mortality table used (experience basis). Always read from table in set\_categ\_variables() according to sex/smoker status .  
  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Mortality  
Variable Type: Floating Point Number  
Default Value: 100  
Length: 0  
Number of Decimals: 2  
Choice List:  
Character Type: Not Applicable  
Valid Range From: 0  
Valid Range To: 1000  
Table Format: Default Row Numbers  
Set Value in Input Manager:  
Variable Sharing: Not Shared  
Category Order: 0

### 5.3.3.1.387 *mort\_mult\_col\_key*

Description: Lookup value code variable  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Mortality  
Variable Type: Character  
Default Value: 0  
Length: 20  
Number of Decimals: 1  
Choice List: 0  
Character Type: Standard  
Valid Range From:  
Valid Range To:  
Table Format: Default Row Numbers  
Set Value in Input Manager:  
Variable Sharing: Not Shared  
Category Order: 0

**5.3.3.1.388**      ***mort\_mult\_end\_age***

Description:	Age from which to phase-out mortality multiplier
Help:	At this age the mortality multiplier will gradually be phased-out (i.e. approach 100%) until the omega-age, so that at very old ages the base mortality table is less effected by the multiplier.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Mortality
Variable Type:	Integer Number
Default Value:	75
Length:	1
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	75
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.389**      ***mort\_mult\_res***

Description:	Mortality multiplier (%) for reserves
Help:	Percentage of basic mortality table used for reserve calculation. Always read from table in set_categ_variables() according to sex/smoker status .
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Mortality
Variable Type:	Floating Point Number
Default Value:	100
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1000
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.390**      ***mort\_mult\_tbl***

Description:	Mortality multiplier table (%)
--------------	--------------------------------

Help:	Percentage of basic mortality table used for all lives, split by M/F, S/NS/A, and product type.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Mortality
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	0
Table Format:	Row Name (numeric)
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.391**     ***mort\_sel\_status***

Description:	Select mortality (Y/N)
Help:	Indicator to determine whether to use select or ultimate experience mortality rates.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Mortality
Variable Type:	Character
Default Value:	N
Length:	1
Number of Decimals:	0
Choice List:	Y,N
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.392**     ***mort\_year\_tt***

Description:	Lookup value code variable
Help:	
Modified On:	7/26/2021 2:40:47 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Mortality
Variable Type:	Integer Number
Default Value:	0

Length:	20
Number of Decimals:	0
Choice List:	0
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	3000
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.393 *omega\_age*

Description:	Maximum age possible in projection
Help:	Highest age in mortality table. Internal logic variable calculated in calc_omega_age.
Modified On:	1/11/2023 11:18:09 AM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Mortality
Variable Type:	Floating Point Number
Default Value:	0
Length:	1
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	120
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.394 *omega\_age\_cmi*

Description:	Maximum age possible in CMI table
Help:	Highest age in mortality table. Internal logic variable calculated in calc_omega_age.
Modified On:	1/11/2023 11:18:39 AM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Mortality
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	120



Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.395 *omega\_age\_dec*

Description:	Maximum age possible in projection
Help:	Highest age in mortality table. Internal logic variable calculated in calc_omega_age.
Modified On:	1/11/2023 11:18:53 AM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Mortality
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	120
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.396 *omega\_age\_w*

Description:	Maximum age possible in projection
Help:	Highest age in mortality table. Internal logic variable calculated in calc_omega_age.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Mortality
Variable Type:	Integer Number
Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	120
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.397**     ***puv\_09\_tbl***

Description:	Composite external source
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Mortality
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.398**     ***puv\_col\_key***

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Mortality
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.399**     ***puv\_row\_key***

Description:	Lookup value code variable
Help:	
Modified On:	7/28/2021 1:24:47 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Mortality
Variable Type:	Character

Default Value:	18_10
Length:	10
Number of Decimals:	1
Choice List:	18_10
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.400     *select\_periods***

Description:	mortality select periods
Help:	
Modified On:	1/9/2023 11:51:19 AM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Mortality
Variable Type:	Floating Point Number
Default Value:	0
Length:	1
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.401     *survive\_tbl***

Description:	survival factors table for extra annuity reserve
Help:	nPx factors (survival to age 65) used for extra annuity reserve, based on adjusted table 4a1 as calculated in Clal's annuity reserve spreadsheet.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Mortality
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable

Valid Range From:	0
Valid Range To:	1
Table Format:	Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.402      *sv\_09\_tbl***

Description:	Composite external source
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Mortality
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.403      *sv\_09\_tbl\_check***

Description:	Composite external source
Help:	
Modified On:	10/17/2021 4:45:49 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Mortality
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.404 sv\_col\_key**

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Mortality
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.405 sv\_row\_key**

Description:	Lookup value code variable
Help:	
Modified On:	7/28/2021 12:36:51 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Mortality
Variable Type:	Character
Default Value:	18_10
Length:	10
Number of Decimals:	1
Choice List:	18_10
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.406 age\_at\_issue**

Description:	Age at issue
Help:	Age at issue. Can be age last in years (e.g. 35) or exact age using decimals (e.g. 35.75). This age is rounded down in the model which is based on age last, incrementing by one year each policy anniversary.

Modified On:	5/5/2021 9:31:40 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Policy Details
Variable Type:	Floating Point Number
Default Value:	30
Length:	0
Number of Decimals:	3
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	120
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.407 **agency\_no**

Description:	Osek Merushe number
Help:	Agency Number (Osek merushe number)
Modified On:	3/15/2020 11:22:34 AM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Policy Details
Variable Type:	Floating Point Number
Default Value:	0
Length:	20
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	9999999999
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.408 **agent\_no**

Description:	agent number and company letter (used as unique index)
Help:	Agent number and first letter of company (a/c) to get unique agent index
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Character
Default Value:	0
Length:	10

Number of Decimals:	0
Choice List:	
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.409 *ben\_class*

Description:	benefit class
Help:	used to determine the type of claim
	dth = death claim, ddth = dd = dread disease, tpd = total & permanent disability, adb = accidental death benefit, health, phi = permanent health insurance, ltc = long term care, fib = family income benefit mortg = mortgage
Modified On:	6/10/2021 12:21:04 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Policy Details
Variable Type:	Character
Default Value:	tpd
Length:	6
Number of Decimals:	0
Choice List:	dth,ddth,dd,tpd,adb,fib,ltc,mortg,phi,health,adif, profil
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.410 *ben\_class\_input*

Description:	benefit class
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details

Variable Type:	Character
Default Value:	tpd
Length:	6
Number of Decimals:	1
Choice List:	dth,ddth,dd,tpd,adb,fib,ltc,mortg,phi,health,adif,profil
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.411      *benefit\_term***

Description:	Policy benefit term (months)
Help:	The original policy (benefit) term in integral months calculated from the issue date to the date of policy expiry.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Integer Number
Default Value:	120
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	1
Valid Range To:	1200
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.412      *benefit\_term\_input***

Description:	Policy benefit term (months)
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Integer Number
Default Value:	120
Length:	0
Number of Decimals:	0
Choice List:	



Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1200
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### **5.3.3.1.413**      *benefits\_curr*

Description:	Number of covers at valuation date
Help:	Current number of in force benefits at the valuation date.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Floating Point Number
Default Value:	1
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1000000
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### **5.3.3.1.414**      *benefits\_curr\_rider*

Description:	benefits_curr_rider
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Floating Point Array
Default Value:	1 1
Length:	25
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	10000
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.415      *bonus\_inforce***

Description:	bonus inforce at valuation date
Help:	variable linked with the reserve field from the inforce file
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100000000
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.416      *channel***

Description:	channel הפצה ערוץ
Help:	
Modified On:	1/9/2022 12:30:57 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Policy Details
Variable Type:	Character
Default Value:	tpd
Length:	6
Number of Decimals:	1
Choice List:	dth,ddth,dd,tpd,adb,fib,lrc,mortg,phi,health,adif,profil
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.417      *chilean***

Description:	Chilean indicator for gimla
--------------	-----------------------------

Help:	Can be age last in years (e.g. 35) or exact age using decimals (e.g. 35.75). This age is rounded down in the model which is based on age last, incrementing by one year each policy anniversary.
Modified On:	5/15/2023 1:48:43 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Policy Details
Variable Type:	Integer Number
Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	120
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.418**      ***commence\_period\_w***

Description:	Period t in which policy commences
Help:	Calculated in startup column.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Integer Number
Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	-600
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.419**      ***company***

Description:	company name
Help:	Use for looking up expenses from expense table. Only used when lookup by prodcode = "Y"
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab

Category:	Policy Details
Variable Type:	Character
Default Value:	clal
Length:	5
Number of Decimals:	0
Choice List:	clal,hasne,briut
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.420**      ***elapsed\_months***

Description:	Elapsed months at valn date
Help:	The number of months, rounded up, from policy inception to the valuation date.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Integer Number
Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1200
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.421**      ***elapsed\_months\_extra***

Description:	months between tarif & origi date
Help:	The number of months, rounded up, from policy inception to the valuation date.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Integer Number
Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	

Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1200
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.422      *error\_code***

Description:	error_code for data record
Help:	1 and 3 are OK any other number will cause the record to be skipped
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Integer Number
Default Value:	1
Length:	1
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.423      *error\_msg***

Description:	error message for skipped records
Help:	
Modified On:	10/3/2021 8:54:19 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Policy Details
Variable Type:	Character
Default Value:	no_error
Length:	60
Number of Decimals:	0
Choice List:	
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared

Category Order: 0

#### **5.3.3.1.424      *fix\_term\_curr\_age\_above\_max\_add\_months***

Description: term fix for private/self-emp and res\_kitzba<=0 -  
if current age is above the max then add how  
many years

Help: Maximum cumulative claim inflation allowed (for  
health products).  
As a percentage.  
For example, 200 means that the claims  
inflation will stop if and when the claims reach  
double the base assumption.

Modified On: 11/29/2021 7:39:56 AM (UTC+02:00)

Modified By: CLAL-INS\joshm

Category: Policy Details

Variable Type: Floating Point Number

Default Value: 80

Length: 1

Number of Decimals: 2

Choice List:

Character Type: Not Applicable

Valid Range From: 0

Valid Range To: 100

Table Format: Default Row Numbers

Set Value in Input Manager: All

Variable Sharing: Not Shared

Category Order: 0

#### **5.3.3.1.425      *fix\_term\_curr\_age\_max***

Description: term fix for private/self-emp and res\_kitzba<=0 -  
if current age is below

Help: Maximum cumulative claim inflation allowed (for  
health products).  
As a percentage.  
For example, 200 means that the claims  
inflation will stop if and when the claims reach  
double the base assumption.

Modified On: 11/29/2021 7:37:09 AM (UTC+02:00)

Modified By: CLAL-INS\joshm

Category: Policy Details

Variable Type: Floating Point Number

Default Value: 80

Length: 1

Number of Decimals: 2

Choice List:

Character Type: Not Applicable

Valid Range From: 0

Valid Range To: 100

Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.426 *fix\_term\_end\_age\_limit*

Description:	term fix for private/self-emp and res_kitzba<=0 - if end age is above
Help:	Maximum cumulative claim inflation allowed (for health products). As a percentage. For example, 200 means that the claims inflation will stop if and when the claims reach double the base assumption.
Modified On:	11/29/2021 7:34:46 AM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Policy Details
Variable Type:	Floating Point Number
Default Value:	80
Length:	1
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.427 *fix\_term\_new\_end\_age*

Description:	term fix for private/self-emp and res_kitzba<=0 - set end age
Help:	Maximum cumulative claim inflation allowed (for health products). As a percentage. For example, 200 means that the claims inflation will stop if and when the claims reach double the base assumption.
Modified On:	11/29/2021 7:37:23 AM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Policy Details
Variable Type:	Floating Point Number
Default Value:	80
Length:	1
Number of Decimals:	2
Choice List:	

Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.428 *flag\_code*

Description:	flag_code for data record
Help:	Code is used to identify special covers/policies. Not used in the model but passed to the output file for summing the results. 1= LTC NB05 old tarif 2= PHI mifali NB05 alone - added to old policy 3= PHI mifali NB05 alone - other new policy 4= PHI mifali NB05 alone - w/o other new policy
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Integer Number
Default Value:	0
Length:	1
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.429 *foreign\_id*

Description:	Foreign / citizen insured identification
Help:	Can be age last in years (e.g. 35) or exact age using decimals (e.g. 35.75). This age is rounded down in the model which is based on age last, incrementing by one year each policy anniversary.
Modified On:	5/15/2023 1:49:05 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Policy Details
Variable Type:	Integer Number
Default Value:	0
Length:	0
Number of Decimals:	0



Choice List:  
Character Type: Not Applicable  
Valid Range From: 0  
Valid Range To: 120  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.430**     ***fund***

Description: Fund (Keren)  
Help: 10 = yod,..., 1 = aleph  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Policy Details  
Variable Type: Character  
Default Value: 52  
Length: 2  
Number of Decimals: 0  
Choice List:  
Character Type: Standard  
Valid Range From: 0  
Valid Range To: 200  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.431**     ***fund\_group***

Description: Fund Group  
Help:  
Modified On: 1/13/2020 1:25:18 PM (UTC+02:00)  
Modified By: CLAL-INS\NinaB  
Category: Policy Details  
Variable Type: Character  
Default Value: P  
Length: 2  
Number of Decimals: 1  
Choice List: P  
Character Type: Standard  
Valid Range From:  
Valid Range To:  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

**5.3.3.1.432      *fund\_name\_temp***

Description:	Fund name for model (for tables)
Help:	Model fund name/number, based on actual fund number from data file
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Character
Default Value:	52
Length:	2
Number of Decimals:	0
Choice List:	
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.433      *fund\_yesodi***

Description:	Fund (Keren) for main policy
Help:	10 = yod,..., 1 = aleph
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Character
Default Value:	52
Length:	2
Number of Decimals:	0
Choice List:	
Character Type:	Standard
Valid Range From:	0
Valid Range To:	200
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.434      *fundgroup\_manual***

Description:	fundgroup_manual
Help:	
Modified On:	8/5/2024 3:35:58 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt

Category:	Policy Details
Variable Type:	Character
Default Value:	P
Length:	2
Number of Decimals:	1
Choice List:	P
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.435     *health\_occ\_perc\_min***

Description:	Minimum extra health loading on qx
Help:	Extra health loading on premium and death rates according to occupation.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Floating Point Number
Default Value:	-20
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	-100
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.436     *health\_perc***

Description:	Extra loadings (health only) on premium/qx
Help:	Extra premium loading (percent of basic premium) for the policy for health conditions. Will only be added to the premium if variable health_occ_in_prem is set to N. Regardless this should also be used for the claim assumption.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Floating Point Number
Default Value:	0
Length:	0

Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	999
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.437**     *ind\_ifrs*

Description:	IFRS indicator
Help:	1 = NB F = Female
Modified On:	6/24/2024 3:30:53 PM (UTC+03:00)
Modified By:	CLAL-INS\Arikt
Category:	Policy Details
Variable Type:	Integer Number
Default Value:	0
Length:	1
Number of Decimals:	0
Choice List:	M
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.438**     *ind\_nb*

Description:	New business indicator
Help:	1 = NB F = Female
Modified On:	4/11/2024 3:33:52 PM (UTC+03:00)
Modified By:	CLAL-INS\Arikt
Category:	Policy Details
Variable Type:	Integer Number
Default Value:	0
Length:	1
Number of Decimals:	0
Choice List:	M
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers

Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.439 *index\_row\_num*

Description:	record number
Help:	index row num
Modified On:	8/4/2021 10:54:32 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Policy Details
Variable Type:	Character
Default Value:	Y
Length:	50
Number of Decimals:	0
Choice List:	Y
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.440 *insured*

Description:	insured
Help:	for health covers 1 = main 0 = child 2 = partner (used for looking up correct premium rate for non-family tariffs)
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Integer Number
Default Value:	1
Length:	1
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	2
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.441      *insured\_id***

Description:	ID no for main insured
Help:	ID no for main insured
Modified On:	8/19/2021 8:14:49 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Policy Details
Variable Type:	Floating Point Number
Default Value:	1
Length:	1
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	1
Valid Range To:	9999999999
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.442      *maasik\_no***

Description:	Maasik number for managers policies
Help:	Agency Number (Osek merushe number)
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Character
Default Value:	0
Length:	13
Number of Decimals:	0
Choice List:	
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.443      *matan\_period\_w***

Description:	Period t in which reached matan term
Help:	Calculated in startup column. Matan period only relevant for benefit class "MATAN".
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details

Variable Type:	Integer Number
Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	10000
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.444**     *matan\_term*

Description:	Matan term in months
Help:	Matan term in months after which the sum insured is reduced by $0.02 \times \text{matan\_term}$ .
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Integer Number
Default Value:	60
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	1
Valid Range To:	100000
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.445**     *maturity\_period\_ann*

Description:	Period t in which annuity ends up
Help:	Calculated in startup column.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Integer Number
Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0

Valid Range To:	600
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.446**     *maturity\_period\_w*

Description:	Period t in which policy matures
Help:	Calculated in startup column.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Integer Number
Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	600
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.447**     *mortg\_pmt\_fix\_w*

Description:	Monthly level mortgage payback
Help:	Monthly level mortgage payback. Calculated in startup.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	10000000000
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0



**5.3.3.1.448      *movement\_flag***

Description:	Type of Movement (death, surrender, no change...)
Help:	Code is used to identify special covers/policies. Not used in the model but passed to the output file for summing the results. 1= LTC NB05 old tarif 2= PHI mifali NB05 alone - added to old policy 3= PHI mifali NB05 alone - other new policy 4= PHI mifali NB05 alone - w/o other new policy
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Integer Number
Default Value:	0
Length:	1
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.449      *movement\_month***

Description:	Calender month of movement
Help:	Code is used to identify special covers/policies. Not used in the model but passed to the output file for summing the results. 1= LTC NB05 old tarif 2= PHI mifali NB05 alone - added to old policy 3= PHI mifali NB05 alone - other new policy 4= PHI mifali NB05 alone - w/o other new policy
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Integer Number
Default Value:	0
Length:	1
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	12
Table Format:	Default Row Numbers
Set Value in Input Manager:	All

Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.450 *movement\_status*

Description:	Movement Status (IF, NB, PU, NC...)
Help:	Code is used to identify special covers/policies. Not used in the model but passed to the output file for summing the results. 1= LTC NB05 old tarif 2= PHI mifali NB05 alone - added to old policy 3= PHI mifali NB05 alone - other new policy 4= PHI mifali NB05 alone - w/o other new policy
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Character
Default Value:	0
Length:	2
Number of Decimals:	0
Choice List:	
Character Type:	Standard
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.451 *occ\_key*

Description:	Occupational key
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Character
Default Value:	3
Length:	1
Number of Decimals:	1
Choice List:	1,2,3
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.452**      ***occ\_perc***

Description:	Extra loadings (occupation only) on premium/qx
Help:	Extra premium loading (percent of basic premium) for the policy for occupation. Will only be added to the premium if variable health_occ_in_prem is set to N. Regardless this should also be used for the claim assumption.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	999
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.453**      ***paid\_up***

Description:	Paid up at valuation date? (Y/N/G/C)
Help:	
Modified On:	5/9/2022 9:00:27 AM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Policy Details
Variable Type:	Character
Default Value:	N
Length:	1
Number of Decimals:	0
Choice List:	N,Y
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.454**      ***paid\_up\_input***

Description:	Paid up at valuation date? (Y/N/G/C)
Help:	
Modified On:	5/9/2022 9:00:46 AM (UTC+03:00)

Modified By:	CLAL-INS\ahuvaa
Category:	Policy Details
Variable Type:	Character
Default Value:	N
Length:	1
Number of Decimals:	1
Choice List:	N,Y
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.455 *par\_nonpar*

Description:	Participating or Non-Participating?
Help:	P = Participating business (i.e. with-profit) N = Non participating business (i.e. non-profit)
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Character
Default Value:	P
Length:	1
Number of Decimals:	0
Choice List:	P,N
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.456 *par\_nonpar\_yesodi*

Description:	Participating or Non-Participating? for yesodi policy
Help:	P = Participating business (i.e. with-profit) N = Non participating business (i.e. non-profit)
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Character
Default Value:	P
Length:	1

Number of Decimals:	0
Choice List:	P,N
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.457**     ***pol\_index***

Description:	Policy index (Pol No, Company, Tarif, Tafkid)
Help:	Policy Index to uniquely identify cover. Made from Policy number, Company code, Tarif and Tafkid
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Character
Default Value:	000001
Length:	30
Number of Decimals:	0
Choice List:	
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.458**     ***pol\_number***

Description:	Policy number
Help:	Policy number.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Character
Default Value:	000001
Length:	12
Number of Decimals:	0
Choice List:	
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers

Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.459 *policies\_curr*

Description:	Number of policies inforce at valn date
Help:	Current number of in force policies at the valuation date.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Floating Point Number
Default Value:	1
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1000000
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.460 *policy\_type*

Description:	policy type
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Character
Default Value:	Managers
Length:	13
Number of Decimals:	0
Choice List:	managers,private,selfemp,health,Managers
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.461 *policy\_type\_orig*

Description:	original policy type
--------------	----------------------

Help:  
 Modified On: 12/26/2022 1:41:14 PM (UTC+02:00)  
 Modified By: CLAL-INS\ahuvaa  
 Category: Policy Details  
 Variable Type: Character  
 Default Value: Managers  
 Length: 13  
 Number of Decimals: 0  
 Choice List: managers,private,selfemp,health,Managers  
 Character Type: Standard  
 Valid Range From: 0  
 Valid Range To: 0  
 Table Format: Default Row Numbers  
 Set Value in Input Manager: All  
 Variable Sharing: Not Shared  
 Category Order: 0

#### **5.3.3.1.462**     ***prem\_profil\_rider\_type***

Description: prem type (0=out, 1=in) per profil rider  
 Help: prem type (0=out, 1=in) per profil rider.  
 Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
 Modified By: CLAL-INS\ninab  
 Category: Policy Details  
 Variable Type: Floating Point Array  
 Default Value: 0  
 Length: 26  
 Number of Decimals: 3  
 Choice List:  
 Character Type: Not Applicable  
 Valid Range From: 0  
 Valid Range To: 1  
 Table Format: Default Row Numbers  
 Set Value in Input Manager:  
 Variable Sharing: Not Shared  
 Category Order: 0

#### **5.3.3.1.463**     ***prem\_term***

Description: Policy premium term (months)  
 Help: Original premium term in months.  
 Set in set\_other\_variables.  
 Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
 Modified By: CLAL-INS\ninab  
 Category: Policy Details  
 Variable Type: Integer Number  
 Default Value: 120  
 Length: 0

Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1200
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.464     *prem\_term\_input***

Description:	Policy premium term (months)
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Integer Number
Default Value:	120
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1200
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.465     *prod\_code***

Description:	Product code
Help:	read in from inforce file
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Character
Default Value:	392
Length:	15
Number of Decimals:	0
Choice List:	
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared



Category Order: 0

#### **5.3.3.1.466     *prod\_code\_adif\_extra\_prem***

Description: Product code for reading extra prem tables for adif  
Help: read in from inforce file  
Modified On: 8/12/2021 9:43:03 AM (UTC+03:00)  
Modified By: CLAL-INS\joshm  
Category: Policy Details  
Variable Type: Character  
Default Value: none  
Length: 15  
Number of Decimals: 0  
Choice List: none  
Character Type: Standard  
Valid Range From: 0  
Valid Range To: 0  
Table Format: Default Row Numbers  
Set Value in Input Manager:  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.467     *prod\_code\_adif\_extra\_prem\_temp***

Description: Product code for reading extra prem tables for adif  
Help: read in from inforce file  
Modified On: 8/12/2021 4:29:34 PM (UTC+03:00)  
Modified By: CLAL-INS\joshm  
Category: Policy Details  
Variable Type: Character  
Default Value: none  
Length: 15  
Number of Decimals: 0  
Choice List: none  
Character Type: Standard  
Valid Range From: 0  
Valid Range To: 0  
Table Format: Default Row Numbers  
Set Value in Input Manager:  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.468     *prod\_code\_base***

Description: Product code of the base (Yessodi) cover  
Help: read in from inforce file

Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Character
Default Value:	prof00
Length:	15
Number of Decimals:	0
Choice List:	
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.469**     ***prod\_code\_old***

Description:	Product code - old - from tarif spec
Help:	read in from inforce file
Modified On:	8/16/2021 2:36:59 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Policy Details
Variable Type:	Character
Default Value:	none
Length:	15
Number of Decimals:	0
Choice List:	none
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.470**     ***prod\_code\_rider***

Description:	prod_code_rider
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Character
Default Value:	rsapir1-p
Length:	15
Number of Decimals:	0
Choice List:	

Character Type:	Standard
Valid Range From:	0
Valid Range To:	99999
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### **5.3.3.1.471     *prod\_code\_rider\_floating***

Description:	prod_code_rider for interface
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Floating Point Array
Default Value:	1 1
Length:	25
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	10000
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### **5.3.3.1.472     *prod\_group\_yessodi\_portfolio***

Description:	ifrs product definition
Help:	
Modified On:	6/8/2023 2:21:07 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Policy Details
Variable Type:	Character
Default Value:	tpd
Length:	6
Number of Decimals:	1
Choice List:	dth,ddth,dd,tpd,adb,fib,ltc,mortg,phi,health,adif,profil
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.473     *prod\_yr\_w***

Description:	production year, used for DAC issues
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Integer Number
Default Value:	2004
Length:	1
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	2004
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.474     *profil\_dyn\_child\_sa***

Description:	Profil dynamic model-Amount of SA for child
Help:	prem type (0=out, 1=in) per profil rider.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Integer Array
Default Value:	0
Length:	26
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	8
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.475     *profil\_dyn\_child\_term***

Description:	Profil dynamic model-Risk reduction term for child
Help:	prem type (0=out, 1=in) per profil rider.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab

Category:	Policy Details
Variable Type:	Integer Array
Default Value:	0
Length:	26
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	8
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.476     *profil\_dyn\_spous\_sa***

Description:	Profil dynamic model-Amount of SA for spous
Help:	prem type (0=out, 1=in) per profil rider.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Integer Array
Default Value:	0
Length:	26
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	8
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.477     *profil\_dyn\_spous\_term***

Description:	Profil dynamic model-Risk reduction term for spous
Help:	prem type (0=out, 1=in) per profil rider.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Integer Array
Default Value:	0
Length:	26
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable

Valid Range From:	0
Valid Range To:	8
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### **5.3.3.1.478     *profil\_dynamic***

Description:	Profil dynamic model (0=No, 1=Yes)
Help:	prem type (0=out, 1=in) per profil rider.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Integer Array
Default Value:	0
Length:	26
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	8
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### **5.3.3.1.479     *profil\_rider\_type***

Description:	death benefit type (1=fixed, 2=extra) SI
Help:	prem type (0=out, 1=in) per profil rider.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Integer Array
Default Value:	0
Length:	26
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	8
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.480**     ***prog\_name***

Description:	prog_name field from data file
Help:	Used for classifying reserves between health and life
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Character
Default Value:	BLANK
Length:	12
Number of Decimals:	0
Choice List:	ADIF,BRIUT,KLASI,BLANK
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.481**     ***rider\_sex***

Description:	Sex of profil rider cover
Help:	M = Male F = Female
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Floating Point Array
Default Value:	1
Length:	25
Number of Decimals:	0
Choice List:	M,F,A
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.482**     ***rider\_type\_w***

Description:	rider_type_w
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details

Variable Type:	Integer Array
Default Value:	0
Length:	25
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	10
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.483**     *riders\_count\_w*

Description:	Number of riders for current Profil policy
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Integer Number
Default Value:	0
Length:	1
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	25
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.484**     *riders\_count\_w\_input*

Description:	Number of riders for current Profil policy
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Integer Number
Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	25



Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.485 *risk\_code*

Description:	product code of risk rider with Meitav (Managers)
Help:	This is the product code of the risk rider (Sapir) that shares the total premium with the policy (Meitav Managers) being run.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Character
Default Value:	394
Length:	20
Number of Decimals:	0
Choice List:	
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.486 *risk\_si*

Description:	sum insured of risk rider with Meitav (Managers)
Help:	This is the sum insured of the risk rider (Sapir) that shares the total premium with the policy (Meitav Managers) being run.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Floating Point Number
Default Value:	0
Length:	1
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	20000000
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared

Category Order: 0

#### **5.3.3.1.487**      ***saving\_max\_perc***

Description: Maximum saving percentage allowed  
Help: Used for adjusting basic savings when there is a fixed premium  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Policy Details  
Variable Type: Floating Point Number  
Default Value: 100  
Length: 1  
Number of Decimals: 0  
Choice List:  
Character Type: Not Applicable  
Valid Range From: 0  
Valid Range To: 100  
Table Format: Default Row Numbers  
Set Value in Input Manager:  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.488**      ***saving\_perc***

Description: Total percentage of savings (basic + extra)  
Help: For Adif: Total percentage of savings (basic + pure savings) .  
For Profil: Percentage of total premium allocated to pure savings (rest goes to normal product).  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Policy Details  
Variable Type: Floating Point Number  
Default Value: 80  
Length: 0  
Number of Decimals: 2  
Choice List:  
Character Type: Not Applicable  
Valid Range From: 0  
Valid Range To: 100  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

**5.3.3.1.489 sex**

Description:	Sex
Help:	M = Male F = Female
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Character
Default Value:	M
Length:	1
Number of Decimals:	0
Choice List:	M,F,A
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.490 smoker\_stat**

Description:	Smoker status
Help:	Smoker status under which the policy has been issued: N = Non smoker, S = Smoker or A = Aggregate
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Character
Default Value:	N
Length:	1
Number of Decimals:	0
Choice List:	N,S,A
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.491 submodel**

Description:	sub model to run
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt

Category:	Policy Details
Variable Type:	Character
Default Value:	TERM
Length:	4
Number of Decimals:	0
Choice List:	TERM,TRAD,UNIT
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.492**      ***sum\_ins\_curr***

Description:	Sum Insured at valn date
Help:	Current sum assured at the valuation date. For mortgage this is the original sum-insured.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Floating Point Number
Default Value:	100000
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100000000
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.493**      ***sum\_ins\_curr\_input***

Description:	Sum Insured at valn date
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Floating Point Number
Default Value:	100000
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable

Valid Range From:	0
Valid Range To:	100000000
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.494**      ***sum\_ins\_curr\_rider***

Description:	sum insured (average) for each rider - array
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Floating Point Array
Default Value:	0
Length:	25
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100000000
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.495**      ***surr\_value\_if***

Description:	Surrender value at valn date from IF file
Help:	units at valuation date (accumulation/reserve) per 1 benefit
Modified On:	8/5/2024 4:12:43 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Policy Details
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	-99999999
Valid Range To:	99999999
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.496**      ***surr\_value\_if\_input***

Description:	Surrender value at valn date from IF file - input
Help:	units at valuation date (accumulation/reserve) per 1 benefit
Modified On:	8/5/2024 4:09:28 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Policy Details
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	-99999999
Valid Range To:	999999999
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.497**      ***tarif***

Description:	tarif
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Integer Number
Default Value:	0
Length:	1
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.498**      ***tarif\_rider***

Description:	tarif per profil rider
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details

Variable Type:	Integer Array
Default Value:	6300 6301 0 0 0 0 0 0 0 0
Length:	26
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	99999
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.499      *temp\_agency\_no***

Description:	Lookup value code variable
Help:	
Modified On:	2/13/2022 10:06:26 AM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Policy Details
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.500      *unit\_value\_accum***

Description:	Accum unit balance at valn date
Help:	units at valuation date (accumulation/reserve) per 1 benefit
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	-9999999

Valid Range To:	999999999
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.501      *unit\_value\_accum\_input***

Description:	Accum unit balance at valn date
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	3
Choice List:	
Character Type:	Not Applicable
Valid Range From:	-999999999
Valid Range To:	999999999
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.502      *unit\_value\_savings***

Description:	Savings unit balance at valn date
Help:	units at valuation date (extra savings account) per 1 benefit
Modified On:	8/5/2024 4:12:20 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Policy Details
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	-999999999
Valid Range To:	999999999
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0



**5.3.3.1.503      *unit\_value\_savings\_input***

Description:	Savings unit balance at valn date - input
Help:	units at valuation date (extra savings account) per 1 benefit
Modified On:	8/5/2024 4:08:04 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Policy Details
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	-99999999
Valid Range To:	999999999
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.504      *year\_prod***

Description:	Year of policy production (prod-date)
Help:	For reporting purposes only
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details
Variable Type:	Integer Number
Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1200
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.505      *year\_start***

Description:	Year of policy start (origi-date)
Help:	For reporting purposes only
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Policy Details

Variable Type:	Integer Number
Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1200
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.506 *adjust\_prem\_and\_claims\_temp*

Description:	Adjust premium and claims by actual-calculated premium ratio
Help:	This is used primarily for PHI, to indicate if the model should increase premiums and claims based on the ratio between the premium in the data and the calculated (standard) premium. It is a way of modeling occupation classes, health/hobby/occupation additions etc. This variable is set in "set_by_prodcode" from the prod_spec_term table. "N" - do not adjust "Y" - adjust
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium
Variable Type:	Character
Default Value:	N
Length:	1
Number of Decimals:	0
Choice List:	Y,N
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.507 *alloc\_limit*

Description:	alloc_limit for maximum premium charge as shekel amount
--------------	---

Help:	Percentage of premium allocated to units. Each element in the array applies for the number of months specified in the alloc_rate_period array. The first element in the array should be entered for period 1. Read in from alloc_rate_tbl in set_common_variables.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium
Variable Type:	Floating Point Number
Default Value:	0
Length:	16
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	150
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.508 *allocation\_limit\_amount*

Description:	maximum DNP - monthly - shekel
Help:	Percentage of premium allocated to units. Each element in the array applies for the number of months specified in the alloc_rate_period array. The first element in the array should be entered for period 1. Read in from alloc_rate_tbl in set_common_variables.
Modified On:	2/13/2025 4:01:21 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Premium
Variable Type:	Floating Point Number
Default Value:	0
Length:	16
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	150
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.509 aloc\_kafuy**

Description:	1-DNP as today
Help:	Percentage of premium allocated to units. Each element in the array applies for the number of months specified in the alloc_rate_period array. The first element in the array should be entered for period 1. Read in from alloc_rate_tbl in set_common_variables.
Modified On:	2/13/2025 3:50:53 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Premium
Variable Type:	Floating Point Number
Default Value:	0
Length:	16
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	150
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.510 amla\_hishtatfut\_dnp**

Description:	Commission as % of DNP - With no VAT
Help:	Percentage of premium allocated to units. Each element in the array applies for the number of months specified in the alloc_rate_period array. The first element in the array should be entered for period 1. Read in from alloc_rate_tbl in set_common_variables.
Modified On:	2/13/2025 3:54:02 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Premium
Variable Type:	Floating Point Number
Default Value:	0
Length:	16
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	150
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.511      *base***

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium
Variable Type:	Floating Point Number
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.512      *basic\_perc\_w***

Description:	basic premium proportion
Help:	basic premium proportion : calculated in startup from saving_perc
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium
Variable Type:	Floating Point Number
Default Value:	50
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.513      *charge\_rate\_tt\_col***

Description:	Lookup value
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium

Variable Type:	Character
Default Value:	0
Length:	22
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.514**      ***charge\_rate\_tt\_row***

Description:	Lookup value
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium
Variable Type:	Character
Default Value:	0
Length:	22
Number of Decimals:	2
Choice List:	
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.515**      ***col\_name***

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	

Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.516 *health\_occ\_in\_prem*

Description:	Health loading included in prem_curr?
Help:	Health loading already included in current premium ? (occupational)
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium
Variable Type:	Character
Default Value:	Y
Length:	1
Number of Decimals:	0
Choice List:	Y,N
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.517 *health\_occ\_perc*

Description:	Extra loadings (health+occ) on premium/qx
Help:	No longer taken from data. Calculated in set other variables as the sum of health and occupational loadings, each of which are taken from the data.
	Extra premium loading (percent of basic premium) for the policy for health conditions and occupation. Will only be added to the premium if variable health_occ_in_prem is set to N. Regardless this should also be used for the claim assumption.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable

Valid Range From:	0
Valid Range To:	999
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.518 *imp\_manual\_alloc\_rate\_term\_dt*

Description:	DNP benefit final date
Help:	Percentage of premium allocated to units. Each element in the array applies for the number of months specified in the alloc_rate_period array. The first element in the array should be entered for period 1. Read in from alloc_rate_tbl in set_common_variables.
Modified On:	3/10/2025 12:17:11 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Premium
Variable Type:	Character
Default Value:	0
Length:	16
Number of Decimals:	2
Choice List:	0
Character Type:	Standard
Valid Range From:	0
Valid Range To:	150
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.519 *mod\_load\_in\_prem*

Description:	modal loading already includ. in prem?
Help:	if N : add to prem_curr in the startup if "level prem" if prem_lookup = "Y", modal loading will always be added to the premium and this value will not affect it. Set in startup based on "prog_name" (Klasi - "N", others - "Y")
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium
Variable Type:	Character
Default Value:	Y
Length:	1
Number of Decimals:	0



Choice List:	Y,N
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.520      *netprem\_max***

Description:	Max net prem as % gross prem
Help:	The net premium is limited to this percentage of the gross (office) premium.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium
Variable Type:	Floating Point Number
Default Value:	1000
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100000
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.521      *pol\_fee\_disc\_perc***

Description:	Policy fee discount as a percentage
Help:	discount applied to polict fee.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	-100
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared

Category Order: 0

#### **5.3.3.1.522     *pol\_number\_i***

Description: Pol Number I  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Premium  
Variable Type: Character  
Default Value: 0  
Length: 15  
Number of Decimals: 1  
Choice List: 0  
Character Type: Standard  
Valid Range From:  
Valid Range To:  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.523     *policy\_fee\_if***

Description: Annual policy fee  
Help: Policy fee set from pol\_fee\_tbl in the startup if  
"read\_from\_tables"=Y  
add to prem\_curr in the startup if "level prem"  
add to prem\_if\_b if YRT  
Modified On: 5/2/2022 9:01:58 AM (UTC+03:00)  
Modified By: CLAL-INS\ahuvaa  
Category: Premium  
Variable Type: Floating Point Number  
Default Value: 10  
Length: 0  
Number of Decimals: 2  
Choice List:  
Character Type: Not Applicable  
Valid Range From: 0  
Valid Range To: 10000000  
Table Format: Default Row Numbers  
Set Value in Input Manager:  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.524     *policy\_fee\_input***

Description: Annual policy fee

Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Premium  
Variable Type: Floating Point Number  
Default Value: 10  
Length: 0  
Number of Decimals: 2  
Choice List:  
Character Type: Not Applicable  
Valid Range From: 0  
Valid Range To: 1000000  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.525     *prate\_level\_tbl***

Description: premium rate table by entry age + term  
Help: level premium rate rate table.  
looked up by age and benefit-term (NOT premium-term)  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Premium  
Variable Type: Floating Point Number  
Default Value: 0  
Length: 0  
Number of Decimals: 1  
Choice List:  
Character Type: Not Applicable  
Valid Range From: 0  
Valid Range To: 10000  
Table Format: Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.526     *prem\_code***

Description: premium lookup code  
Help: This is the index value to llokup premium rates in the prem\_rates table (consolidated premium rates table). It is looked up from the prem-code-map table)  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Premium

Variable Type:	Character
Default Value:	0
Length:	22
Number of Decimals:	0
Choice List:	
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.527**     ***prem\_code\_map\_tbl***

Description:	health prem_code_map_tbl
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Row Name (string)
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.528**     ***prem\_code\_rider***

Description:	premium lookup code for profil riders
Help:	This is the index value to llokup premium rates in the prem_rates table (consolidated premium rates table). It is looked up from the prem-code-map table)
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium
Variable Type:	Character
Default Value:	0
Length:	550
Number of Decimals:	0
Choice List:	

Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.529     *prem\_code\_test***

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium
Variable Type:	Character
Default Value:	0
Length:	20
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.530     *prem\_code\_test\_temp***

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	0
Choice List:	
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.531      *prem\_curr***

Description:	Annual premium at valn date
Help:	Current in force annual premium per policy at the valuation date.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium
Variable Type:	Floating Point Number
Default Value:	2000
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	10000000
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.532      *prem\_curr\_changed***

Description:	Prem Curr Changed
Help:	indicate if the prem_curr parameter has been set to a calculated value
Modified On:	10/5/2021 11:54:31 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Premium
Variable Type:	Character
Default Value:	N
Length:	1
Number of Decimals:	1
Choice List:	N,Y
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.533      *prem\_curr\_if***

Description:	Annual gross premium from data file
Help:	Current in force annual premium per policy at the valuation date.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab

Category:	Premium
Variable Type:	Floating Point Number
Default Value:	2000
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	10000000
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.534**     ***prem\_curr\_input***

Description:	Annual premium at valn date
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium
Variable Type:	Floating Point Number
Default Value:	2000
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	10000000
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.535**     ***prem\_curr\_rider***

Description:	annual premium (current) for each rider - array
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium
Variable Type:	Floating Point Array
Default Value:	0
Length:	25
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0

Valid Range To:	100000
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.536**     ***prem\_freq***

Description:	Premium frequency
Help:	Frequency of premium payment: 1 - Annually 2 - Half-Yearly 4 - Quarterly 12 - Monthly
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium
Variable Type:	Integer Number
Default Value:	12
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	1
Valid Range To:	12
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.537**     ***prem\_if\_rates***

Description:	Lookup code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared



Category Order: 0

#### **5.3.3.1.538      *prem\_init\_different\_temp***

Description: Does initial prem differ from the renew.  
Help: Is the initial premium different from the renewal premium?  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Premium  
Variable Type: Character  
Default Value: N  
Length: 1  
Number of Decimals: 0  
Choice List: Y,N  
Character Type: Standard  
Valid Range From: 0  
Valid Range To: 0  
Table Format: Default Row Numbers  
Set Value in Input Manager:  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.539      *prem\_key\_temp***

Description: Lookup value code variable  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Premium  
Variable Type: Character  
Default Value: 0  
Length: 20  
Number of Decimals: 1  
Choice List: 0  
Character Type: Standard  
Valid Range From:  
Valid Range To:  
Table Format: Default Row Numbers  
Set Value in Input Manager:  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.540      *prem\_lookup\_freq\_temp***

Description: premium lookup frequency

Help:	read in from setup_tbl and set in set_categ_variables() 0 = level premium 1 = YRT 3,5,10 = sub terms, indicates whenever premium rate changes.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium
Variable Type:	Integer Number
Default Value:	3
Length:	1
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.541      *prem\_lookup\_freq\_w***

Description:	Frequency of charge rate renewal for Profil Riders
Help:	Premium lookup frequency in years (eg 1 or 5) for charge or premium rates. Used for Profil riders (elements 0 to 24) and/or Meitav risk rider (element 25).
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium
Variable Type:	Integer Array
Default Value:	1
Length:	26
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	1
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.542      *prem\_lookup\_temp***

Description:	premium lookup (Y/N)
--------------	----------------------

Help:	read in from setup_tbl and set in set_categ_variables() "N" = level premium "Y" = YRT or stepped premium rates
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium
Variable Type:	Character
Default Value:	N
Length:	1
Number of Decimals:	0
Choice List:	Y,N
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### 5.3.3.1.543 ***prem\_newtag\_prop***

Description:	Proportion of new premiums allocated to tagmulim
Help:	
Modified On:	8/31/2020 2:34:20 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Category:	Premium
Variable Type:	Floating Point Number
Default Value:	55
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### 5.3.3.1.544 ***prem\_orig***

Description:	Original premium at valn date
Help:	Current in force annual premium per policy at the valuation date.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium

Variable Type:	Floating Point Number
Default Value:	2000
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	10000000
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.545**     ***prem\_rate\_col***

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.546**     ***prem\_rate\_multiplier\_rider***

Description:	percentage; modifies premium rate for profil riders
Help:	Applied to premium rate for profil riders. Taken from the tarif spec table or set to 100%.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium
Variable Type:	Floating Point Array
Default Value:	1
Length:	26
Number of Decimals:	3
Choice List:	
Character Type:	Not Applicable

Valid Range From:	0
Valid Range To:	2
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### **5.3.3.1.547     *prem\_rate\_row***

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium
Variable Type:	Integer Number
Default Value:	0
Length:	10
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### **5.3.3.1.548     *prem\_rate\_scale\_w***

Description:	scale of Prem rate in table
Help:	scale of prem rate in premium rate table i.e per benefit of a 1000 SI or 100 SI
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium
Variable Type:	Integer Number
Default Value:	1000
Length:	1
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	1
Valid Range To:	1000
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.549      *prem\_rates\_charge\_tt***

Description:	Lookup code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.550      *prem\_rates\_extra\_prm***

Description:	Composite external source
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.551      *prem\_rates\_risk***

Description:	Premium rate table for risk rider
Help:	Premium rate set in set_common_variables(), per 1000 SA.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium

Variable Type:	Character
Default Value:	
Length:	41
Number of Decimals:	0
Choice List:	
Character Type:	Standard
Valid Range From:	0
Valid Range To:	10000
Table Format:	Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.552**     ***prem\_rates\_risk\_1***

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.553**     ***prem\_rates\_risk\_2***

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	

Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.554**     ***prem\_rates\_risk\_rider***

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.555**     ***prem\_rates\_row***

Description:	Lookup code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium
Variable Type:	Integer Number
Default Value:	0
Length:	10
Number of Decimals:	0
Choice List:	0
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0



**5.3.3.1.556      *prem\_rates\_series***

Description:	prem_rates external source Series End
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium
Variable Type:	Integer Number
Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.557      *prem\_rates\_si***

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.558      *prem\_rates\_si\_col***

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium
Variable Type:	Character

Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.559     *prem\_rates\_si\_row***

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.560     *prem\_rates\_temp\_series\_end***

Description:	Lookup value code variable
Help:	
Modified On:	1/6/2022 5:29:54 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Premium
Variable Type:	Integer Number
Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers

Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.561     *prem\_risk\_max***

Description: max perc of prem for extra SI (Sapir)  
Help:  
Modified On: 5/2/2022 11:35:39 AM (UTC+03:00)  
Modified By: CLAL-INS\ahuvaa  
Category: Premium  
Variable Type: Floating Point Number  
Default Value: 0  
Length: 1  
Number of Decimals: 2  
Choice List:  
Character Type: Not Applicable  
Valid Range From: 0  
Valid Range To: 100  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.562     *premium\_rate\_w***

Description: level premium rate per mille  
Help: Premium rate (level) set in startup, per 1000 SA.  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Premium  
Variable Type: Floating Point Number  
Default Value: 10  
Length: 0  
Number of Decimals: 3  
Choice List:  
Character Type: Not Applicable  
Valid Range From: 0  
Valid Range To: 10000  
Table Format: Default Row Numbers  
Set Value in Input Manager:  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.563     *product\_alloc\_rate\_percent***

Description: 1-DNP for product (with no benefits)

Help:	Percentage of premium allocated to units. Each element in the array applies for the number of months specified in the alloc_rate_period array. The first element in the array should be entered for period 1. Read in from alloc_rate_tbl in set_common_variables.
Modified On:	2/13/2025 3:52:50 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Premium
Variable Type:	Floating Point Number
Default Value:	0
Length:	16
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	150
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.564 *promil*

Description:	promil from data file
Help:	Used for free covers (zero premium) as criteria to skip or project the cover.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	10000000
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.565 *reinsur\_simple\_perc*

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab

Category:	Premium
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.566**     *reinsur\_simple\_rider\_cost*

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.567**     *reinsure\_simple\_cost*

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	

Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.568**     ***sal\_inc\_set***

Description:	Lookup value code variable wildcard
Help:	
Modified On:	12/29/2022 4:18:00 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Premium
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.569**     ***sal\_inc\_set\_rider***

Description:	Lookup value code variable wildcard
Help:	
Modified On:	12/29/2022 4:57:05 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Premium
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.570      *sal\_tbl***

Description:	Salary increase table
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Row Name (string)
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.571      *tagmulim\_perc***

Description:	Percentage of tagmulim from premium (after allocation)
Help:	For New Profil: Percentage of tagmulim premium (including basic + pure savings) from the total premium (including basic + pure savings), both after allocation rates. This is only used in order to limit the amount of risk riders, based on the variable rider_max_perc For PRIVATE policies this is not used.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium
Variable Type:	Floating Point Number
Default Value:	54.56
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.572      *term\_in\_profil***

Description:	term_in_profil indicator for sal_inc lookup
Help:	
Modified On:	5/16/2023 11:56:20 AM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Premium
Variable Type:	Character
Default Value:	N
Length:	1
Number of Decimals:	0
Choice List:	Y,N
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.573      *use\_tarif\_spec\_prem***

Description:	Use tarif spec table for premium rates
Help:	Apply SA multipliers (claims cost table)? (use for LTC, PHI, etc. claims) If set_by_prodcod = "Y" then looked up from prod specs table.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium
Variable Type:	Character
Default Value:	N
Length:	1
Number of Decimals:	0
Choice List:	Y,N
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.574      *use\_tarif\_spec\_prem\_rider***

Description:	Use tarif spec table for premium rates for profil riders
--------------	--



Help:	Indicator for premium rates table to use for profil riders. If =1, then use the tarif spec table premium rates lookup, If =0, then use the product code level premium lookup.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium
Variable Type:	Integer Array
Default Value:	0
Length:	25
Number of Decimals:	0
Choice List:	Y,N
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.575      *discount\_perc\_rider***

Description:	charge discount (%) for each rider - array
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium discounts
Variable Type:	Floating Point Array
Default Value:	0
Length:	25
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	10000000
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.576      *discount\_period\_rider***

Description:	charge discount period (months) for each rider - array
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab

Category:	Premium discounts
Variable Type:	Floating Point Array
Default Value:	0
Length:	25
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	10000000
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.577 *prem\_disc\_dcr1\_m*

Description:	Premium decreasing discount1 months
Help:	Month from policy start when Premium discount period ends.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium discounts
Variable Type:	Integer Number
Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1200
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.578 *prem\_disc\_dcr1\_r*

Description:	Premium decreasing discount1 rate
Help:	Premium discount as a percentage of premium . Applied during a defined period (see Prem_disc_month).
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium discounts
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	2

Choice List:	
Character Type:	Not Applicable
Valid Range From:	-100
Valid Range To:	500
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.579      *prem\_disc\_dcr2\_m***

Description:	Premium decreasing discount2 months
Help:	Month from policy start when Premium discount period ends.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium discounts
Variable Type:	Integer Number
Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1200
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.580      *prem\_disc\_dcr2\_r***

Description:	Premium decreasing discount2 rate
Help:	Premium discount as a percentage of premium . Applied during a defined period (see Prem_disc_month).
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium discounts
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	-100
Valid Range To:	500
Table Format:	Default Row Numbers

Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.581      *prem\_disc\_dcr3\_m***

Description:	Premium descreasing discount3 months
Help:	Month from policy start when Premium discount period ends.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium discounts
Variable Type:	Integer Number
Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1200
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.582      *prem\_disc\_dcr3\_r***

Description:	Premium descreasing discount3 rate
Help:	Premium discount as a percentage of premium . Applied during a defined period (see Prem_disc_month).
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium discounts
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	-100
Valid Range To:	500
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.583      *prem\_disc\_dcr4\_m***

Description:	Premium decreasing discount4 months
Help:	Month from policy start when Premium discount period ends.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium discounts
Variable Type:	Integer Number
Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1200
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.584      *prem\_disc\_dcr4\_r***

Description:	Premium decreasing discount4 rate
Help:	Premium discount as a percentage of premium . Applied during a defined period (see Prem_disc_month).
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium discounts
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	-100
Valid Range To:	500
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.585      *prem\_disc\_dcr5\_m***

Description:	Premium decreasing discount5 months
Help:	Month from policy start when Premium discount period ends.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)

Modified By:	CLAL-INS\ninab
Category:	Premium discounts
Variable Type:	Integer Number
Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1200
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.586 *prem\_disc\_dcr5\_r*

Description:	Premium decreasing discount5 rate
Help:	Premium discount as a percentage of premium . Applied during a defined period (see Prem_disc_month).
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium discounts
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	-100
Valid Range To:	500
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.587 *prem\_disc\_month*

Description:	Premium discount period (last month)
Help:	Month from policy start when Premium discount period ends.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium discounts
Variable Type:	Integer Number
Default Value:	0
Length:	0

Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1200
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.588      *prem\_disc\_month\_2***

Description:	Premium discount period (last month)
Help:	Month from policy start when Premium discount period ends.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium discounts
Variable Type:	Integer Number
Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1200
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.589      *prem\_disc\_month\_2\_input***

Description:	Premium discount period (last month)
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium discounts
Variable Type:	Integer Number
Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1200
Table Format:	Default Row Numbers
Set Value in Input Manager:	All

Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.590      *prem\_disc\_month\_input***

Description: Premium discount period (last month)  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Premium discounts  
Variable Type: Integer Number  
Default Value: 0  
Length: 0  
Number of Decimals: 0  
Choice List:  
Character Type: Not Applicable  
Valid Range From: 0  
Valid Range To: 1200  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.591      *prem\_disc\_perc***

Description: Premium discount as a percentage  
Help: Premium discount as a percentage of premium  
. Applied during a defined period (see Prem\_disc\_month).  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Premium discounts  
Variable Type: Floating Point Number  
Default Value: 0  
Length: 0  
Number of Decimals: 2  
Choice List:  
Character Type: Not Applicable  
Valid Range From: -100  
Valid Range To: 500  
Table Format: Default Row Numbers  
Set Value in Input Manager:  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.592      *prem\_disc\_perc\_2***

Description: Premium discount as a percentage



Help:	Premium discount as a percentage of premium . Applied during a defined period (see Prem_disc_month).
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium discounts
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	-100
Valid Range To:	500
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.593     *prem\_disc\_perc\_2\_input***

Description:	Premium discount as a percentage
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium discounts
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	-100
Valid Range To:	500
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.594     *prem\_disc\_perc\_input***

Description:	Premium discount as a percentage
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium discounts
Variable Type:	Floating Point Number
Default Value:	0

Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	-100
Valid Range To:	500
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.595      *prem\_disc\_shimur\_flag***

Description:	Use premium discount shimur
Help:	Premium discount as a percentage of premium . Applied during a defined period (see Prem_disc_month).
Modified On:	2/11/2024 4:56:49 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Premium discounts
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	2
Choice List:	0
Character Type:	Standard
Valid Range From:	-100
Valid Range To:	500
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.596      *prem\_disc\_shimur\_im***

Description:	Lookup value code variable
Help:	
Modified On:	2/11/2024 3:23:26 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Premium discounts
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	

Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.597 *prem\_disc\_step*

Description:	Premium discount decrease type
Help:	0 = no loading 1 = goreem on yesodi 3 = temp discount on all covers in policy 4 = permanent discount on all covers in policy 5 = temp discount on specific cover 6 = permanent discount on specific cover other = goreem
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium discounts
Variable Type:	Integer Number
Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	480
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.598 *prem\_disc\_step1\_m*

Description:	The 1st step prem discount period
Help:	Month from policy start when Premium discount period ends.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium discounts
Variable Type:	Integer Number
Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1200
Table Format:	Default Row Numbers
Set Value in Input Manager:	

Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.599      *prem\_disc\_step1\_r***

Description: The 1st step prem discount rate as a percentage  
Help: Premium discount as a percentage of premium . Applied during a defined period (see Prem\_disc\_month).  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Premium discounts  
Variable Type: Floating Point Number  
Default Value: 0  
Length: 0  
Number of Decimals: 2  
Choice List:  
Character Type: Not Applicable  
Valid Range From: -100  
Valid Range To: 500  
Table Format: Default Row Numbers  
Set Value in Input Manager:  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.600      *prem\_disc\_step2\_m***

Description: The 2nd step prem discount period  
Help: Month from policy start when Premium discount period ends.  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Premium discounts  
Variable Type: Integer Number  
Default Value: 0  
Length: 0  
Number of Decimals: 0  
Choice List:  
Character Type: Not Applicable  
Valid Range From: 0  
Valid Range To: 1200  
Table Format: Default Row Numbers  
Set Value in Input Manager:  
Variable Sharing: Not Shared  
Category Order: 0

**5.3.3.1.601      *prem\_disc\_step2\_r***

Description:	The 2st step additional prem discount rate
Help:	Premium discount as a percentage of premium . Applied during a defined period (see Prem_disc_month).
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium discounts
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	-100
Valid Range To:	500
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.602      *prem\_disc\_step3\_m***

Description:	The 3th step prem discount period
Help:	Month from policy start when Premium discount period ends.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium discounts
Variable Type:	Integer Number
Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1200
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.603      *prem\_disc\_step3\_r***

Description:	The 3th step additional prem discount rate
Help:	Premium discount as a percentage of premium . Applied during a defined period (see Prem_disc_month).

Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium discounts
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	-100
Valid Range To:	500
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.604      *prem\_disc\_step4\_m***

Description:	The 4th step prem discount period
Help:	Month from policy start when Premium discount period ends.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium discounts
Variable Type:	Integer Number
Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1200
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.605      *prem\_disc\_step4\_r***

Description:	The 4th step additional prem discount rate
Help:	Premium discount as a percentage of premium . Applied during a defined period (see Prem_disc_month).
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium discounts
Variable Type:	Floating Point Number
Default Value:	0

Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	-100
Valid Range To:	500
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.606     *prem\_disc\_step5\_m***

Description:	The 5th step prem discount period
Help:	Month from policy start when Premium discount period ends.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium discounts
Variable Type:	Integer Number
Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1200
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.607     *prem\_disc\_step5\_r***

Description:	The 5th step additional prem discount rate
Help:	Premium discount as a percentage of premium . Applied during a defined period (see Prem_disc_month).
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium discounts
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	-100

Valid Range To:	500
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.608**     ***prem\_disc\_type***

Description:	Premium discount type
Help:	0 = no loading 1 = gorem on yesodi 3 = temp discount on all covers in policy 4 = permanent discount on all covers in policy 5 = temp discount on specific cover 6 = permanent discount on specific cover other = gorem
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium discounts
Variable Type:	Integer Number
Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	480
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.609**     ***prem\_disc\_type\_2***

Description:	Premium discount type
Help:	0 = no loading 1 = gorem on yesodi 3 = temp discount on all covers in policy 4 = permanent discount on all covers in policy 5 = temp discount on specific cover 6 = permanent discount on specific cover other = gorem
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium discounts
Variable Type:	Integer Number
Default Value:	0
Length:	0
Number of Decimals:	0



Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	480
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.610**     ***adjust\_prem\_and\_claims***

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Product Details
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.611**     ***age\_incidence***

Description:	code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Product Details
Variable Type:	Integer Number
Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.612**     ***alloc\_rate***

Description:	Percentage of Premium allocated to units
Help:	Percentage of premium allocated to units. Each element in the array applies for the number of months specified in the alloc_rate_period array. The first element in the array should be entered for period 1. Read in from alloc_rate_tbl in set_common_variables.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Product Details
Variable Type:	Floating Point Array
Default Value:	0 100
Length:	16
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	150
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.613**     ***alloc\_rate\_period***

Description:	Months applying to alloc rate
Help:	Each element in this array relates to a corresponding element in the alloc_rate array. The elements in this array specify the number of months for which the allocation rates in alloc_rate apply.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Product Details
Variable Type:	Floating Point Array
Default Value:	6 12
Length:	16
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	600
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.614 alloc\_rate\_row**

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Product Details
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.615 alloc\_rate\_set\_temp**

Description:	Allocation rate assumption set
Help:	Allocation rate assumption set. If read_from_table = "Y", then this variable will be set in set_common_variables.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Product Details
Variable Type:	Character
Default Value:	default
Length:	7
Number of Decimals:	0
Choice List:	
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.616 alloc\_rate\_stri**

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)

Modified By:	CLAL-INS\ninab
Category:	Product Details
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.617**     ***blue\_white***

Description:	Lookup value code variable wildcard
Help:	
Modified On:	7/14/2024 11:48:51 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Product Details
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.618**     ***bonus***

Description:	persistency bonus percentages array
Help:	Persistency bonus rates by policy month (%). Always read from table bonus_tbl.
Modified On:	6/27/2022 5:17:10 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Product Details
Variable Type:	Floating Point Array
Default Value:	0
Length:	1400
Number of Decimals:	2
Choice List:	

Character Type:	Not Applicable
Valid Range From:	-100
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.619 *bonus\_tbl*

Description:	persistency bonus rates table
Help:	Table with persistency bonus rates by policy month. This is always used (even if read_from_tables = "N") - looked up by surr_charge_set
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Product Details
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	0
Table Format:	Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.620 *claims\_cost\_key\_start*

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/10/2021 10:44:38 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Product Details
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All

Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.621**      *claims\_factor*

Description: Lookup value code variable wildcard  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Product Details  
Variable Type: Character  
Default Value: 0  
Length: 10  
Number of Decimals: 1  
Choice List: 0  
Character Type: Standard  
Valid Range From:  
Valid Range To:  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.622**      *claims\_factor\_occ*

Description: Lookup value code variable wildcard  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Product Details  
Variable Type: Character  
Default Value: 0  
Length: 10  
Number of Decimals: 1  
Choice List: 0  
Character Type: Standard  
Valid Range From:  
Valid Range To:  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.623**      *claims\_series\_year*

Description: Lookup value code variable wildcard  
Help:

Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Product Details
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.624**      ***claimskey\_endage***

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Product Details
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.625**      ***claimskey\_sex***

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Product Details
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0

Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.626 *dd\_prop\_cont*

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Product Details
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.627 *death\_ben\_w*

Description:	death benefit?
Help:	read in from setup_tbl and set in set_categ_variables() Y = death benefit N = other benefit (dd, capital disablement etc.)
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Product Details
Variable Type:	Character
Default Value:	N
Length:	1
Number of Decimals:	0
Choice List:	Y,N
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	



Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.628      *decrem\_rates\_tbl***

Description: Decrem Rates Tbl  
Help:  
Modified On: 7/28/2021 11:55:53 AM (UTC+03:00)  
Modified By: CLAL-INS\joshm  
Category: Product Details  
Variable Type: Character  
Default Value: ltc  
Length: 20  
Number of Decimals: 1  
Choice List: ltc  
Character Type: Standard  
Valid Range From:  
Valid Range To:  
Table Format: Default Row Numbers  
Set Value in Input Manager:  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.629      *dur\_down***

Description: code variable  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Product Details  
Variable Type: Integer Number  
Default Value: 0  
Length: 0  
Number of Decimals: 0  
Choice List:  
Character Type: Not Applicable  
Valid Range From:  
Valid Range To:  
Table Format: Default Row Numbers  
Set Value in Input Manager:  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.630      *dur\_up***

Description: code variable  
Help:

Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Product Details
Variable Type:	Integer Number
Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.631**     ***duration\_phi***

Description:	code variable
Help:	
Modified On:	12/8/2020 3:27:14 PM (UTC+02:00)
Modified By:	CLAL-INS\NinaB
Category:	Product Details
Variable Type:	Integer Number
Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.632**     ***groups\_sol***

Description:	groups solvency
Help:	
Modified On:	3/22/2023 2:02:02 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Product Details
Variable Type:	Character
Default Value:	0
Length:	28
Number of Decimals:	0
Choice List:	0

Character Type:	Standard
Valid Range From:	0
Valid Range To:	600
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.633      *incidencerate\_key***

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Product Details
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.634      *lapse\_tarif\_set***

Description:	Lookup value code variable wildcard
Help:	
Modified On:	6/12/2022 3:58:05 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Product Details
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	0
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.635**     *matan\_perc*

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Product Details
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.636**     *matan\_perc\_temp*

Description:	% of SI paid after x years for MATAN
Help:	Percentage of SI in case of survival at mid term (w) (e.g. at matan benefit)
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Product Details
Variable Type:	Floating Point Number
Default Value:	2
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	-100
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.637**     *phi\_type*

Description:	PHI type(P-Pitzuim, S-WP)
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Product Details

Variable Type:	Character
Default Value:	P
Length:	1
Number of Decimals:	0
Choice List:	
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.638**     *pitzui\_shichrur*

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Product Details
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.639**     *prem\_age*

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Product Details
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	

Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.640**     ***prem\_factor***

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Product Details
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.641**     ***prem\_inc***

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Product Details
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.642**     ***prem\_init\_different***

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Product Details
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.643**     ***prem\_key\_start***

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Product Details
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.644**     ***prem\_lookup***

Description:	Lookup code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Product Details
Variable Type:	Character

Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.645     *prem\_lookup\_freq***

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Product Details
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.646     *prem\_lookup\_freq\_trad***

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/29/2021 12:08:48 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Product Details
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers



Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.647 *prem\_lookup\_trad*

Description:	Lookup code variable wildcard
Help:	
Modified On:	8/16/2021 11:05:44 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Product Details
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.648 *prem\_profil\_type*

Description:	premium type for profil
Help:	defines whether premium for a cover (usually phi) is "mitoch hahafraSHOT" (13-7% expenses are deducted) or "michutz" (and expenses not deducted)
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Product Details
Variable Type:	Character
Default Value:	out
Length:	3
Number of Decimals:	0
Choice List:	in,out,mix
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.649**     ***prem\_rates\_series\_end\_im***

Description:	Prem Rates Im
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Product Details
Variable Type:	Integer Number
Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.650**     ***prem\_series\_year***

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Product Details
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.651**     ***premkey\_endage***

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Product Details
Variable Type:	Character

Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.652     *premkey\_insured***

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Product Details
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.653     *premkey\_occ***

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Product Details
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers

Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

### **5.3.3.1.654**     ***premkey\_sex***

Description: Lookup value code variable wildcard  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Product Details  
Variable Type: Character  
Default Value: 0  
Length: 10  
Number of Decimals: 1  
Choice List: 0  
Character Type: Standard  
Valid Range From:  
Valid Range To:  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

### **5.3.3.1.655**     ***premkey\_smoker***

Description: Lookup value code variable wildcard  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Product Details  
Variable Type: Character  
Default Value: 0  
Length: 10  
Number of Decimals: 1  
Choice List: 0  
Character Type: Standard  
Valid Range From:  
Valid Range To:  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

### **5.3.3.1.656**     ***prod\_assumpt\_key\_tbl***

Description: Product code specific assumptions for prod\_code\_base

Help:	Product code specific assumptions. Used only if lookup_by_prodcode = "Y".
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Product Details
Variable Type:	Character
Default Value:	
Length:	56
Number of Decimals:	0
Choice List:	
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Row Name (string)
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.657     *prod\_assumpt\_rider\_exp\_tbl***

Description:	Product code specific assumptions for prod_code_rider
Help:	Product code specific assumptions. Used only if lookup_by_prodcode = "Y".
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Product Details
Variable Type:	Character
Default Value:	
Length:	56
Number of Decimals:	0
Choice List:	
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Row Name (string)
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.658     *prod\_assumpt\_rider\_lapse\_tbl***

Description:	Product code specific assumptions for lapse for profil riders
Help:	Product code specific assumptions. Used only if lookup_by_prodcode = "Y".
Modified On:	6/15/2022 9:38:26 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Product Details

Variable Type:	Character
Default Value:	0
Length:	56
Number of Decimals:	0
Choice List:	0
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Row Name (string)
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.659**     *prod\_spec\_risk\_code*

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Product Details
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.660**     *prod\_specs\_max*

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Product Details
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	

Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.661**     *prod\_specs\_rider*

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Product Details
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	0
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.662**     *prod\_specs\_rider\_col*

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Product Details
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.663      *prodcold***

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/16/2021 11:39:37 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Product Details
Variable Type:	Character
Default Value:	100
Length:	10
Number of Decimals:	1
Choice List:	100
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.664      *prodcode\_par\_nonpar***

Description:	par_nonpar from DWH prodcode table
Help:	Product code specific assumptions. Used only if lookup_by_prodcode = "Y".
Modified On:	5/8/2022 3:34:53 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Product Details
Variable Type:	Character
Default Value:	0
Length:	56
Number of Decimals:	0
Choice List:	0
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Row Name (string)
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.665      *pup\_ind***

Description:	indicator whether the record is pup
Help:	
Modified On:	6/9/2022 12:05:56 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Product Details



Variable Type:	Integer Number
Default Value:	0
Length:	1
Number of Decimals:	0
Choice List:	0
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.666 *pup\_sv\_charge\_rebate*

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Product Details
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.667 *puv\_tbl*

Description:	paid up value table by entry age + dur
Help:	premium rate set in set_common_variables() per 1000 SA.
Modified On:	7/28/2021 1:23:00 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Product Details
Variable Type:	Character
Default Value:	1_puv_100
Length:	39
Number of Decimals:	0
Choice List:	1_puv_100
Character Type:	Standard
Valid Range From:	0

Valid Range To:	10000
Table Format:	Row Name (numeric)
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.668**      *recovery\_rates\_col*

Description:	Lookup value code variable
Help:	
Modified On:	7/20/2021 4:51:50 PM (UTC+03:00)
Modified By:	CLAL-INSjoshm
Category:	Product Details
Variable Type:	Integer Number
Default Value:	0
Length:	10
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.669**      *recovery\_rates\_row*

Description:	Lookup value code variable
Help:	
Modified On:	7/20/2021 4:52:05 PM (UTC+03:00)
Modified By:	CLAL-INSjoshm
Category:	Product Details
Variable Type:	Integer Number
Default Value:	0
Length:	10
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.670      *recovery\_rates\_tbl***

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Product Details
Variable Type:	Floating Point Number
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.671      *reins\_key\_start***

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Product Details
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.672      *res\_basis***

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/9/2021 8:38:23 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Product Details
Variable Type:	Character

Default Value:	No_Reserve
Length:	10
Number of Decimals:	1
Choice List:	No_Reserve,Perc_Prem,Net_Prem
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.673 *rider\_ind*

Description:	indicator whether the record is a rider
Help:	
Modified On:	6/9/2022 11:17:39 AM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Product Details
Variable Type:	Integer Number
Default Value:	0
Length:	1
Number of Decimals:	0
Choice List:	0
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.674 *rider\_max\_perc*

Description:	Max % of (tagmulim) premium allowed to go to riders
Help:	For new Profil. Max % of (tagmulim) premium, after allocation, allowed to go to riders' charges.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Product Details
Variable Type:	Floating Point Number
Default Value:	35
Length:	1
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable

Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.675 *rider\_tarif\_tbl*

Description:	Table to map Profil Riders Tarif code to product code
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Product Details
Variable Type:	Character
Default Value:	
Length:	45
Number of Decimals:	0
Choice List:	
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Row Name (string)
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.676 *risk\_type\_w*

Description:	Risk Type codes for Profil Riders
Help:	Code of risk-type for Profil Riders. Used to lookup relevant decrement rates, and to carry out certain formulae differently. 1 = dth = regular death 2 = adb 3 = tpd 4 = dd 5 = phi 6 = health
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Product Details
Variable Type:	Integer Array
Default Value:	0
Length:	25
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable

Valid Range From:	0
Valid Range To:	10
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### **5.3.3.1.677      *savings\_pol***

Description:	Identify if prod_code_base is of savings policy
Help:	
Modified On:	1/26/2023 11:25:16 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Product Details
Variable Type:	Character
Default Value:	Y
Length:	1
Number of Decimals:	1
Choice List:	N,Y
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### **5.3.3.1.678      *savings\_pol\_prod\_code***

Description:	Identify if prod_code is of savings policy
Help:	
Modified On:	3/23/2023 11:06:49 AM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Product Details
Variable Type:	Character
Default Value:	Y
Length:	1
Number of Decimals:	1
Choice List:	N,Y
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.679      *secondary\_prop\_continue***

Description:	% of policies changing state
Help:	For "achrayut le'chaim" product. The percentage of policies that continue after a claim.  If set_by_procode = "Y" then this is looked up from the product specs table.  For LTC - this is the proportion of lapses becoming paid up policies - but It is automatically set to 100 in set_other_variables.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Product Details
Variable Type:	Floating Point Number
Default Value:	50
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.680      *si\_unit\_w***

Description:	si_unit_w
Help:	Number of units sum-insured (eg 1000) to which charge or premium rates relate. Used for Profil riders (elements 0 to 24) and/or Meitav risk rider (element 25).
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Product Details
Variable Type:	Floating Point Array
Default Value:	1
Length:	26
Number of Decimals:	3
Choice List:	
Character Type:	Not Applicable
Valid Range From:	1
Valid Range To:	1000
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared

Category Order: 0

#### **5.3.3.1.681     *stri***

Description: Lookup value code variable  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Product Details  
Variable Type: Character  
Default Value: 0  
Length: 10  
Number of Decimals: 1  
Choice List: 0  
Character Type: Standard  
Valid Range From:  
Valid Range To:  
Table Format: Default Row Numbers  
Set Value in Input Manager:  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.682     *sum\_inc***

Description: Lookup value code variable wildcard  
Help:  
Modified On: 5/16/2023 10:07:12 AM (UTC+03:00)  
Modified By: CLAL-INS\ahuvaa  
Category: Product Details  
Variable Type: Character  
Default Value: 0  
Length: 10  
Number of Decimals: 1  
Choice List: 0  
Character Type: Standard  
Valid Range From:  
Valid Range To:  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.683     *suminisba\_tbl***

Description: Lookup value code variable  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)



Modified By:	CLAL-INS\ninab
Category:	Product Details
Variable Type:	Floating Point Number
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.684**     ***suminsbas\_col***

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Product Details
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.685**     ***suminsbas\_row***

Description:	Lokup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Product Details
Variable Type:	Integer Number
Default Value:	0
Length:	10
Number of Decimals:	0
Choice List:	0
Character Type:	Not Applicable

Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.686 *surr\_chg\_tbl*

Description:	Surrender charges table
Help:	Contains penalty rates on surrender.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Product Details
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	0
Table Format:	Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.687 *sv\_tbl*

Description:	surrender value table by entry age + dur
Help:	premium rate set in set_common_variables() per 1000 SA.
Modified On:	7/28/2021 12:33:13 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Product Details
Variable Type:	Character
Default Value:	1_100
Length:	39
Number of Decimals:	0
Choice List:	1_100
Character Type:	Standard
Valid Range From:	0
Valid Range To:	10000
Table Format:	Row Name (numeric)
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.688      *tarif\_spec\_row\_key***

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Product Details
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.689      *waiting\_period\_modeled***

Description:	Lookup value code variable wildcard
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Product Details
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.690      *zillmer\_pr\_tbl***

Description:	Zillmer rates table (% of premium)
Help:	Table of zillmer premium rates by policy type and dac purpose (book or taxe). Set for the 10 first policy years ( = 0 after).
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Product Details

Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	0
Table Format:	Row Name (numeric)
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.691**     ***prem\_term\_original***

Description:	Prem Term Original
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Profitability Measures
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.692**     ***comm\_prof\_re***

Description:	Reinsurance profit commission (%)
Help:	Reinsurance profit commission expressed as a % of reins. profit. if"read_from_table"=Y : read from comm_ren-tbl in startup
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Reinsurance
Variable Type:	Floating Point Number
Default Value:	2
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable

Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.693 *comm\_ren\_re*

Description:	Reinsurance Renewal commission (%)
Help:	Reinsurance Renewal commission expressed as a % of reins. premium income. read from table in set_common_variable
Modified On:	8/22/2021 3:04:51 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Reinsurance
Variable Type:	Floating Point Array
Default Value:	50
Length:	150
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.694 *exp\_re\_nom*

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Reinsurance
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.695      *expense\_re\_nom\_temp***

Description:	Nominal reins. exp. as % of premium
Help:	Nominal reinsurance expenses as a % of reinsurance premium. Loaded from comm renewal table in set comm variables
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Reinsurance
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1000
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.696      *interest\_rein***

Description:	int rate paid to reinsurer on reserve
Help:	Annual investment income rate . Set in set_exp_variables from int_rates_tbl. if "read_from_tables" =Y
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Reinsurance
Variable Type:	Floating Point Number
Default Value:	1
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.697      *prem\_per\_unit\_si\_re***

Description:	Extra premium per unit SI - reinsurance
--------------	---

Help:	Additional reinsurance premium per unit sum insured. Product addition and not policy specific, i.e. if RFT its value overridden from table.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Reinsurance
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	8
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1000000
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.698 ***prem\_rates\_re***

Description:	rein premium rates
Help:	
Modified On:	7/22/2021 9:48:13 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Reinsurance
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.699 ***prem\_re\_bw***

Description:	reinsurance premium rates row lookup - bw
Help:	This is the index value to lookup claims cost rates in the claim_cost.tbl. It is looked up initially from the tarif_spec.tbl but may be modified by policy data from the file.
Modified On:	7/15/2024 1:03:11 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt

Category:	Reinsurance
Variable Type:	Character
Default Value:	0
Length:	22
Number of Decimals:	0
Choice List:	
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.700     *prem\_re\_endage***

Description:	reinsurance premium rates row lookup - end age
Help:	This is the index value to lookup claims cost rates in the claim_cost.tbl. It is looked up initially from the tarif_spec.tbl but may be modified by policy data from the file.
Modified On:	6/19/2024 12:26:40 PM (UTC+03:00)
Modified By:	CLAL-INS\Arikt
Category:	Reinsurance
Variable Type:	Character
Default Value:	0
Length:	22
Number of Decimals:	0
Choice List:	
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.701     *prem\_re\_occ***

Description:	reinsurance premium rates row lookup - occ
Help:	This is the index value to lookup claims cost rates in the claim_cost.tbl. It is looked up initially from the tarif_spec.tbl but may be modified by policy data from the file.
Modified On:	6/19/2024 12:26:17 PM (UTC+03:00)
Modified By:	CLAL-INS\Arikt
Category:	Reinsurance
Variable Type:	Character



Default Value:	0
Length:	22
Number of Decimals:	0
Choice List:	
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.702     *prem\_re\_row\_key***

Description:	reinsurance premium rates row lookup code
Help:	This is the index value to lookup claims cost rates in the claim_cost.tbl. It is looked up initially from the tarif_spec.tbl but may be modified by policy data from the file.
Modified On:	7/22/2021 9:50:42 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Reinsurance
Variable Type:	Character
Default Value:	0
Length:	22
Number of Decimals:	0
Choice List:	
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.703     *prem\_re\_sex***

Description:	reinsurance premium rates row lookup - sex
Help:	This is the index value to lookup claims cost rates in the claim_cost.tbl. It is looked up initially from the tarif_spec.tbl but may be modified by policy data from the file.
Modified On:	6/19/2024 2:21:33 PM (UTC+03:00)
Modified By:	CLAL-INS\Arikt
Category:	Reinsurance
Variable Type:	Character
Default Value:	default
Length:	22
Number of Decimals:	0

Choice List:	default
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.704**      ***prem\_re\_wp***

Description:	reinsurance premium rates row lookup - wp
Help:	This is the index value to lookup claims cost rates in the claim_cost.tbl. It is looked up initially from the tarif_spec.tbl but may be modified by policy data from the file.
Modified On:	7/15/2024 1:03:38 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Reinsurance
Variable Type:	Character
Default Value:	0
Length:	22
Number of Decimals:	0
Choice List:	
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.705**      ***re\_clm\_rein\_pc***

Description:	percent of retained claim reinsured
Help:	percentage of retained claim reinsured.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Reinsurance
Variable Type:	Floating Point Number
Default Value:	100
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers

Set Value in Input Manager:  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.706      *re\_clm\_rein\_pc\_rider***

Description: percentage reinsured for each Profil rider  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Reinsurance  
Variable Type: Floating Point Array  
Default Value: 0 0  
Length: 25  
Number of Decimals: 2  
Choice List:  
Character Type: Not Applicable  
Valid Range From: 0  
Valid Range To: 100  
Table Format: Default Row Numbers  
Set Value in Input Manager:  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.707      *re\_clm\_ret\_fix***

Description: claim amount retained (monthly)  
Help: Fixed amount of each monthly claim retained (not reinsured). For non-proportional reinsurance.  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Reinsurance  
Variable Type: Floating Point Number  
Default Value: 100000  
Length: 0  
Number of Decimals: 0  
Choice List:  
Character Type: Not Applicable  
Valid Range From: 0  
Valid Range To: 100000000  
Table Format: Default Row Numbers  
Set Value in Input Manager:  
Variable Sharing: Not Shared  
Category Order: 0

**5.3.3.1.708 re\_cost\_pc\_rider**

Description:	percentage cost of reinsured for each Profil rider
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Reinsurance
Variable Type:	Floating Point Array
Default Value:	0 0
Length:	25
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.709 re\_cost\_perc**

Description:	Percentage cost of reinsurance
Help:	Percentage of cost of reinsurance. This is cost above the claims ceded. For example, a value of 10 implies that the reinsurance premium is 10% higher than the claims paid by reinsurance. Used for "simple" reinsurance method.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Reinsurance
Variable Type:	Floating Point Number
Default Value:	0
Length:	1
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.710 re\_ratio\_w**

Description:	Ratio reinsured
--------------	-----------------

Help:	Proportion reinsured for surplus reinsurance : calculated in startup
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Reinsurance
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	3
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.711 *re\_type*

Description:	reinsurance type
Help:	OT: Orginial Term Left side = quota share reinsurance type Right side= Surplus reinsurance type Eg. OT_YRT: the quota share reinsurance premium is on original terms up to the retention, above YRT premium.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Reinsurance
Variable Type:	Character
Default Value:	YRT
Length:	9
Number of Decimals:	0
Choice List:	OT,YRT,NONE,OT_OT,OT_YRT,OT_NONE,Y RT_OT,YRT_YRT,YRT_NONE,NONE_OT,NO NE_YRT,NONE_NONE,simple,OT_OT
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.712 *rein\_key\_temp*

Description:	Lookup value code variable
Help:	

Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Reinsurance
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.713 *rein\_set*

Description:	Reinsurance assumptions
Help:	Reinsurance assumptions. If read_from_table = "Y", then this variable will be set in set_common_variables.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Reinsurance
Variable Type:	Character
Default Value:	default
Length:	7
Number of Decimals:	0
Choice List:	
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.714 *rein\_set\_input*

Description:	Reinsurance assumptions
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Reinsurance
Variable Type:	Character
Default Value:	default
Length:	7

Number of Decimals:	1
Choice List:	default
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.715**     *reinsur\_comm*

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\inab
Category:	Reinsurance
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.716**     *reinsur\_comm\_key*

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\inab
Category:	Reinsurance
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared

Category Order: 0

#### **5.3.3.1.717      *reinsur\_kod\_tavla***

Description: Lookup value code variable  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Reinsurance  
Variable Type: Character  
Default Value: 0  
Length: 10  
Number of Decimals: 1  
Choice List: 0  
Character Type: Standard  
Valid Range From:  
Valid Range To:  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.718      *reinsurance***

Description: calculate reinsurance ?  
Help: Y = calculate reinsurance  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Reinsurance  
Variable Type: Character  
Default Value: N  
Length: 1  
Number of Decimals: 0  
Choice List: Y,N  
Character Type: Standard  
Valid Range From: 0  
Valid Range To: 0  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.719      *retention\_perc***

Description: Retention Ratio  
Help: Proportion reinsured for surplus reinsurance :  
calculated in startup  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)



Modified By:	CLAL-INS\ninab
Category:	Reinsurance
Variable Type:	Floating Point Number
Default Value:	1
Length:	0
Number of Decimals:	3
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.720 *rider\_tarif\_row\_key*

Description:	Lookup value code variable
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Reinsurance
Variable Type:	Character
Default Value:	0
Length:	10
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.721 *asset\_shock*

Description:	Asset Shock (to replace investment income)
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Reserve
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	4
Choice List:	
Character Type:	Not Applicable

Valid Range From:	-100
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.722**     *err\_sar\_perc*

Description:	ERR as % of sum at risk
Help:	extra-ordinary reserve
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Reserve
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	0.3
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.723**     *err\_spread\_period*

Description:	years to build up err
Help:	number of years over which err is built up
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Reserve
Variable Type:	Floating Point Number
Default Value:	8
Length:	1
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	30
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.724      *res\_adj\_factor***

Description:	Reserve Adjustment factor
Help:	
Modified On:	8/5/2024 3:46:21 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Reserve
Variable Type:	Floating Point Number
Default Value:	0
Length:	116
Number of Decimals:	4
Choice List:	
Character Type:	Not Applicable
Valid Range From:	-1
Valid Range To:	1
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.725      *res\_anndef\_lapse***

Description:	Annual lapse rate for Annuity deficiency reserves calc.
Help:	variable linked with the kitzba reserve field from the inforce file
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Reserve
Variable Type:	Floating Point Number
Default Value:	0.02
Length:	0
Number of Decimals:	4
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100000000
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.726      *res\_anndef\_lapse\_par***

Description:	Annual lapse rate for Annuity deficiency reserves calc. - participating
Help:	variable linked with the kitzba reserve field from the inforce file
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)

Modified By:	CLAL-INS\ninab
Category:	Reserve
Variable Type:	Floating Point Number
Default Value:	0.02
Length:	0
Number of Decimals:	4
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100000000
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.727      *res\_kitzba***

Description:	reserve from inforce for kitzba
Help:	variable linked with the kitzba reserve field from the inforce file
Modified On:	8/5/2024 4:13:07 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Reserve
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100000000
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.728      *res\_kitzba\_input***

Description:	reserve from inforce for kitzba - input
Help:	variable linked with the kitzba reserve field from the inforce file
Modified On:	8/5/2024 4:10:16 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Reserve
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	2

Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100000000
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.729      *res\_perc\_prem***

Description:	Percentage of premium for reserve calculation
Help:	reserve percentage premium : % of annual premium in force. Used in reserve_basic for YRT PREMIUM (prem_lookup_freq =1.) (as a percentage)
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Reserve
Variable Type:	Floating Point Array
Default Value:	0 0 0
Length:	121
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.730      *reserve\_factors\_tbl***

Description:	reserve factors table
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Reserve
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	0

Table Format:	Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.731     *resinforce***

Description:	reserve from inforce
Help:	variable linked with the reserve field from the inforce file
Modified On:	8/5/2024 4:11:49 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Reserve
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100000000
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.732     *resinforce\_input***

Description:	reserve from inforce - input
Help:	variable linked with the reserve field from the inforce file
Modified On:	8/5/2024 4:10:57 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Reserve
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	100000000
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.733      zeroise\_res**

Description:	Zeroise negative reserves (Y/N)?
Help:	Y = Individual negative reserves are set to zero
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Reserve
Variable Type:	Character
Default Value:	Y
Length:	1
Number of Decimals:	0
Choice List:	Y,N
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.734      zillmer\_adj\_factor**

Description:	Adjustment factor for Zillmer (to scale up to actuals)
Help:	The percentage applied to the DAC tax value taken from the data file to adjust it according to the actual DAC held.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Reserve
Variable Type:	Floating Point Number
Default Value:	100
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1000
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.735      zillmer\_si\_book**

Description:	Zillmer rate (%) for BOOKS
Help:	Used in separate column 'zillmer_book' only.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab

Category:	Reserve
Variable Type:	Floating Point Number
Default Value:	3
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	10
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.736 ***zillmer\_si\_tax***

Description:	Zillmer rate (%) for TAX
Help:	Used in separate column 'zillmer_tax' only.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Reserve
Variable Type:	Floating Point Number
Default Value:	3
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	10
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.737 ***chetz\_be\_ind***

Description:	Indicator for calculating Investment income chetz based on BE reserves
Help:	
Modified On:	6/15/2023 2:08:57 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Setup
Variable Type:	Character
Default Value:	N
Length:	1
Number of Decimals:	0
Choice List:	Y,N
Character Type:	Standard



Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.738 *chetz\_be\_ind\_yrs*

Description:	Calendar year to start calculating chetz based on IFRS method
Help:	Valuation occurs at end of valn_month in valn_year
Modified On:	6/21/2023 1:43:12 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Setup
Variable Type:	Integer Number
Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	10000000
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.739 *done\_startup\_w*

Description:	Has startup been done ?
Help:	Has startup run for the first time? Boolean - default = false . At the end of startup this variable is set to 'true'. This variable is used in the context of NB modelling (layering), where a formula must be executed only once and not looped.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Setup
Variable Type:	Character
Default Value:	FALSE
Length:	5
Number of Decimals:	0
Choice List:	False,True
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0

Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.740 *dump\_vars*

Description:	Output all variables to logfile?
Help:	If Y, the program will output all variables to the log stream after the startup has been executed in each model. NB Should only be set to Y for testing purposes, as will create HUGE amounts of output in the log file when run with a large policy file.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Setup
Variable Type:	Character
Default Value:	N
Length:	1
Number of Decimals:	0
Choice List:	Y,N
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.741 *esg\_run*

Description:	Indicator for ESG run
Help:	
Modified On:	5/15/2023 2:05:35 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Setup
Variable Type:	Character
Default Value:	N
Length:	1
Number of Decimals:	1
Choice List:	N,Y
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.742 gross\_up\_historic**

Description:	Gross up historic survivorship?
Help:	Internal logic variable set in startup. Y = In negative periods gross up survivorship.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Setup
Variable Type:	Character
Default Value:	N
Length:	1
Number of Decimals:	0
Choice List:	Y,N
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.743 madad\_current**

Description:	madad at valuation date
Help:	This is the madad at the valuation date.  For Life and Health products: It is applied to the policy fee to adjust it to the current madad. (see madad_base_pol_fee, in Product Specifications screen )  For Health products only: It is applied to the risk premium rates and claim cost tables, to adjust them to the current madad.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Setup
Variable Type:	Floating Point Number
Default Value:	1
Length:	1
Number of Decimals:	4
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0.0001
Valid Range To:	99999
Table Format:	Default Row Numbers

Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.744**     *pol\_type\_annuity\_tu*

Description: Policy type data source - Annuity TU  
Help:  
Modified On: 12/26/2022 2:23:31 PM (UTC+02:00)  
Modified By: CLAL-INS\ahuvaa  
Category: Setup  
Variable Type: Character  
Default Value: Current  
Length: 8  
Number of Decimals: 0  
Choice List: Original,Current  
Character Type: Standard  
Valid Range From: 0  
Valid Range To: 0  
Table Format: Default Row Numbers  
Set Value in Input Manager:  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.745**     *pol\_type\_annuity\_tu\_switch*

Description: Policy type data switch - Annuity TU  
Help:  
Modified On: 12/26/2022 2:51:49 PM (UTC+02:00)  
Modified By: CLAL-INS\ahuvaa  
Category: Setup  
Variable Type: Character  
Default Value: Current  
Length: 8  
Number of Decimals: 0  
Choice List: Original,Current  
Character Type: Standard  
Valid Range From: 0  
Valid Range To: 0  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.746**     *pol\_type\_comm\_hekef*

Description: Policy type data source - Commission Hekef

Help:	DAC amortisation type
Modified On:	12/26/2022 2:23:35 PM (UTC+02:00)
Modified By:	CLAL-INS\lahuvaa
Category:	Setup
Variable Type:	Character
Default Value:	Current
Length:	8
Number of Decimals:	0
Choice List:	Original,Current
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.747     *pol\_type\_comm\_hekef\_switch***

Description:	Policy type data switch - Commission Hekef
Help:	DAC amortisation type
Modified On:	12/26/2022 2:23:03 PM (UTC+02:00)
Modified By:	CLAL-INS\lahuvaa
Category:	Setup
Variable Type:	Character
Default Value:	Current
Length:	8
Number of Decimals:	0
Choice List:	Original,Current
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.748     *pol\_type\_expenses***

Description:	Policy type data source - Expenses
Help:	DAC amortisation type
Modified On:	12/26/2022 2:23:38 PM (UTC+02:00)
Modified By:	CLAL-INS\lahuvaa
Category:	Setup
Variable Type:	Character
Default Value:	Current
Length:	8
Number of Decimals:	0

Choice List:	Original,Current
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.749      *pol\_type\_expenses\_switch***

Description:	Policy type data switch - Expenses
Help:	DAC amortisation type
Modified On:	12/26/2022 2:23:07 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Setup
Variable Type:	Character
Default Value:	Current
Length:	8
Number of Decimals:	0
Choice List:	Original,Current
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.750      *pol\_type\_lapse***

Description:	Policy type data source - Lapse rates
Help:	DAC amortisation type
Modified On:	12/26/2022 2:23:41 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Setup
Variable Type:	Character
Default Value:	Current
Length:	8
Number of Decimals:	0
Choice List:	Original,Current
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.751     *pol\_type\_lapse\_rider***

Description:	Policy type data source - Lapse savings rider rates
Help:	DAC amortisation type
Modified On:	1/19/2023 8:43:07 AM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Setup
Variable Type:	Character
Default Value:	Current
Length:	8
Number of Decimals:	0
Choice List:	Original,Current
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.752     *pol\_type\_lapse\_rider\_switch***

Description:	Policy type data switch - Lapse savings riders rates
Help:	DAC amortisation type
Modified On:	1/19/2023 8:42:51 AM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Setup
Variable Type:	Character
Default Value:	Current
Length:	8
Number of Decimals:	0
Choice List:	Original,Current
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.753     *pol\_type\_lapse\_switch***

Description:	Policy type data switch - Lapse rates
Help:	DAC amortisation type
Modified On:	12/26/2022 2:23:12 PM (UTC+02:00)

Modified By:	CLAL-INS\ahuvaa
Category:	Setup
Variable Type:	Character
Default Value:	Current
Length:	8
Number of Decimals:	0
Choice List:	Original,Current
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.754 *pol\_type\_phi\_incidence*

Description:	Policy type data source - PHI Incidence
Help:	DAC amortisation type
Modified On:	12/26/2022 2:23:45 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Setup
Variable Type:	Character
Default Value:	Current
Length:	8
Number of Decimals:	0
Choice List:	Original,Current
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.755 *pol\_type\_phi\_incidence\_switch*

Description:	Policy type data switch - PHI Incidence
Help:	DAC amortisation type
Modified On:	12/26/2022 2:23:15 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Setup
Variable Type:	Character
Default Value:	Current
Length:	8
Number of Decimals:	0
Choice List:	Original,Current
Character Type:	Standard



Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### **5.3.3.1.756      *pol\_type\_recovery\_rates***

Description:	Policy type data source - recovery rates
Help:	DAC amortisation type
Modified On:	3/13/2023 3:46:37 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Setup
Variable Type:	Character
Default Value:	Current
Length:	8
Number of Decimals:	0
Choice List:	Original,Current
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### **5.3.3.1.757      *pol\_type\_recovery\_rates\_switch***

Description:	Policy type data switch - Recovery rates
Help:	DAC amortisation type
Modified On:	3/13/2023 3:45:22 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Setup
Variable Type:	Character
Default Value:	Current
Length:	8
Number of Decimals:	0
Choice List:	Original,Current
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.758**     ***pol\_type\_sal\_inc***

Description:	Policy type data source - salary increase
Help:	DAC amortisation type
Modified On:	12/26/2022 2:23:48 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Setup
Variable Type:	Character
Default Value:	Current
Length:	8
Number of Decimals:	0
Choice List:	Original,Current
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.759**     ***pol\_type\_sal\_inc\_switch***

Description:	Policy type data switch - salary increase
Help:	DAC amortisation type
Modified On:	12/26/2022 2:23:19 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Setup
Variable Type:	Character
Default Value:	Current
Length:	8
Number of Decimals:	0
Choice List:	Original,Current
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.760**     ***projection\_type***

Description:	Projection type
--------------	-----------------

Help:	Purpose of the projection run: Valn = perform a valuation for an in force policy New_Bus = used to project future new business layers Pricing = project one new business policy Rollup=same as valn but does not reset valuation date account balances
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Setup
Variable Type:	Character
Default Value:	Pricing
Length:	7
Number of Decimals:	0
Choice List:	Valn,New_Bus,Pricing,Rollup
Character Type:	Standard
Valid Range From:	0
Valid Range To:	0
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.761      *projection\_type\_int***

Description:	Projection Type for Interest rates
Help:	
Modified On:	9/12/2019 11:21:01 AM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Category:	Setup
Variable Type:	Character
Default Value:	Valn
Length:	15
Number of Decimals:	1
Choice List:	Valn,Rollup
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.762      *ra\_fact\_dis\_incid\_gross***

Description:	Risk Adjustment Factor Disability Incidence Scenario (gross)
Help:	
Modified On:	3/17/2024 9:44:37 PM (UTC+02:00)

Modified By:	CLAL-INS\Arikt
Category:	Setup
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	8
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.763**     ***ra\_fact\_dis\_incid\_reins***

Description:	Risk Adjustment Factor Disability Scenario (reins) - incidence
Help:	
Modified On:	3/17/2024 3:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Setup
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	8
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.764**     ***ra\_fact\_dis\_termi\_gross***

Description:	Risk Adjustment Factor Disability Termination Scenario (gross)
Help:	
Modified On:	3/17/2024 10:05:00 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Setup
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	8

Choice List:  
Character Type: Not Applicable  
Valid Range From:  
Valid Range To:  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.765      *ra\_fact\_dis\_termi\_reins***

Description: Risk Adjustment Factor Disability Scenario (reins) - termination  
Help:  
Modified On: 3/17/2024 3:44:28 PM (UTC+02:00)  
Modified By: CLAL-INS\Arikt  
Category: Setup  
Variable Type: Floating Point Number  
Default Value: 0  
Length: 0  
Number of Decimals: 8  
Choice List:  
Character Type: Not Applicable  
Valid Range From:  
Valid Range To:  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.766      *ra\_fact\_exp\_gross***

Description: Risk Adjustment Factor Expenses Scenario (gross)  
Help:  
Modified On: 3/22/2023 1:47:39 PM (UTC+02:00)  
Modified By: CLAL-INS\ahuvaa  
Category: Setup  
Variable Type: Floating Point Number  
Default Value: 0  
Length: 0  
Number of Decimals: 8  
Choice List:  
Character Type: Not Applicable  
Valid Range From:  
Valid Range To:  
Table Format: Default Row Numbers  
Set Value in Input Manager: All

Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.767     *ra\_fact\_exp\_reins***

Description: Risk Adjustment Factor Expenses Scenario (Reins)  
Help:  
Modified On: 3/17/2024 3:09:06 PM (UTC+02:00)  
Modified By: CLAL-INS\Arikt  
Category: Setup  
Variable Type: Floating Point Number  
Default Value: 0  
Length: 0  
Number of Decimals: 8  
Choice List:  
Character Type: Not Applicable  
Valid Range From:  
Valid Range To:  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.768     *ra\_fact\_lapse\_gross***

Description: Risk Adjustment Factor lapse Scenario (gross)  
Help:  
Modified On: 3/22/2023 1:46:59 PM (UTC+02:00)  
Modified By: CLAL-INS\ahuvaa  
Category: Setup  
Variable Type: Floating Point Number  
Default Value: 0  
Length: 0  
Number of Decimals: 8  
Choice List:  
Character Type: Not Applicable  
Valid Range From:  
Valid Range To:  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.769     *ra\_fact\_lapse\_reins***

Description: Risk Adjustment Factor lapse Scenario (reins)  
Help:

Modified On:	3/17/2024 3:09:41 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Setup
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	8
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.770      *ra\_fact\_long\_gross***

Description:	Risk Adjustment Factor Longevity Scenario (gross)
Help:	
Modified On:	3/22/2023 1:48:11 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Setup
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	8
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.771      *ra\_fact\_long\_reins***

Description:	Risk Adjustment Factor Longevity Scenario (reins)
Help:	
Modified On:	3/17/2024 3:13:26 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Setup
Variable Type:	Floating Point Number
Default Value:	0
Length:	0

Number of Decimals:	8
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.772     *ra\_fact\_mort\_gross***

Description:	Risk Adjustment Factor Mortality Scenario (gross)
Help:	
Modified On:	3/22/2023 1:48:28 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Setup
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	8
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.3.1.773     *ra\_fact\_mort\_reins***

Description:	Risk Adjustment Factor Mortality Scenario (reins)
Help:	
Modified On:	3/17/2024 3:11:49 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Setup
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	8
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers



Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.774**     ***ra\_fact\_tu\_gross***

Description: Risk Adjustment Factor take-up Scenario (gross)  
Help:  
Modified On: 3/22/2023 1:50:22 PM (UTC+02:00)  
Modified By: CLAL-INS\ahuvaa  
Category: Setup  
Variable Type: Floating Point Number  
Default Value: 0  
Length: 0  
Number of Decimals: 8  
Choice List:  
Character Type: Not Applicable  
Valid Range From:  
Valid Range To:  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.3.1.775**     ***ra\_fact\_tu\_reins***

Description: Risk Adjustment Factor take-up Scenario (reins)  
Help:  
Modified On: 3/17/2024 3:14:33 PM (UTC+02:00)  
Modified By: CLAL-INS\Arikt  
Category: Setup  
Variable Type: Floating Point Number  
Default Value: 0  
Length: 0  
Number of Decimals: 8  
Choice List:  
Character Type: Not Applicable  
Valid Range From:  
Valid Range To:  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

**5.3.3.1.776**     ***rollup\_period***

Description:	Period over which rollup is applied
Help:	
Modified On:	9/12/2019 10:57:05 AM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Category:	Setup
Variable Type:	Integer Number
Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	12
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.777**     ***serv\_units\_dur***

Description:	Lookup value code variable
Help:	
Modified On:	6/11/2023 10:38:48 AM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Setup
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.3.1.778**     ***start\_int\_proj\_after\_rollup***

Description:	Start Int Proj After Rollup
Help:	
Modified On:	9/12/2019 4:04:35 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Category:	Setup
Variable Type:	Character

Default Value:	N
Length:	1
Number of Decimals:	1
Choice List:	N,Y
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.779 *valn\_month*

Description:	Valuation month
Help:	Valuation occurs at end of valn_month in valn_year
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Setup
Variable Type:	Integer Number
Default Value:	12
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	1
Valid Range To:	12
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.1.780 *valn\_year*

Description:	Valuation year
Help:	Valuation occurs at end of valn_month in valn_year
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Setup
Variable Type:	Integer Number
Default Value:	2001
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	1980

Valid Range To:	2100
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.3.2 Columns

#### 5.3.3.2.1 *blank\_test*

Description:	Blank Test
Help:	
Modified On:	11/3/2020 2:14:16 PM (UTC+02:00)
Modified By:	CLAL-INS\NinaB
Category:	
Column Header:	blank_test
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### 5.3.3.2.2 *cashflow\_b\_bef\_ret*

Description:	Cashflow B before retirement
Help:	
Modified On:	8/15/2021 3:07:51 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	
Column Header:	cashflow_b_bef_ret
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.3 claims\_lrc\_q1\_pv**

Description:	PV of disability claims in Q1
Help:	
Modified On:	2/12/2024 4:05:07 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	
Column Header:	claims_lrc_q1_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.4 claims\_lrc\_q2\_pv**

Description:	PV of disability claims in Q2
Help:	
Modified On:	2/12/2024 4:05:12 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	
Column Header:	claims_lrc_q2_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.5 claims\_lrc\_q3\_pv**

Description:	PV of disability claims in Q3
Help:	
Modified On:	2/12/2024 4:05:19 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	
Column Header:	claims_lrc_q3_pv
Combine Groups By:	Sum Both
Combine Periods:	Last

Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.6 *claims\_lrc\_q4\_pv***

Description:	PV of disability claims in Q4
Help:	
Modified On:	2/12/2024 4:05:26 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	
Column Header:	claims_lrc_q4_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.7 *claims\_re\_lrc\_q1\_pv***

Description:	PV of disability claims reins in Q1
Help:	
Modified On:	9/10/2024 4:28:26 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	
Column Header:	claims_re_lrc_q1_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.8 claims\_re\_lrc\_q2\_pv**

Description:	PV of disability claims reins in Q2
Help:	
Modified On:	9/10/2024 4:31:32 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	
Column Header:	claims_re_lrc_q2_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.9 claims\_re\_lrc\_q3\_pv**

Description:	PV of disability claims reins in Q3
Help:	
Modified On:	9/10/2024 4:32:03 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	
Column Header:	claims_re_lrc_q3_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.10 claims\_re\_lrc\_q4\_pv**

Description:	PV of disability claims reins in Q4
Help:	
Modified On:	9/10/2024 4:32:27 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	
Column Header:	claims_re_lrc_q4_pv
Combine Groups By:	Sum Both
Combine Periods:	Last

Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.11**      *claims\_re\_lrc\_yr2plus\_pv*

Description:	PV of disability claims reins in Y2 onwards
Help:	
Modified On:	9/10/2024 4:33:10 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	
Column Header:	claims_re_lrc_yr2plus_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.12**      *expense\_claims\_lrc\_q1\_pv*

Description:	PV of expense claims in Q1
Help:	
Modified On:	11/13/2024 11:01:32 PM (UTC+02:00)
Modified By:	CLAL-INS\arikt
Category:	
Column Header:	expense_claims_lrc_q1_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False



**5.3.3.2.13**      ***expense\_claims\_lrc\_q2\_pv***

Description:	PV of expense claims in Q2
Help:	
Modified On:	11/13/2024 11:02:35 PM (UTC+02:00)
Modified By:	CLAL-INS\arikt
Category:	
Column Header:	expense_claims_lrc_q2_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.14**      ***expense\_claims\_lrc\_q3\_pv***

Description:	PV of expense claims in Q3
Help:	
Modified On:	11/13/2024 11:03:14 PM (UTC+02:00)
Modified By:	CLAL-INS\arikt
Category:	
Column Header:	expense_claims_lrc_q3_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.15**      ***expense\_claims\_lrc\_q4\_pv***

Description:	PV of expense claims in Q4
Help:	
Modified On:	11/13/2024 11:03:46 PM (UTC+02:00)
Modified By:	CLAL-INS\arikt
Category:	
Column Header:	expense_claims_lrc_q4_pv
Combine Groups By:	Sum Both
Combine Periods:	Last

Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.16**      ***expense\_claims\_lrc\_yr2plus\_pv***

Description:	PV of expense claims Year 2 onwards
Help:	
Modified On:	11/14/2024 2:28:09 PM (UTC+02:00)
Modified By:	CLAL-INS\arikt
Category:	
Column Header:	expense_claims_lrc_yr2plus_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.17**      ***initialise***

Description:	Initialise
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\arikt
Category:	
Column Header:	initialise
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.18**      *int\_units\_piz\_active*

Description:	Interest earned on active piz units
Help:	
Modified On:	2/26/2025 12:17:07 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	
Column Header:	int_units_piz_active
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.19**      *int\_units\_piz\_pup*

Description:	Interest earned on paid-up piz units
Help:	
Modified On:	2/26/2025 12:31:42 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	
Column Header:	int_units_piz_pup
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.20**      *outgo\_b\_before\_ret*

Description:	Outgo B before retirement
Help:	
Modified On:	12/8/2021 3:16:26 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	
Column Header:	outgo_b_before_ret
Combine Groups By:	Sum Both
Combine Periods:	Sum

Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.21      *res\_ann\_deficiency***

Description:	Res Ann Deficiency
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	
Column Header:	res_ann_def
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.22      *retirement\_age\_lookup***

Description:	Retirement Age Lookup
Help:	
Modified On:	8/30/2021 8:47:17 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	
Column Header:	retirement_age_lookup
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.23      *retirement\_prop***

Description:	proportion retiring
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	
Column Header:	retirement_prop
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.24      *rider\_perc\_allowed***

Description:	percentage of risk riders allowed (due to limit)
Help:	
Modified On:	1/11/2023 6:58:10 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	
Column Header:	rider_perc_allowed
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.25      *riskadj\_gross\_rel\_q1\_pv***

Description:	PV of gross risk adjustment pv Q1
Help:	
Modified On:	9/10/2024 6:58:21 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	
Column Header:	riskadj_gross_rel_q1_pv
Combine Groups By:	Sum Both
Combine Periods:	Last

Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.26**      ***riskadj\_gross\_rel\_q2\_pv***

Description:	PV of gross risk adjustment pv Q2
Help:	
Modified On:	9/10/2024 6:58:49 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	
Column Header:	riskadj_gross_rel_q2_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.27**      ***riskadj\_gross\_rel\_q3\_pv***

Description:	PV of gross risk adjustment pv Q3
Help:	
Modified On:	9/10/2024 6:59:10 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	
Column Header:	riskadj_gross_rel_q3_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.28**      ***riskadj\_gross\_rel\_q4\_pv***

Description:	PV of gross risk adjustment pv Q4
Help:	
Modified On:	9/10/2024 6:59:29 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	
Column Header:	riskadj_gross_rel_q4_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.29**      ***riskadj\_gross\_rel\_total\_pv***

Description:	PV of gross risk adjustment pv total
Help:	
Modified On:	9/10/2024 6:44:29 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	
Column Header:	riskadj_gross_rel_total_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.30**      ***riskadj\_gross\_rel\_yr2plus\_pv***

Description:	PV of gross risk adjustment pv Year 2 onwards
Help:	
Modified On:	9/10/2024 6:59:54 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	
Column Header:	riskadj_gross_rel_yr2plus_pv
Combine Groups By:	Sum Both
Combine Periods:	Last

Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.31**      *riskadj\_re\_rel\_q1\_pv*

Description:	PV of rein risk adjustment Q1
Help:	
Modified On:	9/10/2024 6:53:06 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	
Column Header:	riskadj_re_rel_q1_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.32**      *riskadj\_re\_rel\_q2\_pv*

Description:	PV of rein risk adjustment Q2
Help:	
Modified On:	9/10/2024 6:54:13 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	
Column Header:	riskadj_re_rel_q2_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False



**5.3.3.2.33**      *riskadj\_re\_rel\_q3\_pv*

Description:	PV of rein risk adjustment Q3
Help:	
Modified On:	9/10/2024 6:55:12 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	
Column Header:	riskadj_re_rel_q3_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.34**      *riskadj\_re\_rel\_q4\_pv*

Description:	PV of rein risk adjustment Q4
Help:	
Modified On:	9/10/2024 6:56:16 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	
Column Header:	riskadj_re_rel_q4_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.35**      *riskadj\_re\_rel\_total\_pv*

Description:	PV of release rein risk adjustment
Help:	
Modified On:	9/10/2024 6:45:04 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	
Column Header:	riskadj_re_rel_total_pv
Combine Groups By:	Sum Both
Combine Periods:	Last

Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.36**      *riskadj\_re\_rel\_yr2plus\_pv*

Description:	PV of rein risk adjustment Year 2 onwards
Help:	
Modified On:	9/10/2024 6:57:08 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	
Column Header:	riskadj_re_rel_yr2plus_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.37**      *surv\_per\_ret*

Description:	Survivorship w.r.t. retirement
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	
Column Header:	surv_per_ret
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.38**      ***surv\_ret***

Description:	Survivorship w.r.t. retirement
Help:	
Modified On:	8/3/2021 3:33:14 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	
Column Header:	surv_ret
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.39**      ***units\_at\_mat***

Description:	Units at maturity
Help:	
Modified On:	11/17/2022 5:30:58 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	
Column Header:	units_at_mat
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.40**      ***reserve***

Description:	Reserve total (excluding ERR)
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Balance Sheet Reserves
Column Header:	reserve
Combine Groups By:	Sum Both
Combine Periods:	Last

Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.41**      ***reserve\_bef\_ret***

Description:	Reserve total (excluding ERR) before retirement
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Balance Sheet Reserves
Column Header:	reserve_bef_ret
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.42**      ***reserve\_extra***

Description:	extra reserve items
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Balance Sheet Reserves
Column Header:	reserve_extra
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.43      *ann\_cost\_pv***

Description:	PV of annuity defeciciency cost at maturity
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Balance Sheet Reserves Annuity Deficiency
Column Header:	annuity_cost_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.44      *net\_prem\_def***

Description:	Net premiums Deficiency in force
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Balance Sheet Reserves Basic Reserve
Column Header:	net_prem_def
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.45      *net\_premium\_e***

Description:	Net premiums in force
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Balance Sheet Reserves Basic Reserve
Column Header:	net_premium_e
Combine Groups By:	Sum Both
Combine Periods:	Last

Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.46      *res\_np\_deficiency***

Description:	Net Premium Reserve Deficiency
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Balance Sheet Reserves Basic Reserve
Column Header:	res_np_def
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.47      *reserve\_annuity***

Description:	Annuity In Payment Reserves
Help:	
Modified On:	9/15/2022 9:56:27 AM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Balance Sheet Reserves Basic Reserve
Column Header:	reserve_annuity
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.48      *reserve\_basic***

Description:	Basic reserve - Net or Gross Premium
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Balance Sheet Reserves Basic Reserve
Column Header:	reserve_basic
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.49      *reserve\_basic\_bef\_ret***

Description:	Basic reserve - Net or Gross Premium before retirement
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Balance Sheet Reserves Basic Reserve
Column Header:	reserve_basic_bef_ret
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.50      *reserve\_basic\_gt\_su***

Description:	Basic reserve - guaranteed - for Service units
Help:	
Modified On:	5/29/2025 12:21:28 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Balance Sheet Reserves Basic Reserve
Column Header:	reserve_basic_gt_su
Combine Groups By:	Sum Both

Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.51**      ***reserve\_claims***

Description:	Basic reserve - Claims In Payment
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Balance Sheet Reserves Basic Reserve
Column Header:	reserve_claims
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.52**      ***reserve\_claims\_retent***

Description:	Basic reserve - Claims In Payment - retention
Help:	
Modified On:	4/11/2024 3:07:18 PM (UTC+03:00)
Modified By:	CLAL-INS\Arikt
Category:	Balance Sheet Reserves Basic Reserve
Column Header:	reserve_claims_retent
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False



**5.3.3.2.53**      ***reserve\_risk\_premium***

Description:	Risk premium for Basic reserve
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Balance Sheet Reserves Basic Reserve
Column Header:	reserve_risk_premium
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.54**      ***ber\_retire\_rm***

Description:	Risk driver for takeup risk
Help:	
Modified On:	8/5/2021 2:44:00 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Capital
Column Header:	ber_retire_rm
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.55**      ***capital\_at\_risk***

Description:	Capital At Risk
Help:	
Modified On:	12/3/2020 2:41:27 PM (UTC+02:00)
Modified By:	CLAL-INS\NinaB
Category:	Capital
Column Header:	capital_at_risk
Combine Groups By:	Sum Both
Combine Periods:	Last

Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.56**      *capital\_at\_risk\_rm*

Description:	RM Capital At Risk
Help:	
Modified On:	12/7/2020 3:37:44 PM (UTC+02:00)
Modified By:	CLAL-INS\NinaB
Category:	Capital
Column Header:	capital_at_risk_rm
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.57**      *claim\_cost\_pv\_rm*

Description:	Risk margin calc for disability claim cost
Help:	
Modified On:	2/14/2024 4:10:47 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Capital
Column Header:	claim_cost_pv_rm
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.58**      ***claim\_cost\_re\_pv\_rm***

Description:	Risk margin calc for disability rein claim cost
Help:	
Modified On:	3/26/2024 12:18:34 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Capital
Column Header:	claim_cost_re_pv_rm
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.59**      ***claims\_annuity\_pv\_rm***

Description:	Risk margin for longevity risk
Help:	
Modified On:	8/15/2021 4:19:40 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Capital
Column Header:	claims_annuity_pv_rm
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.60**      ***claims\_death\_pv\_rm***

Description:	Discounted risk margin for mortality scenario
Help:	
Modified On:	12/7/2020 3:49:04 PM (UTC+02:00)
Modified By:	CLAL-INS\NinaB
Category:	Capital
Column Header:	claims_death_pv_rm
Combine Groups By:	Sum Both
Combine Periods:	Last

Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.61**      *claims\_disability\_pv\_rm*

Description:	Risk margin calc for disability
Help:	
Modified On:	12/7/2020 3:40:40 PM (UTC+02:00)
Modified By:	CLAL-INS\NinaB
Category:	Capital
Column Header:	claims_disability_pv_rm
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.62**      *expense\_pv\_rm*

Description:	Risk driver for expense risk
Help:	
Modified On:	12/7/2020 3:41:56 PM (UTC+02:00)
Modified By:	CLAL-INS\NinaB
Category:	Capital
Column Header:	expense_pv_rm
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.63**      ***inv\_income\_chetz\_pv\_rm***

Description:	Risk margin for chetz
Help:	
Modified On:	12/7/2020 3:42:23 PM (UTC+02:00)
Modified By:	CLAL-INS\NinaB
Category:	Capital
Column Header:	inv_income_chetz_pv_rm
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.64**      ***profit\_book\_vif\_pv\_pos\_rm***

Description:	Risk driver for lapse risk
Help:	
Modified On:	12/7/2020 3:49:02 PM (UTC+02:00)
Modified By:	CLAL-INS\NinaB
Category:	Capital
Column Header:	profit_book_vif_pv_pos_rm
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.65**      ***rein\_claims\_pv\_rm***

Description:	Risk margin calc for rein disability claim cost using VA discount
Help:	
Modified On:	3/17/2024 1:29:21 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Capital
Column Header:	rein_claims_pv_rm
Combine Groups By:	Sum Both

Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### 5.3.3.2.66 ***v\_rm\_cumm***

Description:	Cummulative V for risk margin
Help:	
Modified On:	7/28/2021 5:03:46 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Capital
Column Header:	v_rm_cumm
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### 5.3.3.2.67 ***bonus\_shimur***

Description:	persistence bonus for savings
Help:	extra info for IFRS
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Cashflows
Column Header:	bonus_shimur
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.68**      ***cashflow\_b\_post\_ret***

Description:	Cashflow after retirement (beginning of period)
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Cashflows
Column Header:	cashflow_b_post_ret
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.69**      ***cashflow\_pv\_active***

Description:	Cashflow Pv Active
Help:	
Modified On:	6/15/2023 3:19:30 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Cashflows
Column Header:	cashflow_pv_active
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.70**      ***cashflow\_pv\_active\_chetz***

Description:	Cashflow Pv Active discounted using chetz rates
Help:	
Modified On:	7/14/2024 2:19:59 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Cashflows
Column Header:	cashflow_pv_active_chetz
Combine Groups By:	Sum Both

Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.71      *cashflow\_pv\_active\_e***

Description:	Cashflow Pv Active - all components being discountned EOP
Help:	
Modified On:	7/19/2022 12:58:48 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Cashflows
Column Header:	cashflow_pv_active_e
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.72      *cashflow\_pv\_deferred***

Description:	Cashflow Pv Deferred
Help:	
Modified On:	6/15/2023 3:21:47 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Cashflows
Column Header:	cashflow_pv_deferred
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False



**5.3.3.2.73**      ***cashflow\_pv\_deferred\_chetz***

Description:	Cashflow Pv Deferred chetz
Help:	
Modified On:	7/10/2024 12:49:12 AM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Cashflows
Column Header:	cashflow_pv_deferred_chetz
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.74**      ***cashflow\_pv\_deferred\_chetz\_ifrs***

Description:	Cashflow Pv Deferred chetz - IFRS
Help:	
Modified On:	7/14/2024 6:56:22 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Cashflows
Column Header:	cashflow_pv_deferred_chetz_ifrs
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.75**      ***cashflow\_pv\_deferred\_chetz\_res***

Description:	Cashflow Pv Deferred chetz - Res
Help:	
Modified On:	7/14/2024 6:55:12 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Cashflows
Column Header:	cashflow_pv_deferred_chetz_res
Combine Groups By:	Sum Both

Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.76**      *cashflow\_pv\_deferred\_e*

Description:	Cashflow Pv Deferred - discounted EOP
Help:	
Modified On:	9/22/2022 11:44:46 AM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Cashflows
Column Header:	cashflow_pv_deferred_e
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.77**      *cashflow\_pv\_ifrs*

Description:	Cashflow Pv Active discounted - IFRS
Help:	
Modified On:	7/14/2024 2:04:39 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Cashflows
Column Header:	cashflow_pv_ifrs
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.78**      ***cashflow\_pv\_ifrs\_active***

Description:	Cashflow Pv Active discounted - IFRS - Active
Help:	
Modified On:	7/14/2024 2:24:52 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Cashflows
Column Header:	cashflow_pv_ifrs_active
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.79**      ***cashflow\_pv\_res***

Description:	Cashflow Pv Active discounted - res
Help:	
Modified On:	7/14/2024 2:06:36 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Cashflows
Column Header:	cashflow_pv_res
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.80**      ***cashflow\_pv\_res\_active***

Description:	Cashflow Pv Active discounted - res - Active
Help:	
Modified On:	7/14/2024 2:18:35 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Cashflows
Column Header:	cashflow_pv_res_active
Combine Groups By:	Sum Both
Combine Periods:	Last

Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.81**      ***cashflow\_re\_pv***

Description:	Cashflow Re Pv
Help:	
Modified On:	7/12/2021 12:25:04 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows
Column Header:	cashflow_re_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.82**      ***claims\_annuity\_gt***

Description:	Guaranteed annuity claims for IFRS
Help:	extra info for IFRS
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Cashflows
Column Header:	claims_annuity_gt
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.83**      ***claims\_insurance***

Description:	Claims due to death or disability
Help:	extra info for IFRS
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Cashflows
Column Header:	claims_insurance
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.84**      ***comm\_hekef\_net***

Description:	Hekef commission net of clawback for IFRS
Help:	extra info for IFRS
Modified On:	11/29/2021 8:15:37 AM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows
Column Header:	comm_hekef_net
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.85**      ***comm\_profit***

Description:	Profit commission (calculation needs to be added when data field is added) for IFRS
Help:	extra info for IFRS
Modified On:	1/11/2023 1:56:31 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Cashflows
Column Header:	comm_profit
Combine Groups By:	Sum Both

Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.86      *comm\_reg***

Description:	All monthly commissions related to premiums for IFRS
Help:	extra info for IFRS
Modified On:	11/29/2021 8:17:25 AM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows
Column Header:	comm_reg
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.87      *coverage\_units***

Description:	Coverage units for IFRS - gross - weighted service units
Help:	extra info for IFRS
Modified On:	8/8/2024 4:58:10 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Cashflows
Column Header:	coverage_units
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False

Virtual: False

#### **5.3.3.2.88**      ***coverage\_units\_re***

Description:	Coverage units reinsurance for IFRS - weighted service units
Help:	extra info for IFRS
Modified On:	8/1/2024 12:32:22 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Cashflows
Column Header:	coverage_units_re
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.89**      ***expense\_clm***

Description:	Total claims expenses, including annuity expenses - for IFRS
Help:	extra info for IFRS
Modified On:	12/8/2021 8:57:25 AM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows
Column Header:	expense_clm
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.90**      ***expense\_init***

Description:	Initial expenses - for IFRS
Help:	extra info for IFRS
Modified On:	11/17/2022 5:32:36 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm

Category:	Cashflows
Column Header:	expense_init
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.91      *expense\_pv\_active***

Description:	Expense Pv Active
Help:	
Modified On:	9/13/2022 4:01:48 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Cashflows
Column Header:	expense_pv_active
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.92      *expense\_pv\_active\_no\_inv***

Description:	Expense Pv Active w/o investment expenses
Help:	
Modified On:	9/13/2022 4:02:19 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Cashflows
Column Header:	expense_pv_active_no_inv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes



Override: False  
Virtual: False

#### **5.3.3.2.93**      ***expense\_pv\_deferred***

Description: Expense Pv Deferred  
Help:  
Modified On: 10/26/2021 10:54:31 AM (UTC+03:00)  
Modified By: CLAL-INS\joshm  
Category: Cashflows  
Column Header: expense\_pv\_deferred  
Combine Groups By: Sum Both  
Combine Periods: Last  
Default sliding Size: -1  
Discount Timing: End  
Discount Use: Yes  
Rate Use: Cash Flow  
Rebase Type: Previous  
Retain Value: Yes  
Override: False  
Virtual: False

#### **5.3.3.2.94**      ***expense\_ren***

Description: Renewal expenses, not including annuity expenses - for IFRS  
Help: extra info for IFRS  
Modified On: 12/8/2021 8:58:40 AM (UTC+02:00)  
Modified By: CLAL-INS\joshm  
Category: Cashflows  
Column Header: expense\_ren  
Combine Groups By: Sum Both  
Combine Periods: Sum  
Default sliding Size: -1  
Discount Timing: End  
Discount Use: Yes  
Rate Use: Cash Flow  
Rebase Type: Previous  
Retain Value: Yes  
Override: False  
Virtual: False

#### **5.3.3.2.95**      ***expense\_var\_pv\_active***

Description: Variable Expense Pv Active  
Help:  
Modified On: 11/17/2022 5:32:36 PM (UTC+02:00)  
Modified By: CLAL-INS\joshm

Category:	Cashflows
Column Header:	expense_var_pv_active
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

### 5.3.3.2.96 *fvui*

Description:	PV FVUI (fair value of underlying items) - for IFRS ניכוי לפני החיסכון בגין המבוטח כלפי התחייבות ניהול דמי
Help:	extra info for IFRS
Modified On:	3/27/2023 11:16:09 AM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Cashflows
Column Header:	fvui
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

### 5.3.3.2.97 *int\_cred*

Description:	Return accrued to policy reserve - for IFRS - Note: query participating annuities
Help:	extra info for IFRS
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Cashflows
Column Header:	int_cred
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes

Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.98      *investment\_income\_chetz\_pv\_deferred***

Description:	Chetz investment income Pv Deferred
Help:	
Modified On:	6/19/2023 1:45:53 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Cashflows
Column Header:	investment_income_chetz_pv_deferred
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.99      *investment\_income\_chetz\_pv\_inpay***

Description:	Chetz Investment Income Pv AnnInpayment
Help:	
Modified On:	6/19/2023 1:46:00 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Cashflows
Column Header:	investment_income_chetz_pv_inpay
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.100      *investment\_income\_pv\_active***

Description:	Investment income Pv Active
--------------	-----------------------------

Help:  
 Modified On: 7/18/2022 6:07:11 PM (UTC+03:00)  
 Modified By: CLAL-INS\ahuvaa  
 Category: Cashflows  
 Column Header: investment\_income\_pv\_active  
 Combine Groups By: Sum Both  
 Combine Periods: Last  
 Default sliding Size: -1  
 Discount Timing: End  
 Discount Use: Yes  
 Rate Use: Cash Flow  
 Rebase Type: Previous  
 Retain Value: Yes  
 Override: False  
 Virtual: False

### **5.3.3.2.101**     *investment\_income\_pv\_deferred*

Description: Investment income Pv Deferred  
 Help:  
 Modified On: 9/12/2019 4:46:55 PM (UTC+03:00)  
 Modified By: CLAL-INS\NinaB  
 Category: Cashflows  
 Column Header: investment\_income\_pv\_deferred  
 Combine Groups By: Sum Both  
 Combine Periods: Last  
 Default sliding Size: -1  
 Discount Timing: End  
 Discount Use: Yes  
 Rate Use: Cash Flow  
 Rebase Type: Previous  
 Retain Value: Yes  
 Override: False  
 Virtual: False

### **5.3.3.2.102**     *mgt\_fees\_prem*

Description: Management fees deducted from premium  
 (profil only) - for IFRS  
 Help: extra info for IFRS  
 Modified On: 11/29/2021 8:42:58 AM (UTC+02:00)  
 Modified By: CLAL-INS\joshm  
 Category: Cashflows  
 Column Header: mgt\_fees\_prem  
 Combine Groups By: Sum Both  
 Combine Periods: Sum  
 Default sliding Size: -1  
 Discount Timing: End

Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

### **5.3.3.2.103**     *outgo\_pv\_active*

Description:	Outgo Pv Active
Help:	
Modified On:	9/12/2019 4:47:57 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Category:	Cashflows
Column Header:	outgo_pv_active
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

### **5.3.3.2.104**     *outgo\_pv\_deferred*

Description:	Outgo Pv Deferred
Help:	
Modified On:	9/12/2019 4:47:27 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Category:	Cashflows
Column Header:	outgo_pv_deferred
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.105      *prem\_insurance***

Description:	Premium allocated to insurance / risk - for IFRS
Help:	extra info for IFRS
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Cashflows
Column Header:	prem_insurance
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.106      *prem\_savings***

Description:	Premium allocated to savings - for IFRS
Help:	extra info for IFRS
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Cashflows
Column Header:	prem_savings
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.107      *profit\_book\_pv\_active***

Description:	Profit book Pv Active
Help:	
Modified On:	9/12/2019 4:50:25 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Category:	Cashflows
Column Header:	profit_book_pv_active
Combine Groups By:	Sum Both
Combine Periods:	Last

Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.108     *profit\_book\_pv\_deferred***

Description:	Profit Book Pv Deferred
Help:	
Modified On:	10/26/2021 11:12:03 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows
Column Header:	profit_book_pv_deferred
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.109     *profit\_book\_vif\_gross\_pv\_active***

Description:	Gross Profit book vif Pv Active
Help:	
Modified On:	6/28/2021 2:39:38 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows
Column Header:	profit_book_vif_gross_pv_active
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.110**     ***profit\_book\_vif\_pv\_active***

Description:	Profit book vif Pv Active
Help:	
Modified On:	9/12/2019 4:49:23 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Category:	Cashflows
Column Header:	profit_book_vif_pv_active
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.111**     ***profit\_book\_vif\_pv\_deferred***

Description:	Profit Book Vif Pv Deferred
Help:	
Modified On:	10/26/2021 11:13:29 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows
Column Header:	profit_book_vif_pv_deferred
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.112**     ***profit\_gross\_vif\_pv\_active***

Description:	Gross Profit Net vif Pv Active
Help:	
Modified On:	8/28/2022 4:39:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Cashflows
Column Header:	profit_gross_vif_pv_active
Combine Groups By:	Sum Both
Combine Periods:	Last



Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.113**     ***profit\_net\_vif\_pv\_active***

Description:	Net Profit vif Pv Active
Help:	
Modified On:	8/28/2022 4:37:44 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Cashflows
Column Header:	profit_net_vif_pv_active
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.114**     ***profit\_net\_vif\_pv\_deferred***

Description:	Net Profit Vif Pv Deferred
Help:	
Modified On:	9/12/2019 4:57:52 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Category:	Cashflows
Column Header:	profit_net_vif_pv_deferred
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.115      *reserve\_increase\_pv\_active***

Description:	Reserve increase Pv Active
Help:	
Modified On:	7/18/2022 6:07:32 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Cashflows
Column Header:	reserve_increase_pv_active
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.116      *reserve\_increase\_pv\_deferred***

Description:	Reserve Increase Pv Deferred
Help:	
Modified On:	9/12/2019 4:48:37 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Category:	Cashflows
Column Header:	reserve_increase_pv_deferred
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.117      *reserve\_pv***

Description:	Reserve PV
Help:	
Modified On:	11/24/2021 12:48:01 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows
Column Header:	reserve_pv
Combine Groups By:	Sum Both
Combine Periods:	Last

Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

### 5.3.3.2.118 *rid\_cashflow\_pv*

Description:	PV of cashflows for profil riders
Help:	
Modified On:	1/11/2023 7:02:58 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows
Column Header:	rid_cashflow_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

### 5.3.3.2.119 *service\_units*

Description:	Proxy for runoff of profit - for IFRS - Note: query proportionality
Help:	extra info for IFRS
Modified On:	5/29/2025 12:21:29 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Cashflows
Column Header:	service_units
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.120      *service\_units\_pv***

Description:	PV of service units - for IFRS
Help:	extra info for IFRS
Modified On:	12/8/2021 3:21:35 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows
Column Header:	service_units_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.121      *units\_for\_takeup***

Description:	Units/reserves eligible for annuity takeup - for IFRS
Help:	extra info for IFRS
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Cashflows
Column Header:	units_for_takeup
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.122      *income\_b***

Description:	Income at beginning of the period
Help:	
Modified On:	8/6/2024 6:41:55 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Cashflows Income
Column Header:	income_b
Combine Groups By:	Sum Both

Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.123**     *income\_e*

Description:	Income at end of the period
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Cashflows Income
Column Header:	income_e
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.124**     *income\_pv*

Description:	Income Pv
Help:	
Modified On:	7/12/2021 12:36:51 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows Income
Column Header:	income_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.125**      ***charges\_premium***

Description:	Premium related charges
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Cashflows Income Charges
Column Header:	charges_premium
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.126**      ***charges\_premium\_pv***

Description:	PV Premium related charges
Help:	
Modified On:	9/12/2019 4:16:39 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Category:	Cashflows Income Charges
Column Header:	charges_premium_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.127**      ***cover\_charge***

Description:	Life cover charge (beg)
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Cashflows Income Charges
Column Header:	cover_charge_b
Combine Groups By:	Sum Both
Combine Periods:	Sum

Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.128      *cover\_charge\_pv***

Description:	PV Life cover charges
Help:	
Modified On:	9/12/2019 4:20:20 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Category:	Cashflows Income Charges
Column Header:	cover_charge_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.129      *management\_fee\_pv***

Description:	PV Management fee
Help:	
Modified On:	11/17/2022 5:33:34 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows Income Charges
Column Header:	management_fees_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.130      *management\_fees***

Description:	Management fees
Help:	
Modified On:	11/17/2022 5:33:34 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows Income Charges
Column Header:	management_fees
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.131      *mgt\_var\_no\_bor***

Description:	management fees for same calc results in managment_fees_variable and bor_return for when no bor
Help:	
Modified On:	9/25/2024 12:44:48 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Cashflows Income Charges
Column Header:	mgt_var_no_bor
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.132      *mgt\_var\_no\_bor\_pup***

Description:	management fees for same calc results in managment_fees_variable and bor_return
Help:	
Modified On:	9/25/2024 12:45:06 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Cashflows Income Charges
Column Header:	mgt_var_no_bor_pup



Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

### 5.3.3.2.133 *surr\_charge*

Description:	Surrender penalty received in period
Help:	
Modified On:	11/17/2022 4:46:09 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows Income Charges
Column Header:	surr_charge
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

### 5.3.3.2.134 *investment\_income*

Description:	Investment Income
Help:	
Modified On:	9/10/2024 12:47:24 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Cashflows Income Investment Income
Column Header:	investment_income
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.135**      ***investment\_income\_bef\_ret***

Description:	Investment Income
Help:	
Modified On:	7/21/2024 7:05:40 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Cashflows Income Investment Income
Column Header:	investment_income_bef_ret
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.136**      ***investment\_income\_chetz***

Description:	Investment Income form chetz
Help:	
Modified On:	7/18/2024 10:47:25 AM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Cashflows Income Investment Income
Column Header:	investment_income_chetz
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.137**      ***units\_e\_piz\_int\_active***

Description:	Cummulative investment income on piz units
Help:	
Modified On:	2/26/2025 12:27:34 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Cashflows Income Investment Income
Column Header:	units_e_piz_int_active
Combine Groups By:	Sum Both

Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.138**      *units\_e\_piz\_int\_pup*

Description:	Cummulative investment income on piz units (pup)
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Cashflows Income Investment Income
Column Header:	units_e_piz_int_pup
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.139**      *be\_reserve*

Description:	best estimate reserve
Help:	
Modified On:	5/8/2022 3:37:09 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Cashflows Income Premium
Column Header:	be_reserve
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.140      *discount\_factor\_acc***

Description:	Accumulated discount factor
Help:	
Modified On:	8/16/2024 12:53:53 AM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Cashflows Income Premium
Column Header:	discount_factor_acc
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	0
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.141      *investment\_income\_chetz\_bef\_ret***

Description:	investment income from chetz before retiring
Help:	PV of investment income at the "discount rate", at the beginning of the period. The discount rate in the experience model is the embedded value discount rate
Modified On:	7/25/2024 3:06:30 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Cashflows Income Premium
Column Header:	investment_income_chetz_bef_ret
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	0
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.142      *investment\_income\_chetz\_pv***

Description:	PV of investment income from chetz
--------------	------------------------------------

Help:	PV of investment income at the "discount rate", at the beginning of the period. The discount rate in the experience model is the embedded value discount rate
Modified On:	7/9/2024 2:39:02 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Cashflows Income Premium
Column Header:	investment_income_chetz_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	0
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.143**     *investment\_income\_chetz\_pv\_active*

Description:	PV of investment income from chetz Active
Help:	PV of investment income at the "discount rate", at the beginning of the period. The discount rate in the experience model is the embedded value discount rate
Modified On:	7/10/2024 12:20:45 AM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Cashflows Income Premium
Column Header:	investment_income_chetz_pv_active
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	0
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.144**     *investment\_income\_pv*

Description:	PV of investment income
Help:	PV of investment income at the "discount rate", at the beginning of the period. The discount rate in the experience model is the embedded value discount rate
Modified On:	7/21/2024 11:14:04 AM (UTC+03:00)

Modified By:	CLAL-INS\arikt
Category:	Cashflows Income Premium
Column Header:	investment_income_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	0
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

### 5.3.3.2.145 *pol\_fee*

Description:	Policy Fee
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Cashflows Income Premium
Column Header:	pol_fee
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

### 5.3.3.2.146 *pol\_fee\_pv*

Description:	pv of Policy Fee
Help:	
Modified On:	9/12/2019 4:23:19 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Category:	Cashflows Income Premium
Column Header:	pol_fee_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous

Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.147**     ***prem\_savings\_pv***

Description:	PV of premiums savings
Help:	
Modified On:	3/27/2023 10:58:44 AM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Cashflows Income Premium
Column Header:	prem_savings_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	0
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.148**     ***premium***

Description:	premium (excluding policy fee)
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Cashflows Income Premium
Column Header:	premium
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.149**     ***premium\_disc***

Description:	premium discount
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)

Modified By:	CLAL-INS\Arikt
Category:	Cashflows Income Premium
Column Header:	premium_disc
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

### 5.3.3.2.150 *premium\_disc\_pv*

Description:	PV of premium discounts
Help:	
Modified On:	9/12/2019 4:11:27 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Category:	Cashflows Income Premium
Column Header:	premium_disc_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	0
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

### 5.3.3.2.151 *premium\_disc\_shimur*

Description:	premium discount - shimur
Help:	
Modified On:	4/11/2024 6:37:02 PM (UTC+03:00)
Modified By:	CLAL-INS\Arikt
Category:	Cashflows Income Premium
Column Header:	premium_disc_shimur
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous



Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.152**     ***premium\_disc\_shimur\_pv***

Description:	PV of premium discounts - shimur
Help:	
Modified On:	4/11/2024 6:02:30 PM (UTC+03:00)
Modified By:	CLAL-INS\Arikt
Category:	Cashflows Income Premium
Column Header:	premium_disc_shimur_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	0
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.153**     ***premium\_extra***

Description:	Premium income for extra sum insured
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Cashflows Income Premium
Column Header:	premium_extra
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.154**     ***premium\_gross***

Description:	Premium (including policy fee)
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)

Modified By:	CLAL-INS\Arikt
Category:	Cashflows Income Premium
Column Header:	premium_gross
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

### 5.3.3.2.155 *premium\_gross\_fix*

Description:	fixed premium gross - monthly
Help:	
Modified On:	9/21/2022 2:48:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Cashflows Income Premium
Column Header:	premium_gross_fix
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

### 5.3.3.2.156 *premium\_gross\_var*

Description:	variable premium gross - monthly
Help:	
Modified On:	9/21/2022 2:50:17 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Cashflows Income Premium
Column Header:	premium_gross_var
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous

Retain Value:	Yes
Override:	False
Virtual:	False

### **5.3.3.2.157**     ***premium\_pv***

Description:	PV of premiums
Help:	
Modified On:	9/12/2019 4:06:07 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Category:	Cashflows Income Premium
Column Header:	premium_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	0
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

### **5.3.3.2.158**     ***total\_bor\_acc\_pv***

Description:	PV of Bor
Help:	
Modified On:	6/24/2024 3:14:01 PM (UTC+03:00)
Modified By:	CLAL-INS\Arikt
Category:	Cashflows Income Premium
Column Header:	total_bor_acc_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	0
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

### **5.3.3.2.159**     ***total\_bor\_return\_pv***

Description:	PV of Bor return
Help:	
Modified On:	6/24/2024 3:16:03 PM (UTC+03:00)

Modified By:	CLAL-INS\Arikt
Category:	Cashflows Income Premium
Column Header:	total_bor_return_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	0
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

### 5.3.3.2.160 *outgo\_b*

Description:	Outgo at the beginning of the period
Help:	
Modified On:	12/9/2021 12:36:40 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows Outgo
Column Header:	outgo_b
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

### 5.3.3.2.161 *outgo\_e*

Description:	Outgo at end of period
Help:	
Modified On:	9/12/2024 11:15:39 AM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Cashflows Outgo
Column Header:	outgo_e
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous

Retain Value: Yes  
Override: False  
Virtual: False

#### **5.3.3.2.162**     *outgo\_pv*

Description: Outgo Pv  
Help:  
Modified On: 10/26/2021 11:11:00 AM (UTC+03:00)  
Modified By: CLAL-INS\joshm  
Category: Cashflows|Outgo  
Column Header: outgo\_pv  
Combine Groups By: Sum Both  
Combine Periods: Last  
Default sliding Size: -1  
Discount Timing: End  
Discount Use: Yes  
Rate Use: Cash Flow  
Rebase Type: Previous  
Retain Value: Yes  
Override: False  
Virtual: False

#### **5.3.3.2.163**     *claim\_cost*

Description: PHI claims inpay  
Help:  
Modified On: 3/26/2024 12:22:18 PM (UTC+02:00)  
Modified By: CLAL-INS\Arikt  
Category: Cashflows|Outgo|Claims  
Column Header: claim\_cost  
Combine Groups By: Sum Both  
Combine Periods: Last  
Default sliding Size: -1  
Discount Timing: End  
Discount Use: Yes  
Rate Use: Cash Flow  
Rebase Type: Previous  
Retain Value: Yes  
Override: False  
Virtual: False

#### **5.3.3.2.164**     *claim\_cost\_pv*

Description: PV of Disability Claim cost  
Help:  
Modified On: 1/22/2024 1:24:08 PM (UTC+02:00)

Modified By:	CLAL-INS\Arikt
Category:	Cashflows Outgo Claims
Column Header:	claim_cost_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.165**      *claim\_cost\_re*

Description:	PHI rein claims inpay
Help:	
Modified On:	3/26/2024 12:22:28 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Cashflows Outgo Claims
Column Header:	claim_cost_re
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.166**      *claim\_cost\_re\_pv*

Description:	PV of Reins Disability Claim cost
Help:	
Modified On:	3/17/2024 1:18:30 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Cashflows Outgo Claims
Column Header:	claim_cost_re_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous

Retain Value:	Yes
Override:	False
Virtual:	False

### 5.3.3.2.167 *claims\_annuity*

Description:	Annuity Claims in Period
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Cashflows Outgo Claims
Column Header:	claims_annuity
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

### 5.3.3.2.168 *claims\_annuity\_nogt*

Description:	Annuity Claims in Period, excluding guaranteed payments.
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Cashflows Outgo Claims
Column Header:	claims_annuity_nogt
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

### 5.3.3.2.169 *claims\_annuity\_nogt\_pv*

Description:	PV of Annuity Claims Paid, excluding guaranteed payments.
Help:	

Modified On:	8/12/2024 11:38:33 AM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Cashflows Outgo Claims
Column Header:	claims_annuity_nogt_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

### 5.3.3.2.170 *claims\_annuity\_nogt\_pv\_deferred*

Description:	PV of Deferred Annuity Claims Paid (non-guaranteed)
Help:	
Modified On:	10/26/2021 10:51:38 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows Outgo Claims
Column Header:	claims_annuity_nogt_pv_deferred
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

### 5.3.3.2.171 *claims\_annuity\_pv*

Description:	PV of Annuity Claims Paid
Help:	
Modified On:	10/26/2021 10:51:54 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows Outgo Claims
Column Header:	claims_annuity_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes



Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.172      *claims\_annuity\_pv\_deferred***

Description:	PV of Deferred Annuity Claims Paid
Help:	
Modified On:	8/15/2021 4:12:00 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows Outgo Claims
Column Header:	claims_annuity_pv_deferred
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.173      *claims\_death***

Description:	Total Death Claims in Period
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Cashflows Outgo Claims
Column Header:	claims_death
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.174      *claims\_death\_pv***

Description:	PV of Death Claims Paid (total)
--------------	---------------------------------

Help:  
 Modified On: 9/12/2019 4:18:01 PM (UTC+03:00)  
 Modified By: CLAL-INS\NinaB  
 Category: Cashflows|Outgo|Claims  
 Column Header: claims\_death\_pv  
 Combine Groups By: Sum Both  
 Combine Periods: Last  
 Default sliding Size: -1  
 Discount Timing: End  
 Discount Use: Yes  
 Rate Use: Cash Flow  
 Rebase Type: Previous  
 Retain Value: Yes  
 Override: False  
 Virtual: False

### 5.3.3.2.175 *claims\_disability*

Description: Total disability Claims in Period  
 Help:  
 Modified On: 12/11/2023 2:43:58 PM (UTC+02:00)  
 Modified By: CLAL-INS\Arikt  
 Category: Cashflows|Outgo|Claims  
 Column Header: claims\_disability  
 Combine Groups By: Sum Both  
 Combine Periods: Sum  
 Default sliding Size: -1  
 Discount Timing: End  
 Discount Use: Yes  
 Rate Use: Cash Flow  
 Rebase Type: Previous  
 Retain Value: Yes  
 Override: False  
 Virtual: False

### 5.3.3.2.176 *claims\_disability\_pv*

Description: PV of Disability Claims Paid (total)  
 Help:  
 Modified On: 9/12/2019 4:18:27 PM (UTC+03:00)  
 Modified By: CLAL-INS\NinaB  
 Category: Cashflows|Outgo|Claims  
 Column Header: claims\_disability\_pv  
 Combine Groups By: Sum Both  
 Combine Periods: Last  
 Default sliding Size: -1  
 Discount Timing: End  
 Discount Use: Yes

Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.177**      ***claims\_lrc\_q1***

Description:	Disability Claims in Period from first quarter event
Help:	
Modified On:	2/12/2024 4:01:19 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Cashflows Outgo Claims
Column Header:	claims_lrc_q1
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.178**      ***claims\_lrc\_q2***

Description:	Disability Claims in Period from seconde quarter event
Help:	
Modified On:	2/12/2024 4:02:42 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Cashflows Outgo Claims
Column Header:	claims_lrc_q2
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.179**      ***claims\_lrc\_q3***

Description:	Disability Claims in Period from third quarter event
Help:	
Modified On:	2/12/2024 4:02:50 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Cashflows Outgo Claims
Column Header:	claims_lrc_q3
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.180**      ***claims\_lrc\_q4***

Description:	Disability Claims in Period from forth quarter event
Help:	
Modified On:	2/12/2024 4:02:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Cashflows Outgo Claims
Column Header:	claims_lrc_q4
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.181**      ***claims\_lrc\_yr2plus***

Description:	Disability Claims in Period that not from first year event
Help:	
Modified On:	2/12/2024 4:03:08 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Cashflows Outgo Claims
Column Header:	claims_lrc_yr2plus

Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

### 5.3.3.2.182 *claims\_maturity*

Description:	Total Maturity Claims in Period
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Cashflows\Outgo\Claims
Column Header:	claims_maturity
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

### 5.3.3.2.183 *claims\_maturity\_pv*

Description:	PV of Maturity Claims Paid (total)
Help:	
Modified On:	9/12/2019 4:15:35 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Category:	Cashflows\Outgo\Claims
Column Header:	claims_maturity_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.184**      ***claims\_maturity\_ret***

Description:	Total reserves at retirement
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Cashflows Outgo Claims
Column Header:	claims_maturity_ret
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.185**      ***claims\_maturity\_ret\_pv***

Description:	PV of Maturity Claims Paid at retirement
Help:	
Modified On:	9/18/2022 11:54:13 AM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Cashflows Outgo Claims
Column Header:	claims_maturity_ret_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.186**      ***claims\_pv***

Description:	PV of Claims Paid (total)
Help:	
Modified On:	10/5/2021 2:20:50 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows Outgo Claims
Column Header:	claims_pv
Combine Groups By:	Sum Both

Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.187      *claims\_re\_lrc\_q1***

Description:	Disability Claims in Period from first quarter event - reinsurance
Help:	
Modified On:	2/12/2024 4:03:17 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Cashflows Outgo Claims
Column Header:	claims_re_lrc_q1
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.188      *claims\_re\_lrc\_q2***

Description:	Disability Claims in Period from seconde quarter event - reinsurance
Help:	
Modified On:	2/12/2024 4:03:23 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Cashflows Outgo Claims
Column Header:	claims_re_lrc_q2
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False

Virtual: False

#### **5.3.3.2.189      *claims\_re\_lrc\_q3***

Description: Disability Claims in Period from third quarter event - reinsurance

Help:

Modified On: 2/12/2024 4:03:32 PM (UTC+02:00)

Modified By: CLAL-INS\Arikt

Category: Cashflows|Outgo|Claims

Column Header: claims\_re\_lrc\_q3

Combine Groups By: Sum Both

Combine Periods: Sum

Default sliding Size: -1

Discount Timing: End

Discount Use: Yes

Rate Use: Cash Flow

Rebase Type: Previous

Retain Value: Yes

Override: False

Virtual: False

#### **5.3.3.2.190      *claims\_re\_lrc\_q4***

Description: Disability Claims in Period from forth quarter event - reinsurance

Help:

Modified On: 2/12/2024 4:03:40 PM (UTC+02:00)

Modified By: CLAL-INS\Arikt

Category: Cashflows|Outgo|Claims

Column Header: claims\_re\_lrc\_q4

Combine Groups By: Sum Both

Combine Periods: Sum

Default sliding Size: -1

Discount Timing: End

Discount Use: Yes

Rate Use: Cash Flow

Rebase Type: Previous

Retain Value: Yes

Override: False

Virtual: False

#### **5.3.3.2.191      *claims\_re\_lrc\_yr2plus***

Description: Disability Claims in Period that not from first year event- reinsurance

Help:

Modified On: 2/12/2024 4:03:46 PM (UTC+02:00)



Modified By:	CLAL-INS\Arikt
Category:	Cashflows Outgo Claims
Column Header:	claims_re_lrc_yr2plus
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

### 5.3.3.2.192 *claims\_surrender*

Description:	Surrender claims
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Cashflows Outgo Claims
Column Header:	claims_surrender
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

### 5.3.3.2.193 *claims\_surrender\_pv*

Description:	PV of Surrender Claims Paid (total)
Help:	
Modified On:	9/12/2019 4:20:39 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Category:	Cashflows Outgo Claims
Column Header:	claims_surrender_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous

Retain Value:	Yes
Override:	False
Virtual:	False

### 5.3.3.2.194 *claims\_total*

Description:	Claims Paid (total)
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Cashflows Outgo Claims
Column Header:	claims_total
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

### 5.3.3.2.195 *death\_benefit*

Description:	Total death benefit
Help:	
Modified On:	12/8/2021 3:00:37 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows Outgo Claims
Column Header:	death_benefit
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

### 5.3.3.2.196 *death\_claim\_si*

Description:	Death claims on sum insured
Help:	
Modified On:	1/11/2023 6:58:10 PM (UTC+02:00)

Modified By:	CLAL-INS\joshm
Category:	Cashflows Outgo Claims
Column Header:	death_claim_si
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.197      *death\_claim\_units***

Description:	Death claims on units
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Cashflows Outgo Claims
Column Header:	death_claim_units
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.198      *expense\_claims\_lrc\_q1***

Description:	Expenses disability Claims in Period from first quarter event
Help:	
Modified On:	2/12/2024 4:04:12 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Cashflows Outgo Claims
Column Header:	expense_claims_lrc_q1
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow

Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.199**      ***expense\_claims\_lrc\_q2***

Description:	Expenses disability Claims in Period from seconde quarter event
Help:	
Modified On:	2/12/2024 4:04:32 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Cashflows Outgo Claims
Column Header:	expense_claims_lrc_q2
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.200**      ***expense\_claims\_lrc\_q3***

Description:	Expenses Disability Claims in Period from third quarter event
Help:	
Modified On:	2/12/2024 4:04:44 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Cashflows Outgo Claims
Column Header:	expense_claims_lrc_q3
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.201**      ***expense\_claims\_lrc\_q4***

Description:	Expenses Disability Claims in Period from forth quarter event
Help:	
Modified On:	2/12/2024 4:04:52 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Cashflows Outgo Claims
Column Header:	expense_claims_lrc_q4
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.202**      ***expense\_claims\_lrc\_yr2plus***

Description:	Expenses Disability Claims in Period that not from first year event
Help:	
Modified On:	2/12/2024 4:04:59 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Cashflows Outgo Claims
Column Header:	expense_claims_lrc_yr2plus
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.203**      ***nogt\_annpv***

Description:	Just for output
Help:	
Modified On:	8/12/2024 12:31:19 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Cashflows Outgo Claims
Column Header:	nogt_annpv
Combine Groups By:	Sum Both

Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.204      *surr\_penalty\_e\_bef***

Description:	Surrender penalty inforce, end of period
Help:	
Modified On:	11/17/2022 4:46:21 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows Outgo Claims
Column Header:	surr_penalty_e_bef
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.205      *surr\_value***

Description:	Surrender value inforce
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Cashflows Outgo Claims
Column Header:	surr_value
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.206 comm\_hekef**

Description:	First year lump sum commission with clawback
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Cashflows Outgo Commission
Column Header:	comm_hekef
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.207 comm\_nihul**

Description:	nihul commission
Help:	Nihul Commission
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Cashflows Outgo Commission
Column Header:	comm_nihul
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.208 comm\_nihul\_pv**

Description:	PV of nihul commissions
Help:	PV of (EXTRA commissions less clawback) at the "discount rate", at the beginning of the period. The discount rate in the experience model is the embedded value discount rate.
Modified On:	6/8/2023 3:29:03 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Cashflows Outgo Commission

Column Header:	comm_nihul_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	0
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.209      *comm\_not\_res\_pv***

Description:	PV of commissions - expect reserve commission
Help:	
Modified On:	9/12/2022 3:23:02 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Cashflows Outgo Commission
Column Header:	comm_not_res_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	0
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.210      *comm\_prize***

Description:	First year lump sum commission
Help:	
Modified On:	11/17/2022 5:29:40 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows Outgo Commission
Column Header:	comm_prize
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes



Override: False  
Virtual: False

### 5.3.3.2.211 *comm\_pv*

Description: PV of commissions  
Help:  
Modified On: 9/12/2019 2:49:47 PM (UTC+03:00)  
Modified By: CLAL-INS\NinaB  
Category: Cashflows|Outgo|Commission  
Column Header: comm\_pv  
Combine Groups By: Sum Both  
Combine Periods: Last  
Default sliding Size: 0  
Discount Timing: End  
Discount Use: No  
Rate Use: Cash Flow  
Rebase Type: Current  
Retain Value: Yes  
Override: False  
Virtual: False

### 5.3.3.2.212 *comm\_reg\_riders\_out\_pv*

Description: נוכחי ערך ריידר שוטפות עמלות  
Help: PV of (EXTRA commissions less clawback) at the "discount rate", at the beginning of the period.  
The discount rate in the experience model is the embedded value discount rate.  
Modified On: 6/8/2023 3:29:41 PM (UTC+03:00)  
Modified By: CLAL-INS\ahuvaa  
Category: Cashflows|Outgo|Commission  
Column Header: comm\_reg\_riders\_out\_pv  
Combine Groups By: Sum Both  
Combine Periods: Last  
Default sliding Size: 0  
Discount Timing: End  
Discount Use: No  
Rate Use: Cash Flow  
Rebase Type: Current  
Retain Value: Yes  
Override: False  
Virtual: False

### 5.3.3.2.213 *comm\_regular*

Description: Regular commission

Help:  
 Modified On: 12/11/2023 2:43:58 PM (UTC+02:00)  
 Modified By: CLAL-INS\Arikt  
 Category: Cashflows|Outgo|Commission  
 Column Header: comm\_regular  
 Combine Groups By: Sum Both  
 Combine Periods: Sum  
 Default sliding Size: -1  
 Discount Timing: Beginning  
 Discount Use: Yes  
 Rate Use: Cash Flow  
 Rebase Type: Current  
 Retain Value: Yes  
 Override: False  
 Virtual: False

#### **5.3.3.2.214      *comm\_renewal***

Description: Renewal commission  
 Help:  
 Modified On: 3/13/2025 10:41:04 AM (UTC+02:00)  
 Modified By: CLAL-INS\ahuvaa  
 Category: Cashflows|Outgo|Commission  
 Column Header: comm\_renewal  
 Combine Groups By: Sum Both  
 Combine Periods: Sum  
 Default sliding Size: -1  
 Discount Timing: Beginning  
 Discount Use: Yes  
 Rate Use: Cash Flow  
 Rebase Type: Previous  
 Retain Value: Yes  
 Override: False  
 Virtual: False

#### **5.3.3.2.215      *comm\_renewal\_pv***

Description: PV of renewal commissions  
 Help:  
 Modified On: 1/11/2023 10:16:46 AM (UTC+02:00)  
 Modified By: CLAL-INS\ahuvaa  
 Category: Cashflows|Outgo|Commission  
 Column Header: comm\_renewal\_pv  
 Combine Groups By: Sum Both  
 Combine Periods: Last  
 Default sliding Size: 0  
 Discount Timing: End  
 Discount Use: No

Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

### 5.3.3.2.216 *comm\_reserve*

Description:	Reserve commission
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Cashflows Outgo Commission
Column Header:	comm_reserve
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

### 5.3.3.2.217 *comm\_reserve\_pv*

Description:	PV of reserve commissions
Help:	
Modified On:	9/12/2019 4:17:32 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Category:	Cashflows Outgo Commission
Column Header:	comm_reserve_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	0
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

### 5.3.3.2.218 *comm\_total*

Description:	total commission
--------------	------------------

Help:  
 Modified On: 1/11/2023 10:14:04 AM (UTC+02:00)  
 Modified By: CLAL-INS\ahuvaa  
 Category: Cashflows|Outgo|Commission  
 Column Header: comm\_total  
 Combine Groups By: Sum Both  
 Combine Periods: Sum  
 Default sliding Size: -1  
 Discount Timing: Beginning  
 Discount Use: Yes  
 Rate Use: Cash Flow  
 Rebase Type: Previous  
 Retain Value: Yes  
 Override: False  
 Virtual: False

### 5.3.3.2.219 *profit\_re\_pv*

Description: PV of reinsurance profit  
 Help:  
 Modified On: 9/12/2019 4:08:25 PM (UTC+03:00)  
 Modified By: CLAL-INS\NinaB  
 Category: Cashflows|Outgo|Commission  
 Column Header: profit\_re\_pv  
 Combine Groups By: Sum Both  
 Combine Periods: Last  
 Default sliding Size: -1  
 Discount Timing: End  
 Discount Use: No  
 Rate Use: Cash Flow  
 Rebase Type: Current  
 Retain Value: Yes  
 Override: False  
 Virtual: False

### 5.3.3.2.220 *comm\_clawback*

Description: Commission clawback  
 Help:  
 Modified On: 1/11/2023 7:02:58 PM (UTC+02:00)  
 Modified By: CLAL-INS\joshm  
 Category: Cashflows|Outgo|Commission|Clawback  
 Column Header: comm\_clawback  
 Combine Groups By: Sum Both  
 Combine Periods: Sum  
 Default sliding Size: -1  
 Discount Timing: Beginning  
 Discount Use: Yes

Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.221      *comm\_hekef\_cum***

Description:	cumul of hekef commissions
Help:	
Modified On:	1/11/2023 7:02:58 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows Outgo Commission Clawback
Column Header:	comm_hekef_cum
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.222      *exp\_total***

Description:	Total expenses
Help:	
Modified On:	11/17/2022 5:32:36 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows Outgo Expenses
Column Header:	expense_total
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.223      *expense\_inflation***

Description:	Accumulated Expense Inflation
--------------	-------------------------------

Help:  
 Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
 Modified By: CLAL-INS\ninab  
 Category: Cashflows|Outgo|Expenses  
 Column Header: expense\_inflation  
 Combine Groups By: Sum Both  
 Combine Periods: Last  
 Default sliding Size: -1  
 Discount Timing: End  
 Discount Use: Yes  
 Rate Use: Cash Flow  
 Rebase Type: Previous  
 Retain Value: Yes  
 Override: False  
 Virtual: False

### 5.3.3.2.224 ***expense\_pv***

Description: PV of expenses  
 Help:  
 Modified On: 12/9/2021 12:36:40 PM (UTC+02:00)  
 Modified By: CLAL-INS\joshm  
 Category: Cashflows|Outgo|Expenses  
 Column Header: expense\_pv  
 Combine Groups By: Sum Both  
 Combine Periods: Last  
 Default sliding Size: -1  
 Discount Timing: End  
 Discount Use: Yes  
 Rate Use: Cash Flow  
 Rebase Type: Previous  
 Retain Value: Yes  
 Override: False  
 Virtual: False

### 5.3.3.2.225 ***expense\_total\_pre\_ret***

Description: Total expenses before retirement  
 Help:  
 Modified On: 11/17/2022 5:32:36 PM (UTC+02:00)  
 Modified By: CLAL-INS\joshm  
 Category: Cashflows|Outgo|Expenses  
 Column Header: expense\_total\_pre\_ret  
 Combine Groups By: Sum Both  
 Combine Periods: Sum  
 Default sliding Size: -1  
 Discount Timing: Beginning  
 Discount Use: Yes

Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.226      *expense\_total\_pre\_ret\_no\_inv***

Description:	Total expenses before retirement w/o investment expenses
Help:	
Modified On:	11/17/2022 5:32:36 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows Outgo Expenses
Column Header:	expense_total_pre_ret_no_inv
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.227      *expense\_var\_pv***

Description:	PV of overhead expenses (variable expenses)
Help:	
Modified On:	11/17/2022 5:32:36 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows Outgo Expenses
Column Header:	expense_var_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.228      *expense\_init\_fix\_cvr***

Description:	Fixed Initial expenses (per cover)
--------------	------------------------------------

Help:  
 Modified On: 1/18/2023 5:04:26 PM (UTC+02:00)  
 Modified By: CLAL-INS\joshm  
 Category: Cashflows|Outgo|Expenses|Initial  
 Column Header: expense\_init\_fix\_cvr  
 Combine Groups By: Sum Both  
 Combine Periods: Sum  
 Default sliding Size: -1  
 Discount Timing: Beginning  
 Discount Use: Yes  
 Rate Use: Cash Flow  
 Rebase Type: Previous  
 Retain Value: Yes  
 Override: False  
 Virtual: False

#### **5.3.3.2.229**      ***expense\_initial\_fix***

Description: Fixed Initial expenses  
 Help:  
 Modified On: 11/17/2022 5:32:07 PM (UTC+02:00)  
 Modified By: CLAL-INS\joshm  
 Category: Cashflows|Outgo|Expenses|Initial  
 Column Header: expense\_initial\_fix  
 Combine Groups By: Sum Both  
 Combine Periods: Sum  
 Default sliding Size: -1  
 Discount Timing: Beginning  
 Discount Use: Yes  
 Rate Use: Cash Flow  
 Rebase Type: Previous  
 Retain Value: Yes  
 Override: False  
 Virtual: False

#### **5.3.3.2.230**      ***expense\_initial\_fix\_pol***

Description: Fixed Initial expenses (per policy)  
 Help:  
 Modified On: 11/17/2022 5:32:07 PM (UTC+02:00)  
 Modified By: CLAL-INS\joshm  
 Category: Cashflows|Outgo|Expenses|Initial  
 Column Header: expense\_initial\_fix\_pol  
 Combine Groups By: Sum Both  
 Combine Periods: Sum  
 Default sliding Size: -1  
 Discount Timing: Beginning  
 Discount Use: Yes



Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.231**      ***expense\_initial\_perc***

Description:	Variable Initial expenses
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Cashflows Outgo Expenses Initial
Column Header:	expense_initial_perc
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.232**      ***expense\_claims***

Description:	ExpensVariable claim expenses Clm Perc
Help:	
Modified On:	11/17/2022 4:49:51 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows Outgo Expenses Other expenses
Column Header:	expense_claims
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.233**      ***expense\_claims\_pv***

Description:	PV of claim expenses
--------------	----------------------

Help:  
 Modified On: 10/26/2021 10:53:37 AM (UTC+03:00)  
 Modified By: CLAL-INS\joshm  
 Category: Cashflows|Outgo|Expenses|Other expenses  
 Column Header: expense\_claims\_pv  
 Combine Groups By: Sum Both  
 Combine Periods: Last  
 Default sliding Size: -1  
 Discount Timing: End  
 Discount Use: No  
 Rate Use: Cash Flow  
 Rebase Type: Previous  
 Retain Value: Yes  
 Override: False  
 Virtual: False

#### **5.3.3.2.234**      ***expense\_clm\_perc***

Description: ExpensVariable claim expenses Clm Perc  
 Help:  
 Modified On: 11/17/2022 4:41:40 PM (UTC+02:00)  
 Modified By: CLAL-INS\joshm  
 Category: Cashflows|Outgo|Expenses|Other expenses  
 Column Header: expense\_clm\_perc  
 Combine Groups By: Sum Both  
 Combine Periods: Sum  
 Default sliding Size: -1  
 Discount Timing: End  
 Discount Use: No  
 Rate Use: Cash Flow  
 Rebase Type: Previous  
 Retain Value: Yes  
 Override: False  
 Virtual: False

#### **5.3.3.2.235**      ***expense\_clms\_fix***

Description: Fixed claims expenses  
 Help:  
 Modified On: 12/8/2021 3:02:27 PM (UTC+02:00)  
 Modified By: CLAL-INS\joshm  
 Category: Cashflows|Outgo|Expenses|Other expenses  
 Column Header: expense\_clms\_fix  
 Combine Groups By: Sum Both  
 Combine Periods: Sum  
 Default sliding Size: -1  
 Discount Timing: End  
 Discount Use: No

Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### 5.3.3.2.236 **expense\_pv\_ann**

Description:	PV of expense_ren_perc_ann (Variable renewal expenses from annuity)
Help:	
Modified On:	9/29/2022 2:38:10 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Cashflows Outgo Expenses Other expenses
Column Header:	expense_pv_ann
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### 5.3.3.2.237 **expense\_ren\_fix\_pv**

Description:	PV of fixed renewal expenses
Help:	
Modified On:	3/6/2025 4:56:08 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Cashflows Outgo Expenses Other expenses
Column Header:	expense_ren_fix_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### 5.3.3.2.238 **expense\_ren\_perc\_pv**

Description:	PV of % renewal expenses
--------------	--------------------------

Help:  
 Modified On: 3/9/2025 11:02:02 AM (UTC+02:00)  
 Modified By: CLAL-INS\ahuvaa  
 Category: Cashflows|Outgo|Expenses|Other expenses  
 Column Header: expense\_ren\_perc\_pv  
 Combine Groups By: Sum Both  
 Combine Periods: Last  
 Default sliding Size: -1  
 Discount Timing: End  
 Discount Use: No  
 Rate Use: Cash Flow  
 Rebase Type: Previous  
 Retain Value: Yes  
 Override: False  
 Virtual: False

### 5.3.3.2.239 *comm\_reg\_riders\_out*

Description: Additional commission for riders "out" of profil  
 Help:  
 Modified On: 12/11/2023 2:43:58 PM (UTC+02:00)  
 Modified By: CLAL-INS\Arikt  
 Category: Cashflows|Outgo|Expenses|Renewal  
 Column Header: comm\_reg\_riders\_out  
 Combine Groups By: Sum Both  
 Combine Periods: Sum  
 Default sliding Size: -1  
 Discount Timing: Beginning  
 Discount Use: Yes  
 Rate Use: Cash Flow  
 Rebase Type: Previous  
 Retain Value: Yes  
 Override: False  
 Virtual: False

### 5.3.3.2.240 *expense\_ren\_charge*

Description: Variable renewal expenses on charges  
 Help:  
 Modified On: 12/11/2023 2:43:58 PM (UTC+02:00)  
 Modified By: CLAL-INS\Arikt  
 Category: Cashflows|Outgo|Expenses|Renewal  
 Column Header: expense\_ren\_charge  
 Combine Groups By: Sum Both  
 Combine Periods: Sum  
 Default sliding Size: -1  
 Discount Timing: Beginning  
 Discount Use: Yes

Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.241**      ***expense\_ren\_fix***

Description:	Fixed Renewal expenses
Help:	
Modified On:	6/8/2021 4:42:21 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows Outgo Expenses Renewal
Column Header:	expense_ren_fix
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.242**      ***expense\_ren\_fix\_cvr***

Description:	Fixed Renewal expenses (per cover)
Help:	
Modified On:	5/29/2025 11:33:21 AM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Cashflows Outgo Expenses Renewal
Column Header:	expense_ren_fix_cvr
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.243**      ***expense\_ren\_fix\_pol***

Description:	Fixed Renewal expenses (per policy)
--------------	-------------------------------------

Help:  
Modified On: 5/29/2025 11:33:56 AM (UTC+03:00)  
Modified By: CLAL-INS\arikt  
Category: Cashflows|Outgo|Expenses|Renewal  
Column Header: expense\_ren\_fix\_pol  
Combine Groups By: Sum Both  
Combine Periods: Sum  
Default sliding Size: -1  
Discount Timing: Beginning  
Discount Use: Yes  
Rate Use: Cash Flow  
Rebase Type: Previous  
Retain Value: Yes  
Override: False  
Virtual: False

#### **5.3.3.2.244**      ***expense\_ren\_fix\_pup***

Description: Fixed Renewal expenses (PUP policies)  
Help:  
Modified On: 5/29/2025 11:34:31 AM (UTC+03:00)  
Modified By: CLAL-INS\arikt  
Category: Cashflows|Outgo|Expenses|Renewal  
Column Header: expense\_ren\_fix\_pup  
Combine Groups By: Sum Both  
Combine Periods: Sum  
Default sliding Size: -1  
Discount Timing: Beginning  
Discount Use: Yes  
Rate Use: Cash Flow  
Rebase Type: Previous  
Retain Value: Yes  
Override: False  
Virtual: False

#### **5.3.3.2.245**      ***expense\_ren\_perc***

Description: Variable renewal expenses  
Help:  
Modified On: 3/31/2025 9:02:01 PM (UTC+03:00)  
Modified By: CLAL-INS\arikt  
Category: Cashflows|Outgo|Expenses|Renewal  
Column Header: expense\_ren\_perc  
Combine Groups By: Sum Both  
Combine Periods: Sum  
Default sliding Size: -1  
Discount Timing: Beginning  
Discount Use: Yes

Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.246      *expense\_ren\_perc\_ann***

Description:	Variable renewal expenses from annuity
Help:	
Modified On:	12/8/2021 8:56:18 AM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows Outgo Expenses Renewal
Column Header:	expense_ren_perc_ann
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.247      *expense\_ren\_perc\_bef\_ret***

Description:	Variable renewal expenses before retirement
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Cashflows Outgo Expenses Renewal
Column Header:	expense_ren_perc_bef_ret
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.248      *expense\_ren\_perc\_bef\_ret\_no\_inv***

Description:	Variable renewal expenses before retirement w/o investment expenses
--------------	--

Help:  
 Modified On: 12/11/2023 2:43:58 PM (UTC+02:00)  
 Modified By: CLAL-INS\Arikt  
 Category: Cashflows|Outgo|Expenses|Renewal  
 Column Header: expense\_ren\_perc\_bef\_ret\_no\_inv  
 Combine Groups By: Sum Both  
 Combine Periods: Sum  
 Default sliding Size: -1  
 Discount Timing: Beginning  
 Discount Use: Yes  
 Rate Use: Cash Flow  
 Rebase Type: Previous  
 Retain Value: Yes  
 Override: False  
 Virtual: False

### 5.3.3.2.249 *comm\_claw\_spv*

Description: Supervisor commission clawback  
 Help:  
 Modified On: 1/11/2023 7:02:58 PM (UTC+02:00)  
 Modified By: CLAL-INS\joshm  
 Category: Cashflows|Outgo|Expenses|Supervisors  
 Commission  
 Column Header: comm\_claw\_spv  
 Combine Groups By: Sum Both  
 Combine Periods: Last  
 Default sliding Size: -1  
 Discount Timing: Beginning  
 Discount Use: Yes  
 Rate Use: Cash Flow  
 Rebase Type: Previous  
 Retain Value: Yes  
 Override: False  
 Virtual: False

### 5.3.3.2.250 *comm\_spv\_cum*

Description: Cumulated supervisor commissions  
 Help:  
 Modified On: 1/11/2023 7:02:58 PM (UTC+02:00)  
 Modified By: CLAL-INS\joshm  
 Category: Cashflows|Outgo|Expenses|Supervisors  
 Commission  
 Column Header: comm\_spv\_cum  
 Combine Groups By: Sum Both  
 Combine Periods: Last  
 Default sliding Size: 0



Discount Timing:	Beginning
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

### 5.3.3.2.251 *comm\_supervisor*

Description:	supervisor commission
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Cashflows Outgo Expenses Supervisors Commission
Column Header:	comm_supervisor
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

### 5.3.3.2.252 *reserve\_increase*

Description:	Increase in total reserve
Help:	
Modified On:	1/11/2023 7:47:17 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows Outgo Reserve Increase
Column Header:	reserve_increase
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.253      *reserve\_increase\_bef\_ret***

Description:	Increase in total reserve before retirement
Help:	
Modified On:	7/7/2024 4:31:30 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Cashflows Outgo Reserve Increase
Column Header:	reserve_increase_bef_ret
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.254      *reserve\_increase\_pv***

Description:	PV of Increase in total reserve
Help:	
Modified On:	10/26/2021 11:14:37 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows Outgo Reserve Increase
Column Header:	reserve_increase_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.255      *reserve\_total\_increase\_pv***

Description:	PV of Total Reserve Increase
Help:	PV of Total increase in (net) reserves.
Modified On:	11/17/2022 4:50:03 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows Outgo Reserve Increase
Column Header:	reserve_total_increase_pv
Combine Groups By:	Sum Both
Combine Periods:	Last

Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

### 5.3.3.2.256 *cashflow*

Description:	Cashflow profit for the period
Help:	
Modified On:	12/8/2021 8:28:00 AM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows Profit
Column Header:	cashflow
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

### 5.3.3.2.257 *cashflow\_b*

Description:	Income minus outgo at beg of period
Help:	
Modified On:	10/5/2021 2:20:41 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows Profit
Column Header:	cashflow_b
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.258**      ***cashflow\_e***

Description:	Income minus outgo at end of period
Help:	
Modified On:	10/5/2021 2:20:35 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows Profit
Column Header:	cashflow_e
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.259**      ***cashflow\_profit***

Description:	
Help:	
Modified On:	12/8/2021 8:27:42 AM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows Profit
Column Header:	cashflow_profit
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.260**      ***cashflow\_profit\_bef\_ret***

Description:	Cashflow profit for the period before retirement
Help:	
Modified On:	11/17/2022 4:40:00 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows Profit
Column Header:	cashflow_profit_bef_ret
Combine Groups By:	Sum Both
Combine Periods:	Sum

Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.261**     ***profit\_book\_active\_vif***

Description:	Before Tax Profit (cashflow basis for MCEV) of policy active period (VIF Profit)
Help:	After Tax Profit (cashflow basis for MCEV) without DAC (VIF Profit)
Modified On:	9/12/2024 11:17:04 AM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Cashflows Profit
Column Header:	profit_bk_act_vif
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.262**     ***profit\_book\_bef\_ret***

Description:	Profit Book Bef Ret
Help:	
Modified On:	9/12/2024 11:17:59 AM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Cashflows Profit
Column Header:	profit_book_bef_ret
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.263**     ***profit\_book\_vif***

Description:	Before Tax Profit (cashflow basis for MCEV) without DAC (VIF Profit)
Help:	After Tax Profit (cashflow basis for MCEV) without DAC (VIF Profit)
Modified On:	9/12/2024 11:18:27 AM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Cashflows Profit
Column Header:	profit_book_vif
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.264**     ***profit\_book\_vif\_bef\_ret***

Description:	Before Tax Profit (cashflow basis for MCEV) without DAC (VIF Profit) before retirement
Help:	After Tax Profit (cashflow basis for MCEV) without DAC (VIF Profit)
Modified On:	9/12/2024 11:18:50 AM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Cashflows Profit
Column Header:	profit_book_vif_bef_ret
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.265**     ***profit\_book\_vif\_gross***

Description:	Profit Book Vif Gross
Help:	
Modified On:	12/8/2021 3:19:12 PM (UTC+02:00)

Modified By:	CLAL-INS\joshm
Category:	Cashflows Profit
Column Header:	profit_book_vif_gross
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.266**     ***profit\_book\_vif\_gross\_pv***

Description:	Profit Book Vif Gross Pv
Help:	
Modified On:	10/26/2021 11:13:02 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows Profit
Column Header:	profit_book_vif_gross_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.267**     ***profit\_book\_vif\_post\_ret***

Description:	Before Tax Profit (cashflow basis for MCEV) without DAC (VIF Profit) after retirement
Help:	After Tax Profit (cashflow basis for MCEV) without DAC (VIF Profit)
Modified On:	11/17/2022 4:44:55 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows Profit
Column Header:	profit_book_vif_post_ret
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes

Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.268**     *profit\_book\_vif\_pv\_pos*

Description:	Profit Book Vif Pv Pos
Help:	
Modified On:	7/31/2023 6:08:22 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Cashflows Profit
Column Header:	profit_book_vif_pv_pos
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.269**     *profit\_gross\_vif*

Description:	Profit Gross Vif
Help:	
Modified On:	8/28/2022 4:37:14 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Cashflows Profit
Column Header:	profit_net_vif_gross
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.270**     *profit\_gross\_vif\_pv*

Description:	Profit Gross Vif Pv
--------------	---------------------



Help:  
 Modified On: 10/26/2021 11:13:50 AM (UTC+03:00)  
 Modified By: CLAL-INS\joshm  
 Category: Cashflows|Profit  
 Column Header: profit\_net\_vif\_gross\_pv  
 Combine Groups By: Sum Both  
 Combine Periods: Last  
 Default sliding Size: -1  
 Discount Timing: End  
 Discount Use: Yes  
 Rate Use: Cash Flow  
 Rebase Type: Previous  
 Retain Value: Yes  
 Override: False  
 Virtual: False

### 5.3.3.2.271 *profit\_net\_vif*

Description: After Tax Profit (cashflow basis for MCEV) without DAC (VIF Profit)  
 Help: After Tax Profit (cashflow basis for MCEV) without DAC (VIF Profit)  
 Modified On: 1/8/2025 12:21:48 PM (UTC+02:00)  
 Modified By: CLAL-INS\arikt  
 Category: Cashflows|Profit  
 Column Header: profit\_net\_vif  
 Combine Groups By: Sum Both  
 Combine Periods: Sum  
 Default sliding Size: -1  
 Discount Timing: End  
 Discount Use: Yes  
 Rate Use: Cash Flow  
 Rebase Type: Previous  
 Retain Value: Yes  
 Override: False  
 Virtual: False

### 5.3.3.2.272 *profit\_vif\_net\_bef\_ret*

Description: Profit Vif Net Bef Ret  
 Help:  
 Modified On: 8/28/2022 4:37:32 PM (UTC+03:00)  
 Modified By: CLAL-INS\ahuvaa  
 Category: Cashflows|Profit  
 Column Header: profit\_vif\_net\_bef\_ret  
 Combine Groups By: Sum Both  
 Combine Periods: Sum  
 Default sliding Size: -1

Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.273**      ***comm\_dac***

Description:	Deferrable commission for DAC
Help:	
Modified On:	4/9/2024 5:49:45 PM (UTC+03:00)
Modified By:	CLAL-INS\Arikt
Category:	Cashflows Profit DAC
Column Header:	comm_dac
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.274**      ***dac\_book***

Description:	DAC (books)
Help:	
Modified On:	7/12/2021 12:55:59 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows Profit DAC
Column Header:	dac_book
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.275      *dac\_il\_book***

Description:	DAC books (Israeli GAAP)
Help:	
Modified On:	1/11/2023 6:58:10 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows Profit DAC
Column Header:	dac_il_book
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.276      *dac\_il\_tax***

Description:	DAC tax (Israeli TAX/STAT)
Help:	
Modified On:	1/11/2023 7:44:06 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows Profit DAC
Column Header:	dac_il_tax
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.277      *dac\_increase***

Description:	Increase in DAC books
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Cashflows Profit DAC
Column Header:	dac_increase
Combine Groups By:	Sum Both
Combine Periods:	Sum

Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

### 5.3.3.2.278 *dac\_tax*

Description:	Dac for tax purposes
Help:	
Modified On:	1/11/2023 6:58:10 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows Profit DAC
Column Header:	dac_tax
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

### 5.3.3.2.279 *dac\_tax\_increase*

Description:	Increase in DAC for tax
Help:	
Modified On:	8/9/2021 4:07:13 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows Profit DAC
Column Header:	dac_tax_increase
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.280 exp\_dac**

Description:	expenses used for DAC calculation
Help:	
Modified On:	11/17/2022 5:32:36 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows Profit DAC
Column Header:	exp_dac
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.281 zillmer\_book**

Description:	Zillmer for reported surplus
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Cashflows Profit DAC
Column Header:	zillmer_book
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.282 zillmer\_tax**

Description:	Zillmer for tax purposes
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Cashflows Profit DAC
Column Header:	zillmer_tax
Combine Groups By:	Sum Both
Combine Periods:	Last

Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.283     *bor\_acc***

Description:	Bor accumulated for active UNIT policies
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Charges
Column Header:	bor_acc
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.284     *bor\_acc\_pup***

Description:	Bor accumulated for paid-up UNIT policies
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Charges
Column Header:	bor_acc_pup
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.285      *bor\_har\_retire***

Description:	Management Fees Accum/Deficiency at retirement
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Charges
Column Header:	bor_har_retire
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.286      *bor\_return***

Description:	Bor returned for active policies
Help:	
Modified On:	9/25/2024 12:44:48 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Charges
Column Header:	bor_return
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.287      *bor\_return\_pup***

Description:	Bor returned for paid-up policies
Help:	
Modified On:	3/26/2025 12:24:39 PM (UTC+02:00)
Modified By:	CLAL-INS\arikt
Category:	Charges
Column Header:	bor_return_pup
Combine Groups By:	Sum Both

Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.288**     *har\_acc*

Description:	Har accumulated for active policies
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Charges
Column Header:	har_acc
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.289**     *har\_acc\_pup*

Description:	Har accumulated for paid-up policies
Help:	
Modified On:	3/26/2025 12:52:54 PM (UTC+02:00)
Modified By:	CLAL-INS\arikt
Category:	Charges
Column Header:	har_acc_pup
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False



**5.3.3.2.290**     *har\_return*

Description:	Har returned for active policies
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Charges
Column Header:	har_return
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.291**     *har\_return\_pup*

Description:	Har returned for paid-up policies
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Charges
Column Header:	har_return_pup
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.292**     *manage\_fees\_fixe\_active\_pv*

Description:	PV of fixed management fees for active policies
Help:	
Modified On:	10/18/2021 12:38:54 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Charges
Column Header:	manage_fees_fixed_active_pv
Combine Groups By:	Sum Both
Combine Periods:	Last

Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.293      *manage\_fees\_fixed\_ann\_pv***

Description:	PV of fixed management fees for annuity stage
Help:	
Modified On:	6/29/2021 11:20:49 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Charges
Column Header:	management_fees_fixed_ann_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.294      *manage\_fees\_fixed\_ann\_pv\_def***

Description:	Management Fees Fixed Pv Deferred
Help:	
Modified On:	6/29/2021 11:20:03 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Charges
Column Header:	manage_fees_fixed_ann_pv_def
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.295      *manage\_fees\_fixed\_ann\_pv\_ip***

Description:	Management Fees Fixed Pv Inpayment
Help:	
Modified On:	10/18/2021 12:39:09 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Charges
Column Header:	manage_fees_fixed_ann_pv_ip
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.296      *manage\_fees\_var\_active\_pv***

Description:	PV of variable management fees for active policies
Help:	
Modified On:	10/18/2021 12:39:15 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Charges
Column Header:	manage_fees_var_active_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.297      *manage\_fees\_var\_ann\_pv***

Description:	PV of variable management fees for annuity stage
Help:	
Modified On:	6/29/2021 11:22:01 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Charges
Column Header:	management_fees_var_ann_pv
Combine Groups By:	Sum Both

Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.298      *manage\_fees\_var\_ann\_pv\_def***

Description:	Management Fees Variable Pv Deferred
Help:	
Modified On:	11/17/2022 4:43:37 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Charges
Column Header:	manage_fees_var_ann_pv_def
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.299      *manage\_fees\_var\_ann\_pv\_ip***

Description:	Management Fees Variable Pv Inpayment
Help:	
Modified On:	10/18/2021 12:39:25 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Charges
Column Header:	manage_fees_var_ann_pv_ip
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.300      *management\_fee\_variable***

Description:	Variable management fees for active UNIT policies
Help:	
Modified On:	9/25/2024 12:44:48 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Charges
Column Header:	management_fee_variable
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.301      *management\_fee\_variable\_pup***

Description:	Variable management fees for paid-up UNIT policies
Help:	
Modified On:	9/25/2024 12:45:06 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Charges
Column Header:	management_fee_variable_pup
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.302      *management\_fees\_fixed\_active***

Description:	Fixed management fees for active policies
Help:	
Modified On:	4/10/2024 11:59:32 AM (UTC+03:00)
Modified By:	CLAL-INS\Arikt
Category:	Charges
Column Header:	management_fees_fixed_active
Combine Groups By:	Sum Both

Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.303      *management\_fees\_fixed\_ann***

Description:	Fixed management fees for annuity stage
Help:	
Modified On:	10/2/2022 10:59:19 AM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Charges
Column Header:	management_fees_fixed_ann
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.304      *management\_fees\_var\_active***

Description:	Variable management fees for active policies
Help:	
Modified On:	4/10/2024 11:58:22 AM (UTC+03:00)
Modified By:	CLAL-INS\Arikt
Category:	Charges
Column Header:	management_fees_var_active
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.305      *management\_fees\_var\_ann***

Description:	Variable management fees for annuity stage
Help:	
Modified On:	10/2/2022 10:59:58 AM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Charges
Column Header:	management_fees_var_ann
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.306      *net\_interest\_rate***

Description:	Interest rate net of fixed management fees
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\arikt
Category:	Charges
Column Header:	net_interest_rate
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.307      *new\_pup\_har\_ret***

Description:	
Help:	
Modified On:	3/26/2025 12:17:33 PM (UTC+02:00)
Modified By:	CLAL-INS\arikt
Category:	Charges
Column Header:	new_pup_har_ret
Combine Groups By:	Sum Both
Combine Periods:	Last

Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.308**      *cashflow\_pv\_inpay*

Description:	Cashflow Pv AnnInpayment
Help:	
Modified On:	8/30/2021 2:54:38 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Claims
Column Header:	cashflow_pv_inpay
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.309**      *cashflow\_pv\_inpay\_chetz*

Description:	Cashflow Pv AnnInpayment chetz
Help:	
Modified On:	7/14/2024 2:15:06 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Claims
Column Header:	cashflow_pv_inpay_chetz
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False



**5.3.3.2.310**      ***cashflow\_pv\_inpay\_e***

Description:	Cashflow Pv AnnInpayment discounted EOP
Help:	
Modified On:	7/19/2022 3:20:58 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Claims
Column Header:	cashflow_pv_inpay_e
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.311**      ***claims\_annuity\_nogt\_pv\_inpay***

Description:	Claims Annuity Pv Inpayment (non-guaranteed)
Help:	
Modified On:	10/26/2021 10:51:45 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Claims
Column Header:	claims_annuity_nogt_pv_inpay
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.312**      ***claims\_annuity\_pv\_inpay***

Description:	Claims Annuity Pv Inpayment
Help:	
Modified On:	8/15/2021 4:12:36 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Claims
Column Header:	claims_annuity_pv_inpay
Combine Groups By:	Sum Both
Combine Periods:	Last

Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.313**      *claims\_lrc\_yr2plus\_pv*

Description:	PV of Disability Claims after first year
Help:	
Modified On:	2/12/2024 4:05:32 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Claims
Column Header:	claims_lrc_yr2plus_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.314**      *claims\_pv\_not\_annuity*

Description:	Claims Pv (Not Annuity)
Help:	
Modified On:	8/15/2021 4:12:42 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Claims
Column Header:	claims_pv_not_annuity
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.315**      ***claims\_rate\_per***

Description:	Claims Rate Per
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Claims
Column Header:	claims_rate_per
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.316**      ***expense\_pv\_inpay***

Description:	Expense Pv AnnInpayment
Help:	
Modified On:	10/26/2021 10:54:39 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Claims
Column Header:	expense_pv_inpay
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.317**      ***investment\_income\_pv\_inpay***

Description:	Investment Income Pv AnnInpayment
Help:	
Modified On:	10/26/2021 10:56:16 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Claims
Column Header:	investment_income_pv_inpay
Combine Groups By:	Sum Both
Combine Periods:	Last

Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.318**      ***outgo\_pv\_inpay***

Description:	Outgo Pv AnnInpayment
Help:	
Modified On:	10/26/2021 11:11:18 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Claims
Column Header:	outgo_pv_inpay
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.319**      ***profit\_book\_pv\_inpay***

Description:	Profit Book Pv AnnInpayment
Help:	
Modified On:	10/26/2021 11:12:12 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Claims
Column Header:	profit_book_pv_inpay
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.320**     ***profit\_book\_vif\_pv\_inpay***

Description:	Profit Book Vif Pv AnnInpayment
Help:	
Modified On:	10/26/2021 11:13:35 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Claims
Column Header:	profit_book_vif_pv_inpay
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.321**     ***profit\_net\_vif\_pv\_inpay***

Description:	Net Profit Vif Pv AnnInpayment
Help:	
Modified On:	10/26/2021 11:14:19 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Claims
Column Header:	profit_net_vif_pv_inpay
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.322**     ***reserve\_increase\_pv\_inpay***

Description:	Reserve increase Pv AnnInpayment
Help:	
Modified On:	10/26/2021 11:14:55 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Claims
Column Header:	reserve_increase_pv_inpay
Combine Groups By:	Sum Both
Combine Periods:	Last

Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.323      *comm\_clawback\_pv***

Description:	Comm Clawback Pv
Help:	
Modified On:	9/12/2019 4:27:32 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Category:	Clawback
Column Header:	comm_clawback_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.324      *comm\_reg\_pv***

Description:	Comm Regular Pv
Help:	
Modified On:	6/8/2023 3:27:30 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Commission
Column Header:	comm_reg_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.325      *decrement\_rate\_unit***

Description:	Decrement Rate for units (active stage)
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Decrements
Column Header:	decrement_rate_unit
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.326      *decrement\_rate\_unit\_pup***

Description:	Decrement Rate for units (paid-up stage)
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Decrements
Column Header:	decrement_rate_unit_pup
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.327      *duration\_denominator***

Description:	Denominator for calculation of duration
Help:	
Modified On:	1/11/2023 7:46:07 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Decrements
Column Header:	duration_denominator
Combine Groups By:	Sum Both
Combine Periods:	Last

Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.328      *duration\_numerator***

Description:	Numerator for calculation of duration
Help:	
Modified On:	1/11/2023 7:46:13 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Decrements
Column Header:	duration_numerator
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.329      *surv\_act\_bal***

Description:	Proportion of active policies premium paying - on balance exposure
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Decrements
Column Header:	surv_act_bal
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False



**5.3.3.2.330**      ***surv\_act\_bal\_bef\_ret***

Description:	Proportion of active policies - before retirement - on balance exposure
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Decrements
Column Header:	surv_act_bal_bef_ret
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.331**      ***surv\_act\_cnt***

Description:	Proportion of active policies premium paying - on count exposure
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Decrements
Column Header:	surv_act_cnt
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.332**      ***surv\_act\_cnt\_bef\_ret***

Description:	Proportion of active policies - before retirement - on count exposure
Help:	
Modified On:	1/12/2023 12:26:05 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Decrements
Column Header:	surv_act_cnt_bef_ret

Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### 5.3.3.2.333 *surv\_act\_post\_ret*

Description:	Proportion of active policies
Help:	
Modified On:	1/12/2023 9:57:09 AM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Decrements
Column Header:	surv_act_post_ret
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### 5.3.3.2.334 *surv\_act\_prm*

Description:	Proportion of active policies premium paying - on premium exposure
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Decrements
Column Header:	surv_act_prm
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False

Virtual: False

### **5.3.3.2.335**     *surv\_act\_prm\_bef\_ret*

Description: Proportion of active policies - before retirement  
- on premium exposure

Help:

Modified On: 12/11/2023 2:43:58 PM (UTC+02:00)

Modified By: CLAL-INS\Arikt

Category: Decrements

Column Header: surv\_act\_prm\_bef\_ret

Combine Groups By: Sum Both

Combine Periods: Last

Default sliding Size: -1

Discount Timing: End

Discount Use: No

Rate Use: Cash Flow

Rebase Type: Current

Retain Value: Yes

Override: False

Virtual: False

### **5.3.3.2.336**     *surv\_bal*

Description: Propn of lives in force at end of time t - on  
balance exposure

Help:

Modified On: 12/11/2023 2:43:58 PM (UTC+02:00)

Modified By: CLAL-INS\Arikt

Category: Decrements

Column Header: surv\_bal

Combine Groups By: Sum Both

Combine Periods: Last

Default sliding Size: -1

Discount Timing: End

Discount Use: No

Rate Use: Cash Flow

Rebase Type: Current

Retain Value: Yes

Override: False

Virtual: False

### **5.3.3.2.337**     *surv\_bal\_bef\_ret*

Description: Propn of lives in force at end of time t - before  
retirement - on balance exposure

Help:

Modified On: 12/11/2023 2:43:58 PM (UTC+02:00)

Modified By:	CLAL-INS\Arikt
Category:	Decrements
Column Header:	surv_bal_bef_ret
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

### 5.3.3.2.338 *surv\_cnt*

Description:	Propn of lives in force at end of time t - on count exposure
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Decrements
Column Header:	surv_cnt
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

### 5.3.3.2.339 *surv\_cnt\_bef\_ret*

Description:	Propn of lives in force at end of time t - before retirement - on count exposure
Help:	
Modified On:	1/12/2023 12:26:13 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Decrements
Column Header:	surv_cnt_bef_ret
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No

Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.340**     *surv\_per\_act\_bal*

Description:	Prob of staying active for period - on balance exposure
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Decrements
Column Header:	surv_per_act_bal
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.341**     *surv\_per\_act\_bal\_bef\_ret*

Description:	Prob of staying active for period - before retirement - on balance exposure
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Decrements
Column Header:	surv_per_act_bal_bef_ret
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.342**     ***surv\_per\_act\_cnt***

Description:	Prob of staying active for period - on count exposure
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Decrements
Column Header:	surv_per_act_cnt
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.343**     ***surv\_per\_act\_cnt\_bef\_ret***

Description:	Prob of staying active for period - before retirement - on count exposure
Help:	
Modified On:	1/12/2023 12:26:01 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Decrements
Column Header:	surv_per_act_cnt_bef_ret
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.344**     ***surv\_per\_act\_prm***

Description:	Prob of staying active for period - on premium exposure
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Decrements
Column Header:	surv_per_act_prm

Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.345**     *surv\_per\_act\_prm\_bef\_ret*

Description:	Prob of staying active for period - before retirement - on premium exposure
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Decrements
Column Header:	surv_per_act_prm_bef_ret
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.346**     *surv\_per\_bal*

Description:	Probability of survival for the period - on balance exposure
Help:	
Modified On:	1/11/2023 7:05:22 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Decrements
Column Header:	surv_per_bal
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes

Override: False  
Virtual: False

#### **5.3.3.2.347**     *surv\_per\_bal\_bef\_ret*

Description: Probability of survival for the period - before retirement - on balance exposure

Help:

Modified On: 12/11/2023 2:43:58 PM (UTC+02:00)

Modified By: CLAL-INS\Arikt

Category: Decrements

Column Header: surv\_per\_bal\_bef\_ret

Combine Groups By: Average Both

Combine Periods: Last

Default sliding Size: -1

Discount Timing: End

Discount Use: No

Rate Use: Cash Flow

Rebase Type: Current

Retain Value: Yes

Override: False

Virtual: False

#### **5.3.3.2.348**     *surv\_per\_cnt*

Description: Probability of survival for the period - on count exposure

Help:

Modified On: 1/11/2023 7:06:04 PM (UTC+02:00)

Modified By: CLAL-INS\joshm

Category: Decrements

Column Header: surv\_per\_cnt

Combine Groups By: Average Both

Combine Periods: Last

Default sliding Size: -1

Discount Timing: End

Discount Use: No

Rate Use: Cash Flow

Rebase Type: Current

Retain Value: Yes

Override: False

Virtual: False

#### **5.3.3.2.349**     *surv\_per\_prm\_bef\_ret*

Description: Probability of survival for the period - before retirement - on premium exposure

Help:



Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Decrements
Column Header:	surv_per_prm_bef_ret
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

### 5.3.3.2.350 *surv\_prm*

Description:	Proprn of lives in force at end of time t - on premium exposure
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Decrements
Column Header:	surv_prm
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

### 5.3.3.2.351 *surv\_prm\_bef\_ret*

Description:	Proprn of lives in force at end of time t - before retirement - on premium exposure
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Decrements
Column Header:	surv_prm_bef_ret
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End

Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

### **5.3.3.2.352**     *surv\_pup\_bal*

Description:	Propn of paid-up policies at end of t - on balance exposure
Help:	
Modified On:	1/17/2023 10:31:22 AM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Decrements
Column Header:	surv_pup_bal
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

### **5.3.3.2.353**     *surv\_pup\_bal\_bef\_ret*

Description:	Propn of paid-up policies at end of t - before retirement - on balance exposure
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Decrements
Column Header:	surv_pup_bal_bef_ret
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.354**     ***surv\_pup\_cnt***

Description:	Propn of paid-up policies at end of t - on count exposure
Help:	
Modified On:	1/17/2023 10:31:42 AM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Decrements
Column Header:	surv_pup_cnt
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.355**     ***surv\_pup\_cnt\_bef\_ret***

Description:	Propn of paid-up policies at end of t - before retirement - on count exposure
Help:	
Modified On:	1/12/2023 12:26:10 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Decrements
Column Header:	surv_pup_cnt_bef_ret
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.356**     ***surv\_pup\_post\_ret***

Description:	Propn of paid-up policies at end of t
Help:	
Modified On:	8/29/2021 1:18:56 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Decrements
Column Header:	surv_pup_post_ret
Combine Groups By:	Sum Both

Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.357**     *surv\_pup\_prm*

Description:	Propn of paid-up policies at end of t - on premium exposure
Help:	
Modified On:	1/11/2023 7:02:58 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Decrements
Column Header:	surv_pup_prm
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.358**     *surv\_pup\_prm\_bef\_ret*

Description:	Propn of paid-up policies at end of t - before retirement - on premium exposure
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Decrements
Column Header:	surv_pup_prm_bef_ret
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False

Virtual: False

#### **5.3.3.2.359**      *death\_rate*

Description: Independent monhtly death rate  
Help:  
Modified On: 12/11/2023 2:43:58 PM (UTC+02:00)  
Modified By: CLAL-INS\Arikt  
Category: Decrements|Death  
Column Header: death\_rate  
Combine Groups By: Average Both  
Combine Periods: Last  
Default sliding Size: -1  
Discount Timing: Middle  
Discount Use: No  
Rate Use: Cash Flow  
Rebase Type: Current  
Retain Value: Yes  
Override: False  
Virtual: False

#### **5.3.3.2.360**      *lapse\_factor*

Description: factor applied to alapse rates, by aggent and year  
Help:  
Modified On: 12/14/2022 4:52:26 PM (UTC+02:00)  
Modified By: CLAL-INS\ahuvaa  
Category: Decrements|Lapse  
Column Header: lapse\_factor  
Combine Groups By: Average Both  
Combine Periods: Last  
Default sliding Size: -1  
Discount Timing: End  
Discount Use: No  
Rate Use: Cash Flow  
Rebase Type: Current  
Retain Value: Yes  
Override: False  
Virtual: False

#### **5.3.3.2.361**      *lapse\_rate\_act\_bal*

Description: Monthly Lapse rate in period - for active policy - with balance exposure  
Help:  
Modified On: 12/19/2024 4:04:12 PM (UTC+02:00)  
Modified By: CLAL-INS\arikt

Category:	Decrements Lapse
Column Header:	lapse_rate_act_bal
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

### 5.3.3.2.362 *lapse\_rate\_act\_bal\_dep*

Description:	Dependent lapse rate - for active policies - on balance exposure
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\arikt
Category:	Decrements Lapse
Column Header:	lapse_rate_act_bal_dep
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

### 5.3.3.2.363 *lapse\_rate\_act\_cnt*

Description:	Monthly Lapse rate in period - for active policy - with count exposure
Help:	
Modified On:	12/19/2024 4:02:01 PM (UTC+02:00)
Modified By:	CLAL-INS\arikt
Category:	Decrements Lapse
Column Header:	lapse_rate_act_cnt
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow

Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.364**      ***`lapse_rate_act_cnt_dep`***

Description:	Dependent lapse rate - for active policies - on count exposure
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\arikt
Category:	Decrements\Lapse
Column Header:	<code>lapse_rate_act_cnt_dep</code>
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.365**      ***`lapse_rate_act_prm`***

Description:	Monthly Lapse rate in period - for active policy - with premium exposure
Help:	
Modified On:	12/19/2024 4:03:29 PM (UTC+02:00)
Modified By:	CLAL-INS\arikt
Category:	Decrements\Lapse
Column Header:	<code>lapse_rate_act_prm</code>
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.366**      ***lapse\_rate\_act\_prm\_dep***

Description:	Dependent lapse rate - for active policies - on premium exposure
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Decrements Lapse
Column Header:	lapse_rate_act_prm_dep
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.367**      ***lapse\_rate\_pup\_bal***

Description:	Lapse rate in period for pup policies - with balance exposure
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Decrements Lapse
Column Header:	lapse_rate_pup_bal
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	0
Discount Timing:	Middle
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.368**      ***lapse\_rate\_pup\_bal\_dep***

Description:	Dependent lapse rate for pups - on balance exposure
Help:	
Modified On:	1/11/2023 6:54:51 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Decrements Lapse
Column Header:	lapse_rate_pup_bal_dep



Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	0
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.369**      ***`lapse_rate_pup_cnt`***

Description:	Lapse rate in period for pup policies - with count exposure
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Decrements\Lapse
Column Header:	<code>lapse_rate_pup_cnt</code>
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	0
Discount Timing:	Middle
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.370**      ***`lapse_rate_pup_cnt_dep`***

Description:	Dependent lapse rate for pups - on count exposure
Help:	
Modified On:	1/11/2023 6:54:42 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Decrements\Lapse
Column Header:	<code>lapse_rate_pup_cnt_dep</code>
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	0
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes

Override: False  
Virtual: False

#### **5.3.3.2.371**      ***`lapse_rate_pup_prm`***

Description: Lapse rate in period for pup policies - with premium exposure

Help:

Modified On: 12/11/2023 2:43:58 PM (UTC+02:00)

Modified By: CLAL-INS\Arikt

Category: Decrements\Lapse

Column Header: `lapse_rate_pup_prm`

Combine Groups By: Average Both

Combine Periods: Last

Default sliding Size: 0

Discount Timing: Middle

Discount Use: No

Rate Use: Cash Flow

Rebase Type: Current

Retain Value: Yes

Override: False

Virtual: False

#### **5.3.3.2.372**      ***`lapse_total_bal`***

Description: total lapses including surrenders and paid-up - monthly rates - on balance exposure

Help:

Modified On: 12/11/2023 2:43:58 PM (UTC+02:00)

Modified By: CLAL-INS\Arikt

Category: Decrements\Lapse

Column Header: `lapse_total_bal`

Combine Groups By: Average Both

Combine Periods: Last

Default sliding Size: -1

Discount Timing: End

Discount Use: No

Rate Use: Cash Flow

Rebase Type: Current

Retain Value: Yes

Override: False

Virtual: False

#### **5.3.3.2.373**      ***`lapse_total_prm`***

Description: total lapses including surrenders and paid-up - monthly rates - on premium exposure

Help:

Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Decrements Lapse
Column Header:	lapse_total_prm
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

### 5.3.3.2.374 *pup\_rate\_bal*

Description:	Premium cessation rate in period (monthly) - for balance exposure
Help:	
Modified On:	5/11/2025 5:03:46 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Decrements Pups
Column Header:	pup_rate_bal
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

### 5.3.3.2.375 *pup\_rate\_bal\_dep*

Description:	Dependent pup rate - on balance exposure
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Decrements Pups
Column Header:	pup_rate_bal_dep
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No

Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.376**     *pup\_rate\_cnt*

Description:	Premium cessation rate in period (monthly) - for count exposure
Help:	
Modified On:	5/11/2025 5:04:03 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Decrements Pups
Column Header:	pup_rate_cnt
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.377**     *pup\_rate\_cnt\_dep*

Description:	Dependent pup rate - on count exposure
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Decrements Pups
Column Header:	pup_rate_cnt_dep
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.378**     ***pup\_rate\_prm***

Description:	Premium cessation rate in period (monthly) - for premium exposure
Help:	
Modified On:	5/11/2025 5:04:09 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Decrements Pups
Column Header:	pup_rate_prm
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.379**     ***pup\_rate\_prm\_dep***

Description:	Dependent pup rate - on premium exposure
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Decrements Pups
Column Header:	pup_rate_prm_dep
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.380**     ***expense\_investment***

Description:	Expense Investment
Help:	
Modified On:	4/4/2023 9:50:40 AM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Expenses
Column Header:	expense_investment
Combine Groups By:	Sum Both

Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.381      *expense\_investment\_bef\_ret***

Description:	Expense Investment before retirement
Help:	
Modified On:	9/18/2022 9:39:13 AM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Expenses
Column Header:	expense_investment_bef_ret
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.382      *expense\_investment\_post\_ret***

Description:	Expense Investment after retirement
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Expenses
Column Header:	expense_investment_post_ret
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.383 expense\_investment\_pv**

Description:	Expense Investment Pv
Help:	
Modified On:	9/18/2022 10:07:13 AM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Expenses
Column Header:	expense_investment_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.384 expense\_investment\_pv\_bef\_ret**

Description:	Expense Investment Pv before retirement
Help:	
Modified On:	9/14/2022 1:50:24 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Expenses
Column Header:	expense_investment_pv_bef_ret
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.385 expense\_investment\_pv\_post\_ret**

Description:	Expense Investment Pv after retirement
Help:	
Modified On:	9/29/2022 2:35:42 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Expenses
Column Header:	expense_investment_pv_post_ret
Combine Groups By:	Sum Both
Combine Periods:	Last

Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.386      *expense\_ren\_charge\_pv***

Description:	Expense Ren Charge Pv
Help:	
Modified On:	10/26/2021 10:54:55 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Expenses
Column Header:	expense_ren_charge_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.387      *benefits\_b\_prm***

Description:	No of in force benefits (begin of month) - on premium exposure
Help:	
Modified On:	1/11/2023 7:21:58 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Inforce Details Number of Covers
Column Header:	benefits_b_prm
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False



**5.3.3.2.388**     ***policies\_b***

Description:	No of in force policies (beg of month)
Help:	
Modified On:	1/11/2023 8:02:55 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Inforce Details Number of Covers
Column Header:	policies_b
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.389**     ***policies\_pup\_b***

Description:	No of in force pup policies (beg of mont
Help:	
Modified On:	1/11/2023 8:03:52 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Inforce Details Number of Covers
Column Header:	policies_pup_b
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	0
Discount Timing:	Beginning
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.390**     ***policy\_deaths***

Description:	No of policies expiring due to death
Help:	
Modified On:	1/11/2023 7:02:58 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Inforce Details Number of Covers
Column Header:	policy_deaths
Combine Groups By:	Sum Both
Combine Periods:	Sum

Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

### 5.3.3.2.391 *policy\_surr*

Description:	No of policies expiring due to surrender
Help:	
Modified On:	1/11/2023 8:04:16 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Inforce Details Number of Covers
Column Header:	policy_surr
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

### 5.3.3.2.392 *claims\_retent*

Description:	Claims - retention
Help:	
Modified On:	6/10/2024 7:01:50 PM (UTC+03:00)
Modified By:	CLAL-INS\Arikt
Category:	Inforce Details Sum Assured
Column Header:	claims_retent
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.393      *sum\_at\_risk\_claim***

Description:	Sum at risk for death claims (non-PUPs)
Help:	
Modified On:	1/11/2023 6:58:10 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Inforce Details Sum Assured
Column Header:	sum_at_risk_claim
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.394      *sum\_insured***

Description:	Sum insured - undecrement
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Inforce Details Sum Assured
Column Header:	sum_insured
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.395      *sum\_insured\_if\_e***

Description:	Sum insured in force (end of month)
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Inforce Details Sum Assured
Column Header:	sum_insured_if_e
Combine Groups By:	Sum Both
Combine Periods:	Last

Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.396**      ***sum\_insured\_occ\_gross***

Description:	Sum insured - including occupation loading
Help:	
Modified On:	4/11/2024 2:50:13 PM (UTC+03:00)
Modified By:	CLAL-INS\Arikt
Category:	Inforce Details Sum Assured
Column Header:	sum_insured_occ_gross
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.397**      ***sum\_insured\_occ\_retent***

Description:	Sum insured - including occupation loading - retention
Help:	
Modified On:	4/11/2024 2:53:08 PM (UTC+03:00)
Modified By:	CLAL-INS\Arikt
Category:	Inforce Details Sum Assured
Column Header:	sum_insured_occ_retent
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.398**     *alloc\_units*

Description:	Premium allocation to units
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Inforce Details Unit Fund
Column Header:	alloc_units
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.399**     *interest\_units\_e*

Description:	Interest credited to units
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Inforce Details Unit Fund
Column Header:	interest_units_e
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.400**     *units\_b*

Description:	Unit account after allocation (beg)
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Inforce Details Unit Fund
Column Header:	units_b
Combine Groups By:	Sum Both
Combine Periods:	Last

Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.401      *units\_b\_active***

Description:	Units at beginning of period for active policies
Help:	
Modified On:	4/30/2020 3:32:46 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Category:	Inforce Details Unit Fund
Column Header:	units_b_active
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.402      *units\_b\_bef***

Description:	Unit account before allocation (beg)
Help:	
Modified On:	11/17/2022 4:46:44 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Inforce Details Unit Fund
Column Header:	units_b_bef
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.403      *units\_b\_bef\_pup\_acc***

Description:	Accum units for new PUPs
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Inforce Details Unit Fund
Column Header:	units_b_bef_pup_acc
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.404      *units\_b\_bef\_pup\_sav***

Description:	Accum saving units for new PUPs
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Inforce Details Unit Fund
Column Header:	units_b_bef_pup_sav
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.405      *units\_b\_pup***

Description:	Units at beginning of period for paid-up policies
Help:	
Modified On:	4/30/2020 3:33:39 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Category:	Inforce Details Unit Fund
Column Header:	units_b_pup
Combine Groups By:	Sum Both
Combine Periods:	Last

Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.406**      *units\_bon*

Description:	Extra units from persistency bonus
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Inforce Details Unit Fund
Column Header:	units_bon
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.407**      *units\_e*

Description:	Unit account after fees (end)
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Inforce Details Unit Fund
Column Header:	units_e
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False



**5.3.3.2.408      *units\_e\_bef***

Description:	Unit account before fees (end)
Help:	
Modified On:	11/17/2022 4:47:05 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Inforce Details Unit Fund
Column Header:	units_e_bef
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.409      *units\_e\_hon***

Description:	HON Money (post 2000) Unit account after fees (end)
Help:	Accumulation of hon money (to calculate annuity deficiency reserve and cost). Assumes that 1) new premium goes to kizba money, 2) surrenders are taken proportionately from hon and kizba money
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Inforce Details Unit Fund
Column Header:	units_e_hon
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.410      *units\_e\_hon\_active***

Description:	Active HON Money (post 2000) Unit account after fees (end)
--------------	--

Help:	Accumulation of hon money (to calculate annuity deficiency reserve and cost). Assumes that 1) new premium goes to kizba money, 2) surrenders are taken proportionately from hon and kizba money
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Inforce Details Unit Fund
Column Header:	units_e_hon_active
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

### 5.3.3.2.411 *units\_e\_hon\_pup*

Description:	Paid-up Money (post 2000) Unit account after fees (end)
Help:	Accumulation of hon money (to calculate annuity deficiency reserve and cost). Assumes that 1) new premium goes to kizba money, 2) surrenders are taken proportionately from hon and kizba money
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Inforce Details Unit Fund
Column Header:	units_e_hon_pup
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.412      *units\_e\_kiz***

Description:	Kitzbati Money (post 2000) Unit account after fees (end)
Help:	Accumulation of hon money (to calculate annuity deficiency reserve and cost). Assumes that 1) new premium goes to kizba money, 2) surrenders are taken proportionately from hon and kizba money
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Inforce Details Unit Fund
Column Header:	units_e_kiz
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.413      *units\_e\_new***

Description:	New Money (post 2000) Unit account after fees (end)
Help:	Accumulation of old money (to calculate annuity deficiency reserve and cost). Assumes that 1) new premium goes to new money, 2) surrenders are taken from new money
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Inforce Details Unit Fund
Column Header:	units_e_new
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.414      *units\_e\_newtag***

Description:	New Tagmulim Money (post 2000) Unit account after fees (end)
Help:	Accumulation of old money (to calculate annuity deficiency reserve and cost). Assumes that 1) new premium goes to new money, 2) surrenders are taken from new money
Modified On:	2/26/2025 12:32:12 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Inforce Details Unit Fund
Column Header:	units_e_newtag
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.415      *units\_e\_old***

Description:	Old Money (pre 2000) Unit account after fees (end)
Help:	Accumulation of old money (to calculate annuity deficiency reserve and cost). Assumes that 1) new premium goes to new money, 2) surrenders are taken from new money
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Inforce Details Unit Fund
Column Header:	units_e_old
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.416      *units\_e\_old\_active***

Description:	Active Old Money (pre 2000) Unit account after fees (end)
Help:	Accumulation of old money (to calculate annuity deficiency reserve and cost). Assumes that 1) new premium goes to new money, 2) surrenders are taken from new money
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Inforce Details Unit Fund
Column Header:	units_e_old_active
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.417      *units\_e\_old\_pup***

Description:	Paid-up Old Money (pre 2000) Unit account after fees (end)
Help:	Accumulation of old money (to calculate annuity deficiency reserve and cost). Assumes that 1) new premium goes to new money, 2) surrenders are taken from new money
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Inforce Details Unit Fund
Column Header:	units_e_old_pup
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.418 units\_e\_piz**

Description:	Pizuim Money (post 2000) Unit account after fees (end)
Help:	Accumulation of old money (to calculate annuity deficiency reserve and cost). Assumes that 1) new premium goes to new money, 2) surrenders are taken from new money
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Inforce Details Unit Fund
Column Header:	units_e_piz
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.419 units\_e\_piz\_active**

Description:	New Pizuim Money (active) (post 2000) Unit account after fees (end)
Help:	Accumulation of old money (to calculate annuity deficiency reserve and cost). Assumes that 1) new premium goes to new money, 2) surrenders are taken from new money
Modified On:	2/26/2025 12:33:29 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Inforce Details Unit Fund
Column Header:	units_e_piz_active
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.420      *units\_e\_piz\_newprems***

Description:	New Pizuim Money (new prems) (post 2000) Unit account after fees (end)
Help:	Accumulation of old money (to calculate annuity deficiency reserve and cost). Assumes that 1) new premium goes to new money, 2) surrenders are taken from new money
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Inforce Details Unit Fund
Column Header:	units_e_piz_newprems
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.421      *units\_e\_piz\_pup***

Description:	New Pizuim Money (pup) (post 2000) Unit account after fees (end)
Help:	Accumulation of old money (to calculate annuity deficiency reserve and cost). Assumes that 1) new premium goes to new money, 2) surrenders are taken from new money
Modified On:	2/26/2025 12:34:21 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Inforce Details Unit Fund
Column Header:	units_e_piz_pup
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.422**     ***units\_e\_prat***

Description:	Prat Money (post 2000) Unit account after fees (end)
Help:	Accumulation of old money (to calculate annuity deficiency reserve and cost). Assumes that 1) new premium goes to new money, 2) surrenders are taken from new money
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Inforce Details Unit Fund
Column Header:	units_e_prat
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.423**     ***units\_e\_prat\_active***

Description:	New Prat Money (active) (post 2000) Unit account after fees (end)
Help:	Accumulation of old money (to calculate annuity deficiency reserve and cost). Assumes that 1) new premium goes to new money, 2) surrenders are taken from new money
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Inforce Details Unit Fund
Column Header:	units_e_prat_active
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False



**5.3.3.2.424      *units\_e\_prat\_newprems***

Description:	New Prat Money (new prems) (post 2000) Unit account after fees (end)
Help:	Accumulation of old money (to calculate annuity deficiency reserve and cost). Assumes that 1) new premium goes to new money, 2) surrenders are taken from new money
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Inforce Details Unit Fund
Column Header:	units_e_prat_newprems
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.425      *units\_e\_prat\_pup***

Description:	New Prat Money (pup) (post 2000) Unit account after fees (end)
Help:	Accumulation of old money (to calculate annuity deficiency reserve and cost). Assumes that 1) new premium goes to new money, 2) surrenders are taken from new money
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Inforce Details Unit Fund
Column Header:	units_e_prat_pup
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.426      *prem\_termination\_prop***

Description:	proportion stopping to pay premium
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Lapses
Column Header:	prem_termination_prop
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.427      *mort\_year***

Description:	Mort Year for key for mort tables
Help:	
Modified On:	7/26/2021 2:41:41 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Mortality
Column Header:	mort_year
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.428      *basic\_perc***

Description:	Proportion of premium for basic Adif (be
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Policy Details
Column Header:	basic_perc
Combine Groups By:	Sum Both
Combine Periods:	Last

Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.429**     *age\_last*

Description:	Age last birthday
Help:	
Modified On:	11/30/2023 5:57:09 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Policy Details Insured Details
Column Header:	age_last
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.430**     *interest\_re\_lrc\_q1*

Description:	interest re for q1 event
Help:	
Modified On:	12/2/2024 3:07:28 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Policy Details Insured Details
Column Header:	interest_re_lrc_q1
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.431**     *interest\_re\_lrc\_q2*

Description:	interest re for q2 event
Help:	
Modified On:	12/2/2024 3:07:30 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Policy Details Insured Details
Column Header:	interest_re_lrc_q2
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.432**     *interest\_re\_lrc\_q3*

Description:	interest re for q3 event
Help:	
Modified On:	12/2/2024 3:07:15 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Policy Details Insured Details
Column Header:	interest_re_lrc_q3
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.433**     *interest\_re\_lrc\_q4*

Description:	interest re for q4 event
Help:	
Modified On:	12/2/2024 3:07:38 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Policy Details Insured Details
Column Header:	interest_re_lrc_q4
Combine Groups By:	Sum Both
Combine Periods:	Sum

Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.434**     *interest\_re\_lrc\_yr2plus*

Description:	interest re from second year event
Help:	
Modified On:	12/2/2024 3:07:58 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Policy Details Insured Details
Column Header:	interest_re_lrc_yr2plus
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.435**     *riskadj\_gross*

Description:	calc of risk adj. as sum of all scenarios
Help:	
Modified On:	9/10/2024 7:36:17 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Policy Details Insured Details
Column Header:	riskadj_gross
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.436**     *riskadj\_gross\_rel\_q1*

Description:	riskadj gross for q1 event
Help:	
Modified On:	12/2/2024 10:37:52 AM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Policy Details Insured Details
Column Header:	riskadj_gross_rel_q1
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.437**     *riskadj\_gross\_rel\_q2*

Description:	riskadj gross for q2 event
Help:	
Modified On:	12/2/2024 2:28:39 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Policy Details Insured Details
Column Header:	riskadj_gross_rel_q2
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.438**     *riskadj\_gross\_rel\_q3*

Description:	riskadj gross for q3 event
Help:	
Modified On:	12/2/2024 2:43:37 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Policy Details Insured Details
Column Header:	riskadj_gross_rel_q3
Combine Groups By:	Sum Both
Combine Periods:	Sum

Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.439**     ***riskadj\_gross\_rel\_q4***

Description:	riskadj gross for q4 event
Help:	
Modified On:	12/2/2024 2:43:46 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Policy Details Insured Details
Column Header:	riskadj_gross_rel_q4
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.440**     ***riskadj\_gross\_rel\_total***

Description:	riskadj release - non discounted
Help:	
Modified On:	8/6/2024 3:57:48 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Policy Details Insured Details
Column Header:	riskadj_gross_rel_total
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.441**     *riskadj\_gross\_rel\_yr2plus*

Description:	riskadj gross from second year event
Help:	
Modified On:	12/2/2024 2:36:39 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Policy Details Insured Details
Column Header:	riskadj_gross_rel_yr2plus
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.442**     *riskadj\_net*

Description:	calc of risk adj. as sum of all scenarios
Help:	
Modified On:	3/17/2024 9:56:08 PM (UTC+02:00)
Modified By:	CLAL-INS\arikt
Category:	Policy Details Insured Details
Column Header:	riskadj_net
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.443**     *riskadj\_re*

Description:	calc of risk adj. as sum of all scenarios
Help:	
Modified On:	9/10/2024 7:32:00 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Policy Details Insured Details
Column Header:	riskadj_re
Combine Groups By:	Sum Both
Combine Periods:	Last



Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.444**     *riskadj\_re\_rel\_q1*

Description:	riskadj re for q1 event
Help:	
Modified On:	12/2/2024 2:39:26 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Policy Details Insured Details
Column Header:	riskadj_re_rel_q1
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.445**     *riskadj\_re\_rel\_q2*

Description:	riskadj net for q2 event
Help:	
Modified On:	12/2/2024 2:39:59 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Policy Details Insured Details
Column Header:	riskadj_re_rel_q2
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.446**     *riskadj\_re\_rel\_q3*

Description:	riskadj re for q3 event
Help:	
Modified On:	12/2/2024 2:40:16 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Policy Details Insured Details
Column Header:	riskadj_re_rel_q3
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.447**     *riskadj\_re\_rel\_q4*

Description:	riskadj re for q4 event
Help:	
Modified On:	12/2/2024 2:40:39 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Policy Details Insured Details
Column Header:	riskadj_re_rel_q4
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.448**     *riskadj\_re\_rel\_total*

Description:	riskadj release re - non discounted
Help:	
Modified On:	8/6/2024 3:58:54 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Policy Details Insured Details
Column Header:	riskadj_re_rel_total
Combine Groups By:	Sum Both
Combine Periods:	Sum

Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.449**     *riskadj\_re\_rel\_yr2plus*

Description:	riskadj re from second year event
Help:	
Modified On:	12/2/2024 2:41:05 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Policy Details Insured Details
Column Header:	riskadj_re_rel_yr2plus
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.450**     *premium\_if\_b*

Description:	Premium I.F. at start of month excluding premium for risk rider
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Policy Details Premium
Column Header:	premium_if_b
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.451     *premium\_if\_b\_total***

Description:	Total premium in force at start of period
Help:	
Modified On:	1/11/2023 7:17:40 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Policy Details Premium
Column Header:	premium_if
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.452     *premium\_if\_e***

Description:	Premium in force at end of period
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Policy Details Premium
Column Header:	premium_if_e
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.453     *premium\_if\_riders***

Description:	Premium in force at start of period for risk rider
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Policy Details Premium
Column Header:	premium_if_riders
Combine Groups By:	Sum Both
Combine Periods:	Sum

Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.454      *alloc\_units\_honi***

Description:	Alloc Units Honi
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Premium
Column Header:	alloc_units_honi
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.455      *alloc\_units\_newtag***

Description:	Alloc Units Newtag
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Premium
Column Header:	alloc_units_newtag
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.456      *alloc\_units\_piz***

Description:	Alloc Units Pizuim
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Premium
Column Header:	alloc_units_piz
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.457      *alloc\_units\_prat***

Description:	Alloc Units Prat
Help:	
Modified On:	2/29/2024 10:02:43 AM (UTC+02:00)
Modified By:	CLAL-INS\yonis
Category:	Premium
Column Header:	alloc_units_prat
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.458      *premium\_inc***

Description:	premium increase - based on sal_inc table
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Premium
Column Header:	premium_inc
Combine Groups By:	Average Both
Combine Periods:	Last

Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.459**      ***sum\_ins\_inc***

Description:	Sum insured increase - based on sal_inc table
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Premium
Column Header:	sum_ins_inc
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.460**      ***sum\_ins\_inc\_acc***

Description:	Cummulative Sum insured increase - based on sal_inc table
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Premium
Column Header:	sum_ins_inc_acc
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.461      *prem\_disc\_shimur\_rate***

Description:	Calculates shimur rate
Help:	
Modified On:	3/11/2025 9:30:26 AM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Premium discounts
Column Header:	prem_disc_shimur_rate
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.462      *be\_retire***

Description:	PV of be reserve at retirement..
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Profitability Measures
Column Header:	ber_retire
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.463      *cashflow\_gross\_pv\_pos***

Description:	PV of cashflows - gross - only if positive
Help:	
Modified On:	3/26/2023 2:15:06 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Profitability Measures
Column Header:	cashflow_gross_pv_pos
Combine Groups By:	Sum Both
Combine Periods:	Sum



Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.464**      *cashflow\_pv*

Description:	PV of cashflows.
Help:	
Modified On:	3/22/2023 2:24:59 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Profitability Measures
Column Header:	cashflow_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.465**      *cashflow\_pv\_chetz*

Description:	PV of cashflows discounted using chetz rates
Help:	
Modified On:	7/14/2024 2:07:43 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Profitability Measures
Column Header:	cashflow_pv_chetz
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.466**      ***cashflow\_pv\_e***

Description:	PV of cashflows with all components being discounted EOP
Help:	
Modified On:	9/22/2022 11:31:28 AM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Profitability Measures
Column Header:	cashflow_pv_e
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.467**      ***cashflow\_pv\_pos***

Description:	PV of cashflows - only if positive
Help:	
Modified On:	3/26/2023 2:15:22 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Profitability Measures
Column Header:	cashflow_pv_pos
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.468**      ***profit\_bk\_act\_vif\_pv***

Description:	PV of Book Profit (end of month) VIF basis of active period
Help:	
Modified On:	10/26/2021 11:11:44 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Profitability Measures
Column Header:	profit_bk_act_vif_pv
Combine Groups By:	Sum Both

Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.469**     ***profit\_book\_vif\_pv***

Description:	PV of Book Profit (end of month) VIF basis
Help:	
Modified On:	10/26/2021 11:13:10 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Profitability Measures
Column Header:	profit_book_vif_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.470**     ***profit\_net\_vif\_pv***

Description:	VIF - PV After Tax Profit on cashflow basis (end of month)
Help:	
Modified On:	8/12/2024 12:43:35 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Profitability Measures
Column Header:	profit_net_vif_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.471**     *ret\_prop\_col*

Description:	Ret Prop Col
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Profitability Measures
Column Header:	ret_prop_col
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.472**     *cashflow\_re\_b*

Description:	Cashflow Re B
Help:	
Modified On:	8/6/2024 6:52:59 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Reinsurance
Column Header:	cashflow_re_b
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.473**     *cashflow\_re\_e*

Description:	Cashflow Re E
Help:	
Modified On:	9/12/2024 11:14:51 AM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Reinsurance
Column Header:	cashflow_re_e
Combine Groups By:	Sum Both

Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.474**      ***claims\_re***

Description:	Reinsurance Claims Paid
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Reinsurance
Column Header:	claims_re
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	0
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.475**      ***comm\_re***

Description:	Initial and renewal reins commission
Help:	
Modified On:	1/5/2025 3:17:00 PM (UTC+02:00)
Modified By:	CLAL-INS\arikt
Category:	Reinsurance
Column Header:	comm_re
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	0
Discount Timing:	Beginning
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.476      *comm\_re\_prof***

Description:	Reinsurance profit commission
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Reinsurance
Column Header:	comm_re_prof
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	0
Discount Timing:	Beginning
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.477      *interest\_re***

Description:	Interest on reinsurance reserve
Help:	
Modified On:	9/12/2024 11:35:50 AM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Reinsurance
Column Header:	interest_re
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	0
Discount Timing:	Beginning
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.478      *interest\_re\_pv***

Description:	PV of Interest on reinsurance reserve
Help:	
Modified On:	10/26/2021 10:55:17 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Reinsurance
Column Header:	interest_re_pv
Combine Groups By:	Sum Both
Combine Periods:	Last

Default sliding Size:	0
Discount Timing:	Beginning
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.479**     ***premium\_re***

Description:	Reins Premium income at begin. of period
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Reinsurance
Column Header:	premium_re
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	0
Discount Timing:	Beginning
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.480**     ***rein\_claims\_pv***

Description:	PV of reinsurance claims
Help:	
Modified On:	9/12/2019 4:10:18 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Category:	Reinsurance
Column Header:	rein_claims_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.481      *rein\_comm\_pv***

Description:	PV of reinsurance commission
Help:	
Modified On:	9/12/2019 4:10:37 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Category:	Reinsurance
Column Header:	rein_comm_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.482      *rein\_prem\_pv***

Description:	PV of reinsurance premium
Help:	
Modified On:	9/12/2019 4:10:57 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Category:	Reinsurance
Column Header:	rein_prem_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.483      *reserve\_re***

Description:	Reinsurance reserves
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Reinsurance
Column Header:	reserve_re
Combine Groups By:	Sum Both
Combine Periods:	Last



Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.484      *reserve\_re\_increase***

Description:	Increase in reinsurance reserve
Help:	
Modified On:	9/12/2024 11:37:03 AM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Reinsurance
Column Header:	reserve_re_increase
Combine Groups By:	Sum Both
Combine Periods:	Sum
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.485      *reserve\_re\_increase\_pv***

Description:	PV of Increase in reinsurance reserve
Help:	
Modified On:	8/6/2024 7:43:57 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Reinsurance
Column Header:	reserve_re_increase_pv
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.486      *startup***

Description:	Startup
Help:	
Modified On:	8/12/2024 11:24:19 AM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Setup
Column Header:	startup
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.487      *cal\_duration***

Description:	Calendar duration (in years) of policy
Help:	
Modified On:	1/5/2025 3:26:30 PM (UTC+02:00)
Modified By:	CLAL-INS\arikt
Category:	Timing
Column Header:	cal_duration
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.488      *cal\_month***

Description:	Month of the calendar year
Help:	
Modified On:	8/15/2021 2:47:28 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Timing
Column Header:	cal_month
Combine Groups By:	Average Both
Combine Periods:	Last

Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.489**     *cal\_year*

Description:	Calendar year
Help:	
Modified On:	10/4/2021 3:33:57 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Timing
Column Header:	cal_year
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.490**     *pol\_month*

Description:	Month of the policy year
Help:	
Modified On:	10/3/2021 1:47:05 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Timing
Column Header:	pol_month
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.491**     *pol\_year*

Description:	Policy year
Help:	
Modified On:	10/3/2021 1:47:12 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Timing
Column Header:	pol_year
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.492**     *pol\_year\_ext*

Description:	Policy year + extra months
Help:	
Modified On:	6/9/2022 10:47:30 AM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Timing
Column Header:	pol_year_ext
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.3.2.493**     *proj\_month*

Description:	Projection month
Help:	
Modified On:	8/15/2021 9:07:35 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Timing
Column Header:	proj_month
Combine Groups By:	Average Both
Combine Periods:	Last

Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.494**     *proj\_year*

Description:	Projection year
Help:	
Modified On:	8/12/2021 5:12:37 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Timing
Column Header:	proj_year
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.3.2.495**     *proj\_year\_rollup*

Description:	Projection year for rollup (0 throughout the rollup period, starting at 1 thereafter)
Help:	Calender year index used for referencing arrays according to calender year (inv_rate, expense_multipliers ...)
Modified On:	11/17/2022 4:45:08 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Timing
Column Header:	proj_year_rollup
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False

Virtual: False

### 5.3.3.3 External Functions

#### 5.3.3.3.1 *call\_extra\_scalars*

Description: Call Extra Scalars to be populated in output  
Help:  
Modified On: 12/14/2023 3:01:28 PM (UTC+02:00)  
Modified By: CLAL-INS\Arikt  
Category:

#### 5.3.3.3.2 *monthly\_rate*

Description: Convert annual to monthly  
Help: Given an annual interest rate, expressed as i% this external formula returns  $i(12)/12\%$ , i.e. the monthly rate which compounds to give the annual rate, eg  $\text{monthly\_rate}(10)=0.007974$   
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Setup

#### 5.3.3.3.3 *set\_accum\_fund*

Description: Set accumulation fund var for basic unit  
Help: This external function sets the variables for accumulation fund of basic part of units  
Modified On: 6/2/2022 9:58:50 AM (UTC+03:00)  
Modified By: CLAL-INS\ahuvaa  
Category: Setup

#### 5.3.3.3.4 *set\_accum\_pup\_fund*

Description: Set accum fund vars for PUP (basic unit)  
Help: This external function sets the variables for accumulation fund for paid-up policies. (basic units)  
Modified On: 6/2/2022 9:59:22 AM (UTC+03:00)  
Modified By: CLAL-INS\ahuvaa  
Category: Setup

#### 5.3.3.3.5 *set\_by\_prodcode*

Description: Set variables based on PRODCODE

Help: External function to reset certain variables, with values from tables, before running projection. It is for variables that depend on the product (referred to by "prodcode"), such as product-specifications. These variables from the default or screen are over-written by values coming from tables, if the variable "lookup\_by\_product" is set to "Y". This function is called from startup.

Modified On: 6/17/2025 10:39:55 AM (UTC+03:00)

Modified By: CLAL-INS\arikt

Category: Setup

#### **5.3.3.3.6 set\_from\_data**

Description: Set assumptions from data file

Help: External function to reset certain assumptions, with values from tables, before running projection. Assumptions from the default or screen are over-written by values coming from assumption tables, if the variable "read\_from\_tables" is set to "Y". This function is called from startup.

Modified On: 3/12/2025 1:40:08 PM (UTC+02:00)

Modified By: CLAL-INS\ahuvaa

Category: Setup

#### **5.3.3.3.7 set\_from\_tables**

Description: Setup variables [read] from tables

Help: External function to reset certain assumptions, with values from tables, before running projection. Assumptions from the default or screen are over-written by values coming from assumption tables, if the variable "read\_from\_tables" is set to "Y". This function is called from startup.

Modified On: 4/25/2023 10:41:49 AM (UTC+03:00)

Modified By: CLAL-INS\ahuvaa

Category: Setup

#### **5.3.3.3.8 set\_other\_variables**

Description: Setup other variables

Help: External function to calculate and/or adjust variables before running projection. This function is called from startup.

Modified On: 7/7/2024 6:07:08 PM (UTC+03:00)

Modified By: CLAL-INS\arikt  
Category: Setup

#### **5.3.3.3.9 set\_profil\_rider\_variables**

Description: Setup variables for Profil riders  
Help: External function to calculate and/or adjust variables before running projection.  
This function is called from startup.  
Modified On: 12/24/2024 9:54:07 AM (UTC+02:00)  
Modified By: CLAL-INS\arikt  
Category: Setup

#### **5.3.3.3.10 set\_reinsurance**

Description: Set variables for reinsurance  
Help: This function sets the variables used for reinsurance.  
This function is called by startup.  
Modified On: 8/1/2024 11:20:33 AM (UTC+03:00)  
Modified By: CLAL-INS\arikt  
Category: Setup

#### **5.3.3.3.11 set\_saving**

Description: Set accumulation fund var (pur saving)  
Help: This external function sets the variables for accumulation fund of pure saving units  
Modified On: 12/27/2022 1:07:22 PM (UTC+02:00)  
Modified By: CLAL-INS\ahuvaa  
Category: Setup

#### **5.3.3.3.12 set\_saving\_pup**

Description: Set accum fund vars for PUP (p. saving)  
Help: This external function sets the variables for accumulation fund for paid-up policies. (pure saving units)  
Modified On: 6/2/2022 9:59:34 AM (UTC+03:00)  
Modified By: CLAL-INS\ahuvaa  
Category: Setup

#### **5.3.3.3.13 validate\_data**

Description: Validate projection parameters before calculations



Help:	This function carries out data validation checks and halts program execution if errors are detected. Is only used if do data validation is set to "Y".
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Setup

### 5.3.3.4 Temporary Tables

#### 5.3.3.4.1 *charge\_amount\_tt*

Description:	charges deducted for profil riders (income)
Help:	
Modified On:	2/18/2024 9:24:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	
Number of Rows:	1200
Number of Columns:	25
Calculation Methods:	Cell
Reset Variables:	
Resetting Type:	Reset Temporary Table for every model point and/or projection loop
Override:	False
Virtual:	False

#### 5.3.3.4.2 *charge\_rate\_tt*

Description:	annual charge rate for Profil riders
Help:	
Modified On:	7/20/2021 4:14:46 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	
Number of Rows:	100
Number of Columns:	25
Calculation Methods:	Cell
Reset Variables:	
Resetting Type:	Reset Temporary Table for every model point and/or projection loop
Override:	False
Virtual:	False

#### 5.3.3.4.3 *claim\_amount\_tt*

Description:	claims paid on profil riders
Help:	
Modified On:	7/21/2021 1:04:23 PM (UTC+03:00)

Modified By:	CLAL-INS\joshm
Category:	
Number of Rows:	1200
Number of Columns:	25
Calculation Methods:	Cell
Reset Variables:	
Resetting Type:	Reset Temporary Table for every model point and/or projection loop
Override:	False
Virtual:	False

#### **5.3.3.4.4 *claim\_cost\_tt***

Description:	claims cost factors for profil riders
Help:	
Modified On:	7/21/2021 1:39:02 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	
Number of Rows:	100
Number of Columns:	25
Calculation Methods:	Cell
Reset Variables:	
Resetting Type:	Reset Temporary Table for every model point and/or projection loop
Override:	False
Virtual:	False

#### **5.3.3.4.5 *claims\_mult\_tt***

Description:	claims multipliers for profil riders by policy year
Help:	
Modified On:	7/27/2021 11:04:59 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	
Number of Rows:	100
Number of Columns:	25
Calculation Methods:	Cell
Reset Variables:	
Resetting Type:	Reset Temporary Table for every model point and/or projection loop
Override:	False
Virtual:	False

#### **5.3.3.4.6 *decrement\_tt***

Description:	profil riders monthly decrement rates
Help:	
Modified On:	7/26/2021 1:48:33 PM (UTC+03:00)

Modified By:	CLAL-INS\joshm
Category:	
Number of Rows:	1200
Number of Columns:	25
Calculation Methods:	Cell
Reset Variables:	
Resetting Type:	Reset Temporary Table for every model point and/or projection loop
Override:	False
Virtual:	False

#### **5.3.3.4.7 *prem\_rates\_extra\_tt***

Description:	Adif Premium rates for extra risk.
Help:	
Modified On:	7/3/2022 7:42:02 AM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	
Number of Rows:	120
Number of Columns:	4
Calculation Methods:	Cell
Reset Variables:	
Resetting Type:	Reset Temporary Table for every model point and/or projection loop
Override:	False
Virtual:	False

#### **5.3.3.4.8 *sum\_ins\_basic\_tt***

Description:	Adif Basic Sum Insured amounts per 100 monthly premium
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	
Number of Rows:	120
Number of Columns:	4
Calculation Methods:	Cell
Reset Variables:	
Resetting Type:	Reset Temporary Table for every model point and/or projection loop
Override:	False
Virtual:	False

#### **5.3.3.4.9 *surr\_charge\_tt***

Description:	Surrender charge rates
Help:	

Modified On:	4/11/2024 4:43:36 PM (UTC+03:00)
Modified By:	CLAL-INS\Arikt
Category:	
Number of Rows:	1000
Number of Columns:	2
Calculation Methods:	Cell
Reset Variables:	
Resetting Type:	Reset Temporary Table for every model point and/or projection loop
Override:	False
Virtual:	False

#### **5.3.3.4.10**      ***claim\_rate\_tt***

Description:	claims rate for profil riders
Help:	
Modified On:	2/18/2024 9:24:59 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Decrements Death
Number of Rows:	100
Number of Columns:	25
Calculation Methods:	Cell
Reset Variables:	
Resetting Type:	Reset Temporary Table for every model point and/or projection loop
Override:	False
Virtual:	False

#### **5.3.3.4.11**      ***sum\_insured\_rider\_tt***

Description:	Sum Insured for Profil riders
Help:	
Modified On:	12/24/2024 9:53:22 AM (UTC+02:00)
Modified By:	CLAL-INS\arikt
Category:	Inforce Details Sum Assured
Number of Rows:	1200
Number of Columns:	25
Calculation Methods:	Cell
Reset Variables:	
Resetting Type:	Reset Temporary Table for every model point and/or projection loop
Override:	False
Virtual:	False

### **5.3.3.5 Scalars**

### 5.3.3.5.1 *fund\_type*

Description:	Fund Type
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	
Rebase:	N/A
Type:	Character
Override:	False
Virtual:	False

### 5.3.3.5.2 *min\_retirement\_age*

Description:	Minimum Retirement Age
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	
Rebase:	N/A
Type:	Integer
Override:	False
Virtual:	False

### 5.3.3.5.3 *mult\_age\_ind*

Description:	Indicator to use multiple retirement ages
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	
Rebase:	N/A
Type:	Integer
Override:	False
Virtual:	False

### 5.3.3.5.4 *use\_uw\_date*

Description:	Use UW Date
Help:	
Modified On:	3/19/2024 6:59:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	
Rebase:	N/A
Type:	Character
Override:	False
Virtual:	False

**5.3.3.5.5 profit\_net\_vif\_pv12**

Description:	VIF PV after tax profit at t=12
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Balance Sheet Reserves
Rebase:	N/A
Type:	Double
Override:	False
Virtual:	False

**5.3.3.5.6 res\_prop\_kitzba**

Description:	Proportion of Reserve for Kitzba (as opposed to Hon)
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Balance Sheet Reserves Annuity Deficiency
Rebase:	N/A
Type:	Double
Override:	False
Virtual:	False

**5.3.3.5.7 res\_prop\_kitzba\_newtag**

Description:	Proportion of Kizba Reserve which is New Tagmulim
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Balance Sheet Reserves Annuity Deficiency
Rebase:	N/A
Type:	Double
Override:	False
Virtual:	False

**5.3.3.5.8 res\_prop\_kitzba\_oldtag**

Description:	Proportion of Kizba Reserve which is Old Tagmulim
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Balance Sheet Reserves Annuity Deficiency
Rebase:	N/A
Type:	Double

Override: False  
Virtual: False

#### **5.3.3.5.9 *res\_prop\_kitzba\_piz***

Description: Proportion of Kizba Reserve which is Pizzuim  
Help:  
Modified On: 12/11/2023 2:43:58 PM (UTC+02:00)  
Modified By: CLAL-INS\Arikt  
Category: Balance Sheet|Reserves|Annuity Deficiency  
Rebase: N/A  
Type: Double  
Override: False  
Virtual: False

#### **5.3.3.5.10 *res\_prop\_kitzba\_prat***

Description: Proportion of Kizba Reserve which is Prat  
Help:  
Modified On: 12/11/2023 2:43:58 PM (UTC+02:00)  
Modified By: CLAL-INS\Arikt  
Category: Balance Sheet|Reserves|Annuity Deficiency  
Rebase: N/A  
Type: Double  
Override: False  
Virtual: False

#### **5.3.3.5.11 *res\_total\_increase1***

Description: Total reserve increase of the first month  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Balance Sheet|Reserves|Annuity Deficiency  
Rebase: N/A  
Type: Double  
Override: False  
Virtual: False

#### **5.3.3.5.12 *resanndef\_atmat***

Description: Annuity defeciciency reserve at maturity  
Help:  
Modified On: 12/11/2023 2:43:58 PM (UTC+02:00)  
Modified By: CLAL-INS\Arikt  
Category: Balance Sheet|Reserves|Annuity Deficiency  
Rebase: N/A

Type:	Double
Override:	False
Virtual:	False

#### **5.3.3.5.13**      ***reserve\_opening\_difference***

Description:	Opening difference between the input and calculated reserve. Used in the profit for month 1.
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Balance Sheet Reserves Annuity Deficiency
Rebase:	N/A
Type:	Double
Override:	False
Virtual:	False

#### **5.3.3.5.14**      ***premium\_disc\_pv\_start***

Description:	PV of premium discounts to policy start date
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Cashflows Income Premium
Rebase:	N/A
Type:	Double
Override:	False
Virtual:	False

#### **5.3.3.5.15**      ***premium\_nb\_sp***

Description:	Extra Single Premium for NB to get opening account balance
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Cashflows Income Premium
Rebase:	N/A
Type:	Double
Override:	False
Virtual:	False

#### **5.3.3.5.16**      ***premium\_pv\_st\_date***

Description:	PV of premiums to policy start date
Help:	



Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Cashflows Income Premium
Rebase:	N/A
Type:	Double
Override:	False
Virtual:	False

#### **5.3.3.5.17**      ***claims\_re\_yr1***

Description:	claims reinsurance in first year of projection
Help:	
Modified On:	1/10/2022 12:48:58 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows Outgo Claims
Rebase:	N/A
Type:	Double
Override:	False
Virtual:	False

#### **5.3.3.5.18**      ***claims\_total\_yr1***

Description:	claims total in first year of projection
Help:	
Modified On:	1/10/2022 12:51:00 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows Outgo Claims
Rebase:	N/A
Type:	Double
Override:	False
Virtual:	False

#### **5.3.3.5.19**      ***comm\_re\_prof\_yr1***

Description:	profit commission reinsurance in first year of projection
Help:	
Modified On:	1/10/2022 12:52:13 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows Outgo Claims
Rebase:	N/A
Type:	Double
Override:	False
Virtual:	False

**5.3.3.5.20      *comm\_re\_yr1***

Description: commission reinsurance in first year of projection  
Help:  
Modified On: 1/10/2022 12:50:32 PM (UTC+02:00)  
Modified By: CLAL-INS\joshm  
Category: Cashflows|Outgo|Claims  
Rebase: N/A  
Type: Double  
Override: False  
Virtual: False

**5.3.3.5.21      *comm\_total\_yr1***

Description: commission total in first year of projection  
Help:  
Modified On: 1/10/2022 12:50:52 PM (UTC+02:00)  
Modified By: CLAL-INS\joshm  
Category: Cashflows|Outgo|Claims  
Rebase: N/A  
Type: Double  
Override: False  
Virtual: False

**5.3.3.5.22      *expense\_total\_yr1***

Description: expense total in first year of projection  
Help:  
Modified On: 1/10/2022 12:45:48 PM (UTC+02:00)  
Modified By: CLAL-INS\joshm  
Category: Cashflows|Outgo|Claims  
Rebase: N/A  
Type: Double  
Override: False  
Virtual: False

**5.3.3.5.23      *prem\_discount\_py1***

Description: Premium discount for the first policy year  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Cashflows|Outgo|Claims  
Rebase: N/A  
Type: Double  
Override: False  
Virtual: False

**5.3.3.5.24      *prem\_discount\_py2***

Description:	Premium discount for the second policy year
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Cashflows Outgo Claims
Rebase:	N/A
Type:	Double
Override:	False
Virtual:	False

**5.3.3.5.25      *prem\_discount\_py3***

Description:	Premium discount for the third policy year
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Cashflows Outgo Claims
Rebase:	N/A
Type:	Double
Override:	False
Virtual:	False

**5.3.3.5.26      *premium\_gross\_yr1***

Description:	premium gross total in first year of projection
Help:	
Modified On:	1/10/2022 12:28:32 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows Outgo Claims
Rebase:	N/A
Type:	Double
Override:	False
Virtual:	False

**5.3.3.5.27      *premium\_re\_yr1***

Description:	premium reinsurance in first year of projection
Help:	
Modified On:	1/10/2022 12:49:44 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Cashflows Outgo Claims
Rebase:	N/A
Type:	Double
Override:	False

Virtual: False

#### **5.3.3.5.28**      *profit\_net\_vif\_yr0*

Description: Profit VIF in year 0 (current valuation year).  
Used as VNB actual profit component

Help:

Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)

Modified By: CLAL-INS\ninab

Category: Cashflows|Outgo|Claims

Rebase: N/A

Type: Double

Override: False

Virtual: False

#### **5.3.3.5.29**      *comm\_nihul\_pv\_start*

Description: PV of Nihul Commissions at policy start

Help: No of new policies starting in the valuation year  
(12 months prior to the valuation date)

Modified On: 8/30/2021 8:42:46 AM (UTC+03:00)

Modified By: CLAL-INS\joshm

Category: Cashflows|Outgo|Commission

Rebase: N/A

Type: Double

Override: False

Virtual: False

#### **5.3.3.5.30**      *comm\_prizes\_new*

Description: Lump sum commissions of new policies starting  
in the valuation year

Help: No of new policies starting in the valuation year  
(12 months prior to the valuation date)

Modified On: 11/17/2022 5:29:40 PM (UTC+02:00)

Modified By: CLAL-INS\joshm

Category: Cashflows|Outgo|Commission

Rebase: N/A

Type: Double

Override: False

Virtual: False

#### **5.3.3.5.31**      *comm\_pv\_start*

Description: PV of Commissions at policy start

Help: No of new policies starting in the valuation year  
(12 months prior to the valuation date)

Modified On: 8/30/2021 8:42:43 AM (UTC+03:00)

Modified By:	CLAL-INS\joshm
Category:	Cashflows Outgo Commission
Rebase:	N/A
Type:	Double
Override:	False
Virtual:	False

#### **5.3.3.5.32**      ***exp\_inflation\_mthly***

Description:	Monthly expense inflation rate
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Cashflows Outgo Expenses
Rebase:	N/A
Type:	Double
Override:	False
Virtual:	False

#### **5.3.3.5.33**      ***units\_to\_ann***

Description:	Unit value passed to annuity
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Charges
Rebase:	N/A
Type:	Double
Override:	False
Virtual:	False

#### **5.3.3.5.34**      ***claims\_pv\_st***

Description:	Claims PV St
Help:	
Modified On:	11/30/2023 4:40:39 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Claims
Rebase:	N/A
Type:	Double
Override:	False
Virtual:	False

#### **5.3.3.5.35**      ***comm\_clawback\_pv\_start***

Description:	Comm Clawback Pv Start
Help:	

Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Clawback
Rebase:	N/A
Type:	Double
Override:	False
Virtual:	False

#### **5.3.3.5.36**      *charges\_premium\_pv\_st*

Description:	מכירה לעת מהוון חיסכון בגין מפרמיה ניהול דמי
Help:	
Modified On:	7/25/2022 9:53:13 AM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Commission
Rebase:	N/A
Type:	Double
Override:	False
Virtual:	False

#### **5.3.3.5.37**      *comm\_hekef\_new*

Description:	Comm Hekef New
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Commission
Rebase:	N/A
Type:	Double
Override:	False
Virtual:	False

#### **5.3.3.5.38**      *comm\_init\_new*

Description:	Comm Init New
Help:	
Modified On:	11/17/2022 5:29:40 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Commission
Rebase:	N/A
Type:	Double
Override:	False
Virtual:	False

#### **5.3.3.5.39**      *comm\_reg\_pv\_st*

Description:	Comm Reg Pv St
--------------	----------------

Help:  
Modified On: 7/25/2022 9:40:32 AM (UTC+03:00)  
Modified By: CLAL-INS\ahuvaa  
Category: Commission  
Rebase: N/A  
Type: Double  
Override: False  
Virtual: False

#### **5.3.3.5.40** *comm\_reg\_riders\_out\_pv\_st*

Description: מכירה לעת מהוון ריידר שוטפות עמלות  
Help:  
Modified On: 7/25/2022 9:52:20 AM (UTC+03:00)  
Modified By: CLAL-INS\ahuvaa  
Category: Commission  
Rebase: N/A  
Type: Double  
Override: False  
Virtual: False

#### **5.3.3.5.41** *comm\_ren\_pv\_st*

Description: Comm Ren Pv St  
Help:  
Modified On: 11/16/2021 10:55:54 AM (UTC+02:00)  
Modified By: CLAL-INS\joshm  
Category: Commission  
Rebase: N/A  
Type: Double  
Override: False  
Virtual: False

#### **5.3.3.5.42** *comm\_res\_pv\_st*

Description: Comm Res Pv St  
Help:  
Modified On: 11/16/2021 10:55:42 AM (UTC+02:00)  
Modified By: CLAL-INS\joshm  
Category: Commission  
Rebase: N/A  
Type: Double  
Override: False  
Virtual: False

**5.3.3.5.43      *management\_fee\_pv\_st***

Description: מכירה לעת מהוון מצבירה ניהול דמי סך  
Help:  
Modified On: 7/25/2022 10:02:09 AM (UTC+03:00)  
Modified By: CLAL-INS\ahuvaa  
Category: Commission  
Rebase: N/A  
Type: Double  
Override: False  
Virtual: False

**5.3.3.5.44      *proj\_task\_loop\_num\_scalar***

Description: loop number - for ESG runs  
Help:  
Modified On: 1/11/2023 1:26:57 PM (UTC+02:00)  
Modified By: CLAL-INS\ahuvaa  
Category: Commission  
Rebase: N/A  
Type: Double  
Override: False  
Virtual: False

**5.3.3.5.45      *duration***

Description: Duration (macham) in months  
Help:  
Modified On: 12/8/2020 3:28:58 PM (UTC+02:00)  
Modified By: CLAL-INS\NinaB  
Category: Decrements  
Rebase: N/A  
Type: Double  
Override: False  
Virtual: False

**5.3.3.5.46      *expense\_init\_new***

Description: Expense Init New  
Help:  
Modified On: 11/17/2022 5:32:36 PM (UTC+02:00)  
Modified By: CLAL-INS\joshm  
Category: Expenses  
Rebase: N/A  
Type: Double  
Override: False  
Virtual: False



**5.3.3.5.47**      ***expense\_pv\_start***

Description:	Expense Pv Start
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Expenses
Rebase:	N/A
Type:	Double
Override:	False
Virtual:	False

**5.3.3.5.48**      ***policies\_new***

Description:	No of new policies starting in the valuation year
Help:	No of new policies starting in the valuation year (12 months prior to the valuation date)
Modified On:	1/8/2023 5:10:28 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Inforce Details Number of Covers
Rebase:	N/A
Type:	Double
Override:	False
Virtual:	False

**5.3.3.5.49**      ***premium\_new***

Description:	Premium of new policies starting in the valuation year
Help:	No of new policies starting in the valuation year (12 months prior to the valuation date)
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Inforce Details Number of Covers
Rebase:	N/A
Type:	Double
Override:	False
Virtual:	False

**5.3.3.5.50**      ***origidate***

Description:	Origidate Scalar
Help:	month and year of origi date
Modified On:	1/6/2022 5:38:34 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Policy Details
Rebase:	N/A
Type:	Double

Override: False  
Virtual: False

#### **5.3.3.5.51      *yob***

Description: year of birthday  
Help: month and year of origi date  
Modified On: 5/17/2023 3:58:18 PM (UTC+03:00)  
Modified By: CLAL-INS\ahuvaa  
Category: Policy Details  
Rebase: N/A  
Type: Double  
Override: False  
Virtual: False

#### **5.3.3.5.52      *prem\_alloc\_pv***

Description: PV of Premium for Savings  
Help: Proportion of Premium for Savings (as opposed to risk and expense chanrges)  
Modified On: 12/11/2023 2:43:58 PM (UTC+02:00)  
Modified By: CLAL-INS\Arikt  
Category: Policy Details|Premium  
Rebase: N/A  
Type: Double  
Override: False  
Virtual: False

#### **5.3.3.5.53      *premium\_1***

Description: Premium of the month1 include policy fee  
Help: Proportion of Premium for Savings (as opposed to risk and expense chanrges)  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category: Policy Details|Premium  
Rebase: N/A  
Type: Double  
Override: False  
Virtual: False

#### **5.3.3.5.54      *reins\_simple\_rider\_row***

Description: Lookup scalar  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab

Category:	Premium
Rebase:	N/A
Type:	Character
Override:	False
Virtual:	False

#### **5.3.3.5.55**      ***reins\_simple\_row***

Description:	Lookup scalar
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Premium
Rebase:	N/A
Type:	Character
Override:	False
Virtual:	False

#### **5.3.3.5.56**      ***ktest***

Description:	Identifies the ktest population
Help:	same as group in input
Modified On:	1/10/2022 2:01:49 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Product Details
Rebase:	N/A
Type:	Character
Override:	False
Virtual:	False

#### **5.3.3.5.57**      ***portfolio***

Description:	Portfolio
Help:	same as group in input
Modified On:	11/29/2021 1:08:17 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Product Details
Rebase:	N/A
Type:	Character
Override:	False
Virtual:	False

#### **5.3.3.5.58**      ***prod\_specs\_max\_perc***

Description:	Lookup scalar
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)

Modified By:	CLAL-INS\ninab
Category:	Product Details
Rebase:	N/A
Type:	Character
Override:	False
Virtual:	False

#### **5.3.3.5.59**      ***cashflow\_re\_pv\_st***

Description:	Cashflow Re Pv St
Help:	
Modified On:	11/30/2023 4:44:22 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Reinsurance
Rebase:	N/A
Type:	Double
Override:	False
Virtual:	False

#### **5.3.3.5.60**      ***reins\_comm1***

Description:	Reinsure commission of the first month
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Reinsurance
Rebase:	N/A
Type:	Double
Override:	False
Virtual:	False

#### **5.3.3.5.61**      ***reinsur\_clm\_cost***

Description:	Row lookup value scalar
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Reinsurance
Rebase:	N/A
Type:	Character
Override:	False
Virtual:	False

#### **5.3.3.5.62**      ***reinsur\_kodtavla***

Description:	Column Lookup value scalar
Help:	

Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Reinsurance
Rebase:	N/A
Type:	Character
Override:	False
Virtual:	False

#### **5.3.3.5.63**      ***reserve\_rein\_opening***

Description:	Reserve Rein Opening
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Reinsurance
Rebase:	N/A
Type:	Double
Override:	False
Virtual:	False

#### **5.3.3.5.64**      ***reserve\_manual***

Description:	Reserve Manual
Help:	
Modified On:	8/14/2024 10:39:35 AM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Reserve
Rebase:	N/A
Type:	Double
Override:	False
Virtual:	False

#### **5.3.3.5.65**      ***datetime\_stamp***

Description:	Date and time of run
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Setup
Rebase:	N/A
Type:	Character
Override:	False
Virtual:	False

#### **5.3.3.5.66**      ***sex\_smoker\_code***

Description:	Code for sex and smoker status of insured 1.
--------------	--

Help:	Codes returned are: M NS = 0 M Sm = 1 F NS = 2 F Sm = 3
Modified On:	4/21/2021 4:54:38 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Setup
Rebase:	N/A
Type:	Integer
Override:	False
Virtual:	False

#### **5.3.3.5.67**      ***stamp\_output***

Description:	Output file string
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Setup
Rebase:	N/A
Type:	Character
Override:	False
Virtual:	False

#### **5.3.3.5.68**      ***value\_date***

Description:	Valuation date
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Setup
Rebase:	N/A
Type:	Character
Override:	False
Virtual:	False

#### **5.3.4**              ***sub\_2\_cflow***

Description:	
Help:	
Base Model Class:	none
Model References	All
Read File:	Before Start Up
Modified On:	9/25/2024 2:30:23 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt

### 5.3.4.1 Variables

#### 5.3.4.1.1 *prop\_gteedint\_post\_maturity*

Description:	Prop of participating policies getting Guar int rate post maturity
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	1
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### 5.3.4.1.2 *sv\_tbl\_check*

Description:	Sv Tbl Check
Help:	
Modified On:	10/17/2021 4:46:35 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

## 5.3.4.2 Columns

### 5.3.4.2.1 *annuity\_if\_b\_bef\_ret*

Description:	Annuity If B Bef Ret
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	
Column Header:	annuity_if_b_bef_ret
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

### 5.3.4.2.2 *bonus*

Description:	Israeli Bonus/Malus declared
Help:	new bonus declared in month t to be added to accumulated bonus.
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	
Column Header:	bonus
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

### 5.3.4.2.3 *bonus\_b*

Description:	Bonus of previous period after survival (but before new bonus) (active)
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)



Modified By:	CLAL-INS\Arikt
Category:	
Column Header:	bonus_b
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.4 bonus\_b\_pup**

Description:	Bonus of previous period after survival (but before new bonus) (pup)
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	
Column Header:	bonus_b_pup
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.5 bonus\_if**

Description:	Bonus/Malus In Force
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	
Column Header:	bonus_if
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow

Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.6 bonus\_if\_pup**

Description:	Bonus/Malus of PUP In Force
Help:	accumulated bonus.
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	
Column Header:	bonus_if_pup
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.7 bonus\_pup**

Description:	Israeli Bonus/Malus declared for paid up
Help:	new bonus declared in month t to be added to accumulated bonus.
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	
Column Header:	bonus_pup
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.8 bonus\_rate**

Description:	Bonus Rate (active)
Help:	

Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	
Column Header:	bonus_rate
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.9 bonus\_rate\_mat**

Description:	Bonus Rate after maturity
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	
Column Header:	bonus_rate_mat
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.10 bonus\_rate\_mthly**

Description:	Monthly bonus rate
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	
Column Header:	bonus_rate_mthly
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow

Rebase Type:	None
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.11      *bonus\_rate\_pup***

Description:	Bonus Rate (pup)
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	
Column Header:	bonus_rate_pup
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.12      *bor\_acc***

Description:	Accumulated management fees owing (bor) (active)
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	
Column Header:	bor_acc
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.13      *bor\_acc\_mat***

Description:	Bor after Maturity age
Help:	

Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	
Column Header:	bor_acc_mat
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.14**      ***bor\_acc\_pup***

Description:	Accumulated management fees owing (bor) (pup)
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	
Column Header:	bor_acc_pup
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.15**      ***bor\_return***

Description:	Owned management fees repaid
Help:	
Modified On:	9/25/2024 2:29:54 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	
Column Header:	bor_return
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes

Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.16      *bor\_return\_mat***

Description:	Bor Returned after maturity
Help:	
Modified On:	9/25/2024 2:40:10 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	
Column Header:	bor_return_mat
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.17      *bor\_return\_pup***

Description:	Owned management fees repaid (pup)
Help:	
Modified On:	9/25/2024 2:33:05 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	
Column Header:	bor_return_pup
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.18      *claims\_death***

Description:	Total Death Claims in Period
--------------	------------------------------

Help:  
Modified On: 12/11/2023 2:43:58 PM (UTC+02:00)  
Modified By: CLAL-INS\Arikt  
Category:  
Column Header: claims\_death  
Combine Groups By: Sum Both  
Combine Periods: Last  
Default sliding Size: -1  
Discount Timing: End  
Discount Use: Yes  
Rate Use: Cash Flow  
Rebase Type: Previous  
Retain Value: Yes  
Override: False  
Virtual: False

#### **5.3.4.2.19**      *claims\_maturity*

Description: Total Maturity Claims in Period  
Help:  
Modified On: 12/11/2023 2:43:58 PM (UTC+02:00)  
Modified By: CLAL-INS\Arikt  
Category:  
Column Header: claims\_maturity  
Combine Groups By: Sum Both  
Combine Periods: Last  
Default sliding Size: -1  
Discount Timing: End  
Discount Use: Yes  
Rate Use: Cash Flow  
Rebase Type: Current  
Retain Value: Yes  
Override: False  
Virtual: False

#### **5.3.4.2.20**      *claims\_surrender*

Description: Total Surrender Claims in Period  
Help:  
Modified On: 12/11/2023 2:43:58 PM (UTC+02:00)  
Modified By: CLAL-INS\Arikt  
Category:  
Column Header: claims\_surrender  
Combine Groups By: Sum Both  
Combine Periods: Last  
Default sliding Size: -1  
Discount Timing: End  
Discount Use: Yes

Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.21**      ***death\_claims\_bon***

Description:	Death claims on Bonus/Malus
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	
Column Header:	death_claims_bon
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.22**      ***death\_claims\_si***

Description:	Death claims on sum insured
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	
Column Header:	death_claims_si
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.23**      ***har\_acc***

Description:	Accumulated management fees available for return (har) (active)
--------------	---



Help:  
Modified On: 12/11/2023 2:43:58 PM (UTC+02:00)  
Modified By: CLAL-INS\Arikt  
Category:  
Column Header: har\_acc  
Combine Groups By: Sum Both  
Combine Periods: Last  
Default sliding Size: -1  
Discount Timing: End  
Discount Use: Yes  
Rate Use: Cash Flow  
Rebase Type: Previous  
Retain Value: Yes  
Override: False  
Virtual: False

#### **5.3.4.2.24**      *har\_acc\_mat*

Description: Har after Maturity age  
Help:  
Modified On: 12/11/2023 2:43:58 PM (UTC+02:00)  
Modified By: CLAL-INS\Arikt  
Category:  
Column Header: har\_acc\_mat  
Combine Groups By: Sum Both  
Combine Periods: Last  
Default sliding Size: -1  
Discount Timing: End  
Discount Use: Yes  
Rate Use: Cash Flow  
Rebase Type: Previous  
Retain Value: Yes  
Override: False  
Virtual: False

#### **5.3.4.2.25**      *har\_acc\_pup*

Description: Accumulated management fees available for return (har) (pup)  
Help:  
Modified On: 12/11/2023 2:43:58 PM (UTC+02:00)  
Modified By: CLAL-INS\Arikt  
Category:  
Column Header: har\_acc\_pup  
Combine Groups By: Sum Both  
Combine Periods: Last  
Default sliding Size: -1  
Discount Timing: End

Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.26**      *har\_return*

Description:	Variable management fees returned (active)
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	
Column Header:	har_return
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.27**      *har\_return\_mat*

Description:	Har Returned after maturity
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	
Column Header:	har_return_mat
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.4.2.28**      ***har\_return\_pup***

Description:	Variable management fees returned (pup)
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	
Column Header:	har_return_pup
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.4.2.29**      ***int\_cred***

Description:	Interest credited for bonus
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	
Column Header:	int_cred
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.4.2.30**      ***int\_cred\_mat***

Description:	Int Cred after maturity
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	
Column Header:	int_cred_mat
Combine Groups By:	Sum Both
Combine Periods:	Last

Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.31**      *int\_cred\_pup*

Description:	Interest credited for bonus (pup)
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	
Column Header:	int_cred_pup
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.32**      *int\_post\_mat*

Description:	interest rate accumulating after original maturity age up to age 80
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	
Column Header:	int_post_mat
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.4.2.33      *int\_res\_deduct***

Description:	Reserving interest deducted from bonus (active)
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	
Column Header:	int_res_deduct
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.4.2.34      *int\_res\_deduct\_pup***

Description:	Reserving interest deducted from bonus (pup)
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	
Column Header:	int_res_deduct_pup
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.4.2.35      *maturities\_bon***

Description:	Maturity payments on Bonus/Malus
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	
Column Header:	maturities_bon
Combine Groups By:	Sum Both

Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.36**      ***maturities\_si***

Description:	Maturity payments on sum insured
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	
Column Header:	maturities_si
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.37**      ***mgt\_fee\_fix***

Description:	Fixed management fees (active)
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	
Column Header:	mgt_fee_fix
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.4.2.38**      ***mgt\_fee\_fix\_mat***

Description:	Fixed management fees after maturity
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	
Column Header:	mgt_fee_fix_mat
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.4.2.39**      ***mgt\_fee\_fix\_pup***

Description:	Fixed management fees (pup)
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	
Column Header:	mgt_fee_fix_pup
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.4.2.40**      ***mgt\_fee\_var***

Description:	Variable management fees (active)
Help:	
Modified On:	9/25/2024 2:30:09 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	
Column Header:	mgt_fee_var
Combine Groups By:	Sum Both
Combine Periods:	Last

Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.41**      ***mgt\_fee\_var\_mat***

Description:	Variable management fees after maturity
Help:	
Modified On:	9/25/2024 2:39:17 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	
Column Header:	mgt_fee_var_mat
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.42**      ***mgt\_fee\_var\_no\_bor***

Description:	Variable management fees (active) when no bor for same calc in bor_return and management_fees_variable
Help:	
Modified On:	9/25/2024 2:30:19 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	
Column Header:	mgt_fee_var_no_bor
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False



**5.3.4.2.43 mgt\_fee\_var\_no\_bor\_mat**

Description:	Variable management fees (active) when no bor for same calc in bor_return and management_fees_variable
Help:	
Modified On:	9/25/2024 2:34:25 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	
Column Header:	mgt_fee_var_no_bor_mat
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.4.2.44 mgt\_fee\_var\_no\_bor\_pup**

Description:	Variable management fees (active) when no bor for same calc in bor_return and management_fees_variable
Help:	
Modified On:	9/25/2024 2:31:02 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	
Column Header:	mgt_fee_var_no_bor_pup
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.4.2.45 mgt\_fee\_var\_pup**

Description:	Variable management fees (pup)
Help:	
Modified On:	9/25/2024 2:32:42 PM (UTC+03:00)

Modified By:	CLAL-INS\arikt
Category:	
Column Header:	mgt_fee_var_pup
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.46**      ***net\_interest\_rate***

Description:	Interest rate net of fixed management fees
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\arikt
Category:	
Column Header:	net_interest_rate
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.47**      ***puv\_factor***

Description:	PaidUp value factor
Help:	Surrender value at EOM for policies IF at EOM
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\arikt
Category:	
Column Header:	puv_factor
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous

Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.48      *res\_to\_bonus***

Description:	Reserve on which bonus is earned
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	
Column Header:	res_to_bonus
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.49      *res\_to\_bonus\_pup***

Description:	Reserve on which bonus is earned
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	
Column Header:	res_to_bonus_pup
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.50      *surr\_value***

Description:	Surrender value inforce EOM
Help:	Surrender value at EOM for policies IF at EOM
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)

Modified By:	CLAL-INS\Arikt
Category:	
Column Header:	surr_value
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.51**      ***surr\_value\_pup***

Description:	Surrender value of PUP inforce EOM
Help:	Surrender value at EOM for policies IF at EOM
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	
Column Header:	surr_value_pup
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.52**      ***sv\_factor***

Description:	Surrender factor
Help:	Surrender value at EOM for policies IF at EOM
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	
Column Header:	sv_factor
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous

Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.53      *zillmer\_book***

Description:	Zillmer for reported surplus
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Balance Sheet Dac
Column Header:	zillmer_book
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	0
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.54      *zillmer\_tax***

Description:	Zillmer for tax purposes
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Balance Sheet Dac
Column Header:	zillmer_tax
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	0
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.55      *ann\_takeup\_rate***

Description:	annuitization rate
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)

Modified By:	CLAL-INS\Arikt
Category:	Balance Sheet Reserves
Column Header:	ann_takeup_rate
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.56      *reserve***

Description:	Reserve total (excluding ERR)
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Balance Sheet Reserves
Column Header:	reserve
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.57      *reserve\_extra***

Description:	extra reserve items
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Balance Sheet Reserves
Column Header:	reserve_extra
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous

Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.58      *ann\_factor\_weighted***

Description:	Ann Factor Weighted
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Balance Sheet Reserves Basic Reserve
Column Header:	ann_factor_weighted
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.59      *ann\_factor\_weighted\_int0***

Description:	Ann Factor Weighted with 0 interest
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Balance Sheet Reserves Basic Reserve
Column Header:	ann_factor_weighted_int0
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.60      *annuity\_factor***

Description:	Annuity factor - ax
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)

Modified By:	CLAL-INS\Arikt
Category:	Balance Sheet Reserves Basic Reserve
Column Header:	annuity_factor
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.61      *annuity\_factor\_int0***

Description:	Annuity factor - ax with 0 interest
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Balance Sheet Reserves Basic Reserve
Column Header:	annuity_factor_int0
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.62      *ass\_factor\_weighted***

Description:	Ass Factor Weighted
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Balance Sheet Reserves Basic Reserve
Column Header:	ass_factor_weighted
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous



Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.63      *ass\_factor\_weighted\_int0***

Description:	Ass Factor Weighted with 0 interest rate
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Balance Sheet Reserves Basic Reserve
Column Header:	ass_factor_weighted_int0
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.64      *assurance\_factor***

Description:	Assurance factor - Ax
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Balance Sheet Reserves Basic Reserve
Column Header:	assurance_factor
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.65      *assurance\_factor\_int0***

Description:	Assurance factor - Ax with interest rate = 0
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)

Modified By:	CLAL-INS\Arikt
Category:	Balance Sheet Reserves Basic Reserve
Column Header:	assurance_factor_int0
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.66**      ***net\_premium\_b***

Description:	Net premiums at beginning of month
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Balance Sheet Reserves Basic Reserve
Column Header:	net_premium_b
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.67**      ***net\_premium\_e***

Description:	Net premiums in force
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Balance Sheet Reserves Basic Reserve
Column Header:	net_premium_e
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous

Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.68**      *res\_basic\_act\_newtag*

Description:	Res Basic Act Newtag
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Balance Sheet Reserves Basic Reserve
Column Header:	res_basic_act_newtag
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.69**      *res\_basic\_act\_old*

Description:	Res Basic Act Old
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Balance Sheet Reserves Basic Reserve
Column Header:	res_basic_act_old
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.70**      *res\_basic\_act\_piz*

Description:	Res Basic Act Piz
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)

Modified By:	CLAL-INS\Arikt
Category:	Balance Sheet Reserves Basic Reserve
Column Header:	res_basic_act_piz
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.71**      ***res\_basic\_act\_piz\_int***

Description:	Res Basic Act Piz with interest added
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Balance Sheet Reserves Basic Reserve
Column Header:	res_basic_act_piz_int
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.72**      ***res\_basic\_act\_prat***

Description:	Res Basic Act Prat
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Balance Sheet Reserves Basic Reserve
Column Header:	res_basic_act_prat
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous

Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.73      *res\_basic\_pup\_newtag***

Description:	Res Basic PUP Newtag
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Balance Sheet Reserves Basic Reserve
Column Header:	res_basic_pup_newtag
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.74      *res\_basic\_pup\_old***

Description:	Res Basic PUP Old
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Balance Sheet Reserves Basic Reserve
Column Header:	res_basic_pup_old
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.75      *res\_basic\_pup\_piz***

Description:	Res Basic PUP Piz
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)

Modified By:	CLAL-INS\Arikt
Category:	Balance Sheet Reserves Basic Reserve
Column Header:	res_basic_pup_piz
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.76**      *res\_basic\_pup\_piz\_int*

Description:	Res Basic PUP Piz with interest
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Balance Sheet Reserves Basic Reserve
Column Header:	res_basic_pup_piz_int
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.77**      *res\_basic\_pup\_prat*

Description:	Res Basic PUP Prat
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Balance Sheet Reserves Basic Reserve
Column Header:	res_basic_pup_prat
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous

Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.78**      *reserve\_basic*

Description:	Basic reserve - Net or Gross Premium
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Balance Sheet Reserves Basic Reserve
Column Header:	reserve_basic
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.79**      *reserve\_basic\_prem\_if*

Description:	Basic reserve of premium inforce- Net or Gross Premium
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Balance Sheet Reserves Basic Reserve
Column Header:	reserve_basic_prem_if
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.80**      *reserve\_basic\_pup*

Description:	Basic reserve of PUP- Net or Gross Premium
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)

Modified By:	CLAL-INS\Arikt
Category:	Balance Sheet Reserves Basic Reserve
Column Header:	reserve_basic_pup
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.81      *reserve\_risk\_premium***

Description:	Risk premium for Basic reserve
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Balance Sheet Reserves Basic Reserve
Column Header:	reserve_risk_premium
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.82      *vnp***

Description:	Value of net premium - VNP
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Balance Sheet Reserves Basic Reserve
Column Header:	vnp
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous



Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.83**      ***vsa***

Description:	Value of sum assured - VSA
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Balance Sheet Reserves Basic Reserve
Column Header:	vsa
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.84**      ***death\_rate***

Description:	Independent death rate in period
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Decrements Death
Column Header:	death_rate
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	Middle
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.85**      ***pol\_fee***

Description:	Policy Fee
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)

Modified By:	CLAL-INS\Arikt
Category:	Income Statement Income Premium
Column Header:	pol_fee
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.86**      ***premium***

Description:	Premium (including policy fee)
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Income Statement Income Premium
Column Header:	premium
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.87**      ***premium\_gross***

Description:	Gross premium income during period
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Income Statement Income Premium
Column Header:	premium_gross
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous

Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.88**      *claims\_rate\_per*

Description:	Claims Rate Per
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Income Statement\Outgo\Claims
Column Header:	claims_rate_per
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.89**      *premium\_if\_b*

Description:	Premium in force at start of period
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Inforce Details\Premium
Column Header:	premium_if_b
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.90**      *premium\_if\_e*

Description:	Premium in force at end of period
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)

Modified By:	CLAL-INS\Arikt
Category:	Inforce Details Premium
Column Header:	premium_if_e
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.91      *annuity\_if\_b***

Description:	Monthly annuity in force for GIMLA (start of month)
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Inforce Details Sum Assured
Column Header:	annuity_if_b
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.92      *claims\_ret***

Description:	Claims Ret
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Inforce Details Sum Assured
Column Header:	claims_ret
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow

Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.93**      ***sum\_at\_risk\_if***

Description:	Sum at risk in force
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Inforce Details Sum Assured
Column Header:	sum_at_rsk_if
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.94**      ***sum\_insured***

Description:	Sum insured - undecrement
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Inforce Details Sum Assured
Column Header:	sum_insured
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.95**      ***sum\_insured\_if\_b***

Description:	Sum insured in force (start of month)
Help:	

Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Inforce Details Sum Assured
Column Header:	sum_insured_if_b
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.96      *sum\_insured\_if\_b\_pup***

Description:	SI IF (start of mth) - paid up
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Inforce Details Sum Assured
Column Header:	sum_insured_if_b_pup
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.97      *sum\_insured\_if\_e***

Description:	Sum insured in force (end of month)
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Inforce Details Sum Assured
Column Header:	sum_insured_if_e
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow

Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.98**      *claims\_re*

Description:	Reins Claim outgo in the mid of the per.
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\arikt
Category:	Reinsurance
Column Header:	claims_re
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	0
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.99**      *comm\_re*

Description:	Initial and renewal reins commission
Help:	
Modified On:	1/5/2025 3:19:12 PM (UTC+02:00)
Modified By:	CLAL-INS\arikt
Category:	Reinsurance
Column Header:	comm_re
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	0
Discount Timing:	Beginning
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.100**      *comm\_re\_prof*

Description:	Reinsurance profit commission
Help:	

Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Reinsurance
Column Header:	comm_re_prof
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	0
Discount Timing:	Beginning
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.101**      ***exp\_re\_nom***

Description:	reinsurance nominal expenses
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Reinsurance
Column Header:	exp_re_nom
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	0
Discount Timing:	Beginning
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.102**      ***premium\_if\_b\_re***

Description:	Reins premium in force at start of perio
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Reinsurance
Column Header:	premium_if_b_re
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	0
Discount Timing:	Beginning
Discount Use:	No
Rate Use:	Cash Flow



Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.103**     ***premium\_re***

Description:	Reins Premium income at begin. of period
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Reinsurance
Column Header:	premium_re
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	0
Discount Timing:	Beginning
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.104**     ***profit\_re***

Description:	reinsurance profit for the period
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Reinsurance
Column Header:	profit_re
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	0
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.105**     ***sum\_insured\_re***

Description:	Sum reinsured - undecrementd
Help:	

Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Reinsurance
Column Header:	sum_insured_re
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	0
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.106     *startup***

Description:	Startup
Help:	This column is always called first when running a projection.
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Setup
Column Header:	startup
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.4.2.107     *pol\_sub\_year***

Description:	Policy sub year/term
Help:	
Modified On:	12/11/2023 2:43:58 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Timing
Column Header:	pol_sub_year
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	0
Discount Timing:	Beginning
Discount Use:	No

Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

### 5.3.4.3 External Functions

#### 5.3.4.3.1 *monthly\_rate*

Description:	Convert annual to monthly
Help:	Given an annual interest rate, expressed as i% this external formula returns $i(12)/12\%$ , i.e. the monthly rate which compounds to give the annual rate, eg <code>monthly_rate(10)=0.007974</code>
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Setup

#### 5.3.4.3.2 *set\_other\_variables*

Description:	Setup other variables
Help:	External function to calculate and/or adjust variables before running projection. This function is called from startup.
Modified On:	1/11/2023 11:33:41 AM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Setup

#### 5.3.4.3.3 *set\_premium\_si*

Description:	Modify (and calculate) premium and SA
Help:	This function calculates the current premium or the current sum insured. This function is called by startup.
Modified On:	6/2/2022 1:40:52 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Setup

#### 5.3.4.3.4 *validate\_data*

Description:	Validate Data
Help:	This function carries out data validation checks on the input data and skips the model point if errors are detected, via the throw command.
Modified On:	8/25/2022 2:12:48 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa

Category:

Setup

### 5.3.4.4 Temporary Tables

#### 5.3.4.4.1 *res\_cx*

Description:	Reserve Commutation Cx
Help:	
Modified On:	7/4/2022 2:26:45 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Balance Sheet Reserves Basic Reserve
Number of Rows:	121
Number of Columns:	2
Calculation Methods:	Cell
Reset Variables:	
Resetting Type:	Reset Temporary Table for every model point and/or projection loop
Override:	False
Virtual:	False

#### 5.3.4.4.2 *res\_dx*

Description:	Reserve Commutation Dx
Help:	
Modified On:	7/4/2022 2:30:47 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Balance Sheet Reserves Basic Reserve
Number of Rows:	121
Number of Columns:	2
Calculation Methods:	Cell
Reset Variables:	
Resetting Type:	Reset Temporary Table for every model point and/or projection loop
Override:	False
Virtual:	False

#### 5.3.4.4.3 *res\_lx*

Description:	Reserve Commutation lx
Help:	
Modified On:	3/13/2023 4:02:12 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Balance Sheet Reserves Basic Reserve
Number of Rows:	121
Number of Columns:	1
Calculation Methods:	Cell

Reset Variables:	
Resetting Type:	Reset Temporary Table for every model point and/or projection loop
Override:	False
Virtual:	False

#### 5.3.4.4.4 res\_mx

Description:	Reserve Commutation Mx
Help:	
Modified On:	7/4/2022 2:21:30 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Balance Sheet Reserves Basic Reserve
Number of Rows:	121
Number of Columns:	2
Calculation Methods:	Cell
Reset Variables:	
Resetting Type:	Reset Temporary Table for every model point and/or projection loop
Override:	False
Virtual:	False

#### 5.3.4.4.5 res\_nx

Description:	Reserve Commutation Nx
Help:	
Modified On:	7/4/2022 2:35:05 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Balance Sheet Reserves Basic Reserve
Number of Rows:	121
Number of Columns:	2
Calculation Methods:	Cell
Reset Variables:	
Resetting Type:	Reset Temporary Table for every model point and/or projection loop
Override:	False
Virtual:	False

#### 5.3.4.4.6 res\_vx

Description:	Reserve Interest vx
Help:	
Modified On:	10/21/2020 3:44:35 PM (UTC+03:00)
Modified By:	CLAL-INS\NinaB
Category:	Balance Sheet Reserves Basic Reserve
Number of Rows:	131
Number of Columns:	2
Calculation Methods:	Cell

Reset Variables:	
Resetting Type:	Reset Temporary Table for every model point and/or projection loop
Override:	False
Virtual:	False

### 5.3.4.5 Scalars

#### 5.3.4.5.1 *interest\_rein\_mthly*

Description:	Monthly interest rate on reinsurance reserves
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	
Rebase:	N/A
Type:	Double
Override:	False
Virtual:	False

#### 5.3.4.5.2 *int\_rate\_res\_hy*

Description:	1+ Half-yearly interest rate for reserves
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Balance Sheet Reserves
Rebase:	N/A
Type:	Double
Override:	False
Virtual:	False

#### 5.3.4.5.3 *int\_rate\_res\_mthly*

Description:	Monthly interest rate for reserves
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Balance Sheet Reserves
Rebase:	N/A
Type:	Double
Override:	False
Virtual:	False

**5.3.4.5.4 res\_prop\_mat\_newtag**

Description:	Res Prop Mat Newtag
Help:	
Modified On:	8/22/2021 3:26:44 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Balance Sheet Reserves Basic Reserve
Rebase:	N/A
Type:	Double
Override:	False
Virtual:	False

**5.3.4.5.5 res\_prop\_mat\_oldtag**

Description:	Res Prop Mat Oldtag
Help:	
Modified On:	8/22/2021 3:26:24 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Balance Sheet Reserves Basic Reserve
Rebase:	N/A
Type:	Double
Override:	False
Virtual:	False

**5.3.4.5.6 res\_prop\_mat\_piz**

Description:	Res Prop Mat Piz
Help:	
Modified On:	8/22/2021 3:26:47 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Balance Sheet Reserves Basic Reserve
Rebase:	N/A
Type:	Double
Override:	False
Virtual:	False

**5.3.4.5.7 res\_prop\_mat\_prat**

Description:	Res Prop Mat Prat
Help:	
Modified On:	8/22/2021 3:26:50 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Balance Sheet Reserves Basic Reserve
Rebase:	N/A
Type:	Double
Override:	False
Virtual:	False

**5.3.4.5.8 ann\_factor\_pol**

Description:	Policy Annuity factor at retirement.
Help:	
Modified On:	8/1/2022 12:44:55 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Inforce Details Sum Assured
Rebase:	N/A
Type:	Double
Override:	False
Virtual:	False

**5.3.4.5.9 sum\_insured\_newmoney**

Description:	Sum Insured Newmoney
Help:	
Modified On:	6/10/2021 12:57:28 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Inforce Details Sum Assured
Rebase:	N/A
Type:	Double
Override:	False
Virtual:	False

**5.3.4.5.10 sum\_insured\_newtag**

Description:	Sum Insured Newtag
Help:	
Modified On:	6/10/2021 12:57:28 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Inforce Details Sum Assured
Rebase:	N/A
Type:	Double
Override:	False
Virtual:	False

**5.3.4.5.11 sum\_insured\_oldtag**

Description:	Sum Insured Oldtag
Help:	
Modified On:	6/10/2021 12:57:28 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Inforce Details Sum Assured
Rebase:	N/A
Type:	Double
Override:	False
Virtual:	False



**5.3.4.5.12**      ***sum\_insured\_piz***

Description:	Sum Insured Piz
Help:	
Modified On:	6/10/2021 12:57:29 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Inforce Details Sum Assured
Rebase:	N/A
Type:	Double
Override:	False
Virtual:	False

**5.3.4.5.13**      ***sum\_insured\_piz\_int0***

Description:	Sum Insured Piz with 0 interest
Help:	
Modified On:	6/10/2021 12:57:29 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Inforce Details Sum Assured
Rebase:	N/A
Type:	Double
Override:	False
Virtual:	False

**5.3.4.5.14**      ***sum\_insured\_prat***

Description:	Sum Insured Prat
Help:	
Modified On:	6/10/2021 12:57:29 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Inforce Details Sum Assured
Rebase:	N/A
Type:	Double
Override:	False
Virtual:	False

**5.3.5**      ***sub\_array***

Description:	
Help:	
Base Model Class:	none
Model References	All
Read File:	Before Start Up
Modified On:	8/19/2021 8:25:08 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm

### 5.3.5.1 Variables

#### 5.3.5.1.1 *aml\_ni\_1\_6*

Description:	Aml Ni 1 6
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### 5.3.5.1.2 *amla\_1\_6*

Description:	Amla 1 6
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### 5.3.5.1.3 *amla\_7*

Description:	Amla 7
--------------	--------

Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category:  
Variable Type: Floating Point Number  
Default Value: 0  
Length: 0  
Number of Decimals: 1  
Choice List:  
Character Type: Not Applicable  
Valid Range From:  
Valid Range To:  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.5.1.4 amla\_ni\_7**

Description: Amla Ni 7  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category:  
Variable Type: Floating Point Number  
Default Value: 0  
Length: 0  
Number of Decimals: 1  
Choice List:  
Character Type: Not Applicable  
Valid Range From:  
Valid Range To:  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.5.1.5 dynamic**

Description: Dynamic  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category:  
Variable Type: Integer Number  
Default Value: 0  
Length: 0  
Number of Decimals: 0

Choice List:  
Character Type: Not Applicable  
Valid Range From:  
Valid Range To:  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.5.1.6 lod\_amt\_1**

Description: Lod Amt 1  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category:  
Variable Type: Floating Point Number  
Default Value: 0  
Length: 0  
Number of Decimals: 1  
Choice List:  
Character Type: Not Applicable  
Valid Range From:  
Valid Range To:  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.5.1.7 lod\_pe\_r\_1**

Description: Lod Pe R 1  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category:  
Variable Type: Integer Number  
Default Value: 0  
Length: 0  
Number of Decimals: 0  
Choice List:  
Character Type: Not Applicable  
Valid Range From:  
Valid Range To:  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

**5.3.5.1.8 *pol\_number\_i***

Description:	Pol Number I
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	
Variable Type:	Character
Default Value:	0
Length:	15
Number of Decimals:	1
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.5.1.9 *pr\_cov\_cal***

Description:	Pr Cov Cal
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	
Variable Type:	Integer Number
Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.5.1.10 *prem\_cover***

Description:	Prem Cover
Help:	
Modified On:	8/19/2021 8:25:39 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	

Variable Type:	Floating Point Number
Default Value:	0
Length:	36
Number of Decimals:	2
Choice List:	0
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.5.1.11      *prem\_cover\_input***

Description:	Prem Cover
Help:	
Modified On:	8/19/2021 8:25:52 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	
Variable Type:	Character
Default Value:	0
Length:	50
Number of Decimals:	2
Choice List:	0
Character Type:	Standard
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.5.1.12      *prm\_in\_ppn***

Description:	Prm In Ppn
Help:	is set to 0 or 1 0 = rider premium is out 1 = rider premium is in
Modified On:	8/8/2021 9:22:27 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	1
Choice List:	
Character Type:	Not Applicable

Valid Range From:	0
Valid Range To:	10
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.5.1.13**      ***retention***

Description:	Retention
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	
Variable Type:	Floating Point Number
Default Value:	0
Length:	0
Number of Decimals:	3
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

#### **5.3.5.1.14**      ***rid\_sex***

Description:	Rid Sex
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	
Variable Type:	Integer Number
Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.5.1.15**      ***risk\_type***

Description:	Risk Type
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	
Variable Type:	Integer Number
Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.5.1.16**      ***sum\_as***

Description:	Sum As
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	
Variable Type:	Integer Number
Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

**5.3.5.1.17**      ***tarif***

Description:	Tarif
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	
Variable Type:	Integer Number



Default Value:	0
Length:	0
Number of Decimals:	0
Choice List:	
Character Type:	Not Applicable
Valid Range From:	
Valid Range To:	
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.5.2 Columns

#### 5.3.5.2.1 *initial\_formula*

Description:	Default Formula
Help:	
Modified On:	8/19/2021 9:56:52 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	
Column Header:	init_fml
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	0
Discount Timing:	Beginning
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

### 5.3.5.3 External Functions

<No External Functions Exist>

### 5.3.5.4 Temporary Tables

<No Temporary Tables Exist>

### 5.3.5.5 Scalars

<No Scalars Exist>

### 5.3.6 sub1\_cflow

Description:  
Help:  
Base Model Class: none  
Model References: All  
Read File: Before Start Up  
Modified On: 5/29/2025 11:23:49 AM (UTC+03:00)  
Modified By: CLAL-INS\arikt

#### 5.3.6.1 Variables

##### 5.3.6.1.1 prem\_disc\_scenario

Description: Prem Disc for Scenario  
Help:  
Modified On: 11/25/2020 2:48:22 PM (UTC+02:00)  
Modified By: CLAL-INS\NinaB  
Category:  
Variable Type: Floating Point Number  
Default Value: 0  
Length: 0  
Number of Decimals: 2  
Choice List:  
Character Type: Not Applicable  
Valid Range From: 0  
Valid Range To: 100  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

##### 5.3.6.1.2 pup\_ltc\_key

Description: Pup Ltc Key  
Help:  
Modified On: 9/9/2021 4:41:42 PM (UTC+03:00)  
Modified By: CLAL-INS\joshm  
Category: Decrements  
Variable Type: Character  
Default Value: 0  
Length: 10  
Number of Decimals: 1  
Choice List: 0  
Character Type: Standard

Valid Range From:  
Valid Range To:  
Table Format: Default Row Numbers  
Set Value in Input Manager:  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.6.1.3 *pup\_ltc\_tbl***

Description: Pup Ltc Tbl  
Help:  
Modified On: 9/9/2021 3:50:50 PM (UTC+03:00)  
Modified By: CLAL-INS\joshm  
Category: Decrements  
Variable Type: Floating Point Number  
Default Value: 0  
Length: 0  
Number of Decimals: 1  
Choice List:  
Character Type: Not Applicable  
Valid Range From:  
Valid Range To:  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

#### **5.3.6.1.4 *pup\_ltc\_tbl\_next***

Description: Pup Ltc Tbl  
Help:  
Modified On: 9/9/2021 4:27:02 PM (UTC+03:00)  
Modified By: CLAL-INS\joshm  
Category: Decrements  
Variable Type: Floating Point Number  
Default Value: 0  
Length: 0  
Number of Decimals: 1  
Choice List:  
Character Type: Not Applicable  
Valid Range From:  
Valid Range To:  
Table Format: Default Row Numbers  
Set Value in Input Manager: All  
Variable Sharing: Not Shared  
Category Order: 0

**5.3.6.1.5 inv\_rate\_clm\_mth\_t**

Description:	monthly investment rates for claims in payment
Help:	monthly rate
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Income Statement Income Investment Income
Variable Type:	Floating Point Array
Default Value:	0
Length:	120
Number of Decimals:	6
Choice List:	
Character Type:	Not Applicable
Valid Range From:	-1
Valid Range To:	1
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.6.1.6 mgt\_fee\_fixed\_clm**

Description:	fixed management fee for phi claims net bonus calc.
Help:	Annual rate.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Setup
Variable Type:	Floating Point Number
Default Value:	0.6
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	20
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

**5.3.6.1.7 mgt\_fee\_var\_clm**

Description:	variable management fee for phi claims net bonus calc.
Help:	Annual rate.
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Setup

Variable Type:	Floating Point Number
Default Value:	15
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	50
Table Format:	Default Row Numbers
Set Value in Input Manager:	
Variable Sharing:	Not Shared
Category Order:	0

### 5.3.6.1.8 *pizui\_prop\_pup\_stat\_c*

Description:	Pitzui proportion for PHI in claims
Help:	
Modified On:	5/29/2025 11:25:28 AM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Setup
Variable Type:	Floating Point Number
Default Value:	0.6
Length:	0
Number of Decimals:	2
Choice List:	
Character Type:	Not Applicable
Valid Range From:	0
Valid Range To:	20
Table Format:	Default Row Numbers
Set Value in Input Manager:	All
Variable Sharing:	Not Shared
Category Order:	0

## 5.3.6.2 Columns

### 5.3.6.2.1 *bonus\_rate\_acc\_mthly*

Description:	Accumulate monthly bonus rate
Help:	
Modified On:	8/9/2021 10:19:44 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	
Column Header:	bonus_rate_acc_mthly
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1

Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	None
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.6.2.2 bonus\_rate\_mthly**

Description:	Monthly bonus rate
Help:	
Modified On:	9/10/2024 4:09:50 PM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	
Column Header:	bonus_rate_mthly
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	None
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.6.2.3 err**

Description:	Extraordinary Risk Reserve
Help:	
Modified On:	8/26/2021 2:34:52 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Balance Sheet ERR
Column Header:	err
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### 5.3.6.2.4 *reserve*

Description:	Reserve total (excluding ERR)
Help:	
Modified On:	10/18/2021 10:12:11 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Balance Sheet Reserves
Column Header:	reserve
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### 5.3.6.2.5 *annuity\_factor*

Description:	Annuity factor - ax
Help:	
Modified On:	8/9/2021 10:19:18 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Balance Sheet Reserves Basic Reserve
Column Header:	annuity_factor
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### 5.3.6.2.6 *assurance\_factor*

Description:	Assurance factor - Ax
Help:	
Modified On:	8/9/2021 10:19:26 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Balance Sheet Reserves Basic Reserve
Column Header:	assurance_factor
Combine Groups By:	Average Both
Combine Periods:	Last

Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.6.2.7 net\_prem\_deficiency\_b**

Description:	Net premium Deficiency at beginning of month
Help:	
Modified On:	5/3/2022 3:16:03 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Balance Sheet Reserves Basic Reserve
Column Header:	net_prem_deficiency_b
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.6.2.8 net\_premium\_b**

Description:	Net premiums at beginning of month
Help:	
Modified On:	8/9/2021 10:21:43 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Balance Sheet Reserves Basic Reserve
Column Header:	net_premium_b
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False



**5.3.6.2.9 net\_premium\_e**

Description:	Net premiums in force
Help:	
Modified On:	8/9/2021 10:21:53 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Balance Sheet Reserves Basic Reserve
Column Header:	net_premium_e
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.6.2.10 res\_np\_deficiency**

Description:	NP Reserve Deficiency
Help:	NP Deficiency reserve based on difference between calculated NP and gross premium * factor (eg 90%)
Modified On:	8/9/2021 10:23:05 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Balance Sheet Reserves Basic Reserve
Column Header:	res_np_deficiency
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.6.2.11 reserve\_basic**

Description:	Basic reserve - Net or Gross Premium
Help:	
Modified On:	5/3/2022 3:16:03 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Balance Sheet Reserves Basic Reserve
Column Header:	reserve_basic
Combine Groups By:	Sum Both

Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.6.2.12**      ***reserve\_basic\_claims***

Description:	Basic reserves of claims inpayment
Help:	
Modified On:	1/16/2024 9:21:30 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Balance Sheet Reserves Basic Reserve
Column Header:	reserve_basic_claims
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.6.2.13**      ***reserve\_risk\_premium***

Description:	Risk premium for Basic reserve
Help:	
Modified On:	3/20/2024 12:37:54 AM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Balance Sheet Reserves Basic Reserve
Column Header:	reserve_risk_premium
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.6.2.14 vnp**

Description:	Value of net premium - VNP
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Balance Sheet Reserves Basic Reserve
Column Header:	vnp
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.6.2.15 vsa**

Description:	Value of sum assured - VSA
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Balance Sheet Reserves Basic Reserve
Column Header:	vsa
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.6.2.16 surv**

Description:	Propn of lives in force at end of time t
Help:	
Modified On:	11/15/2022 4:50:49 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Decrements
Column Header:	surv
Combine Groups By:	Average Both
Combine Periods:	Last

Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.6.2.17**      ***surv\_2***

Description:	Propn of lives in force - secondary
Help:	
Modified On:	3/12/2023 12:26:50 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Decrements
Column Header:	surv_2
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.6.2.18**      ***surv\_2\_no\_dec***

Description:	Propn of lives in force - secondary w/o decrem
Help:	
Modified On:	9/20/2022 12:55:03 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Decrements
Column Header:	surv_2_no_dec
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.6.2.19**      ***surv\_no\_dec***

Description:	Proprn of lives in force at end of time t - w/o decrem
Help:	
Modified On:	11/15/2022 4:58:19 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Decrements
Column Header:	surv_no_dec
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.6.2.20**      ***surv\_no\_dth***

Description:	Proprn of lives in force at end of time t w/o death
Help:	
Modified On:	11/15/2022 4:58:24 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Decrements
Column Header:	surv_no_dth
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.6.2.21**      ***surv\_per***

Description:	Probability of survival for the period
Help:	
Modified On:	11/15/2022 4:51:17 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Decrements
Column Header:	surv_per
Combine Groups By:	Average Both

Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.6.2.22**      ***surv\_per\_no\_dec***

Description:	Probability of survival for the period - w/o decrement
Help:	
Modified On:	11/15/2022 4:58:31 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Decrements
Column Header:	surv_per_no_dec
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.6.2.23**      ***surv\_per\_no\_dth***

Description:	Probability of survival for the period w/o death
Help:	
Modified On:	11/15/2022 4:58:34 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Decrements
Column Header:	surv_per_no_dth
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.6.2.24      *death\_rate***

Description:	Independent death rate in period
Help:	
Modified On:	8/26/2021 2:30:12 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Decrement Death
Column Header:	death_rate
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	Middle
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.6.2.25      *death\_rate\_dep***

Description:	Dependent death rate
Help:	
Modified On:	8/9/2021 10:21:07 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Decrement Death
Column Header:	death_rate_dep
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	0
Discount Timing:	Middle
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.6.2.26      *lapse\_rate***

Description:	Lapse rate in period
Help:	
Modified On:	12/19/2024 4:09:09 PM (UTC+02:00)
Modified By:	CLAL-INS\arikt
Category:	Decrement Lapse
Column Header:	lapse_rate
Combine Groups By:	Average Both

Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.6.2.27      *decrem\_rate***

Description:	Monthly decrement rate
Help:	
Modified On:	3/19/2024 7:07:08 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Decrements Other decrements
Column Header:	decrem_rate
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	0
Discount Timing:	Middle
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.6.2.28      *decrem\_rate\_dep***

Description:	dependent decrement rate
Help:	
Modified On:	8/9/2021 10:21:20 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Decrements Other decrements
Column Header:	decrem_rate_dep
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	0
Discount Timing:	Middle
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False



**5.3.6.2.29**      ***pol\_fee***

Description:	Policy Fee
Help:	
Modified On:	1/8/2023 4:50:24 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Income Statement Income Premium
Column Header:	pol_fee
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.6.2.30**      ***prem\_gross\_no\_scen***

Description:	Premium (including policy fee), not including premium discount scenario
Help:	
Modified On:	11/25/2020 3:03:50 PM (UTC+02:00)
Modified By:	CLAL-INS\NinaB
Category:	Income Statement Income Premium
Column Header:	prem_gross_no_scen
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.6.2.31**      ***premium***

Description:	premium (excluding policy fee)
Help:	
Modified On:	8/24/2021 10:59:39 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Income Statement Income Premium
Column Header:	premium
Combine Groups By:	Sum Both

Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.6.2.32**      ***premium\_disc***

Description:	premium discount
Help:	
Modified On:	3/11/2025 9:30:19 AM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Income Statement Income Premium
Column Header:	premium_disc
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.6.2.33**      ***premium\_disc\_no\_scen***

Description:	premium discount with no discount margin added
Help:	
Modified On:	8/10/2021 10:25:49 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Income Statement Income Premium
Column Header:	premium_disc_no_scen
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.6.2.34**      ***premium\_disc\_no\_shimur***

Description:	premium discount without shimur
Help:	
Modified On:	4/11/2024 6:35:41 PM (UTC+03:00)
Modified By:	CLAL-INS\Arikt
Category:	Income Statement Income Premium
Column Header:	premium_disc_no_shimur
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.6.2.35**      ***premium\_gross***

Description:	Premium (including policy fee)
Help:	
Modified On:	8/9/2021 10:22:24 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Income Statement Income Premium
Column Header:	premium_gross
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.6.2.36**      ***claims\_inflation***

Description:	Accumulated Claims Inflation
Help:	
Modified On:	8/9/2021 10:20:21 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Income Statement Outgo Claims
Column Header:	claims_inflation
Combine Groups By:	Sum Both

Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.6.2.37**      *claims\_inpay*

Description:	Claims inpayment
Help:	
Modified On:	1/30/2024 3:57:32 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Income Statement\Outgo\Claims
Column Header:	claims_inpay
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.6.2.38**      *claims\_inpay\_other*

Description:	Claims inpayment from event thta's not occurring in the first quarter
Help:	
Modified On:	3/15/2023 3:11:23 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Income Statement\Outgo\Claims
Column Header:	claims_inpay_other
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.6.2.39**      ***claims\_inpay\_q1***

Description:	Claims inpayment from event occurring in the first quarter
Help:	
Modified On:	1/16/2024 8:47:15 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Income Statement Outgo Claims
Column Header:	claims_inpay_q1
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.6.2.40**      ***claims\_inpay\_q2***

Description:	Claims inpayment from event occurring in the second quarter
Help:	
Modified On:	1/16/2024 8:47:15 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Income Statement Outgo Claims
Column Header:	claims_inpay_q2
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.6.2.41**      ***claims\_inpay\_q3***

Description:	Claims inpayment from event occurring in the third quarter
Help:	
Modified On:	1/16/2024 8:47:13 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt

Category:	Income Statement Outgo Claims
Column Header:	claims_inpay_q3
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.6.2.42**      *claims\_inpay\_q4*

Description:	Claims inpayment from event occurring in the forth quarter
Help:	
Modified On:	1/16/2024 8:47:14 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Income Statement Outgo Claims
Column Header:	claims_inpay_q4
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.6.2.43**      *claims\_rate\_per*

Description:	Claims Rate Per
Help:	
Modified On:	11/15/2022 9:03:20 AM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Income Statement Outgo Claims
Column Header:	claims_rate_per
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous

Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.6.2.44**      *claims\_rate\_per\_other*

Description:	Claims rate per from event thta's not occurring in the first quarter
Help:	
Modified On:	3/15/2023 3:11:30 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Income Statement Outgo Claims
Column Header:	claims_rate_per_other
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.6.2.45**      *claims\_rate\_per\_q1*

Description:	Claims_rate_per from event occurring in the first quarter
Help:	
Modified On:	1/16/2024 8:55:09 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Income Statement Outgo Claims
Column Header:	claims_rate_per_q1
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.6.2.46**      *claims\_rate\_per\_q2*

Description:	Claims_rate_per from event occurring in the second quarter
--------------	--

Help:  
Modified On: 1/16/2024 8:54:34 PM (UTC+02:00)  
Modified By: CLAL-INS\Arikt  
Category: Income Statement\Outgo\Claims  
Column Header: claims\_rate\_per\_q2  
Combine Groups By: Sum Both  
Combine Periods: Last  
Default sliding Size: -1  
Discount Timing: End  
Discount Use: Yes  
Rate Use: Cash Flow  
Rebase Type: Previous  
Retain Value: Yes  
Override: False  
Virtual: False

#### **5.3.6.2.47**      ***claims\_rate\_per\_q3***

Description: Claims\_rate\_per from event occurring in the third quarter

Help:  
Modified On: 1/16/2024 8:54:01 PM (UTC+02:00)  
Modified By: CLAL-INS\Arikt  
Category: Income Statement\Outgo\Claims  
Column Header: claims\_rate\_per\_q3  
Combine Groups By: Sum Both  
Combine Periods: Last  
Default sliding Size: -1  
Discount Timing: End  
Discount Use: Yes  
Rate Use: Cash Flow  
Rebase Type: Previous  
Retain Value: Yes  
Override: False  
Virtual: False

#### **5.3.6.2.48**      ***claims\_rate\_per\_q4***

Description: Claims\_rate\_per from event occurring in the forth quarter

Help:  
Modified On: 1/16/2024 8:53:44 PM (UTC+02:00)  
Modified By: CLAL-INS\Arikt  
Category: Income Statement\Outgo\Claims  
Column Header: claims\_rate\_per\_q4  
Combine Groups By: Sum Both  
Combine Periods: Last  
Default sliding Size: -1



Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.6.2.49      *claims\_total***

Description:	Claims Paid (total)
Help:	
Modified On:	11/30/2022 5:45:09 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Income Statement Outgo Claims
Column Header:	claims_total
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.6.2.50      *premium\_if\_b***

Description:	Premium in force at start of period
Help:	
Modified On:	1/18/2023 1:22:35 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Inforce Details Premium
Column Header:	premium_if_b
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.6.2.51**      ***premium\_if\_b\_2***

Description:	Prem IF at start of period - secondary
Help:	
Modified On:	11/16/2022 8:31:44 AM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Inforce Details Premium
Column Header:	premium_if_b_2
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.6.2.52**      ***premium\_if\_e***

Description:	Premium in force at end of period
Help:	
Modified On:	8/9/2021 10:22:51 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Inforce Details Premium
Column Header:	premium_if_e
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.6.2.53**      ***sum\_at\_risk\_if***

Description:	Sum at risk in force
Help:	
Modified On:	8/9/2021 10:23:49 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Inforce Details Sum Assured
Column Header:	sum_at_rsk_if
Combine Groups By:	Sum Both
Combine Periods:	Last

Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.6.2.54      *sum\_insured***

Description:	Sum insured - undecrement
Help:	
Modified On:	1/9/2023 12:01:59 PM (UTC+02:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Inforce Details Sum Assured
Column Header:	sum_insured
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.6.2.55      *sum\_insured\_if\_b***

Description:	Sum insured in force (start of month)
Help:	
Modified On:	8/24/2021 10:59:44 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Inforce Details Sum Assured
Column Header:	sum_insured_if_b
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.6.2.56      *sum\_insured\_if\_b\_2***

Description:	SI IF (start of mth) - secondary
Help:	
Modified On:	9/12/2021 2:13:48 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Inforce Details Sum Assured
Column Header:	sum_insured_if_b_2
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.6.2.57      *sum\_insured\_if\_b\_2\_no\_dec***

Description:	SI IF (start of mth) - secondary w/o decrem
Help:	
Modified On:	9/20/2022 1:00:40 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Inforce Details Sum Assured
Column Header:	sum_insured_if_b_2_no_dec
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.6.2.58      *sum\_insured\_if\_b\_no\_dec***

Description:	Sum insured in force (start of month) w/o decrement
Help:	
Modified On:	9/20/2022 11:31:12 AM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Inforce Details Sum Assured
Column Header:	sum_insured_if_b_no_dec
Combine Groups By:	Sum Both

Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### 5.3.6.2.59 *sum\_insured\_if\_b\_no\_dth*

Description:	Sum insured in force (start of month) w/o death
Help:	
Modified On:	9/20/2022 1:09:46 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Inforce Details Sum Assured
Column Header:	sum_insured_if_b_no_dth
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### 5.3.6.2.60 *pup\_ltc\_col*

Description:	Pup Ltc Col
Help:	
Modified On:	9/9/2021 3:00:41 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Lapses
Column Header:	pup_ltc_col
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.6.2.61**      ***pup\_ltc\_col\_next***

Description:	Pup Ltc Col
Help:	
Modified On:	9/9/2021 5:42:08 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Lapses
Column Header:	pup_ltc_col_next
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.6.2.62**      ***claims\_re***

Description:	Reinsurance Claims Paid
Help:	
Modified On:	6/10/2021 5:03:50 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Reinsurance
Column Header:	claims_re
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	0
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.6.2.63**      ***comm\_re***

Description:	Initial and renewal reins commission
Help:	
Modified On:	1/5/2025 3:18:25 PM (UTC+02:00)
Modified By:	CLAL-INS\arikt
Category:	Reinsurance
Column Header:	comm_re
Combine Groups By:	Sum Both
Combine Periods:	Last

Default sliding Size:	0
Discount Timing:	Beginning
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.6.2.64**      *comm\_re\_prof*

Description:	Reinsurance profit commission
Help:	Reinsurance profit commission Paid every month
Modified On:	7/26/2022 4:37:30 PM (UTC+03:00)
Modified By:	CLAL-INS\ahuvaa
Category:	Reinsurance
Column Header:	comm_re_prof
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	0
Discount Timing:	Beginning
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.6.2.65**      *exp\_re\_nom*

Description:	reinsurance nominal expenses
Help:	
Modified On:	6/10/2021 5:08:30 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Reinsurance
Column Header:	exp_re_nom
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	0
Discount Timing:	Beginning
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.6.2.66**      ***interest\_re***

Description:	Interest on the reinsurance reserve
Help:	
Modified On:	1/1/2025 3:09:01 PM (UTC+02:00)
Modified By:	CLAL-INS\arikt
Category:	Reinsurance
Column Header:	interest_re
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	0
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.6.2.67**      ***premium\_if\_b\_re***

Description:	Reins premium in force at start of perio
Help:	
Modified On:	1/11/2023 7:21:58 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Reinsurance
Column Header:	premium_if_b_re
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	0
Discount Timing:	Beginning
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.6.2.68**      ***premium\_re***

Description:	Reins Premium income at begin. of period
Help:	
Modified On:	1/22/2024 4:22:16 PM (UTC+02:00)
Modified By:	CLAL-INS\arikt
Category:	Reinsurance
Column Header:	premium_re
Combine Groups By:	Sum Both
Combine Periods:	Last



Default sliding Size:	0
Discount Timing:	Beginning
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.6.2.69**      ***profit\_re***

Description:	reinsurance profit for the period
Help:	
Modified On:	5/5/2021 11:40:57 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Reinsurance
Column Header:	profit_re
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	0
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.6.2.70**      ***reserve\_re***

Description:	Reinsurance Reserve
Help:	
Modified On:	11/15/2022 8:31:05 AM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Reinsurance
Column Header:	reserve_re
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.6.2.71      *reserve\_re\_increase***

Description:	Increase in reinsurance reserve
Help:	
Modified On:	9/12/2024 11:49:10 AM (UTC+03:00)
Modified By:	CLAL-INS\arikt
Category:	Reinsurance
Column Header:	reserve_re_increase
Combine Groups By:	Sum Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	Yes
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.6.2.72      *startup***

Description:	Startup
Help:	
Modified On:	1/16/2024 2:03:04 PM (UTC+02:00)
Modified By:	CLAL-INS\arikt
Category:	Setup
Column Header:	startup
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	End
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Previous
Retain Value:	Yes
Override:	False
Virtual:	False

**5.3.6.2.73      *pol\_sub\_year***

Description:	Policy sub year/term
Help:	
Modified On:	8/9/2021 10:21:59 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Timing
Column Header:	pol_sub_year
Combine Groups By:	Average Both
Combine Periods:	Last

Default sliding Size:	0
Discount Timing:	Beginning
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

#### **5.3.6.2.74**      ***proj\_month***

Description:	Projection month
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Timing
Column Header:	proj_month
Combine Groups By:	Average Both
Combine Periods:	Last
Default sliding Size:	-1
Discount Timing:	Beginning
Discount Use:	No
Rate Use:	Cash Flow
Rebase Type:	Current
Retain Value:	Yes
Override:	False
Virtual:	False

### **5.3.6.3 External Functions**

#### **5.3.6.3.1** ***monthly\_rate***

Description:	Convert annual to monthly
Help:	Given an annual interest rate, expressed as i% this external formula returns $i(12)/12\%$ , i.e. the monthly rate which compounds to give the annual rate, eg <code>monthly_rate(10)=0.007974</code>
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Setup

#### **5.3.6.3.2** ***set\_other\_variables***

Description:	Setup other variables
Help:	External function to calculate and/or adjust variables before running projection. This function is called from startup.

Modified On: 1/11/2023 11:32:03 AM (UTC+02:00)  
Modified By: CLAL-INS\ahuvaa  
Category: Setup

### 5.3.6.3.3 *validate\_data*

Description: Validate Data  
Help: This function carries out data validation checks on the input data and skips the model point if errors are detected, via the throw command.  
Modified On: 8/25/2022 2:11:54 PM (UTC+03:00)  
Modified By: CLAL-INS\ahuvaa  
Category: Setup

## 5.3.6.4 Temporary Tables

### 5.3.6.4.1 *claims\_inpay\_res*

Description: basic reserves for claims in payment of PHI  
Help:  
Modified On: 1/24/2024 11:59:28 AM (UTC+02:00)  
Modified By: CLAL-INS\Arikt  
Category: Balance Sheet|Reserves|Basic Reserve  
Number of Rows: 5  
Number of Columns: 5  
Calculation Methods: Cell  
Reset Variables:  
Resetting Type: Reset Temporary Table for every model point and/or projection loop  
Override: False  
Virtual: False

### 5.3.6.4.2 *res\_cx*

Description: Reserve Commutation Cx  
Help:  
Modified On: 3/20/2024 12:38:07 AM (UTC+02:00)  
Modified By: CLAL-INS\Arikt  
Category: Balance Sheet|Reserves|Basic Reserve  
Number of Rows: 130  
Number of Columns: 1  
Calculation Methods: Cell  
Reset Variables:  
Resetting Type: Reset Temporary Table for every model point and/or projection loop  
Override: False  
Virtual: False

#### 5.3.6.4.3 res\_dx

Description:	Reserve Commutation Dx
Help:	
Modified On:	7/28/2021 4:14:53 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Balance Sheet Reserves Basic Reserve
Number of Rows:	130
Number of Columns:	1
Calculation Methods:	Cell
Reset Variables:	
Resetting Type:	Reset Temporary Table for every model point and/or projection loop
Override:	False
Virtual:	False

#### 5.3.6.4.4 res\_lx

Description:	Reserve Commutation lx
Help:	
Modified On:	3/20/2024 12:38:22 AM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	Balance Sheet Reserves Basic Reserve
Number of Rows:	130
Number of Columns:	1
Calculation Methods:	Cell
Reset Variables:	
Resetting Type:	Reset Temporary Table for every model point and/or projection loop
Override:	False
Virtual:	False

#### 5.3.6.4.5 res\_mx

Description:	Reserve Commutation Mx
Help:	
Modified On:	7/28/2021 4:15:15 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Balance Sheet Reserves Basic Reserve
Number of Rows:	130
Number of Columns:	1
Calculation Methods:	Cell
Reset Variables:	
Resetting Type:	Reset Temporary Table for every model point and/or projection loop
Override:	False
Virtual:	False

**5.3.6.4.6 res\_nx**

Description:	Reserve Commutation Nx
Help:	
Modified On:	7/28/2021 4:15:20 PM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Balance Sheet Reserves Basic Reserve
Number of Rows:	130
Number of Columns:	1
Calculation Methods:	Cell
Reset Variables:	
Resetting Type:	Reset Temporary Table for every model point and/or projection loop
Override:	False
Virtual:	False

**5.3.6.4.7 res\_vx**

Description:	Reserve Interest vx
Help:	
Modified On:	7/22/2021 10:50:57 AM (UTC+03:00)
Modified By:	CLAL-INS\joshm
Category:	Balance Sheet Reserves Basic Reserve
Number of Rows:	131
Number of Columns:	1
Calculation Methods:	Cell
Reset Variables:	
Resetting Type:	Reset Temporary Table for every model point and/or projection loop
Override:	False
Virtual:	False

**5.3.6.4.8 claims\_cost\_factors**

Description:	Claims cost factors at point of claim for PHI & LTC
Help:	
Modified On:	11/1/2021 5:40:14 PM (UTC+02:00)
Modified By:	CLAL-INS\joshm
Category:	Income Statement Outgo Claims
Number of Rows:	106
Number of Columns:	2
Calculation Methods:	Cell
Reset Variables:	
Resetting Type:	Reset Temporary Table for every model point and/or projection loop
Override:	False

Virtual: False

#### **5.3.6.4.9 *claims\_inpay\_pv***

Description: Discount to time t of future Claims of PHI & LTC

Help:

Modified On: 2/1/2024 6:45:48 PM (UTC+02:00)

Modified By: CLAL-INS\Arikt

Category: Income Statement\Outgo\Claims

Number of Rows: 5

Number of Columns: 5

Calculation Methods: Cell

Reset Variables:

Resetting Type: Reset Temporary Table for every model point and/or projection loop

Override: False

Virtual: False

#### **5.3.6.4.10 *claims\_inpay\_rate***

Description: Claims in payment rates for PHI & LTC

Help:

Modified On: 1/29/2024 6:04:27 PM (UTC+02:00)

Modified By: CLAL-INS\Arikt

Category: Income Statement\Outgo\Claims

Number of Rows: 5

Number of Columns: 5

Calculation Methods: Cell

Reset Variables:

Resetting Type: Reset Temporary Table for every model point and/or projection loop

Override: False

Virtual: False

#### **5.3.6.4.11 *claims\_inpay\_rate\_pv***

Description:

Help:

Modified On: 1/31/2024 9:57:01 PM (UTC+02:00)

Modified By: CLAL-INS\Arikt

Category: Income Statement\Outgo\Claims

Number of Rows: 5

Number of Columns: 5

Calculation Methods: Cell

Reset Variables:

Resetting Type: Reset Temporary Table for every model point and/or projection loop

Override: False  
Virtual: False

#### **5.3.6.4.12**      *claims\_inpay\_res\_factor*

Description:  
Help:  
Modified On: 1/31/2024 9:57:22 PM (UTC+02:00)  
Modified By: CLAL-INS\Arikt  
Category: Income Statement\Outgo\Claims  
Number of Rows: 5  
Number of Columns: 5  
Calculation Methods: Cell  
Reset Variables:  
Resetting Type: Reset Temporary Table for every model point and/or projection loop  
  
Override: False  
Virtual: False

#### **5.3.6.4.13**      *claims\_inpayment*

Description: Claims in payment for PHI & LTC  
Help:  
Modified On: 5/29/2025 11:28:53 AM (UTC+03:00)  
Modified By: CLAL-INS\arikt  
Category: Income Statement\Outgo\Claims  
Number of Rows: 5  
Number of Columns: 5  
Calculation Methods: Cell  
Reset Variables:  
Resetting Type: Reset Temporary Table for every model point and/or projection loop  
  
Override: False  
Virtual: False

### **5.3.6.5 Scalars**

#### **5.3.6.5.1** *claims\_inflation\_mthly*

Description: Monthly claims inflation percentage  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category:  
Rebase: N/A  
Type: Double



Override: False  
Virtual: False

#### **5.3.6.5.2 *interest\_rein\_mthly***

Description: Monthly interest rate on reinsurance reserves  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category:  
Rebase: N/A  
Type: Double  
Override: False  
Virtual: False

#### **5.3.6.5.3 *sexcode***

Description: Code for sex of insured 1. 0=Male, 1=Female  
Help:  
Modified On: 8/27/2019 4:00:59 PM (UTC+03:00)  
Modified By: CLAL-INS\ninab  
Category:  
Rebase: N/A  
Type: Integer  
Override: False  
Virtual: False

#### **5.3.6.5.4 *t\_start***

Description: projection start point  
Help:  
Modified On: 11/15/2022 9:03:39 AM (UTC+02:00)  
Modified By: CLAL-INS\joshm  
Category:  
Rebase: N/A  
Type: Integer  
Override: False  
Virtual: False

#### **5.3.6.5.5 *temp\_tbl\_size***

Description: temporary table size for phi & ltc  
Help:  
Modified On: 11/15/2022 9:03:35 AM (UTC+02:00)  
Modified By: CLAL-INS\joshm  
Category:  
Rebase: N/A

Type:	Integer
Override:	False
Virtual:	False

#### **5.3.6.5.6 wp\_phi**

Description:	phi wating period
Help:	
Modified On:	1/29/2024 6:12:32 PM (UTC+02:00)
Modified By:	CLAL-INS\Arikt
Category:	
Rebase:	N/A
Type:	Integer
Override:	False
Virtual:	False

#### **5.3.6.5.7 int\_rate\_res\_hy**

Description:	Half-yearly interest rate for reserves
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Balance Sheet Reserves
Rebase:	N/A
Type:	Double
Override:	False
Virtual:	False

#### **5.3.6.5.8 int\_rate\_res\_mthly**

Description:	Monthly interest rate for reserves
Help:	
Modified On:	8/27/2019 4:00:59 PM (UTC+03:00)
Modified By:	CLAL-INS\ninab
Category:	Balance Sheet Reserves
Rebase:	N/A
Type:	Double
Override:	False
Virtual:	False

## **5.4 Custom Outputs**

<No Custom Outputs Exist>

## 5.5 Externs and DLLs/Libs

<No Externs and DLLs/Libs Exist>

## 6 Appendix

### 6.1 Formulas

All lines are shown per Formula.

#### 6.1.1 Model Classes

##### 6.1.1.1 ann\_cflow

##### 6.1.1.1.1 Columns

##### 6.1.1.1.1.1 cashflow\_b\_post\_ret

```
if (t <= commence_period_w || t > maturity_period_ann)
    return 0.0;
```

```
if (annuity_pmt_curr_tot == 0)
    return 0.0;
```

```
return - pmt_total(t) - expense_ren_perc_post_ret(t);
```

##### 6.1.1.1.1.2 res\_ann\_deficiency

```
if (t < 0 || t >= maturity_period_w || zeroise_ann_def=="Y")
    return 0.0;
```

```
if (life->retirement_age_lookup(1) > takeup_age)
    return 0.;
```

```
if(life->submodel == "TERM" || annuity_pmt_curr_tot == 0)
    return NO_AVG;
```

```
if(life->submodel == "TRAD" && (annuitization_rate<=0.00001 || !(eq(life->ben_class,"GIMLA")))) //
TRAD
```

```
    return 0.0;
```

```
if(life->annuitization_rate<=0.00001 || life->res_prop_kitzba<=0.0) // *** need way to distinguish
between policies with and without guarantees, and with and without kitzva option
    return 0.0;
```

```
int age_takeup_local = takeup_age;
```

```
double tagold_money = 0.0;
double new_money = 0.0;
double reserve_noret = 0.0;
double units_e_old_noret = 0.0;
double units_e_noret = 0.0;
double units_e_hon_noret = 0.0;
double tagold_money_temp = 0.0;
```

```

double prop_new_money = 0.0;
double prat_money = 0.0;
double piz_money = 0.0;
double tagnew_money = 0.0;

if(dump_vars == "Y"){

    log_strm << t<<": Takeup age: "<< age_takeup_local<<endl;
    log_strm<<"Old units: "<<life->units_e_old(t)<<endl;
    log_strm<<"Ret prop col t+1: "<<life->ret_prop_col(t+1)<<endl;
    log_strm<<"Ret prop array: "<<ret_prop_array[ann_index_map[takeup_age]]<<endl;
    log_strm<<"Retirement_prop: "<<retirement_prop<<endl;

}

//Split of current savings balance into prat, oldtag, newtag and piz
if(life->submodel == "TRAD"){ //TRAD
    if(life->ret_prop_col(t+1) > SMALL_DOUBLE){
        reserve_noret = trad->reserve_basic(t)/life->ret_prop_col(t+1) *
ret_prop_array[ann_index_map[takeup_age]] * retirement_prop;
    }

    tagold_money = (trad->res_basic_act_old(t) + trad->res_basic_pup_old(t))
                    /life->ret_prop_col(t+1)
                    * ret_prop_array[ann_index_map[takeup_age]]
                    * retirement_prop;

    tagold_money_temp = tagold_money;

    new_money = max(reserve_noret * life->res_prop_kitzba- tagold_money, 0.0); // new money
in-force at time t

    if(dump_vars == "Y"){
        log_strm << t<<": Takeup age: "<< age_takeup_local<<endl;
        log_strm<<"Tagold money: "<<tagold_money<<endl;
        log_strm<<"New money: "<<new_money<<endl;
    }

    //Add bonus for before mat date

    tagnew_money = (trad->res_basic_act_newtag(t) + trad->res_basic_pup_newtag(t))
                    /life->ret_prop_col(t+1)
                    * ret_prop_array[ann_index_map[takeup_age]]
                    * retirement_prop;

    piz_money = (trad->res_basic_act_piz(t) + trad->res_basic_pup_piz(t))
                 /life->ret_prop_col(t+1)
                 * ret_prop_array[ann_index_map[takeup_age]]
                 * retirement_prop;

    prat_money = (trad->res_basic_act_prat(t) + trad->res_basic_pup_prat(t))
                  /life->ret_prop_col(t+1)
                  * ret_prop_array[ann_index_map[takeup_age]]
                  * retirement_prop;

    if(t < life->mat_period_original && (tagold_money + new_money) > 0){

```

```

tagold_money = tagold_money
                + (trad->bonus_if(t) + trad->bonus_if_pup(t) )
                /life->ret_prop_col(t+1)
                * ret_prop_array[ann_index_map[takeup_age]]
                * retirement_prop
                * tagold_money
                / (tagold_money_temp + new_money);

tagnew_money = tagnew_money
                + (trad->bonus_if(t) + trad->bonus_if_pup(t) )
                /life->ret_prop_col(t+1)
                * ret_prop_array[ann_index_map[takeup_age]]
                * retirement_prop
                * (tagnew_money + piz_money) // Add piz part of bonus to
tagnew
                / (tagold_money_temp + new_money);

piz_money =      piz_money; //No bonus on piz

prat_money =    prat_money
                + (trad->bonus_if(t) + trad->bonus_if_pup(t) )
                /life->ret_prop_col(t+1)
                * ret_prop_array[ann_index_map[takeup_age]]
                * retirement_prop
                * prat_money
                / (tagold_money_temp + new_money);

new_money = new_money
                + (trad->bonus_if(t) + trad->bonus_if_pup(t) )
                /life->ret_prop_col(t+1)
                * ret_prop_array[ann_index_map[takeup_age]]
                * retirement_prop
                * new_money
                / (tagold_money_temp + new_money);

}

if(dump_vars == "Y"){
log_strm << "Including bonus:"<<endl;
log_strm<<"Tagold money: "<<tagold_money<<endl;
log_strm<<"New money: "<<new_money<<endl;
}

}else{
    if(life->ret_prop_col(t+1) > SMALL_DOUBLE){
        tagold_money = life->units_e_old(t)/life->ret_prop_col(t+1) *
ret_prop_array[ann_index_map[takeup_age]] * retirement_prop;
        tagnew_money = life->units_e_newtag(t)/life->ret_prop_col(t+1) *
ret_prop_array[ann_index_map[takeup_age]] * retirement_prop;
        prat_money = life->units_e_prat(t)/life->ret_prop_col(t+1) *
ret_prop_array[ann_index_map[takeup_age]] * retirement_prop;
        piz_money = life->units_e_piz(t)/life->ret_prop_col(t+1) *
ret_prop_array[ann_index_map[takeup_age]] * retirement_prop;
    }
}

```

```

    }

}

if(dump_vars == "Y"){
    log_strm<<"Old money: "<<tagold_money<<endl;
    log_strm<<"Prat money: "<<prat_money<<endl;
    log_strm<<"Piz money: "<<piz_money<<endl;
    log_strm<<"Tag new money: "<<tagnew_money<<endl;
}

// Set T factors (annuity deficiency percentage)

double ann_ratio_tagnew = ann_ratio_res ;
double ann_ratio_other = ann_ratio_res * (1+antisel_margin) ; // extra margin on annuity factor
for old money and if past maturity age (assume selective mortality for annuitants without tax
incentive)

if(dump_vars == "Y"){
    log_strm<<"T-factor: "<<ann_ratio_res<<endl;
}

// Set Discount Factor
double annuity_def_res_inv_margin = life->ann_def_res_inv_margin;
if (eq(life->par_nonpar,"P"))
    annuity_def_res_inv_margin = life->ann_def_res_inv_margin_par;

double discount_factor = pow(1. + annuity_def_res_inv_margin/100. , -max(0., age_takeup_local -
life->age_last(t))) ;

// Set Survival factor, allowing for mortality and lapses
double survive = life->survive_tbl; // survival factor (mortality only) from t until age 65 (based
on regulatory mortality)

if(dump_vars == "Y"){
    log_strm<<"Discount factor: "<<discount_factor<<endl;
    log_strm<<"Survival factor: "<<survive<<endl;
}

if (life->age_last(t) < life->min_retirement_age){
    if (eq(life->par_nonpar, "N"))
        survive = survive * pow( 1 - life->res_anndef_lapse, max(0., life->min_retirement_age
- life->age_last(t)) );
    else
        survive = survive * pow( 1 - life->res_anndef_lapse_par, max(0., life-
>min_retirement_age - life->age_last(t)) );
}

if(dump_vars == "Y"){
    log_strm<<"Survival factor with lapses: "<<survive<<endl;
}

double take_up_pizz = annuity_takeup_piz_res/100. ;

```

```

double res_factor = max(ann_ratio_other * discount_factor-1,0);
double res_tag_factor = max(ann_ratio_tagnew * discount_factor-1,0);

if(dump_vars == "Y"){
    log_strm<<"Res factor other: "<<res_factor<<endl;
    log_strm<<"Res factor newtag: "<<res_tag_factor<<endl;
}

if (life->submodel != "TRAD" && (atoi(life->fund) <53 || (atoi(life->fund) > 400 &&
xint(atoi(life->fund)/10)< 53))) {
    res_factor = ann_ratio_other * discount_factor-1 ;
    res_tag_factor = ann_ratio_tagnew * discount_factor-1 ;
}

// reset annuitisation rate
if (life->submodel == "TRAD" && trad->reserve_basic(t)>0.0)
    annuitization_rate = (prat_money * annuity_takeup_old/100.
        + piz_money * take_up_pizz
        + tagold_money * annuity_takeup_old/100.
        + tagnew_money * annuity_takeup_new_tag/100 ) / trad->reserve_basic(t);

double extra_res = (
    prat_money * annuity_takeup_prat_res/100. *res_factor
    + piz_money * take_up_pizz * res_factor
    + tagold_money * annuity_takeup_old_res/100. * res_factor
    + tagnew_money * annuity_takeup_new_tag_res/100 *
res_tag_factor
    ) * survive;

return max(extra_res,0.0);

```

#### 6.1.1.1.1.3 claims\_annuity\_nogt\_pv

```

if (t < commence_period_w || t > maturity_period_ann)
    return 0.0;

if (annuity_pmt_curr_tot == 0)
    return 0.0;

int proj_yr = xint(life->proj_year(t+1));
if(eq(life->projection_type_int, "Rollup"))
    proj_yr = xint(life->proj_year_rollup(t+1));

if (t >= maturity_period_w)
    return claims_annuity_nogt_pv(t+1) * life->ann_v_month_t[proj_yr]
        + pmt_total_nogt(t+1);

return 0.0;

```

#### 6.1.1.1.1.4 claims\_annuity\_pv

```

if (t < commence_period_w || t > maturity_period_ann)
    return 0.0;

if (annuity_pmt_curr_tot == 0)
    return 0.0;

```

```

int proj_yr = xint(life->proj_year(t+1));
if(eq(life->projection_type_int, "Rollup"))
    proj_yr = xint(life->proj_year_rollup(t+1));

if (t >= maturity_period_w)

    return claims_annuity_pv(t+1) * life->ann_v_month_t[proj_yr]
        + pmt_total(t+1);

return 0.0;

```

#### 6.1.1.1.1.5 int\_rate\_annuity\_reserve

```

if (t <= commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if (annuity_pmt_curr_tot == 0)
    return NO_AVG;

int inv_year = max(min(life->proj_year(t), 100), 0);
double inv_prop = life->free_inv_prop_t[inv_year];

return freeinv_rate_res_ann/100.
    * inv_prop
    + life->invinc/100.
    * (1 - inv_prop);

```

#### 6.1.1.1.1.6 res\_basic\_jl\_1

```

if (t < commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if (eq(joint_life_status,"Single") || ann_pmt_curr_jl == 0)
    return NO_AVG;

double res_1 = res_payment_pv_1(t+1);

if (surv_1_res(t+1) > 0)
    res_1 = res_1 * surv_1(t+1) / surv_1_res(t+1);

if (t == maturity_period_w)
    return res_1 * (1.+exp_res/100.);

return (res_1)
    *(1.+exp_res/100.) * bonus_index_jl_1(t);

```

#### 6.1.1.1.1.7 res\_payment\_1

```

if (t <= commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if (annuity_pmt_curr_tot == 0)
    return NO_AVG;

if (age_pol_1(t) > life->omega_age_w) // omega age allows for table adjustment
    return 0.0;

```



```
return ann_pmt_curr_jl * surv_1_res(t-1);
```

#### 6.1.1.1.1.8 res\_payment\_2

```
if (t <= commence_period_w || t > maturity_period_ann)
  return NO_AVG;
```

```
if (annuity_pmt_curr_tot == 0)
  return NO_AVG;
```

```
if (age_pol_2(t) > life->omega_age_w) // omega age allows for table adjustment
  return 0.0;
```

```
if (t > 0)
  return ann_pmt_curr_jl * surv_2_joint_life_res(t-1);
```

```
return 0.0;
```

#### 6.1.1.1.1.9 res\_payment\_pv\_1

```
if (t <= commence_period_w || t > maturity_period_ann)
  return NO_AVG;
```

```
if (annuity_pmt_curr_tot == 0)
  return NO_AVG;
```

```
double v = pow(1. + int_rate_annuity_reserve(t+1), -1./12.);
```

```
return res_payment_pv_1(t+1) * v
      + res_payment_1(t+1); //Paid at start of month, so does not need to be discounted
```

#### 6.1.1.1.1.10 res\_payment\_pv\_2

```
if (t <= commence_period_w || t > maturity_period_ann)
  return NO_AVG;
```

```
if (annuity_pmt_curr_tot == 0)
  return NO_AVG;
```

```
double v = pow(1. + int_rate_annuity_reserve(t+1), -1./12.);
```

```
return res_payment_pv_2(t+1) * v
      + res_payment_2(t+1); //Paid at start of month, so does not need to be discounted
```

#### 6.1.1.1.1.11 ann\_takeup\_rate

```
if ((t < 0) || (t >= maturity_period_w))
  return 0.0;
```

```
if(life->submodel == "TERM")
  return NO_AVG;
```

```
if (life->submodel=="TRAD" && (annuitization_rate<=0.00001 || !(eq(life->ben_class,"GIMLA"))))
  return 0.0;
```

```
if(life->annuitization_rate<=0.00001 || life->res_prop_kitzba<=0.0) // *** need way to distinguish
between policies with and without guarantees, and with and without kitzva option
  return 0.0;
```

```

if (life->age_last(t)>80.0)
    return 0.0;

// reset annuitisation rate
double ann_rate = 0.0;

if (life->submodel == "TRAD")
{
    double tagold_money = 0.0;
    double new_money = 0.0;
    double prat_money = 0.0;
    double piz_money = 0.0;
    double tagnew_money = 0.0;

    //Split of current savings balance into prat, oldtag, newtag and piz

    if (trad->reserve_basic(t) > 0){
        tagold_money = trad->res_basic_act_old(t) + trad->res_basic_pup_old(t)//Base part
            + (trad->res_basic_act_old(t) + trad->res_basic_pup_old(t))
            / trad->reserve_basic(t)
            * (trad->bonus_if(t) + trad->bonus_if_pup(t)); //Part of bonus
        belonging to old

        tagnew_money = (trad->res_basic_act_newtag(t) + trad->res_basic_pup_newtag(t))
            + (trad->res_basic_act_newtag(t) + trad-
>res_basic_pup_newtag(t)
            + trad->res_basic_act_piz(t) + trad-
>res_basic_pup_piz(t))
            / trad->reserve_basic(t)
            * (trad->bonus_if(t) + trad->bonus_if_pup(t)); //Part of
        bonus belonging to newtag;

        piz_money = (trad->res_basic_act_piz(t) + trad->res_basic_pup_piz(t));

        prat_money = trad->res_basic_act_prat(t) + trad->res_basic_pup_prat(t)
            + (trad->res_basic_act_prat(t) + trad->res_basic_pup_prat(t))
            / trad->reserve_basic(t)
            * (trad->bonus_if(t) + trad->bonus_if_pup(t)); //Part of bonus
        belonging to prat;;
    }

    if(dump_vars == "Y"){
        log_strm << t<<": Takeup age: "<< takeup_age<<endl;
        log_strm<<"Old money: "<<tagold_money<<endl;
        log_strm<<"Newtag money: "<<tagnew_money<<endl;
        log_strm<<"Piz money: "<<piz_money<<endl;
        log_strm<<"Prat money: "<<prat_money<<endl;
        log_strm<<"Total res: "<<trad->reserve_basic(t)<<endl;
    }
}

```

```

if ( trad->reserve_basic(t)>0.0){
  ann_rate = (prat_money * annuity_takeup_prat/100.
    + piz_money * annuity_takeup_piz/100.
    + tagold_money * annuity_takeup_old/100.
    + tagnew_money * annuity_takeup_new_tag/100. ) / (trad->reserve_basic(t) + trad-
    >bonus_if(t) + trad->bonus_if_pup(t));

  if (life->margin_add=="Y") {
    ann_rate = (prat_money * min(annuity_takeup_prat/100.*(1 + life-
    >margin_annuity_takeup/100.), life->annuity_takeup_max)
    + piz_money * min(annuity_takeup_piz/100.*(1 + life-
    >margin_annuity_takeup/100.), life->annuity_takeup_max)
    + tagold_money * min(annuity_takeup_old/100.*(1 + life-
    >margin_annuity_takeup/100.), life->annuity_takeup_max)
    + tagnew_money * min(annuity_takeup_new_tag/100.*(1 + life-
    >margin_annuity_takeup/100.), life->annuity_takeup_max) ) / (trad->reserve_basic(t) + trad-
    >bonus_if(t) + trad->bonus_if_pup(t));
  }
  return ann_rate;
}

}
else
{

if (life->units_e(t)>0.0){
  ann_rate = (life->units_e_prat(t) * annuity_takeup_prat/100.
    + life->units_e_piz(t) * annuity_takeup_piz/100.
    + life->units_e_old(t) * annuity_takeup_old/100.
    + life->units_e_newtag(t) * annuity_takeup_new_tag/100. ) / life-
    >units_e(t);
  //
  if (life->margin_add=="Y") {
    ann_rate = (life->units_e_prat(t) * min(annuity_takeup_prat/100.*(1 + life-
    >margin_annuity_takeup/100.), life->annuity_takeup_max)
    + life->units_e_piz(t) * min(annuity_takeup_piz/100.*(1 + life-
    >margin_annuity_takeup/100.), life->annuity_takeup_max)
    + life->units_e_old(t) * min(annuity_takeup_old/100.*(1 + life-
    >margin_annuity_takeup/100.), life->annuity_takeup_max)
    + life->units_e_newtag(t) * min(annuity_takeup_new_tag/100.*(1 + life-
    >margin_annuity_takeup/100.), life->annuity_takeup_max) ) / life->units_e(t);

  }
  return ann_rate;
}

}
return 0.0;

```

#### 6.1.1.1.1.12 ann\_certain\_fac

```

if (t < commence_period_w || t > maturity_period_ann)
  return NO_AVG;

if (annuity_pmt_curr_tot == 0)
  return NO_AVG;

if (eq(life->paid_up,"G") && (gteed_term ==0 || t > gteed_term))

```

```

    return NO_AVG;

if (!eq(life->paid_up,"G") && t > commence_period_w + gteed_term)
    return NO_AVG;

double v_rate_res_m = pow(res_vx_ann(age_pol_1(t)+1,sexcode_1),1./12.);

if(res_vx_ann(age_pol_1(t),sexcode_1) > 0.)
    v_rate_res_m =
pow((res_vx_ann(age_pol_1(t)+1,sexcode_1)/res_vx_ann(age_pol_1(t),sexcode_1)),1./12.);

return 1+ ann_certain_fac(t+1)*v_rate_res_m;

```

#### 6.1.1.1.13 ann\_defer\_fac

```

if (t < commence_period_w || t > maturity_period_ann || annuity_pmt_curr_tot == 0)
    return NO_AVG;

if((life->ann_maslul<=100 || life->ann_death == 1 ) && eq(life->paid_up,"G"))
    return NO_AVG;

if (t < commence_period_w)
    return 0.0;

if (t == commence_period_w)
    return ann_defer_fac(t+1);

int age_ann_defer = xint(age_ann_start_1+ gteed_term/12);
int temp_gteed_term = commence_period_w + gteed_term;
if (eq(life->paid_up,"G")){
    age_ann_defer = life->age_at_issue + xint(life->ann_maslul/100.);
    temp_gteed_term = gteed_term;
}

if (age_pol_1(t) > 120 || (age_ann_start_1+ gteed_term/12 > 120))
    return 0.0;

if (t > temp_gteed_term )
    return annuity_fac_1(t);

return (res_nx_ann(age_ann_defer,sexcode_1)/ res_dx_ann(age_ann_defer,sexcode_1)-
11./24.)*res_dx_ann(age_ann_defer,sexcode_1)/res_dx_ann(xint(age_pol_1(t)),sexcode_1)*12.;

```

#### 6.1.1.1.14 annuity\_fac\_1

```

if (t <= commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if (annuity_pmt_curr_tot == 0)
    return NO_AVG;

if (age_pol_1(t) == 121)
    return annuity_fac_1(t-1);

if (age_pol_1(t) > 121)
    return 0.0;

return (res_nx_ann(xint(age_pol_1(t)),sexcode_1)/ res_dx_ann(xint(age_pol_1(t)),sexcode_1)-
11./24.)*12.;

```

**6.1.1.1.15 assurance\_fac\_1**

```
if (t < commence_period_w || t > maturity_period_ann)
    return NO_AVG;
```

```
if (death_ben == "N" || annuity_pmt_curr_tot == 0)
    return NO_AVG;
```

```
int age_now = xint(age_pol_1(t));
```

```
return res_mx_ann(age_now,0)/ res_dx_ann(age_now,0);
```

**6.1.1.1.16 bonus\_index**

```
if (t < commence_period_w || t > maturity_period_ann)
    return NO_AVG;
```

```
if (annuity_pmt_curr_tot == 0)
    return NO_AVG;
```

```
if (t == commence_period_w+1 || (eq(life->paid_up,"G")&& t <= 1))
    return 1.0;
```

```
if (mgt_fee_fixed_temp == 0.0 && mgt_fee_variable == 0.0 && mgt_fee_fixed_max == 0.0)
    return 1.0;
```

```
int proj_yr = xint(life->proj_year(t));
if(eq(life->projection_type_int, "Rollup"))
    proj_yr = xint(life->proj_year_rollup(t));
proj_yr= max(proj_yr, 0);
```

```
double mgt_fixed = 0.0;
double bonus_period = 0.0;
```

```
double temp_inv_rate = 0.0;
double temp_ann_inv_rate = 0.0;
```

```
if (life->margin_add_asset == "Y" && t == 1 && life->par_nonpar == "P"){
    temp_inv_rate = life->asset_shock;
    temp_ann_inv_rate = life->asset_shock;
}
```

```
else{
    temp_inv_rate = life->inv_rate_mth_t[proj_yr];
    temp_ann_inv_rate = life->ann_inv_rate_mth_t[proj_yr];
}
```

```
if(mgt_fixed_max_mth != 0.0 && eq(life->ben_class,"profil")){
```

```
    if (temp_inv_rate <= rate_tarif_mth)
        mgt_fixed = 0.0;
```

```
    else
        mgt_fixed = min(temp_inv_rate - rate_tarif_mth, mgt_fixed_max_mth);
```

```
        if ((life->inv_rate_mth_t[max(xint(life->proj_year(t-13)), 0)] - rate_tarif_mth)<
mgt_fixed_max_mth && life->inv_rate_mth_t[max(xint(life->proj_year(t-13)), 0)] > rate_tarif_mth)
            mgt_fixed = min(mgt_fixed_max_mth, mgt_fixed + life->inv_rate_mth_t[max(xint(life->proj_year(t-13)), 0)] - rate_tarif_mth - mgt_fixed_max_mth);
```

```

    }
else
    mgt_fixed = mgt_fixed_mth;

bonus_period = (((1+ temp_ann_inv_rate)
                *(1-mgt_fixed) -1)
                *(1-mgt_fee_variable/100.) - rate_tarif_mth) + 1;
if(eq(life->paid_up,"G")){
    bonus_period = (((1+ temp_ann_inv_rate)
                    *(1-mgt_fixed) -1)
                    *(1-mgt_fee_variable/100.)+1)/(1+ rate_tarif_mth);
}
return bonus_index(t-1) * bonus_period;

```

#### 6.1.1.1.17 res\_basic\_dth

```

if (t < commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if (annuity_pmt_curr_tot == 0)
    return NO_AVG;

double reserve_temp = 0.0;
double interpol = life->pol_month(t)/12.;
double reserve_fact = 0.0;

double ax_down = assurance_fac_1(t);
double ax_up = 0.0;
if(t+12 <= maturity_period_ann)
    ax_up = assurance_fac_1(t+12);

reserve_fact = ax_down*(1.-interpol) + ax_up * interpol;

if (t == maturity_period_w)
    return reserve_fact*death_ben_curr*(1.+exp_res/100.)*surv_1(t);

return reserve_fact*death_ben_curr*(1.+exp_res/100.)*surv_1(t) * bonus_index_dth(t);

```

#### 6.1.1.1.18 res\_basic\_gt

```

if (t < commence_period_w || t > maturity_period_ann)
    return 0.0;

if (annuity_pmt_curr_tot == 0)
    return 0.0;

double reserve_temp = 0.0;
double interpol = life->pol_month(t)/12.;
double reserve_fact = 0.0;

double ax_down = ann_defer_fac(t);
double ax_up = 0.0;
if(t+12 <= maturity_period_ann)
    ax_up = ann_defer_fac(t+12);

reserve_fact = ax_down*(1.-interpol) + ax_up * interpol;

```

```

if (t == maturity_period_w)
    return (reserve_fact*surv_1(t)+ ann_certain_fac(t+1))*(1.+exp_res/100.)*ann_pmt_curr_gteed;

return (reserve_fact*surv_1(t)+ ann_certain_fac(t+1))*(1.+exp_res/100.)*ann_pmt_curr_gteed *
bonus_index_gteed(t);

```

#### 6.1.1.1.1.19 res\_basic\_gt\_su

```

if (t < commence_period_w || t > maturity_period_ann)
    return 0.0;

if(surv_gteed(t)<1.)
    return 0.0;

return res_basic_gt(t);

```

#### 6.1.1.1.1.20 res\_basic\_jl

```

if (t < commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if (eq(joint_life_status,"Single") || annuity_pmt_curr_tot == 0)
    return NO_AVG;

double res_1 = res_payment_pv_1(t+1);

if (surv_1_res(t+1) > 0)
    res_1 = res_1 * surv_1(t+1) / surv_1_res(t+1);

double res_2 = res_payment_pv_2(t+1);

if (surv_2_joint_life_res(t+1) > 0)
    res_2 = res_2 * surv_2_joint_life(t+1) / surv_2_joint_life_res(t+1);

if (t == maturity_period_w)
    return (res_1 + res_2)
        *(1.+exp_res/100.);

return (res_1 * bonus_index_jl_1(t) + res_2 * bonus_index_jl_2(t))
        *(1.+exp_res/100.);

```

#### 6.1.1.1.1.21 res\_basic\_jl\_2

```

if (t < commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if (eq(joint_life_status,"Single") || ann_pmt_curr_jl == 0)
    return NO_AVG;

double res_2 = res_payment_pv_2(t+1);

if (surv_2_joint_life_res(t+1) > 0)
    res_2 = res_2 * surv_2_joint_life(t+1) / surv_2_joint_life_res(t+1);

if (t == maturity_period_w)
    return res_2 * (1.+exp_res/100.);

return (res_2)

```

```
*(1.+exp_res/100.) * bonus_index_jl_2(t);
```

#### 6.1.1.1.1.22 res\_basic\_nogt

```
if (t < commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if (annuity_pmt_curr_tot == 0)
    return NO_AVG;

double reserve_temp = 0.0;
double interpol = life->pol_month(t)/12.;

if (t==commence_period_w)
    interpol=1.0;
double reserve_fact = 0.0;

double ax_down = annuity_fac_1(t);
double ax_up = 0.0;
if(t+12 <= maturity_period_ann)
    ax_up = annuity_fac_1(t+12);

reserve_fact = ax_down*(1.-interpol) + ax_up * interpol;

if (t == maturity_period_w)
    return reserve_fact*(1.+exp_res/100.)*ann_pmt_curr*surv_1(t);

return reserve_fact*(1.+exp_res/100.)*ann_pmt_curr*surv_1(t) * bonus_index_no_gtee(t);
```

#### 6.1.1.1.1.23 reserve\_basic

```
if (t < commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if (annuity_pmt_curr_tot == 0)
    return NO_AVG;

return (res_basic_nogt(t)
        + res_basic_gt(t)
        + res_basic_jl(t)
        + res_basic_dth(t))*antisel_weight_res;
```

#### 6.1.1.1.1.24 bonus\_index\_dth

```
if (t < commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if (death_ben == "N")
    return NO_AVG;

if (t == commence_period_w+1 || (eq(life->paid_up,"G")&& t <= 1))
    return 1.0;

if(fund_type == "N")
    return 1.0;

if (res_basic_dth(t-1) == 0.)
    return bonus_index_dth(t-1);
```



```

double bonus_period = 0.0;

bonus_period = int_cred_dth(t)
               - mgt_fee_fixed_dth(t)
               - mgt_fee_var_dth(t)
               - tarif_deduction_dth(t);

bonus_period = bonus_period / res_basic_dth(t-1);

return bonus_index_dth(t-1)
       * (1+bonus_period);

```

#### 6.1.1.1.1.25 bonus\_index\_gteed

```

if (t < commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if (ann_pmt_curr_gteed == 0)
    return NO_AVG;

if (t == commence_period_w+1 || (eq(life->paid_up,"G")&& t <= 1))
    return 1.0;

if(fund_type == "N")
    return 1.0;

if (res_basic_gt(t-1) == 0.)
    return bonus_index_gteed(t-1);

double bonus_period = 0.0;

bonus_period = int_cred_gteed(t)
               - mgt_fee_fixed_gtd(t)
               - mgt_fee_var_gtd(t)
               - tarif_deduction_gteed(t);

bonus_period = bonus_period / res_basic_gt(t-1);

return bonus_index_gteed(t-1)
       * (1+bonus_period);

```

#### 6.1.1.1.1.26 bonus\_index\_jl\_1

```

if (t < commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if (ann_pmt_curr_jl == 0)
    return NO_AVG;

if (t == commence_period_w+1 || (eq(life->paid_up,"G")&& t <= 1))
    return 1.0;

if(fund_type == "N")
    return 1.0;

if (res_basic_jl_1(t-1) == 0.)
    return bonus_index_jl_1(t-1);

```

```

double bonus_period = 0.0;

bonus_period = int_cred_jl1(t)
               - mgt_fee_fixed_jl1(t)
               - mgt_fee_var_jl1(t)
               - tarif_deduction_jl1(t);

bonus_period = bonus_period / res_basic_jl_1(t-1);

return bonus_index_jl_1(t-1)
       * (1+bonus_period);

```

#### 6.1.1.1.1.27 **bonus\_index\_jl\_2**

```

if (t < commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if (ann_pmt_curr_jl == 0)
    return NO_AVG;

if (t == commence_period_w+1 || (eq(life->paid_up,"G")&& t <= 1))
    return 1.0;

if(fund_type == "N")
    return 1.0;

if (res_basic_jl_2(t-1) == 0.)
    return bonus_index_jl_1(t-1);

double bonus_period = 0.0;

bonus_period = int_cred_jl2(t)
               - mgt_fee_fixed_jl2(t)
               - mgt_fee_var_jl2(t)
               - tarif_deduction_jl2(t);

bonus_period = bonus_period / res_basic_jl_2(t-1);

return bonus_index_jl_2(t-1)
       * (1+bonus_period);

```

#### 6.1.1.1.1.28 **bonus\_index\_no\_gtee**

```

if (t < commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if (ann_pmt_curr == 0)
    return NO_AVG;

if (t == commence_period_w+1 || (eq(life->paid_up,"G")&& t <= 1))
    return 1.0;

if(fund_type == "N")
    return 1.0;

if (res_basic_nogt(t-1) == 0.)
    return bonus_index_no_gtee(t-1);

```

```
double bonus_period = 0.0;

bonus_period = int_cred_no_gteed(t)
               - mgt_fee_fixed_nogt(t)
               - mgt_fee_var_nogt(t)
               - tarif_deduction_no_gteed(t);

bonus_period = bonus_period / res_basic_nogt(t-1);

return bonus_index_no_gtee(t-1)
       * (1+bonus_period);
```

#### 6.1.1.1.1.29 **bor\_acc\_dth**

```
if (t< commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if (death_ben == "N")
    return NO_AVG;

if(fund_type == "N" || mgt_fee_var == 0.0)
    return 0.0;

return bor_acc_notg(t); //At present there can only be death benefit on non-guaranteed single life
```

#### 6.1.1.1.1.30 **bor\_acc\_gtd**

```
if (t< commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if(fund_type == "N" || mgt_fee_var == 0.0 || ann_pmt_curr_gteed == 0.)
    return 0.0;

if (t == commence_period_w || (eq(life->paid_up,"G")&& t <= 0)){

    if (init_bor_har< 0){

        if(eq(life->paid_up,"G"))
            return (-1)* init_bor_har;

        return init_bor_har
            * (-1)//Bor is managed as positive number
            * gtee_ppn_temp/100.;

    }

    return 0.0;

}

if (har_accum_gtd(t) > 0.)
    return 0.0;

double bor = bor_acc_gtd(t-1);

if (t > commence_period_w+1)
    bor = bor
```

```

        * surv_per_gteed(t-1);

if (net_interest_rate(t)< 0){

    double new_bor = - mgt_fee_variable/100.
        * (int_cred_gteed(t) - mgt_fee_fixed_gtd(t));

    if(har_return_gtd(t) > 0)
        new_bor = max(new_bor - har_return_gtd(t), 0);

    bor = bor
        + new_bor;//Addition to bor

}

if (net_interest_rate(t) > 0.0)
    bor = bor + bor_return_gtd(t);//Bor returned

return max(bor, 0.0); //Shouldn't really be possible for bor to be negative, but just in case

```

#### 6.1.1.1.1.31 **bor\_acc\_jl1**

```

if (t< commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if(fund_type == "N" || mgt_fee_var == 0.0 || ann_pmt_curr_jl == 0.)
    return 0.0;

if (t == commence_period_w || (eq(life->paid_up,"G")&& t <= 0)){

    if (init_bor_har< 0){

        if(eq(life->paid_up,"G"))
            return (-1)* init_bor_har;

        return init_bor_har
            * (-1)//Bor is managed as positive number
            * joint_life_ppn_temp/100.;

    }

    return 0.0;

}

if (har_acc_jl1(t) > 0.)
    return 0.0;

double bor = bor_acc_jl1(t-1);

if (t > commence_period_w+1)

```

```

        bor = bor
            * surv_per_1(t-1);

if (net_interest_rate(t)< 0){

    double new_bor = - mgt_fee_variable/100.
        * (int_cred_jl1(t) - mgt_fee_fixed_jl1(t));

    if (har_return_jl1(t) > 0)
        new_bor = max(new_bor - har_return_jl1(t), 0);

    bor = bor
        + new_bor;//Addition to bor

}

if (net_interest_rate(t) > 0.0)
    bor = bor + bor_return_jl1(t);//Bor returned

return max(bor, 0.0);//Shouldn't really be possible for bor to be negative, but just in case

```

#### 6.1.1.1.1.32 **bor\_acc\_jl2**

```

if (t<= commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if(fund_type == "N" || mgt_fee_var == 0.0 || ann_pmt_curr_jl == 0. || (eq(life->paid_up,"G")&& t <=
0) || (har_acc_jl2(t)+har_acc_jl1(t)) > 0.)
    return 0.0;

double bor = bor_acc_jl2(t-1);

if (t > commence_period_w+1)
    bor = bor
        * (1. - death_rate_2_b3(t-1));

bor = bor
    + bor_acc_jl1(t-1)
    * surv_2(t-1)
    * death_rate_1(t-1);

if (net_interest_rate(t)< 0){

    double new_bor = - mgt_fee_variable/100.
        * (int_cred_jl2(t) - mgt_fee_fixed_jl2(t));

    if(har_return_jl2(t) > 0)
        new_bor = max(new_bor - har_return_jl2(t), 0);

    bor = bor + new_bor;//Addition to bor

```

```

}

if (net_interest_rate(t) > 0.0)
    bor = bor + bor_return_jl2(t); //Bor returned

return max(bor, 0.0); //Shouldn't really be possible for bor to be negative, but just in case

```

#### 6.1.1.1.1.33 bor\_acc\_notg

```

if (t < commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if(fund_type == "N" || mgt_fee_var == 0.0 || ann_pmt_curr == 0.)
    return 0.0;

if (t == commence_period_w || (eq(life->paid_up,"G") && t <= 0)){
    if (init_bor_har < 0){
        if(eq(life->paid_up,"G"))
            return (-1)* init_bor_har;

        return init_bor_har
            * (-1) //Bor is managed as positive number
            * no_gtee_ppn_temp/100.;
    }

    return 0.0;
}

if (har_acc_nogt(t) > 0.)
    return 0.0;

double bor = bor_acc_notg(t-1);

if (t > commence_period_w+1)
    bor = bor
        * surv_per_1(t-1); //No decrement at initial period

if (net_interest_rate(t) < 0){
    double new_bor = - mgt_fee_variable/100.
        * (int_cred_no_gteed(t) - mgt_fee_fixed_nogt(t));

    if((har_return_nogt(t) + har_return_dth(t)) > 0)
        new_bor = max(new_bor- har_return_nogt(t) - har_return_dth(t), 0);

    bor = bor

```

```

        + new_bor;//Addition to bor
    }

    if (net_interest_rate(t) > 0.0)
        bor = bor + bor_return_nogt(t) + bor_return_dth(t);//Bor returned (including part for death)

    return max(bor, 0.0);//Shouldn't really be possible for bor to be negative, but just in case

```

#### 6.1.1.1.1.34 bor\_return\_dth

```

if (t <= commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if (net_interest_rate(t) < 0.0 || bor_acc_dth(t-1) == 0 || death_ben == "N")
    return 0.0;

double mgt_fee_pos = -mgt_var_no_bor_dth(t); //Management fees available for return

//mgt_fee_pos = mgt_fee_variable/100.
//          * (int_cred_dth(t) - mgt_fee_fixed_dth(t))
//          * (-1.);

if (t == commence_period_w + 1)
    return max(mgt_fee_pos,
               bor_acc_dth(t-1)
               * (-1));

return max(mgt_fee_pos,
           bor_acc_dth(t-1)
           * (-1)
           * surv_per_1(t-1) - bor_return_nogt(t)); //Cannot return more than outstanding bor,
less returned on nogt

```

#### 6.1.1.1.1.35 bor\_return\_gtd

```

if (t <= commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if (net_interest_rate(t) < 0.0 || bor_acc_gtd(t-1) == 0 || ann_pmt_curr_gteed == 0)
    return 0.0;

double mgt_fee_pos = 0.0; //Management fees available for return

mgt_fee_pos = -mgt_var_no_bor_gtd(t);

if (t == commence_period_w + 1)
    return max(mgt_fee_pos,
               bor_acc_gtd(t-1)
               * (-1));

return max(mgt_fee_pos,

```

```

    bor_acc_gtd(t-1)
    * (-1)
    * surv_per_gteed(t-1)); //Cannot return more than outstanding bor

```

#### 6.1.1.1.1.36 bor\_return\_jl1

```

if (t <= commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if (net_interest_rate(t) < 0.0 || bor_acc_jl1(t-1) == 0 || ann_pmt_curr_jl == 0)
    return 0.0;

double mgt_fee_pos = -mgt_var_no_bor_jl1(t); //Management fees available for return

//mgt_fee_pos = mgt_fee_variable/100.
//      * (int_cred_jl1(t) - mgt_fee_fixed_jl1(t))
//      * (-1.);

if (t == commence_period_w + 1)
    return max(mgt_fee_pos,
               bor_acc_jl1(t-1)
               * (-1));

return max(mgt_fee_pos,
           bor_acc_jl1(t-1)
           * (-1)
           * surv_per_1(t-1)); //Cannot return more than outstanding bor

```

#### 6.1.1.1.1.37 bor\_return\_jl2

```

if (t <= commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if (net_interest_rate(t) < 0.0 || (bor_acc_jl2(t-1) + bor_acc_jl1(t-1)) == 0 || ann_pmt_curr_jl == 0)
    return 0.0;

double mgt_fee_pos = -mgt_var_no_bor_jl2(t); //Management fees available for return

//mgt_fee_pos = mgt_fee_variable/100.
//      * (int_cred_jl2(t) - mgt_fee_fixed_jl2(t))
//      * (-1.);

return max(mgt_fee_pos,
           bor_acc_jl2(t-1)
           * (-1)
           * (1. - death_rate_2_b3(t-1))
           + bor_acc_jl1(t-1)
           * (-1)
           * surv_2(t-1)
           * death_rate_1(t-1)
           ); //Cannot return more than outstanding bor

//This is wrong...

```

#### 6.1.1.1.1.38 bor\_return\_nogt

```

if (t <= commence_period_w || t > maturity_period_ann)
    return NO_AVG;

```



```
if (net_interest_rate(t) < 0.0 || bor_acc_notg(t-1) == 0 || ann_pmt_curr == 0)
    return 0.0;
```

```
double mgt_fee_pos = 0.0; //Management fees available for return
```

```
mgt_fee_pos = -mgt_var_no_bor_nogt(t);
```

```
if (t == commence_period_w + 1)
    return max(mgt_fee_pos,
               bor_acc_notg(t-1)
               * (-1));
```

```
return max(mgt_fee_pos,
           bor_acc_notg(t-1)
           * (-1)
           * surv_per_1(t-1)); //Cannot return more than outstanding bor
```

#### 6.1.1.1.1.39 cashflow\_pv

```
if (t < commence_period_w || t > maturity_period_ann)
    return 0.0;
```

```
if (annuity_pmt_curr_tot == 0)
    return 0.0;
```

```
int proj_yr = xint(life->proj_year(t+1));
if(eq(life->projection_type_int, "Rollup"))
    proj_yr = xint(life->proj_year_rollup(t+1));
```

```
if (t >= maturity_period_w)
    return cashflow_pv(t+1) * life->ann_v_month_t[proj_yr]
           + cashflow_b_post_ret(t+1);
```

```
return 0.0;
```

#### 6.1.1.1.1.40 cashflow\_pv\_chetz

```
if (t < commence_period_w || t > maturity_period_ann || life->free_inv_prop_t[1] >= 1.)
    return 0.0;
```

```
if (annuity_pmt_curr_tot == 0)
    return 0.0;
```

```
int proj_yr = xint(life->proj_year(t+1));
if(eq(life->projection_type_int, "Rollup"))
    proj_yr = xint(life->proj_year_rollup(t+1));
```

```
if (t >= maturity_period_w)
    return cashflow_pv_ifrs(t)*(1.-life->max_chetz) + cashflow_pv_res(t)*(life->max_chetz) ;
```

```
return 0.0;
```

**6.1.1.1.1.41 cashflow\_pv\_e**

```

if (t < commence_period_w || t > maturity_period_ann)
    return 0.0;

if (annuity_pmt_curr_tot == 0)
    return 0.0;

int proj_yr = xint(life->proj_year(t+1));
if(eq(life->projection_type_int, "Rollup"))
    proj_yr = xint(life->proj_year_rollup(t+1));

if (t >= maturity_period_w)
    return (cashflow_pv_e(t+1)+ cashflow_b_post_ret(t+1)) * life->ann_v_month_t[proj_yr]
        ;

return 0.0;

```

**6.1.1.1.1.42 cashflow\_pv\_ifrs**

```

if (t < commence_period_w || t > maturity_period_ann || life->free_inv_prop_t[1] >= 1.)
    return 0.0;

if (annuity_pmt_curr_tot == 0)
    return 0.0;

int proj_yr = xint(life->proj_year(t+1));
if(eq(life->projection_type_int, "Rollup"))
    proj_yr = xint(life->proj_year_rollup(t+1));

if (t >= maturity_period_w)
    return cashflow_pv_ifrs(t+1) * life->ann_v_month_t_ifrs[proj_yr]
        + cashflow_b_post_ret(t+1);

return 0.0;

```

**6.1.1.1.1.43 cashflow\_pv\_res**

```

if (t < commence_period_w || t > maturity_period_ann || life->free_inv_prop_t[1] >= 1.)
    return 0.0;

if (annuity_pmt_curr_tot == 0)
    return 0.0;

int proj_yr = xint(life->proj_year(t+1));
if(eq(life->projection_type_int, "Rollup"))
    proj_yr = xint(life->proj_year_rollup(t+1));

if (t >= maturity_period_w)
    return cashflow_pv_res(t+1) * life->v_month_t_int_res
        + cashflow_b_post_ret(t+1);

return 0.0;

```

**6.1.1.1.1.44 har\_acc\_dth**

```

if (t< commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if(fund_type == "N" || mgt_fee_var == 0. || death_ben == "N")
    return 0.0;

return har_acc_nogt(t);

```

**6.1.1.1.1.45 har\_acc\_jl1**

```

if (t< commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if(fund_type == "N" || mgt_fee_var == 0. || ann_pmt_curr_jl == 0.)
    return 0.0;

if (t == commence_period_w || (eq(life->paid_up,"G")&& t <= 0)){

    if(life->cal_month(t) == 12. || init_bor_har < 0)//If year-end, no accumulation. If there is
    bor, no accumulation
        return 0.0;

    if(eq(life->paid_up,"G"))
        return init_bor_har;

    return init_bor_har
        * joint_life_ppn_temp/100.;

}

double har = 0.0;

if(life->cal_month(t) > 1){
    har = har_acc_jl1(t-1);

    if (t > commence_period_w + 1)
        har = har
            * surv_per_1(t-1);

}

har = har + mgt_fee_var_jl1(t);//management fees paid this month

har = har - har_return_jl1(t); //Deduct management fees returned

return max(har, 0.0);

```

**6.1.1.1.1.46 har\_acc\_jl2**

```

if (t< commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if(fund_type == "N" || mgt_fee_var == 0. || ann_pmt_curr_jl == 0.)
    return 0.0;

```

```

if (t == commence_period_w || (eq(life->paid_up,"G")&& t <= 0)){

    if(life->cal_month(t) == 12. || init_bor_har < 0)//If year-end, no accumulation. If there is
    bor, no accumulation
        return 0.0;

    //return initial_management_fee_owing_paid_to_ann
    //      * joint_life_ppn_temp/100.;
    return 0.0; // assume no initial bor/har
    //Note: this does not account for annuities in payment where the main life has already died
    - this is not accounted for elsewhere either

}

double har = 0.0;

if(life->cal_month(t) > 1){
    har = har_acc_jl2(t-1)
        * (1. - death_rate_2_b3(t-1)); //Only being paid on widowed lives anyway

//Add har passed from first life
    har = har + har_acc_jl1(t-1)
        * death_rate_1(t-1)
        * surv_2(t-1);

}

har = har + mgt_fee_var_jl2(t);//management fees paid this month

har = har - har_return_jl2(t); //Deduct management fees returned

return max(har, 0.0);

```

#### 6.1.1.1.1.47 har\_acc\_nogt

```

if (t< commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if(fund_type == "N" || mgt_fee_var == 0. || ann_pmt_curr == 0.)
    return 0.0;

if (t == commence_period_w || (eq(life->paid_up,"G")&& t <= 0)){

    if(life->cal_month(t) == 12. || init_bor_har < 0)//If year-end, no accumulation. If there is
    bor, no accumulation
        return 0.0;

    if(eq(life->paid_up,"G"))
        return init_bor_har;

    return init_bor_har
        * no_gtee_ppn_temp/100.;

}

double har = 0.0;

```

```

if(life->cal_month(t) > 1){
    har = har_acc_nogt(t-1);
    if (t > commence_period_w + 1)
        har = har
            * surv_per_1(t-1);
}

har = har + mgt_fee_var_nogt(t); //management fees paid this month

har = har - har_return_nogt(t) - har_return_dth(t); //Deduct management fees returned (for death as well)

return max(har, 0.0);

```

#### 6.1.1.1.1.48 har\_accum\_gtd

```

if (t< commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if(fund_type == "N" || mgt_fee_var == 0. || ann_pmt_curr_gteed == 0.)
    return 0.0;

if (t == commence_period_w || (eq(life->paid_up,"G")&& t <= 0)){
    if(life->cal_month(t) == 12. || init_bor_har < 0) //If year-end, no accumulation. If there is bor, no accumulation
        return 0.0;

    if(eq(life->paid_up,"G"))
        return init_bor_har;

    return init_bor_har
        * gtee_ppn_temp/100.;
}

double har = 0.0;

if(life->cal_month(t) > 1){
    har = har_accum_gtd(t-1);
    if (t > commence_period_w + 1)
        har = har
            * surv_per_gteed(t-1);
}

har = har + mgt_fee_var_gtd(t); //management fees paid this month

har = har - har_return_gtd(t); //Deduct management fees returned

```

```
return max(har, 0.0);
```

#### 6.1.1.1.1.49 har\_return\_dth

```
if (t< commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if(fund_type == "N" || mgt_fee_var == 0.0 || death_ben == "N")
    return 0.0;

if(har_acc_dth(t-1) == 0.0)
    return 0; //Nothing collected to return

if(net_interest_rate(t) > 0.0)
    return 0.0; //No need to return if interest is positive

if (life->cal_month(t) == 1)
    return 0.0; //Do not carry over to new year

double har_ret = mgt_fee_variable/100.
    * (int_cred_dth(t) - mgt_fee_fixed_dth(t)) * (-1.);

return min(har_ret,
           har_acc_dth(t-1) * surv_per_1(t-1) + har_return_nogt(t));
```

#### 6.1.1.1.1.50 har\_return\_gtd

```
if (t< commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if(fund_type == "N" || mgt_fee_var == 0.0 || ann_pmt_curr_gteed == 0.)
    return 0.0;

if(har_accum_gtd(t-1) == 0.0)
    return 0; //Nothing collected to return

if(net_interest_rate(t) > 0.0)
    return 0.0; //No need to return if interest is positive

if (life->cal_month(t) == 1)
    return 0.0; //Do not carry over to new year

double har_ret = mgt_fee_variable/100.
    * (int_cred_gteed(t) - mgt_fee_fixed_gtd(t)) * (-1.);

return min(har_ret,
           har_accum_gtd(t-1) * surv_per_gteed(t-1));
```

#### 6.1.1.1.1.51 har\_return\_jl1

```
if (t< commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if(fund_type == "N" || mgt_fee_var == 0.0 || ann_pmt_curr_jl1 == 0.)
```

```

    return 0.0;

if(har_acc_jl1(t-1) == 0.0)
    return 0; //Nothing collected to return

if(net_interest_rate(t) > 0.0)
    return 0.0; //No need to return if interest is positive

if (life->cal_month(t) == 1)
    return 0.0; //Do not carry over to new year

double har_ret = mgt_fee_variable/100.
    * (int_cred_jl1(t) - mgt_fee_fixed_jl1(t)) * (-1.);

return min(har_ret,
    har_acc_jl1(t-1) * surv_per_1(t-1));

```

#### 6.1.1.1.1.52 har\_return\_jl2

```

if (t< commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if(fund_type == "N" || mgt_fee_var == 0.0 || ann_pmt_curr_jl == 0.)
    return 0.0;

if((har_acc_jl2(t-1) + har_acc_jl1(t-1)) == 0.0)
    return 0; //Nothing collected to return

if(net_interest_rate(t) > 0.0)
    return 0.0; //No need to return if interest is positive

if (life->cal_month(t) == 1)
    return 0.0; //Do not carry over to new year

double har_ret = mgt_fee_variable/100.
    * (int_cred_jl2(t) - mgt_fee_fixed_jl2(t)) * (-1.);

return min(har_ret,
    har_acc_jl2(t-1)
    * (1. - death_rate_2_b3(t-1))
    + har_acc_jl1(t-1) // Add mgt fees owing from first life
    * surv_2(t-1)
    * death_rate_1(t-1)
    );

```

#### 6.1.1.1.1.53 har\_return\_nogt

```

if (t< commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if(fund_type == "N" || mgt_fee_var == 0.0 || ann_pmt_curr == 0.)
    return 0.0;

if(har_acc_nogt(t-1) == 0.0)
    return 0; //Nothing collected to return

```

```
if(net_interest_rate(t) > 0.0)
    return 0.0; //No need to return if interest is positive

if (life->cal_month(t) == 1)
    return 0.0; //Do not carry over to new year

double har_ret = mgt_fee_variable/100.
    * (int_cred_no_gteed(t) - mgt_fee_fixed_nogt(t)) * (-1.);

return min(har_ret,
           har_acc_nogt(t-1) * surv_per_1(t-1));
```

#### **6.1.1.1.1.54 mgt\_var\_no\_bor\_dth**

```
if (t< commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if(fund_type == "N" || death_ben == "N")
    return 0.0;

if (net_interest_rate(t) > 0.0)
    return mgt_fee_variable/100.
        * (int_cred_dth(t) - mgt_fee_fixed_dth(t));

return 0.0;
```

#### **6.1.1.1.1.55 mgt\_var\_no\_bor\_gtd**

```
if (t< commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if(fund_type == "N" || ann_pmt_curr_gteed == 0.)
    return 0.0;

if (net_interest_rate(t) > 0.0)
    return mgt_fee_variable/100.
        * (int_cred_gteed(t) - mgt_fee_fixed_gtd(t));

return 0.0;
```

#### **6.1.1.1.1.56 mgt\_var\_no\_bor\_jl1**

```
if (t< commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if(fund_type == "N" || ann_pmt_curr_jl1 == 0.)
    return 0.0;

if (net_interest_rate(t) > 0.0)
    return mgt_fee_variable/100.
        * (int_cred_jl1(t) - mgt_fee_fixed_jl1(t));
```



```
return 0.0;
```

#### 6.1.1.1.1.57 mgt\_var\_no\_bor\_jl2

```
if (t< commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if(fund_type == "N" || ann_pmt_curr_jl == 0.)
    return 0.0;

if (net_interest_rate(t) > 0.0)
    return mgt_fee_variable/100.
        * (int_cred_jl2(t) - mgt_fee_fixed_jl2(t));
```

```
return 0.0;
```

#### 6.1.1.1.1.58 mgt\_var\_no\_bor\_nogt

```
if (t< commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if(fund_type == "N" || mgt_fee_var == 0.0 || ann_pmt_curr == 0.)
    return 0.0;

if (net_interest_rate(t) > 0.0)
    return mgt_fee_variable/100.
        * (int_cred_no_gteed(t) - mgt_fee_fixed_nogt(t));
```

```
return 0.0;
```

#### 6.1.1.1.1.59 net\_interest\_rate

```
if (t< commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if (annuity_pmt_curr_tot == 0.)
    return NO_AVG;

int proj_yr = xint(life->proj_year(t));
if(eq(life->projection_type_int, "Rollup"))
    proj_yr = xint(life->proj_year_rollup(t));
proj_yr= max(proj_yr, 0);

double temp_ann_inv_rate = life->ann_inv_rate_mth_t[proj_yr];

if (life->margin_add_asset == "Y" && t == 1 && life->par_nonpar == "P" && !eq(life->submodel,
"ANN") )
    temp_ann_inv_rate = life->asset_shock;

if (life->margin_add_asset == "Y" && t == 2 && life->par_nonpar == "P" && eq(life->submodel,
"ANN") )
    temp_ann_inv_rate = life->asset_shock;

return (1 + temp_ann_inv_rate)
        * (1-mgt_fixed_mth)
        -1;
```

**6.1.1.1.1.60 mgt\_fee\_fixed\_pv**

```

if (t < commence_period_w || t > maturity_period_ann)
    return 0.0;

if (annuity_pmt_curr_tot == 0)
    return 0.0;

int proj_yr = xint(life->proj_year(t+1));
if(eq(life->projection_type_int, "Rollup"))
    proj_yr = xint(life->proj_year_rollup(t+1));

if (t >= maturity_period_w)
    return (
        mgt_fee_fixed_pv(t+1)
        + mgt_fee_fixed_dth(t+1)
        + mgt_fee_fixed_gtd(t+1)
        + mgt_fee_fixed_jl1(t+1)
        + mgt_fee_fixed_jl2(t+1)
        + mgt_fee_fixed_nogt(t+1)
        )
        * life->ann_v_month_t[proj_yr];

return 0.0;

```

**6.1.1.1.1.61 mgt\_fee\_var\_pv**

```

if (t < commence_period_w || t > maturity_period_ann)
    return 0.0;

if (annuity_pmt_curr_tot == 0)
    return 0.0;

int proj_yr = xint(life->proj_year(t+1));
if(eq(life->projection_type_int, "Rollup"))
    proj_yr = xint(life->proj_year_rollup(t+1));

if (t >= maturity_period_w)
    return (
        mgt_fee_var_pv(t+1)
        + mgt_fee_var_dth(t+1)
        + mgt_fee_var_gtd(t+1)
        + mgt_fee_var_jl1(t+1)
        + mgt_fee_var_jl2(t+1)
        + mgt_fee_var_nogt(t+1)
        )
        * life->ann_v_month_t[proj_yr];

return 0.0;

```

**6.1.1.1.1.62 investment\_income**

```

if (t <= life->commence_period_w || t > maturity_period_ann)
    return 0.0;

if (annuity_pmt_curr_tot == 0)
    return 0.0;

int proj_yr = xint(life->proj_year(t));

```

```

if(eq(life->projection_type_int, "Rollup"))
    proj_yr = xint(life->proj_year_rollup(t));

double temp_inv_rate = 0.0;
double inv_inc_chetz = 0.0;

if (life->free_inv_prop_t[0] < 1.0) {
    temp_inv_rate = life->ann_inv_rate_rf_mth_t[proj_yr];
    inv_inc_chetz = investment_income_chetz(t); }

else temp_inv_rate = life->ann_inv_rate_mth_t[proj_yr];

if(t > maturity_period_w)
    return temp_inv_rate * (reserve_basic(t-1) + cashflow_b_post_ret(t)) + inv_inc_chetz;

return 0.;

```

#### 6.1.1.1.1.63 investment\_income\_chetz

```

if (t <= life->commence_period_w || t > maturity_period_ann)
    return 0.0;

if (life->free_inv_prop_t[1] >= 1. || annuity_pmt_curr_tot == 0)
    return 0.0;

int proj_yr = xint(life->proj_year(t));
if(eq(life->projection_type_int, "Rollup"))
    proj_yr = xint(life->proj_year_rollup(t));

double temp_reserve=0;
int period=0;
double Cal_yr_mth=100*life->cal_year(t) +life->cal_month(t);

if (life->chetz_be_ind == "Y" && Cal_yr_mth>=life->chetz_be_ind_yrs){
    temp_reserve=-cashflow_pv_chetz(t-1);
    period=maturity_period_w+1;}
    else{
        temp_reserve=reserve_basic(t-1) + cashflow_b_post_ret(t);
        period=maturity_period_w;}

if ( t > period)
    return (life->ann_inv_rate_mth_t[proj_yr] -life->ann_inv_rate_rf_mth_t[proj_yr])
        * (temp_reserve);

return 0.;

```

#### 6.1.1.1.1.64 reserve\_bonus\_units\_e\_0

```

if (t > 1200)
    return NO_AVG;

return accum->units_e(0) * life->bonus[benefit_term]/100. *
ret_prop_array[ann_index_map[takeup_age]] * retirement_prop;

```

#### 6.1.1.1.1.65 reserve\_bonus\_units\_e\_t

```

if (annuity_pmt_curr_tot == 0)

```

```

    return NO_AVG;

double units_e_noret = 0;

if(life->ret_prop_col(t+1) > SMALL_DOUBLE){
    units_e_noret = accum->units_e(t)/life->ret_prop_col(t+1) *
ret_prop_array[ann_index_map[takeup_age]] * retirement_prop;
}

return units_e_noret * life->bonus[benefit_term]/100;

```

#### 6.1.1.1.1.66 investment\_income\_chetz\_pv

```

if (t < commence_period_w || t > maturity_period_ann)
    return 0.0;

if (life->free_inv_prop_t[1] >= 1. || annuity_pmt_curr_tot == 0)
    return 0.0;

int proj_yr = xint(life->proj_year(t+1));
if(eq(life->projection_type_int, "Rollup"))
    proj_yr = xint(life->proj_year_rollup(t+1));

if (t >= maturity_period_w)
    return (investment_income_chetz_pv(t+1) + investment_income_chetz(t+1))* life-
>ann_v_month_t[proj_yr];

return 0.;

```

#### 6.1.1.1.1.67 investment\_income\_pv

```

if (t < commence_period_w || t > maturity_period_ann)
    return 0.0;

if (annuity_pmt_curr_tot == 0)
    return 0.0;

int proj_yr = xint(life->proj_year(t+1));
if(eq(life->projection_type_int, "Rollup"))
    proj_yr = xint(life->proj_year_rollup(t+1));

if (t >= maturity_period_w)
    return (investment_income_pv(t+1) + investment_income(t+1))* life->ann_v_month_t[proj_yr];

return 0.;

```

#### 6.1.1.1.1.68 outgo\_b\_post\_ret\_pv

```

if (t < commence_period_w || t > maturity_period_ann)
    return 0.0;

if (annuity_pmt_curr_tot == 0)
    return 0.0;

int proj_yr = xint(life->proj_year(t+1));
if(eq(life->projection_type_int, "Rollup"))
    proj_yr = xint(life->proj_year_rollup(t+1));

if (t >= maturity_period_w)

```

```

        return outgo_b_post_ret_pv(t+1) * life->ann_v_month_t[proj_yr]
            + expense_ren_perc_post_ret(t+1) + pmt_total(t+1);
return 0.0;

```

#### 6.1.1.1.1.69 expense\_investment\_pv

```

if (t < commence_period_w || t > maturity_period_ann)
    return 0.0;

if (annuity_pmt_curr_tot == 0)
    return 0.0;

int proj_yr = xint(life->proj_year(t+1));
if(eq(life->projection_type_int, "Rollup"))
    proj_yr = xint(life->proj_year_rollup(t+1));

if (t >= maturity_period_w)
    return expense_investment_pv(t+1) * life->ann_v_month_t[proj_yr]
        + expense_investment_post_ret(t+1);
return 0.0;

```

#### 6.1.1.1.1.70 expense\_ren\_perc\_post\_ret\_pv

```

if (t < commence_period_w || t > maturity_period_ann)
    return 0.0;

if (annuity_pmt_curr_tot == 0)
    return 0.0;

int proj_yr = xint(life->proj_year(t+1));
if(eq(life->projection_type_int, "Rollup"))
    proj_yr = xint(life->proj_year_rollup(t+1));

if (t >= maturity_period_w)
    return expense_ren_perc_post_ret_pv(t+1) * life->ann_v_month_t[proj_yr]
        + expense_ren_perc_post_ret(t+1);
return 0.0;

```

#### 6.1.1.1.1.71 expense\_investment\_post\_ret

```

if (t <= commence_period_w || t > maturity_period_ann)
    return 0;

if (annuity_pmt_curr_tot == 0)
    return 0;

double result = life->exp_ren_res / 1200
                * reserve_basic(t)
                * life->free_inv_prop_t[life->proj_year(t)]
                * life->expense_inflation(t);

double margin = 0;
if(life->margin_add=="Y")
    margin = life->margin_exp_ren_pc;

return result * (1 + margin / 100);

```

**6.1.1.1.1.72 expense\_ren\_perc\_post\_ret**

```

if (t <= commence_period_w || t > maturity_period_ann)
    return 0.0;

if (annuity_pmt_curr_tot == 0)
    return 0.0;

double result = ( life->exp_ren_res / 1200 * reserve_basic(t) * life->free_inv_prop_t[life-
>proj_year(t)]
                + life->exp_ren_perc_annuity/100 * pmt_total(t) )
                * life->expense_inflation(t);

double margin = 0;
if(life->margin_add=="Y")
    margin = life->margin_exp_ren_pc;

return result * (1 + margin / 100);

```

**6.1.1.1.1.73 reserve\_increase**

```

if (t <= commence_period_w || t > maturity_period_ann)
    return 0.0;

if (annuity_pmt_curr_tot == 0)
    return 0.0;

return reserve_basic(t) - reserve_basic(t-1);

```

**6.1.1.1.1.74 sel\_death\_rate\_1**

```

if (t <= commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if (annuity_pmt_curr_tot == 0)
    return NO_AVG;

//double sel_death_rate_col=0;
double sel_rate = 0.0;
double sel_death_rate_col_temp=0;

if (inlist(life->paid_up,"G"))
    sel_death_rate_col_temp=life->pol_year_ext(t);
else
    sel_death_rate_col_temp=age_pol_1(t) - takeup_age + 1;

sel_death_rate_col=sel_death_rate_col_temp;

sel_rate = sel_ret_qx_im_dth_1;

return sel_rate;

```

**6.1.1.1.1.75 sel\_death\_rate\_2**

```

if (t <= commence_period_w || t > maturity_period_ann)
    return NO_AVG;

```

```

if (annuity_pmt_curr_tot == 0)
    return NO_AVG;

//double sel_death_rate_col=0;
double sel_rate = 0.0;
double sel_death_rate_col_temp=0;

if (inlist(life->paid_up,"G"))
    sel_death_rate_col_temp=life->pol_year_ext(t);
else
    sel_death_rate_col_temp=age_pol_1(t) - takeover_age + 1;

sel_death_rate_col=sel_death_rate_col_temp;

sel_rate = sel_ret_qx_im_dth_2;

return sel_rate;

```

#### 6.1.1.1.1.76 profit\_book\_pv

```

if (t < commence_period_w || t > maturity_period_ann)
    return 0.0;

if (annuity_pmt_curr_tot == 0)
    return 0.0;

int proj_yr = xint(life->proj_year(t+1));
if(eq(life->projection_type_int, "Rollup"))
    proj_yr = xint(life->proj_year_rollup(t+1));

if (t > maturity_period_w)
    return (profit_book_pv(t+1)
            +investment_income(t+1)
            -reserve_increase(t+1))
            * life->ann_v_month_t[proj_yr]
            + cashflow_b_post_ret (t+1);

return 0.0;

```

#### 6.1.1.1.1.77 profit\_book\_vif\_post\_ret

```

if (t <= commence_period_w || t > maturity_period_ann)
    return 0.0;

if (annuity_pmt_curr_tot == 0)
    return 0.0;

return cashflow_b_post_ret(t) + investment_income(t)
        - reserve_increase(t);

```

#### 6.1.1.1.1.78 profit\_net\_vif\_post\_ret\_pv

```

if (t < commence_period_w || t > maturity_period_ann)
    return 0.0;

if (annuity_pmt_curr_tot == 0)
    return 0.0;

```

```

if(t >= maturity_period_w){

    int proj_yr = xint(life->proj_year(t+1));
    if(eq(life->projection_type_int, "Rollup"))
        proj_yr = xint(life->proj_year_rollup(t+1));

    return (profit_book_vif_post_ret(t+1) *(1-life->tax_rate/ 100.)
        + profit_net_vif_post_ret_pv(t+1)) * life->ann_v_month_t[proj_yr];}

return 0.;

```

#### 6.1.1.1.1.79 **surv\_gteed**

```

if (t < commence_period_w || t > maturity_period_ann )
    return NO_AVG;

if(life->ann_maslul<=100  && eq(life->paid_up,"G"))
    return NO_AVG;

double temp = surv_per_gteed(t);

// At valuation date
if (t == 0)
    return 1.0;

if (t > commence_period_w && fabs(surv_gteed(t-1)) < .0000001)
    // No surv in previous period
    return NO_AVG;

int temp_gteed_term = commence_period_w + gteed_term - 1;
if (eq(life->paid_up,"G")){
    temp_gteed_term = gteed_term-1;
}

if (t > commence_period_w && t <=(temp_gteed_term) )
    return 1.0;

return surv_1(t);

```

#### 6.1.1.1.1.80 **surv\_jl\_lastsurv**

```

if (t < commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if (!eq(joint_life_status, "Joint Life"))
    return NO_AVG;

// At valuation date
if (t == commence_period_w)
    return 1.0;

if (t > commence_period_w && fabs(surv_jl_lastsurv(t-1)) < .0000001)
    // No surv in previous period
    return NO_AVG;

if (t > commence_period_w)
    return surv_1(t) + surv_2_joint_life(t);

```



```
return surv_jl_lastsurv(t+1);
```

#### 6.1.1.1.1.81 **surv\_per\_gteed**

```
if (t < commence_period_w || t > maturity_period_ann )
    return NO_AVG;

if(life->ann_maslul<=100  && eq(life->paid_up,"G"))
    return NO_AVG;

// At valuation date
if (t == 0)
    return 1.0;

if (t > commence_period_w && fabs(surv_gteed(t-1)) < .0000001)
    // No surv in previous period
    return NO_AVG;

int temp_gteed_term = commence_period_w + gteed_term - 1;
if (eq(life->paid_up,"G")){
    temp_gteed_term = gteed_term-1;
}

if (t > commence_period_w && t <=(temp_gteed_term) )
    return 1.0; //Still in guaranteed period

if (t == temp_gteed_term + 1)
    return surv_1(t); //For first period after end of guaranteed, take accumulated survival
    proportion to that point

return surv_per_1(t);
```

#### 6.1.1.1.1.82 **death\_rate\_1**

```
if (t <= commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if (annuity_pmt_curr_tot == 0)
    return NO_AVG;

// Assume all lives die at omega age
if (age_pol_1(t) > life->omega_age_w)
    return 1.0;

double rate = 0.0;
if (sex1 == "M" )
    rate = death_rates_ann_m_1;
else
    rate = death_rates_ann_f_1;
// Increase the mort rates back with 4% of SD
rate = rate /0.96*(1.-(qx_sd_comp + qx_sd_random)/100.)/1000. * sel_death_rate_1(t);

if (life-> margin_add=="Y")
    rate = rate * (1+life->margin_ann_mort_pc/100.);

if(rate > 1.)
    return 1.0;
```

```
return 1 - pow(1- rate , 1.0/12.0);
```

#### 6.1.1.1.1.83 death\_rate\_res\_1

```
if (t <= commence_period_w || t > maturity_period_ann )
    return NO_AVG;

if (annuity_pmt_curr_tot == 0)
    return NO_AVG;

// Assume all lives die at omega age
if (age_pol_1(t) > life->omega_age_w)
    return 1.0;

double rate = 0.0;
if (sex1 == "M" )
    rate = death_rates_ann_m_res_1;
else
    rate = death_rates_ann_f_res_1;
// Increase the mort rates back with 4% of SD
rate = rate /0.96*(1.-(qx_sd_comp_res + qx_sd_random_res)/100.)/1000.*mort_fac_res_ann/100. *
sel_death_rate_1(t);

if(rate > 1.)
    return 1.0;

return 1 - pow(1- rate , 1.0/12.0);
```

#### 6.1.1.1.1.84 surv\_1

```
if (t < commence_period_w || t > maturity_period_ann)
    return NO_AVG;

// At valuation date
if (t == commence_period_w || (eq(life->paid_up,"G")&& t<=0))
    return 1.0;

if (t > commence_period_w && fabs(surv_1(t-1)) < .0000001)
    // No surv in previous period
    return NO_AVG;

return surv_1(t-1) * surv_per_1(t);
```

#### 6.1.1.1.1.85 surv\_1\_res

```
if (t < commence_period_w || t > maturity_period_ann)
    return NO_AVG;

// At valuation date
if (t == commence_period_w || (eq(life->paid_up,"G")&& t<=0))
    return 1.0;

if (t > commence_period_w && fabs(surv_1(t-1)) < .0000001)
    // No surv in previous period
    return NO_AVG;

return surv_1_res(t-1) * (1-death_rate_res_1(t));
```

**6.1.1.1.1.86      surv\_per\_1**

```
// Assume decrements are
// distributed uniformly within
// period of the projection.

if (t <= commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if (t > commence_period_w && fabs(surv_1(t-1)) < .0000001)
    return NO_AVG;

return 1. - death_rate_1(t);
```

**6.1.1.1.1.87      death\_rate\_2**

```
if (t <= commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if (eq(joint_life_status, "Single") || annuity_pmt_curr_tot == 0)
    return NO_AVG;

// Assume all lives die at omega age
if (age_pol_2(t) > life->omega_age_w)
    return 1.0;

double rate = 0.0;
if (sex2 == "M" )
    rate = death_rates_ann_m_2;
else
    rate = death_rates_ann_f_2;

// Increase the mort rates back with 4% of SD
rate = rate / 0.96 * (1. - (qx_sd_comp + qx_sd_random) / 100.) / 100. * sel_death_rate_2(t);
if (life-> margin_add == "Y")
    rate = rate * (1 + life->margin_ann_mort_pc / 100.);

if(rate > 1.)
    return 1.0;

return 1 - pow(1- rate , 1./12.);
```

**6.1.1.1.1.88      death\_rate\_2\_b3**

```
if (t <= commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if (eq(joint_life_status, "Single") || annuity_pmt_curr_tot == 0)
    return NO_AVG;

// Assume all lives die at omega age
if (age_pol_2(t) > life->omega_age_w)
    return 1.0;

double rate = 0.0;
if (sex2 == "M" )
    rate = death_rates_ann_m_b3_2;
else
    rate = death_rates_ann_f_b3_2;
```

```
// Increase the mort rates back with 4% of SD
rate = rate / 0.96 * (1. - (qx_sd_comp + qx_sd_random) / 100.) / 1000.;
if (life-> margin_add == "Y")
    rate = rate * (1 + life-> margin_ann_mort_pc / 100.);

if(rate > 1.)
    return 1.0;

return 1 - pow(1 - rate , 1./12.);
```

#### 6.1.1.1.1.89 death\_rate\_res\_2

```
if (t <= commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if (eq(joint_life_status, "Single") || annuity_pmt_curr_tot == 0)
    return NO_AVG;

// Assume all lives die at omega age
if (age_pol_2(t) > life->omega_age_w)
    return 1.0;

double rate = 0.0;
if (sex2 == "M" )
    rate = death_rates_ann_m_res_2;
else
    rate = death_rates_ann_f_res_2;
// Increase the mort rates back with 4% of SD
rate = rate / 0.96 * (1. - (qx_sd_comp_res + qx_sd_random_res) / 100.) / 1000. * mort_fac_res_ann / 100. *
sel_death_rate_2(t);

if(rate > 1.)
    return 1.0;

return 1 - pow(1 - rate , 1./12.);
```

#### 6.1.1.1.1.90 death\_rate\_res\_2\_b3

```
if (t <= commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if (eq(joint_life_status, "Single") || annuity_pmt_curr_tot == 0)
    return NO_AVG;

// Assume all lives die at omega age
if (age_pol_2(t) > life->omega_age_w)
    return 1.0;

double rate = 0.0;
if (sex2 == "M" )
    rate = death_rates_ann_m_res_b3_2;
else
    rate = death_rates_ann_f_res_b3_2;
// Increase the mort rates back with 4% of SD
rate = rate / 0.96 * (1. - (qx_sd_comp_res + qx_sd_random_res) / 100.) / 1000. * mort_fac_res_ann / 100.;

if(rate > 1.)
    return 1.0;
```

```
return 1 - pow(1- rate , 1./12.);
```

#### 6.1.1.1.1.91 **surv\_2**

```
if (t < commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if (!eq(joint_life_status, "Joint Life"))
    return NO_AVG;

// At valuation date
if (t == commence_period_w || (eq(life->paid_up,"G")&& t<=0))
    return 1.0;

if (t > commence_period_w && fabs(surv_2(t-1)) < .0000001)
    // No surv in previous period
    return NO_AVG;

if (t > 0)
    return surv_2(t-1) * surv_per_2(t);

return 0.0; //Unconditional return
```

#### 6.1.1.1.1.92 **surv\_2\_joint\_life**

```
if (t < commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if (!eq(joint_life_status, "Joint Life"))
    return NO_AVG;

// At valuation date
if (t == commence_period_w || (eq(life->paid_up,"G")&& t<=0))
    return 0.0;

if (t > commence_period_w && fabs(surv_2(t-1)) < .0000001)
    // No surv in previous period
    return NO_AVG;

if (t==1)
    return (1-surv_1(t))
           * surv_2(t)
           * life2_ppn_temp/100;

return surv_2_joint_life(t-1)
       *(1- death_rate_2_b3(t-1))
       + surv_1(t-1) * surv_2(t)
       * death_rate_1(t-1)
       * life2_ppn_temp/100;
```

#### 6.1.1.1.1.93 **surv\_2\_joint\_life\_res**

```
if (t < commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if (!eq(joint_life_status, "Joint Life"))
    return NO_AVG;
```

```
// At valuation date
if (t == commence_period_w || (eq(life->paid_up,"G")&& t<=0))
    return 0.0;

if (t > commence_period_w && fabs(surv_2_res(t-1)) < .0000001)
    // No surv in previous period
    return NO_AVG;

if (t==1)
    return (1-surv_1_res(t))
           * surv_2_res(t)
           * life2_ppn_temp/100;

return surv_2_joint_life_res(t-1)
       *(1- death_rate_res_2_b3(t-1))
       + surv_1_res(t-1) * surv_2_res(t)
       * death_rate_res_1(t-1)
       * life2_ppn_temp/100;
```

#### 6.1.1.1.1.94 **surv\_2\_res**

```
if (t < commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if (!eq(joint_life_status, "Joint Life"))
    return NO_AVG;

// At valuation date
if (t == commence_period_w || (eq(life->paid_up,"G")&& t<=0))
    return 1.0;

if (t > commence_period_w && fabs(surv_2(t-1)) < .0000001)
    // No surv in previous period
    return NO_AVG;

if (t > 0)
    return surv_2(t-1) * (1 - death_rate_res_2(t));

return 0.0; //Unconditional return
```

#### 6.1.1.1.1.95 **surv\_per\_2**

```
// Assume decrements are
// distributed uniformly within
// period of the projection.

if (t <= commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if (eq(joint_life_status, "Single"))
    return NO_AVG;

if (t > commence_period_w && fabs(surv_2(t-1)) < .0000001)
    return NO_AVG;

return 1. - death_rate_2(t);
```

**6.1.1.1.196      dth\_ben\_if\_b**

```

if (t <= commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if (death_ben == "N" || annuity_pmt_curr_tot == 0)
    return NO_AVG;

if (t > commence_period_w){
    if(t == 1 && life->paid_up == "G")
        return 0.;
    return double(death_ben_curr) * death_rate_1(t-1) * surv_1(t-2);
}
return 0.0;

```

**6.1.1.1.197      dth\_ben\_if\_b\_final**

```

if (t <= commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if (annuity_pmt_curr_tot == 0)
    return NO_AVG;

return dth_ben_if_b(t) * bonus_index_dth(t) * antisel_weight;

```

**6.1.1.1.198      pmt\_total**

```

if (t <= commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if (annuity_pmt_curr_tot == 0)
    return NO_AVG;

if (t == maturity_period_w + 1 && dump_vars == "Y")
    log_strm<<"Antiselection proportion: "<<antisel_ppn<<endl;

return ann_pay_no_gteed_final(t)
    + ann_pay_gteed_if_final(t)
    + ann_pay_jl_if_1_final(t)
    + ann_pay_jl_if_2_final(t)
    + dth_ben_if_b_final(t);

```

**6.1.1.1.199      pmt\_total\_nogt**

```

if (t <= commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if (annuity_pmt_curr_tot == 0)
    return NO_AVG;

double temp_gteed_pmt=0.;
if(surv_gteed(t)<1.)
    temp_gteed_pmt = ann_pay_gteed_if_final(t);

return ann_pay_no_gteed_final(t)
    + temp_gteed_pmt
    + ann_pay_jl_if_1_final(t)
    + ann_pay_jl_if_2_final(t)
    + dth_ben_if_b_final(t);

```

**6.1.1.1.100 int\_cred\_dth**

```
if (t< commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if (fund_type == "N" || death_ben == "N")
    return 0.0;

int proj_yr = xint(life->proj_year(t));
if(eq(life->projection_type_int, "Rollup"))
    proj_yr = xint(life->proj_year_rollup(t));
proj_yr= max(proj_yr, 0);

double temp_ann_inv_rate = life->ann_inv_rate_mth_t[proj_yr];

if (life->margin_add_asset == "Y" && t == 1 && life->par_nonpar == "P" && !eq(life->submodel,
"ANN") )
    temp_ann_inv_rate = life->asset_shock;

if (life->margin_add_asset == "Y" && t == 2 && life->par_nonpar == "P" && eq(life->submodel,
"ANN") )
    temp_ann_inv_rate = life->asset_shock;

return temp_ann_inv_rate
    * res_basic_dth(t-1);
```

**6.1.1.1.101 int\_cred\_gteed**

```
if (t< commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if (fund_type == "N" || ann_pmt_curr_gteed == 0.)
    return 0.0;

int proj_yr = xint(life->proj_year(t));
if(eq(life->projection_type_int, "Rollup"))
    proj_yr = xint(life->proj_year_rollup(t));
proj_yr= max(proj_yr, 0);

double temp_ann_inv_rate = life->ann_inv_rate_mth_t[proj_yr];

if (life->margin_add_asset == "Y" && t == 1 && life->par_nonpar == "P" && !eq(life->submodel,
"ANN") )
    temp_ann_inv_rate = life->asset_shock;

if (life->margin_add_asset == "Y" && t == 2 && life->par_nonpar == "P" && eq(life->submodel,
"ANN") )
    temp_ann_inv_rate = life->asset_shock;

return temp_ann_inv_rate
    * res_basic_gt(t-1);
```



**6.1.1.1.102 int\_cred\_jl1**

```

if (t< commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if (fund_type == "N" || ann_pmt_curr_jl == 0.)
    return 0.0;

int proj_yr = xint(life->proj_year(t));
if(eq(life->projection_type_int, "Rollup"))
    proj_yr = xint(life->proj_year_rollup(t));
proj_yr= max(proj_yr, 0);

double temp_ann_inv_rate = life->ann_inv_rate_mth_t[proj_yr];

if (life->margin_add_asset == "Y" && t == 1 && life->par_nonpar == "P" && !eq(life->submodel,
"ANN") )
    temp_ann_inv_rate = life->asset_shock;

if (life->margin_add_asset == "Y" && t == 2 && life->par_nonpar == "P" && eq(life->submodel,
"ANN") )
    temp_ann_inv_rate = life->asset_shock;

return temp_ann_inv_rate
    * res_basic_jl_1(t-1);

```

**6.1.1.1.103 int\_cred\_jl2**

```

if (t< commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if (fund_type == "N" || ann_pmt_curr_jl == 0.)
    return 0.0;

int proj_yr = xint(life->proj_year(t));
if(eq(life->projection_type_int, "Rollup"))
    proj_yr = xint(life->proj_year_rollup(t));
proj_yr= max(proj_yr, 0);

double temp_ann_inv_rate = life->ann_inv_rate_mth_t[proj_yr];

if (life->margin_add_asset == "Y" && t == 1 && life->par_nonpar == "P" && !eq(life->submodel,
"ANN") )
    temp_ann_inv_rate = life->asset_shock;

if (life->margin_add_asset == "Y" && t == 2 && life->par_nonpar == "P" && eq(life->submodel,
"ANN") )
    temp_ann_inv_rate = life->asset_shock;

return temp_ann_inv_rate
    * res_basic_jl_2(t-1);

```

**6.1.1.1.104 int\_cred\_no\_gteed**

```

if (t < commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if (fund_type == "N" || ann_pmt_curr == 0.)
    return 0.0;

int proj_yr = xint(life->proj_year(t));
if(eq(life->projection_type_int, "Rollup"))
    proj_yr = xint(life->proj_year_rollup(t));
proj_yr = max(proj_yr, 0);

double temp_ann_inv_rate = life->ann_inv_rate_mth_t[proj_yr];

if (life->margin_add_asset == "Y" && t == 1 && life->par_nonpar == "P" && !eq(life->submodel,
"ANN") )
    temp_ann_inv_rate = life->asset_shock;

if (life->margin_add_asset == "Y" && t == 2 && life->par_nonpar == "P" && eq(life->submodel,
"ANN") )
    temp_ann_inv_rate = life->asset_shock;

return temp_ann_inv_rate
    * res_basic_nogt(t-1);

```

**6.1.1.1.105 mgt\_fee\_fixed\_dth**

```

if (t < commence_period_w || t > maturity_period_ann)
    return 0.0;

if ( death_ben == "N")
    return NO_AVG;

if (fund_type == "N")
    return 0.0;

return mgt_fixed_mth
    * (res_basic_dth(t-1) + int_cred_dth(t));

```

**6.1.1.1.106 mgt\_fee\_fixed\_gtd**

```

if (t < commence_period_w || t > maturity_period_ann)
    return 0.0;

if ( ann_pmt_curr_gteed == 0.)
    return NO_AVG;

if (fund_type == "N")
    return 0.0;

return mgt_fixed_mth
    * (res_basic_gt(t-1) + int_cred_gteed(t));

```

**6.1.1.1.107 mgt\_fee\_fixed\_jl1**

```

if (t < commence_period_w || t > maturity_period_ann)

```

```

        return 0.0;

if ( ann_pmt_curr_jl == 0.)
    return NO_AVG;

if (fund_type == "N")
    return 0.0;

return mgt_fixed_mth
    * (res_basic_jl_1(t-1) + int_cred_jl1(t));

```

#### 6.1.1.1.1.108 mgt\_fee\_fixed\_jl2

```

if (t < commence_period_w || t > maturity_period_ann)
    return 0.0;

if ( ann_pmt_curr_jl == 0.)
    return NO_AVG;

if (fund_type == "N")
    return 0.0;

return mgt_fixed_mth
    * (res_basic_jl_2(t-1) + int_cred_jl2(t));

```

#### 6.1.1.1.1.109 mgt\_fee\_fixed\_nogt

```

if (t < commence_period_w || t > maturity_period_ann)
    return 0.0;

if ( ann_pmt_curr == 0.)
    return NO_AVG;

if (fund_type == "N")
    return 0.0;

return mgt_fixed_mth
    * (res_basic_nogt(t-1) + int_cred_no_gteed(t));

```

#### 6.1.1.1.1.110 mgt\_fee\_var\_dth

```

if (t < commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if(fund_type == "N" || death_ben == "N")
    return 0.0;

double mgt_var = mgt_var_no_bor_dth(t);

//if (net_interest_rate(t) > 0.0)
//    mgt_var = mgt_fee_variable/100.
//    * (int_cred_dth(t) - mgt_fee_fixed_dth(t)); // Management fee (assuming no adjustment)

mgt_var = mgt_var + bor_return_dth(t); //Deduct bor to be returned

mgt_var = max(mgt_var, 0);

mgt_var = mgt_var - har_return_dth(t);

```

```
if (abs(mgt_var) < SMALL_DOUBLE)
    return 0.0; //Remove small rounding issues
```

```
return mgt_var;
```

#### **6.1.1.1.1.111 mgt\_fee\_var\_gtd**

```
if (t< commence_period_w || t > maturity_period_ann)
    return NO_AVG;
```

```
if(fund_type == "N" || ann_pmt_curr_gteed == 0.)
    return 0.0;
```

```
double mgt_var = mgt_var_no_bor_gtd(t);
```

```
//if (net_interest_rate(t) > 0.0)
//    mgt_var = mgt_fee_variable/100.
//    * (int_cred_gteed(t) - mgt_fee_fixed_gtd(t)); // Management fee (assuming no
adjustment)
```

```
mgt_var = mgt_var + bor_return_gtd(t); //Deduct bor to be returned
```

```
mgt_var = max(mgt_var, 0);
```

```
mgt_var = mgt_var - har_return_gtd(t);
```

```
if (abs(mgt_var) < SMALL_DOUBLE)
    return 0.0; //Remove small rounding issues
```

```
return mgt_var;
```

#### **6.1.1.1.1.112 mgt\_fee\_var\_jl1**

```
if (t< commence_period_w || t > maturity_period_ann)
    return NO_AVG;
```

```
if(fund_type == "N" || ann_pmt_curr_jl1 == 0.)
    return 0.0;
```

```
double mgt_var = mgt_var_no_bor_jl1(t);
```

```
//if (net_interest_rate(t) > 0.0)
//    mgt_var = mgt_fee_variable/100.
//    * (int_cred_jl1(t) - mgt_fee_fixed_jl1(t)); // Management fee (assuming no adjustment)
```

```
mgt_var = mgt_var + bor_return_jl1(t); //Deduct bor to be returned
```

```
mgt_var = max(mgt_var, 0);
```

```
mgt_var = mgt_var - har_return_jl1(t);
```

```
if (abs(mgt_var) < SMALL_DOUBLE)
    return 0.0; //Remove small rounding issues
```

```
return mgt_var;
```

#### **6.1.1.1.1.113 mgt\_fee\_var\_jl2**

```
if (t< commence_period_w || t > maturity_period_ann)
```

```

        return NO_AVG;

    if(fund_type == "N" || ann_pmt_curr_jl == 0.)
        return 0.0;

    double mgt_var = mgt_var_no_bor_jl2(t);

    //if (net_interest_rate(t) > 0.0)
    //    mgt_var = mgt_fee_variable/100.
    //    * (int_cred_jl2(t) - mgt_fee_fixed_jl2(t)); // Management fee (assuming no adjustment)

    mgt_var = mgt_var + bor_return_jl2(t); //Deduct bor to be returned

    mgt_var = max(mgt_var, 0);

    mgt_var = mgt_var - har_return_jl2(t);

    if (abs(mgt_var) < SMALL_DOUBLE)
        return 0.0; //Remove small rounding issues

    return mgt_var;

```

#### **6.1.1.1.114 mgt\_fee\_var\_nogt**

```

    if (t < commence_period_w || t > maturity_period_ann)
        return NO_AVG;

    if(fund_type == "N" || mgt_fee_var == 0.0 || ann_pmt_curr == 0.)
        return 0.0;

    double mgt_var = mgt_var_no_bor_nogt(t);

    //if (net_interest_rate(t) > 0.0)
    //    mgt_var = mgt_fee_variable/100.
    //    * (int_cred_no_gteed(t) - mgt_fee_fixed_nogt(t)); // Management fee (assuming no
    //    adjustment)

    mgt_var = mgt_var + bor_return_nogt(t); //Deduct bor to be returned

    mgt_var = max(mgt_var, 0);

    mgt_var = mgt_var - har_return_nogt(t);

    if (abs(mgt_var) < SMALL_DOUBLE)
        return 0.0; //Remove small rounding issues

    return mgt_var;

```

#### **6.1.1.1.115 tarif\_deduction\_dth**

```

    if (t < commence_period_w || t > maturity_period_ann )
        return NO_AVG;

    if (death_ben == "N")
        return NO_AVG;

```

```
if (fund_type == "N")
    return 0.0;
```

```
return rate_tarif_mth
    * res_basic_dth(t-1);
```

#### **6.1.1.1.116      tarif\_deduction\_gteed**

```
if (t < commence_period_w || t > maturity_period_ann )
    return NO_AVG;
```

```
if (annuity_pmt_curr_tot == 0.)
    return NO_AVG;
```

```
if (fund_type == "N")
    return 0.0;
```

```
return rate_tarif_mth
    * res_basic_gt(t-1);
```

#### **6.1.1.1.117      tarif\_deduction\_jl1**

```
if (t < commence_period_w || t > maturity_period_ann )
    return NO_AVG;
```

```
if (ann_pmt_curr_jl == 0.)
    return NO_AVG;
```

```
if (fund_type == "N")
    return 0.0;
```

```
return rate_tarif_mth
    * res_basic_jl_1(t-1);
```

#### **6.1.1.1.118      tarif\_deduction\_jl2**

```
if (t < commence_period_w || t > maturity_period_ann )
    return NO_AVG;
```

```
if (ann_pmt_curr_jl == 0.)
    return NO_AVG;
```

```
if (fund_type == "N")
    return 0.0;
```

```
return rate_tarif_mth
    * res_basic_jl_2(t-1);
```

#### **6.1.1.1.119      tarif\_deduction\_no\_gteed**

```
if (t < commence_period_w || t > maturity_period_ann )
    return NO_AVG;
```

```
if (ann_pmt_curr == 0.)
    return NO_AVG;
```

```
if (fund_type == "N")
    return 0.0;
```

```

return rate_tarif_mth
    * res_basic_nogt(t-1);

```

#### 6.1.1.1.1.120 ann\_pay\_gteed\_if

```

if (t <= commence_period_w || t > maturity_period_ann )
    return NO_AVG;

```

```

if (ann_pmt_curr_gteed == 0)
    return NO_AVG;

```

```

return ann_pmt_curr_gteed * surv_gteed (t-1) ;

```

#### 6.1.1.1.1.121 ann\_pay\_gteed\_if\_final

```

if (t <= commence_period_w || t > maturity_period_ann )
    return NO_AVG;

```

```

if (ann_pmt_curr_gteed == 0)
    return NO_AVG;

```

```

return ann_pay_gteed_if(t) * bonus_index_gteed(t) * antisel_weight ;

```

#### 6.1.1.1.1.122 ann\_pay\_jl\_if\_1

```

if (t <= commence_period_w || t > maturity_period_ann)
    return NO_AVG;

```

```

if (eq(joint_life_status,"Single") || ann_pmt_curr_jl == 0)
    return NO_AVG;

```

```

return ann_pmt_curr_jl * surv_1 (t-1) ;

```

#### 6.1.1.1.1.123 ann\_pay\_jl\_if\_1\_final

```

if (t <= commence_period_w || t > maturity_period_ann)
    return NO_AVG;

```

```

if (eq(joint_life_status,"Single") || ann_pmt_curr_jl == 0)
    return NO_AVG;

```

```

return ann_pay_jl_if_1(t) * bonus_index_jl_1(t) * antisel_weight ;

```

#### 6.1.1.1.1.124 ann\_pay\_jl\_if\_2

```

if (t <= commence_period_w || t > maturity_period_ann)
    return NO_AVG;

```

```

if (eq(joint_life_status,"Single") || ann_pmt_curr_jl == 0)
    return NO_AVG;

```

```

if (t > (maturity_period_w + 1))
    return ann_pay_jl_if_2 (t - 1)
        * (1 - death_rate_2_b3(t-1))
        + //ann_pay_jl_if_1 (t - 1)
        //Reproduce old
        ann_pmt_curr_jl * surv_1(t-2)
        * death_rate_1(t-1)
        * surv_2(t-1)

```

```
* life2_ppn_temp/100.;
```

```
return 0.0 ;
```

#### **6.1.1.1.1.125 ann\_pay\_jl\_if\_2\_final**

```
if (t <= commence_period_w || t > maturity_period_ann)
  return NO_AVG;
```

```
if (annuity_pmt_curr_tot == 0)
  return NO_AVG;
```

```
return ann_pay_jl_if_2(t) * bonus_index_jl_2(t) * antisel_weight;
```

#### **6.1.1.1.1.126 ann\_pay\_no\_gteed**

```
if (t <= commence_period_w || t > maturity_period_ann)
  return NO_AVG;
```

```
if (ann_pmt_curr == 0)
  return NO_AVG;
```

```
return ann_pmt_curr * surv_1 (t-1) ;
```

#### **6.1.1.1.1.127 ann\_pay\_no\_gteed\_final**

```
if (t <= commence_period_w || t > maturity_period_ann)
  return NO_AVG;
```

```
if (ann_pmt_curr == 0)
  return NO_AVG;
```

```
return ann_pay_no_gteed(t) * bonus_index_no_gtee(t) * antisel_weight; ;
```

#### **6.1.1.1.1.128 death\_rates\_row\_1**

```
if (t <= commence_period_w || t > maturity_period_ann)
  return NO_AVG;
```

```
return min(xint(age_pol_1(t)),110);
```

#### **6.1.1.1.1.129 death\_rates\_row\_2**

```
if (t <= commence_period_w || t > maturity_period_ann)
  return NO_AVG;
```

```
return min(xint(age_pol_2(t)),120);
```

#### **6.1.1.1.1.130 units\_for\_takeup**

```
if (t != maturity_period_w)
  return 0.0;
```

```
if(life->submodel == "TRAD")
  return trad->claims_maturity(t) * retirement_prop;
```

```
if(life->submodel == "TERM")
  return 0.0;
```

```
return life->units_at_mat(t) * retirement_prop;
```



**6.1.1.1.1.131 age\_pol\_1**

```

if (t <= commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if (annuity_pmt_curr_tot == 0)
    return NO_AVG;

if( t == commence_period_w +1 ){
    if(eq(life->paid_up,"G"))
        return life->age_at_issue;
    return age_ann_start_1 ;
}
return age_pol_1(t-1) + xint(life->pol_month(t-1)/12) ;

```

**6.1.1.1.1.132 age\_pol\_2**

```

if (t <= commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if (annuity_pmt_curr_tot == 0)
    return NO_AVG;

//For single life policies, the second life is irrelevant
if (eq(joint_life_status,"Single"))
    return NO_AVG;

if( t == commence_period_w +1 ){
    if(eq(life->paid_up,"G"))
        return life->age_at_issue - age_diff_temp;
    return age_ann_start_2 ;
}
return age_pol_2(t-1) + xint(life->pol_month(t-1)/12) ;

```

**6.1.1.1.1.133 profit\_book\_vif\_post\_ret\_pv**

```

if (t < commence_period_w || t > maturity_period_ann)
    return 0.0;

if (annuity_pmt_curr_tot == 0)
    return 0.0;

int proj_yr = xint(life->proj_year(t+1));
if(eq(life->projection_type_int, "Rollup"))
    proj_yr = xint(life->proj_year_rollup(t+1));

if(t >= maturity_period_w){
    return (profit_book_vif_post_ret(t+1) + profit_book_vif_post_ret_pv(t+1)) * life-
>ann_v_month_t[proj_yr];}

return 0.;

```

**6.1.1.1.1.134 reserve\_increase\_pv**

```

if (t < commence_period_w || t > maturity_period_ann)
    return 0.0;

if (annuity_pmt_curr_tot == 0)
    return 0.0;

```

```

int proj_yr = xint(life->proj_year(t+1));
if(eq(life->projection_type_int, "Rollup"))
    proj_yr = xint(life->proj_year_rollup(t+1));

if (t >= maturity_period_w)
    return (reserve_increase_pv(t+1) + reserve_increase(t+1))* life->ann_v_month_t[proj_yr];

return 0.0;

```

#### **6.1.1.1.1.135 startup**

```

start_extrns
    extern map <int, int> ann_index_map;
    extern map <int, double> ret_prop_map;
    extern SmartArray <double> ret_prop_array;
end_extrns

t_low = - (life->elapsed_months + 100); //arbitrarily setting to start calcs 100 months before
purchase date

if (eq(life->submodel,"TERM") || life->res_prop_kitzba <= 0.0)
    return 0;

initialise_variables();

if (!eq(life->error_msg,"no_error"))
    commence_period_w = -1;

return 0;

```

### **6.1.1.1.2 External Functions**

#### **6.1.1.1.2.1 initialise\_variables**

```

void initialise_variables(void)
{
    double ann_fac_gtee_temp=0.0;
    int ann_fac_base_y = 0;
    double ann_redn_fac = 0.0;
    double adjfac=0.0;

    benefit_term = life->benefit_term_input;

    sex1 = life->sex;
    ann_series = life->ann_series;
    // Set annuity takeup rates (if found in fund table)
    if(life->submodel == "TRAD")
        annuitization_rate = life->annuitization_rate;

    adjfac = ann_tu_old;

    if(adjfac != 2.) {
        annuity_takeup_old = ann_tu_old;
        annuity_takeup_new_tag =ann_tu_newtag;
        annuity_takeup_prat =ann_tu_prat;
        annuity_takeup_piz =ann_tu_piz;
    }
}

```

```

// set annuity take up rates for reserving basis
adjfac =ann_tu_old_res;

if(adjfac != 2.) {
    annuity_takeup_old_res = ann_tu_old_res;
    annuity_takeup_new_tag_res =ann_tu_newtag_res;
    annuity_takeup_prat_res =ann_tu_prat_res;
    annuity_takeup_piz_res = ann_tu_piz_res;
}

if (eq(life->submodel,"UNIT")) {
    if(!inlist(life->policy_type,"private,selfemp") || life->res_kitzba >0.){
        if(life->age_at_issue + life->elapsed_months/12. < takeover_age)
            benefit_term = (takeup_age - life->age_at_issue) * 12;
    }
}

if(eq(life->submodel,"TRAD")) {
    if(!inlist(life->policy_type,"private,selfemp") || life->res_kitzba >0.){
        if(life->age_at_issue + life->elapsed_months/12. < takeover_age)
            benefit_term = (takeup_age - life->age_at_issue) * 12;
    }
}

maturity_period_w = life->commence_period_w + benefit_term;
maturity_period_ann = maturity_period_w;

age_ann_start_1 = floor(life->age_at_issue) + xint(benefit_term/12);
year_ann_start = life->year_prod + xint(benefit_term/12);

if (life->submodel!="TRAD" && life->paid_up=="G"){
    maturity_period_w = 0;
    age_ann_start_1 = floor(life->age_at_issue) + xint(life->elapsed_months/12);
    year_ann_start = life->year_prod + xint(life->elapsed_months/12) ;
    if (freeinv_res_ann_tarif == "Y")
        freeinv_rate_res_ann = int_tarif;
    else
        freeinv_rate_res_ann = freeinv_res_ann_inpay;
}

// set annuity code

double check = 0.0;

if (life->submodel=="TRAD"){
    temp_annuity_code = xstring(ann_series)+"_"+life->sex + "_" + xstring(age_ann_start_1);
    check =annuity_details_temp_tbl;
    /*the following code should be here but to maintain the error in line 20, we need to put in
th else part!*/
    if (check==-99999.) {
        if(sex1 == "M")
            annuity_code = xstring(ann_series)+"_"+life->sex + "_67" ;
        else
            annuity_code = xstring(ann_series)+"_"+life->sex + "_64" ;
    }
    else

```

```

        annuity_code = temp_annuity_code;
}else{
    temp_annuity_code = xstring(temp_fund_rates_tbl) + "_" + life->sex + "_" +
xstring(age_ann_start_1);
    check = annuity_details_temp_tbl;

    if (check==-99999.)
        annuity_code = xstring(temp_fund_rates_tbl)+"_" + life->sex + "_80" ;

    else
        annuity_code = temp_annuity_code;
}

ann_fac_no_gtee_temp = ann_fac_no_gtee;
ann_fac_joint_temp = ann_fac_joint;
ann_fac_base_y =base_year;
ann_redn_fac = redn_factor;
ann_fac_gtee_temp = ann_fac_gtee_value;

if(ann_fac_base_y > 0){
    if(ann_redn_fac < 0)
        ann_fac_gtee = ann_fac_gtee_temp/(1. + ann_redn_fac/100.*(year_ann_start -
ann_fac_base_y));

    else
        ann_fac_gtee = ann_fac_gtee_temp * (1. + ann_redn_fac/100.*(year_ann_start -
ann_fac_base_y));
}
else
    ann_fac_gtee = ann_fac_gtee_temp;

int_rate_res_ann = int_res_ann;
if (freeinv_res_ann_tarif == "Y")
    freeinv_rate_res_ann = int_tarif;
else
    freeinv_rate_res_ann = freeinv_res_ann;
mort_fac_res_ann = res_ann_mort_fac;
exp_res = res_ann_exp;
int_tarif_temp = int_tarif;
mgt_fee_fixed_temp = mgt_fee_fixed;
mgt_fee_variable = mgt_fee_var;
mgt_fee_fixed_max = mgt_fee_max;
gteed_term = (gteed_prd) * 12;
life2_ppn_temp = life2_ppn;

if (life->margin_add=="Y")
    mort_fac_res_ann = mort_fac_res_ann * (1+life->margin_res_ann_mort_fac/100);

age_diff_temp = age_diff;
if(eq(life->paid_up,"G")){
    age_diff_temp = atof(life->maasik_no);
    if(life->submodel !="TRAD"){
        if(life->ann_maslul > 100 )
            gteed_term = max(xint(life->ann_maslul/100.)*12 + life->commence_period_w ,
0) ;
        else
            gteed_term = 0 ;
    }
}

```

```

    }
}

joint_life_status = "Single";
if(ann_fac_joint_temp > 0)
    joint_life_status = "joint life";

if(life->submodel != "TRAD" && eq(life->paid_up,"G")){
    if(life->ann_maslul > 25 && life->ann_maslul <= 100){
        joint_life_status = "joint life";
        life2_ppn_temp = life->ann_maslul;
    }
    else
        joint_life_status = "Single";
}
death_ben = "N";
if(ann_fac_dthben > 0)
    death_ben = "Y";

if(life->submodel != "TRAD" && eq(life->paid_up,"G")){
    if(life->ann_maslul == 25){
        death_ben = "Y";
        death_ben_curr = life->sum_ins_curr * life->benefits_curr * int(life->ann_maslul);
    }

    else {
        death_ben = "N";
        death_ben_curr = 0.;
    }
}

maturity_period_ann = maturity_period_w + (110 - age_ann_start_1) * 12 ;
commence_period_w = maturity_period_w;

if(life->submodel != "TRAD" && eq(life->paid_up,"G")){
    commence_period_w = life->commence_period_w;
    maturity_period_ann = benefit_term + life->commence_period_w;
}

if (!eq(joint_life_status, "Single"))
    maturity_period_ann = maturity_period_w + (110 - min(age_ann_start_1, age_ann_start_1 -
age_diff_temp)) * 12 ;

life->maturity_period_ann = maturity_period_ann;

set_work_variables();

return;
}

```

#### 6.1.1.1.2.2 set\_work\_variables

```

void set_work_variables (void) {

yob_1 = year_ann_start - age_ann_start_1;

if (!eq(joint_life_status,"Single")){

```

```

    age_ann_start_2 = age_ann_start_1 - age_diff_temp;
    yob_2 = year_ann_start - age_ann_start_2;
}

    if( sex1=="M")
        sex2="F";
    else
        sex2="M";

if (life->submodel == "TRAD")
    ann_series_prop = ann_series;
else
    ann_series_prop = temp_fund_rates_tbl;

gtee_ppn_temp = gtee_ppn;
no_gtee_ppn_temp = no_gtee_ppn;
joint_life_ppn_temp = joint_life_ppn;

}

```

### 6.1.1.1.3 Temporary Tables

#### 6.1.1.1.3.1 qx\_final\_res

```

// r = Rows are ages
// c = sex, 0= Male 1 = Female 2 = MaleB3 3 = FemaleB3

double qx=0.0;

if (c==0){
    if(sex1 == "M" ){

        if(r < age_ann_start_1)
            qx=0.0;
        else{
            life->mort_year_tt = year_ann_start+ r- age_ann_start_1;
            dth_rts_m_row_key_tt = min( xint(r),120);
            qx = death_rates_ann_m_res_tt;
        }
    }
    else{

        if(r<age_ann_start_2)
            qx =0.0;
        else{
            life->mort_year_tt = year_ann_start+ r- age_ann_start_2;
            dth_rts_m_row_key_tt = min( xint(r),120);
            qx = death_rates_ann_m_res_tt;
        }
    }
}

if (c==1) {
    if(sex1 == "M" ){

        if(r < age_ann_start_2)
            qx = 0.0;
        else{

```

```

        life->mort_year_tt = year_ann_start+ r- age_ann_start_2;
        dth_rts_m_row_key_tt = min( xint(r),120);
        qx = death_rates_ann_f_res_tt;
    }
}
else{

    if(r < age_ann_start_1)
        qx = 0.0;
    else{
        life->mort_year_tt = year_ann_start+ r- age_ann_start_1;
        dth_rts_m_row_key_tt = min( xint(r),120);
        qx = death_rates_ann_f_res_tt;
    }
}

}

if (c==2){
    if(sex1 == "M" ){

        if(r < age_ann_start_1)
            qx=0.0;
        else{
            life->mort_year_tt = year_ann_start+ r- age_ann_start_1;
            dth_rts_m_row_key_tt = min( xint(r),120);
            qx = death_rates_ann_m_res_b3_tt;
        }
    }
    else{

        if(r<age_ann_start_2)
            qx =0.0;
        else{
            life->mort_year_tt = year_ann_start+ r- age_ann_start_2;
            dth_rts_m_row_key_tt = min( xint(r),120);
            qx = death_rates_ann_m_res_b3_tt;
        }
    }
}

if (c==3) {
    if(sex1 == "M" ){

        if(r < age_ann_start_2)
            qx = 0.0;
        else{
            life->mort_year_tt = year_ann_start+ r- age_ann_start_2;
            dth_rts_m_row_key_tt = min( xint(r),120);
            qx = death_rates_ann_f_res_b3_tt;
        }
    }
    else{

        if(r < age_ann_start_1)
            qx = 0.0;
        else {
            life->mort_year_tt = year_ann_start+ r- age_ann_start_1;

```

```

        dth_rts_m_row_key_tt = min( xint(r),120);
        qx = death_rates_ann_f_res_b3_tt;
    }
}

return qx/0.96*(1.-(qx_sd_comp_res + qx_sd_random_res)/100.)*mort_fac_res_ann/100./1000.;

```

#### 6.1.1.1.3.2 res\_cx\_ann

```

// Commutation Function Cx = v^(x+1) * lx - l(x+1)
// r = current age in years

if (r == 120)
    return res_dx_ann(r,0)/(1.+int_rate_res_ann/100.);

double d = res_lx_ann(r,sexcode_1) - res_lx_ann(r+1,sexcode_1); //deaths aged r

if (int_rate_res_ann == 0.0)
    return d;

return d * res_vx_ann(r+1,sexcode_1);

```

#### 6.1.1.1.3.3 res\_dx\_ann

```

// Commutation Function Dx Yearly Dx = lx * v^x
// r = current age in years

if (r > 120)
    return 0.0;

return res_lx_ann(r, c) * res_vx_ann(r,c);

```

#### 6.1.1.1.3.4 res\_lx\_ann

```

// Commutation Function lx

if (r <= 20) // cannot look up a zero or negative starting age
    return 100.0; // radix
if (min(r ,r -age_diff_temp) > life->omega_age_w) // omega age allows for table adjustment
    return 0.0;

// Single life
if (c<2) {
    double q = qx_final_res(r -1, c);

    return res_lx_ann(r-1, c) * (1. - q);
}

// Joint Life
double q1 = qx_final_res(r -1, sexcode_1);
double q2 = qx_final_res(max(min(r -1-age_diff_temp,119),0), sexcode_2);

return res_lx_ann(r-1, c) * (1. - q1)* (1. - q2);

```

#### 6.1.1.1.3.5 res\_mx\_ann

```

if (r>120)

```



```
return 0.0;
```

```
if (r==120)
```

```
    return res_cx_ann(r,0);
```

```
return res_cx_ann(r,0) + res_mx_ann(r+1,0);
```

#### 6.1.1.1.3.6 res\_nx\_ann

```
//Nx
```

```
if (r >= 120)
```

```
return res_dx_ann(r, c);
```

```
return res_nx_ann(r+1, c) + res_dx_ann(r, c);
```

#### 6.1.1.1.3.7 res\_vx\_ann

```
// Commutation Function vx = v^(x)
```

```
// r = current age in years
```

```
double temp_inv_inc = life->invinc/100.;
```

```
double temp_free_inv_prop_1 = life->free_inv_prop_t[min(max(year_ann_start-life->valn_year + r -  
age_ann_start_1 - 1,0),100)];
```

```
double temp_free_inv_prop_2 = life->free_inv_prop_t[min(max(year_ann_start-life->valn_year + r -  
age_ann_start_2 - 1,0),100)];
```

```
double temp_res_rate_1 = freeinv_rate_res_ann/100.*temp_free_inv_prop_1 + temp_inv_inc*(1-  
temp_free_inv_prop_1);
```

```
double temp_res_rate_2 = freeinv_rate_res_ann/100.*temp_free_inv_prop_2 + temp_inv_inc*(1-  
temp_free_inv_prop_2);
```

```
if (c == sexcode_2){
```

```
    if (r < age_ann_start_2) // cannot look up a zero or negative starting age
```

```
        return 1.0;
```

```
    if( r == age_ann_start_2 && life->paid_up=="G")
```

```
        return 1.0;
```

```
    return res_vx_ann(r-1,sexcode_2) /(1.+temp_res_rate_2);
```

```
}
```

```
if (c == sexcode_1){
```

```
    if (r < age_ann_start_1) // cannot look up a zero or negative starting age
```

```
        return 1.0;
```

```
    if( r == age_ann_start_1 && life->paid_up=="G")
```

```
        return 1.0;
```

```
    return res_vx_ann(r-1,sexcode_1) /(1.+temp_res_rate_1);
```

```
}
```

```
if (r < age_ann_start_1) // cannot look up a zero or negative starting age
```

```
    return 1.0;
```

```
if( r == age_ann_start_1 && life->paid_up=="G")
```

```
    return 1.0;
```

```
return res_vx_ann(r-1,2) /(1.+ temp_res_rate_1);
```

#### 6.1.1.1.4 Scalars

##### 6.1.1.1.4.1 ann\_pmt\_curr

```
if(eq(life->paid_up,"G")){
  if(life->ann_maslul == 25 || life->ann_maslul <= 1)
    return life->sum_ins_curr * life->benefits_curr * retirement_prop;
  return 0.0;
}

if(ann_fac_no_gtee_temp != 0.0)
  return initial_annuity_purchase * no_gtee_ppn_temp/ann_fac_no_gtee_temp;

return 0.0;
```

##### 6.1.1.1.4.2 ann\_pmt\_curr\_gteed

```
if(eq(life->paid_up,"G")){
  if(life->ann_maslul > 100 )
    return life->sum_ins_curr * life->benefits_curr * retirement_prop;
  return 0.0;
}

if(ann_fac_gtee != 0.0)
  return initial_annuity_purchase * gtee_ppn_temp/ann_fac_gtee;

return 0.0;
```

##### 6.1.1.1.4.3 ann\_pmt\_curr\_jl

```
if(eq(life->paid_up,"G")){
  if(life->ann_maslul > 25 &&life->ann_maslul <= 100)
    return life->sum_ins_curr * life->benefits_curr * retirement_prop;
  return 0.0;
}

if(ann_fac_joint_temp != 0.0)
  return initial_annuity_purchase * joint_life_ppn_temp/ann_fac_joint_temp;

return 0.0;
```

##### 6.1.1.1.4.4 antisel\_weight

```
if (annuity_pmt_curr_tot == 0)
  return 0;

return 1. + antisel_ppn/100.* antisel_margin ;
```

##### 6.1.1.1.4.5 antisel\_weight\_res

```
if (annuity_pmt_curr_tot == 0)
  return 0;

return 1. + antisel_ppn_res/100.* antisel_margin ;
```

##### 6.1.1.1.4.6 fund\_type

```
if(life->par_npar == 0)
  return "N";
```

```
return "P";
```

#### **6.1.1.1.4.7 mgt\_fixed\_max\_mth**

```
return mgt_fee_fixed_max/1200.;
```

#### **6.1.1.1.4.8 mgt\_fixed\_mth**

```
return mgt_fee_fixed_temp/1200;
```

#### **6.1.1.1.4.9 rate\_tarif\_mth**

```
return pow((1.+ int_tarif_temp/100),1./12.)-1.;
```

#### **6.1.1.1.4.10 sexcode\_1**

```
if (sex1=="F")
    return 1;
```

```
return 0;
```

#### **6.1.1.1.4.11 sexcode\_2**

```
if (sex2=="F")
    return 1;
```

```
return 0;
```

#### **6.1.1.1.4.12 annuity\_pmt\_curr\_tot**

```
return ann_pmt_curr
    + ann_pmt_curr_gteed
    + ann_pmt_curr_jl
    + death_ben_curr;
```

#### **6.1.1.1.4.13 ann\_ratio\_res**

```
//int age_takeup_local = takeup_age;
//int yob_local = life->yob;
//int fund = fund_t_factor;
```

```
double temp = annuity_value_res_tbl;
```

```
return max(1.0,temp+1);
```

#### **6.1.1.1.4.14 antisel\_ppn**

```
if(!eq(life->paid_up,"G")){
    if (initial_annuity_purchase == 0.0)
        return 0.;

    return 100. - 100.*no_antisel_at_ann / initial_annuity_purchase;
}

if (life->submodel!="TRAD" && life->paid_up=="G")
    return life->promil;

return 100.;
```

**6.1.1.1.4.15 antisel\_ppn\_res**

```

if(!eq(life->paid_up,"G")){
  if (initial_annuity_purchase == 0.0)
    return 0.;

  return 100. - 100.*no_antisel_at_ann_for_res / initial_annuity_purchase;
}

if (life->submodel!="TRAD" && life->paid_up=="G")
  return life->promil;

return 100.;

```

**6.1.1.1.4.16 no\_antisel\_at\_ann**

```

if (eq(life->submodel,"TERM,ANN"))
  return 0.0;

if (life->submodel=="TRAD" && (annuitization_rate<=0.00001 || !(eq(life->ben_class,"GIMLA"))))
  return 0.0;

if(life->annuitization_rate<=0.00001 || life->res_prop_kitzba<=0.0) // *** need way to distinguish
between policies with and without guarantees, and with and without kitzva option
  return 0.0;

double tagnew_money = 0.0;
double piz_no_as = 0.0;
double prat_no_as = 0.0;
double old_no_as = 0.0;

if (eq(life->submodel, "TRAD") && trad->reserve_basic(maturity_period_w-1) != 0){

  tagnew_money = trad->res_basic_act_newtag(maturity_period_w-1) + trad-
>res_basic_pup_newtag(maturity_period_w-1);

  tagnew_money = tagnew_money
+ (trad->res_basic_act_newtag(maturity_period_w-1) +
trad->res_basic_pup_newtag(maturity_period_w-1)
+ trad->res_basic_act_piz(maturity_period_w-1) + trad-
>res_basic_pup_piz(maturity_period_w-1))
/ trad->reserve_basic(maturity_period_w-1)
* (trad->bonus_if(maturity_period_w-1) + trad-
>bonus_if_pup(maturity_period_w-1)); //Part of bonus belonging to newtag;

  piz_no_as = trad->res_basic_act_piz(maturity_period_w-1) + trad-
>res_basic_pup_piz(maturity_period_w-1);

  prat_no_as = trad->res_basic_act_prat(maturity_period_w-1) + trad-
>res_basic_pup_prat(maturity_period_w-1);

  prat_no_as = prat_no_as
+ (trad->res_basic_act_prat(maturity_period_w-1) + trad-
>res_basic_pup_prat(maturity_period_w-1))
/ trad->reserve_basic(maturity_period_w-1)
* (trad->bonus_if(maturity_period_w-1) + trad-
>bonus_if_pup(maturity_period_w-1)); //Part of bonus belonging to prat;

```

---

```

    old_no_as = trad->res_basic_act_old(maturity_period_w-1) + trad-
>res_basic_pup_old(maturity_period_w-1);

    old_no_as = old_no_as
+ (trad->res_basic_act_old(maturity_period_w-1) + trad-
>res_basic_pup_old(maturity_period_w-1))
/ trad->reserve_basic(maturity_period_w-1)
* (trad->bonus_if(maturity_period_w-1) + trad-
>bonus_if_pup(maturity_period_w-1)); //Part of bonus belonging to prat;
}

if (eq(life->submodel, "UNIT")){ //Tagnew units before retirement

    tagnew_money = life->units_e_newtag(maturity_period_w-1);

    piz_no_as = life->units_e_piz(maturity_period_w-1);

    prat_no_as = life->units_e_prat(maturity_period_w-1);

    old_no_as = life->units_e_old(maturity_period_w-1);
}

if (life->margin_add!="Y") { //No margins on takeup

    tagnew_money = tagnew_money * annuity_takeup_new_tag/100.;
    piz_no_as = piz_no_as * annuity_takeup_piz/100.;
    prat_no_as = prat_no_as * annuity_takeup_prat/100.;
    old_no_as = old_no_as * annuity_takeup_old/100.;
}

else
{ //Margins on takeup

    tagnew_money = tagnew_money * min(annuity_takeup_new_tag/100.*(1 + life-
>margin_annuity_takeup/100.), life->annuity_takeup_max);
    piz_no_as = piz_no_as * min(annuity_takeup_piz/100.*(1 + life->margin_annuity_takeup/100.),
life->annuity_takeup_max);
    prat_no_as = prat_no_as * min(annuity_takeup_prat/100.*(1 + life-
>margin_annuity_takeup/100.), life->annuity_takeup_max);
    old_no_as = old_no_as * min(annuity_takeup_old/100.*(1 + life->margin_annuity_takeup/100.),
life->annuity_takeup_max);
}

piz_no_as = piz_no_as * (1. - life->piz_antiselection_adj/100.);
prat_no_as = prat_no_as * (1. - life->prat_antiselection_adj/100.);
old_no_as = old_no_as * (1. - life->old_antiselection_adj/100.);

if (life->surv_per_ret(maturity_period_w-1) > 0)
    return (tagnew_money + piz_no_as + prat_no_as + old_no_as)
        * retirement_prop
        / life->surv_per_ret(maturity_period_w-1);

return (tagnew_money + piz_no_as + prat_no_as + old_no_as)
    * retirement_prop;

```

**6.1.1.1.4.17 no\_antisel\_at\_ann\_for\_res**

```

if (eq(life->submodel,"TERM,ANN"))
    return 0.0;

if (life->submodel=="TRAD" && (annuitization_rate<=0.00001 || !(eq(life->ben_class,"GIMLA"))))
    return 0.0;

if(life->annuitization_rate<=0.00001 || life->res_prop_kitzba<=0.0) // *** need way to distinguish
between policies with and without guarantees, and with and without kitzva option
    return 0.0;

double tagnew_money = 0.0;

if (eq(life->submodel, "TRAD")){

    tagnew_money = trad->res_basic_act_newtag(maturity_period_w-1) + trad-
>res_basic_pup_newtag(maturity_period_w-1);

    if(trad->reserve_basic(maturity_period_w-1) !=0)
        tagnew_money = tagnew_money
+ (trad->res_basic_act_newtag(maturity_period_w-1) +
trad->res_basic_pup_newtag(maturity_period_w-1)
+ trad->res_basic_act_piz(maturity_period_w-1) + trad-
>res_basic_pup_piz(maturity_period_w-1))
/ trad->reserve_basic(maturity_period_w-1)
* (trad->bonus_if(maturity_period_w-1) + trad-
>bonus_if_pup(maturity_period_w-1)); //Part of bonus belonging to newtag;
}

if (eq(life->submodel, "UNIT")){ //Tagnew units before retirement

    tagnew_money = life->units_e_newtag(maturity_period_w-1);
}

if (life->margin_add!="Y") { //No margins on takeover

    tagnew_money = tagnew_money * annuity_takeup_new_tag/100.;
}

else
{//Margins on takeover

    tagnew_money = tagnew_money * min(annuity_takeup_new_tag/100.*(1 + life-
>margin_annuity_takeup/100.), life->annuity_takeup_max);
}

if (life->surv_per_ret(maturity_period_w-1) > 0)
    return tagnew_money
        * retirement_prop
        / life->surv_per_ret(maturity_period_w-1);

return tagnew_money
    * retirement_prop;

```

**6.1.1.1.4.18 init\_bor\_har**

```

if ( life->submodel == "TERM" || fund_type == "N")

```

```

    return 0.0;

if (eq(life->paid_up, "G"))
    return life->resinforce * life->benefits_curr * life->mgt_deficit_perc * mgt_fee_var/100.;

return life->bor_har_retire(maturity_period_w+1) * ann_takeup_rate(maturity_period_w-1) ;

```

#### 6.1.1.1.4.19 retirement\_prop

```

if (life->retirement_age_lookup(1) > takeup_age)
    return 0.;

if(life->mult_age_ind == 1){
    return retirement_rate/100.;
}

if(takeup_age == life->takeup_age)
    return 1.;

return 0.;

```

#### 6.1.1.1.4.20 temp\_fund\_scalar

```

if(eq(life->prod_code,"a72") && (atoi(life->fund) < 100 || inlist(life->fund, "521,523,527")))
    return xstring(min(atoi(life->fund),50));

if(eq(life->prod_code,"a80-00honi") && (atoi(life->fund) < 100 || inlist(life->fund,
"521,523,527")))
    return xstring(min(atoi(life->fund),51));

if(eq(life->prod_code,"asav") && (atoi(life->fund) <= 52 || inlist(life->fund, "521,523,527")))
    return xstring(min(atoi(life->fund),50));

return life->fund;

```

#### 6.1.1.1.4.21 initial\_annuity\_purchase

```

if(life->submodel == "TRAD")
    return trad->claims_maturity(maturity_period_w) * ann_takeup_rate(maturity_period_w-1) *
retirement_prop;

if(life->submodel == "TERM")
    return 0.0;

if(commence_period_w == life->maturity_period_w)
    return life->units_at_mat(maturity_period_w) * ann_takeup_rate(maturity_period_w-1);

return life->units_at_mat(maturity_period_w) * ann_takeup_rate(maturity_period_w-1) *
retirement_prop;

```

### 6.1.1.2 fund\_cflow

#### 6.1.1.2.1 Columns

##### 6.1.1.2.1.1 decrement\_rate

```

if (t <= commence_period_w || t > maturity_period_w)
    return 0.0;

```

```

double qx = life->death_rate(t);

double rx = life->retirement_prop(t);

double px = life->prem_termination_prop(t);

if (!inlist(unit_type, "Accum_pup,Saving_pup"))
    return 1.-
        (1.-qx)
        *(1.-px) //Money exits active fund at premium termination rate
        *(1.-lapse_rate_bal(t));

return 1.-
    (1.-qx)
    *(1.-rx)//Money exits pupped fund at retirement rate
    *(1.-lapse_rate_bal(t));

```

#### 6.1.1.2.1.2 lapse\_rate\_bal

```

// returning dependent lapse rate from top model

//for pup policies
if (inlist(unit_type, "Accum_pup,Saving_pup"))
    return life->lapse_rate_pup_bal(t);
// for others
if (inlist(unit_type, "Accum_prem,Saving"))
    return life->lapse_rate_act_bal(t);

return 0.0; //Unconditional return

```

#### 6.1.1.2.1.3 premium

```

if (t <= commence_period_w || t > maturity_period_w)
    return 0.0;

if (inlist(unit_type, "Accum_pup,Saving_pup")) {
    if (t==commence_period_w+1)
        return life->premium_gross(t);
    else
        return 0.0;
} // end if

return life->premium(t);

```

#### 6.1.1.2.1.4 death\_claims\_units

```

if (t <= commence_period_w || t > maturity_period_w)
    return 0.0;

// The death benefit equals the unit value at the end of the month

return units_e_bef(t) * life->death_rate(t);

```



**6.1.1.2.1.5 claims\_surrender**

```

if (t <= commence_period_w || t > maturity_period_w)
    return 0.0;

if(life->lapse_total_bal(t)>=1. && inlist(unit_type, "Accum_prem,Saving"))
    return 0.;

return surr_value(t) * lapse_rate_bal(t);

```

**6.1.1.2.1.6 surr\_charge**

```

if (t <= commence_period_w || t > maturity_period_w)
    return 0.0;

if (eq(life->ben_class,"Adif") || (life->pup_sv_charge_rebate_temp==0.0)) // penalty includes the
penalty on policies becoming paid-up
    return surr_penalty_e_bef(t) * life->lapse_total_bal(t); // surrenders and PUPs
else // For Profil there is no penalty on policies becoming paid-up, only later when surrendered
    return surr_penalty_e_bef(t) * lapse_rate_bal(t); // lapse_rate is only surrenders (not
PUPs)

```

**6.1.1.2.1.7 surr\_penalty\_e\_bef**

```

if (t <= commence_period_w || t > maturity_period_w)
    return 0.0;

// The surrender penalty is calculated before
// surrender and death claims have been paid

double surr_penalty = 0.0;

if (inlist(unit_type, "Accum_prem,Saving") && t+elapsed_months >=0)
    surr_penalty = (units_e_bef(t) - death_claims_units(t))
        * surr_chg_perc_units[t+elapsed_months] / 100.;

// calculate surrender penalty less rebate for Profil paid-up (no surrender penalty for pure
savings)
if (eq(unit_type, "Accum_pup") && (life->pup_sv_charge_rebate_temp > 0.0)) {
    for (int i = 1; i<=t; i++) // i is projection month
        if (i+elapsed_months >=0)
            surr_penalty = surr_penalty + pup_units_tt(i+elapsed_months, t-i)
                * max(0.0 , (surr_chg_perc_units[i + elapsed_months] // surrender
charge (full)
                - life->pup_sv_charge_rebate_temp * floor((t-i+1)/12.)) / 100.); //
surrender charge rebate at time of surrender

// (rebate increases after each full year,
surrender month included in count of years since made paid-up, but month when paid-up is not
included)
}
// limit the maximum surrender penalty to the nominal value of the units
surr_penalty = min(surr_penalty, units_e_bef(t));

return surr_penalty;

```

**6.1.1.2.1.8 surr\_value**

```

if (t <= commence_period_w || t > maturity_period_w)
    return 0.0;

```

```

double units_bonus = 0.0;
if (eq(unit_type,"Accum_prem"))
    units_bonus = life->units_bon(t);

// The surrender value and value of units are calculated before
// surrender claims have been paid after death claims paid.

return units_e_bef(t)
    - surr_penalty_e_bef(t)
    - death_claims_units(t)
    + units_bonus;

```

#### 6.1.1.2.1.9 comm\_regular

```

if (t <= commence_period_w || t > maturity_period_w)
    return 0.0;

if(inlist(unit_type,"Accum_prem,Saving")) {
    double prem_perc_unit = 100. * life->basic_perc(t);
    if (eq(unit_type, "Saving"))
        prem_perc_unit = 100. - prem_perc_unit;
    return premium(t)* prem_perc_unit/100. * comm_regular_pc[xint(pol_year(t))] / 100.;
}
else
    return 0.0;

```

#### 6.1.1.2.1.10 comm\_renewal

```

if (t <= commence_period_w || t > maturity_period_w)
    return 0.0;

if(inlist(unit_type,"Accum_prem,Saving") && xint(pol_year(t))>=comm_renewal_year) {
    double prem_perc_unit = life->basic_perc(t)*100.;
    if (eq(unit_type, "Saving"))
        prem_perc_unit = 100. - prem_perc_unit;
    return comm_ren_perc_prem / 100. * premium(t)* prem_perc_unit/100.;
}
else
    return 0.0;

```

#### 6.1.1.2.1.11 comm\_reserve

```

if (t <= commence_period_w || t > maturity_period_w)
    return 0.0;

// % of unit reserve commission are assumed
// to be payable monthly at the beginning of month
if (life->commres_addvat == "Y")
    return units_b_bef(t) * comm_perc_res[xint(pol_year(t))] / 1200. * (1+life->vat/100.);

return units_b_bef(t) * comm_perc_res[xint(pol_year(t))] / 1200.;

```

#### 6.1.1.2.1.12 int\_cred\_units\_e

```

if (t <= commence_period_w || t > maturity_period_w)
    return 0.0;

double temp_inv_rate_m = 0.0;

```

```

int proj_yr = xint(life->proj_year(t));
if(eq(life->projection_type_int, "Rollup"))
    proj_yr = xint(life->proj_year_rollup(t));
proj_yr = max(proj_yr, 0);

if (life->margin_add_asset == "Y" && t == 1 && life->submodel == "UNIT" && life->par_nonpar == "P")
    temp_inv_rate_m = life->asset_shock;
else
    temp_inv_rate_m = life->inv_rate_mth_t[proj_yr];

return units_b(t) * temp_inv_rate_m;

```

#### 6.1.1.2.1.13 pol\_year

```
return life->pol_year(t);
```

#### 6.1.1.2.1.14 units\_b

```

if (t <= commence_period_w || t > maturity_period_w)
    return 0.0;

return units_b_bef(t) + alloc_units(t) - cover_charge(t) ;

```

#### 6.1.1.2.1.15 units\_b\_bef

```

if (t <= commence_period_w || t > maturity_period_w)
    return 0.0;

double new_bonus = 0.0;
if (eq(unit_type, "Accum_pup"))
    new_bonus = life->units_bon(t-1) * life->pup_rate_bal_dep(t-1) * life->surv_per_ret(t-1);

// SV penalty is only taken off units becoming paid-up for Adif, not for Profil.
// For Adif the units moved from prem-paying to PUP are after removing the SV penalty,
// and the SV penalty is removed from the prem-paying units in the formula surr_charge.
// For Profil the surrender charge is not taken off when made paid-up. Later if the paid-up units
// are surrendered, a penalty is applied.
double sv_charge = 0.0;
if (eq(life->ben_class,"Adif") || (life->pup_sv_charge_rebate_temp==0.0))
    sv_charge = surr_chg_perc_units[t+elapsed_months]/100.;

double new_pup = 0.0;
// In the PUP fund:
// Add the unit balance in respect of policies
// which became paid-up in month t-1 to
// the unit balance of policies which were
// already paid up at the beginning of t-1
if (inlist(unit_type, "Accum_pup,Saving_pup")){
    new_pup = units_b_bef_pup(t) * (1.0 - sv_charge) // take off surrender penalty
              + new_bonus; // add bonus accrued up to the paying up month

    /*if(t==1){
        log_strm<<"Unit type: "<<unit_type<<endl;
        log_strm<<"New pup: "<<new_pup<<endl;
        log_strm<<"units_b_bef_pup: "<<units_b_bef_pup(t)<<endl;
        log_strm<<"sv_charge: "<<sv_charge<<endl;
        log_strm<<"Bonus: "<<new_bonus<<endl;
    }

```

```

    }*/
}
else {
    // In the premium-paying fund:
    // Deduct the unit balance in respect of policies
    // which became paid-up in month t-1 from
    // the unit balance of policies which were
    // premium paying at the beginning of t-1
    new_pup = - units_b_bef_pup(t) * (1.0 - sv_charge); // take off surrender penalty
    // Do not need to remove bonus, because bonus is accrued externally to the units in
    bonus_if(t) and there it is reduced.
}

return units_e(t-1) + new_pup;

```

#### 6.1.1.2.1.16 units\_b\_bef\_pup

```

if (inlist(unit_type, "Accum_pup,Accum_prem"))
    return life->units_b_bef_pup_acc(t);
else
    return life->units_b_bef_pup_sav(t);

```

#### 6.1.1.2.1.17 units\_e

```

if (t <= commence_period_w || t >= maturity_period_w)
    return 0.0;

if (t == 0 && !eq(projection_type,"Rollup")) {
    double temp = 0.0;
    if ((paid_up=="Y") && (inlist(unit_type, "Accum_pup,Saving_pup")))
        temp = 1.0;
    if ((paid_up=="N") && (inlist(unit_type, "Accum_prem,Saving")))
        temp = 1.0;

    return unit_value_if * benefits_curr
        * temp;
}

// Management fees are deducted at the end of each month, are
// expressed as a % of the value of the units and are deducted
// from the units.
// Assume that unit value can be negative

double units_bonus = 0.0; // claims_surrender includes persistency bonus, but this must not come
// off units which excludes the bonus
if (eq(unit_type,"Accum_prem"))
    units_bonus = life->units_bon(t);

if (t == 0 && eq(projection_type,"Rollup")) {

    return (units_e_bef(t) - death_claims_units(t) - claims_surrender(t)
        + units_bonus * lapse_rate_bal(t) - surr_charge(t)
        + premium_nb_sp) * (life->surv_per_ret(t));
}

return (units_e_bef(t)
    - death_claims_units(t)
    - claims_surrender(t)

```

```

    + units_bonus * lapse_rate_bal(t)
    - surr_charge(t)) * (life->surv_per_ret(t));

```

#### 6.1.1.2.1.18 units\_e\_bef

```

if (t <= commence_period_w || t > maturity_period_w)
    return 0.0;

// If projection start, reset to be start account value
if (t==0 && !eq(projection_type,"Rollup"))
    return units_e(t);

return units_b(t) + int_cred_units_e(t) - management_fee(t);

```

#### 6.1.1.2.1.19 alloc\_units

```

if (t <= commence_period_w || t > maturity_period_w)
    return 0.0;

if (premium(t) == 0.0)
    return 0.0;

if (inlist(unit_type, "Accum_prem,Saving")) {
    double prem_perc_unit = min(100.,life->basic_perc(t)*100.);
    if (eq(unit_type, "Saving"))
        prem_perc_unit = 100. - prem_perc_unit;
    return premium(t)*prem_perc_unit/100.*allocation_rate(t);
}
else // paid up
    if (t==commence_period_w+1 && (paid_up=="Y") ) {
        double prem_perc_unit = min(100.,life->basic_perc(t)*100.);
        if (eq(unit_type, "Saving_pup"))
            prem_perc_unit = 100. - prem_perc_unit;
        return premium(t)*prem_perc_unit/100.;
    }
    else
        return 0.0;

```

#### 6.1.1.2.1.20 allocation\_rate

```

if (t < commence_period_w || t > maturity_period_w)
    return 0.0;

// do not calculate for paid-up units
if (eq(unit_type, "Accum_pup") || eq(unit_type, "Saving_pup") || paid_up=="Y" )
    return 0.0;

// set up allocation rate periods
double dur = t + elapsed_months;
int n = alloc_rate_period[1];
int i=1;

// set up allocation rates
while ((dur >n) && (i<14)) {
    i = i + 1;
    n = n + alloc_rate_period[i];
}

double rate = 0.0;

```

```
//check if there is a maximum premium charge
if (life->alloc_limit>0.0) {
    if (life->premium(t)== 0.0 )
        rate = 1.0;
    else
        rate = max(alloc_rate[i]/100.,1.0-(life->alloc_limit * life->surv_ret(t-1)/life-
>premium(t)));
}
else
    rate = alloc_rate[i] / 100.;

double margin_alloc = 0.0;

if(life->margin_add_discount == "Y"){

    life->margin_disc_col_key = "DN_prem_" + life->ben_class;

    margin_alloc = - life->mgt_fee_disc/100.;

}

return min(rate + margin_alloc, 1.);
```

#### 6.1.1.2.1.21 cover\_charge

```
if (inlist(unit_type,"Accum_pup,Saving_pup")) // *** may need to have riders for paid-up as well?
    return 0.;

double prop = 0.;
double units = units_b_bef(t) + alloc_units(t);
if (units > 0.) {

    double tot_units = 0.;
    if (eq(unit_type, "Accum_prem"))
        tot_units = units + saving->units_b_bef(t) + saving->alloc_units(t);
    if (eq(unit_type, "Saving"))
        tot_units = units + accum->units_b_bef(t) + accum->alloc_units(t);
    if (tot_units > 0.) {
        prop = units/tot_units;
    }
}

return prop * life->cover_charge(t);
```

#### 6.1.1.2.1.22 int\_rate\_net\_cumm

```
//Can probably be removed, but leave in just for info

if (t <= commence_period_w || t > maturity_period_w)
    return 0.0;

if(t==0)
    return 0.0;

double int_prev = 0.0;

if(life->cal_month(t) != 1){
```

```

        if(t==1)
            int_prev = life->mgt_deficit_perc;
        else
            int_prev = int_rate_net_cumm(t-1);
    }

```

```

return (1+net_interest_rate(t))*
        (1+int_prev)
        -1.;

```

#### 6.1.1.2.1.23 management\_fee\_fixed

```

if (t <= commence_period_w || t > maturity_period_w)
    return 0.0;

//Management fee discount
double margin_mgt = 0.0;

if(life->margin_add_discount == "Y" && par_nonpar!="N"){

    life->margin_disc_col_key = "DN_acc_" + life->ben_class;

    margin_mgt = life->mgt_fee_disc/1200.;

}

//No split if no variable management fees
if (mgt_fee_variable == 0){

    if (margin_mgt != 0.0)
        margin_mgt = max(margin_mgt, -management_fee_rate(t));

    return (units_b(t) + int_cred_units_e(t))
           * (management_fee_rate (t) + margin_mgt);

}

if (margin_mgt != 0.0)
    margin_mgt = max(margin_mgt, - mgt_fee_fixed/1200.);

return (units_b(t) + int_cred_units_e(t))
       * (mgt_fee_fixed/1200. + margin_mgt);

```

#### 6.1.1.2.1.24 management\_fee\_variable

```

if (t <= 0 || t > maturity_period_w)
    return 0.0;

if(mgt_fee_variable == 0.0)
    return 0.0;

if (inlist(unit_type, "Accum_pup,Saving_pup")){

```

```

        if(life->units_b_pup(t) > 0)
            return life->management_fee_variable_pup(t)
                * units_b(t)
                / life->units_b_pup(t);
    }
    else
    {
        if(life->units_b_active(t) > 0)
            return life->management_fee_variable(t)
                * units_b(t)
                / life->units_b_active(t);
    }

    return 0.0;

```

#### 6.1.1.2.1.25 net\_interest\_rate

```

if (t <= commence_period_w || t > maturity_period_w)
    return 0.0;

double temp_inv_rate_m = 0.0;

int proj_yr = xint(life->proj_year(t));
if(eq(life->projection_type_int, "Rollup"))
    proj_yr = xint(life->proj_year_rollup(t));
proj_yr = max(proj_yr, 0);

if (life->margin_add_asset == "Y" && t == 1 && life->submodel == "UNIT" && life->par_nonpar == "P")
    temp_inv_rate_m = life->asset_shock;
else
    temp_inv_rate_m = life->inv_rate_mth_t[proj_yr];

return (1+temp_inv_rate_m)
    * (1- mgt_fee_fixed/1200.)
    -1.;

```

#### 6.1.1.2.1.26 management\_fee

```

if (t <= commence_period_w || t > maturity_period_w)
    return 0.0;

return management_fee_variable(t)
    + management_fee_fixed (t);

```

#### 6.1.1.2.1.27 management\_fee\_rate

```

if (t <= commence_period_w || t > maturity_period_w)
    return 0.0;

int proj_yr = xint(life->proj_year(t));
int proj_yr_dn = xint(life->proj_year(t-1));

if(eq(life->projection_type_int, "Rollup")){
    proj_yr = xint(life->proj_year_rollup(t));
    proj_yr_dn = xint(life->proj_year_rollup(t-1));
}

```



```

    }

    int proj_yr_pos = max(proj_yr, 0);
    double temp_inv_rate_m = 0.0;

    if (par_nonpar=="N" || eq(life->ben_class,"adif")){
        if (t>2 && (proj_yr ==proj_yr_dn))
            return management_fee_rate(t-1);}

    if (life->margin_add_asset == "Y" && t == 1 && life->submodel == "UNIT" && life->par_nonpar ==
    "P"){
        temp_inv_rate_m = life->asset_shock;
        if (life->dump_vars == "Y")
            log_strm<<"asset shock at time "<<t<<": "<<temp_inv_rate_m<<endl;
    }
    else
        temp_inv_rate_m = life->inv_rate_mth_t[proj_yr_pos];

    if (par_nonpar=="N") // for non-participating funds, mgt_fee_fixed is the credited interest rate
    (net return to policyholder)
        return life->inv_rate_mth_t[proj_yr_pos] - (pow(1. + mgt_fee_fixed / 100. , 1. / 12.) - 1.);

    if (eq(life->ben_class,"adif")) {
        return temp_inv_rate_m - (1.-mgt_fee_variable/100.)*
            ((1.+temp_inv_rate_m)*(1.- mgt_fee_fixed/1200.) - 1.);
    }

    if( life->year_start >= 2013) //different monthly-compounded management fee for before/after 2013
        return pow(1. + management_fee_rate_annual(t) , 1. / 12.) - 1.;
    else
        return 1.- pow(1. - management_fee_rate_annual(t) , 1. / 12.);

```

#### 6.1.1.2.1.28 management\_fee\_rate\_annual

```

if (t <= commence_period_w || t > maturity_period_w)
    return 0.0;

if (par_nonpar=="N" || eq(life->ben_class,"adif"))
    return 0.0;

if (t>1 && (life->pol_month(t) > 1))
    return management_fee_rate_annual(t-1);

double mgtfee_rate_Data = mgt_fee_fixed / 100.;
double mgtfee_rate = mgtfee_rate_Data;
double local_surv =0.0;

if (decrements_apply=="N")
    local_surv = 1.0;
    else
        local_surv = life->surv_act_bal(t-1) + life->surv_pup_bal(t-1);

if (t <= 1 || life->mgtfee_age == -99999. || local_surv==0)
    return mgtfee_rate_Data;

if (life->mgtfee_age>0.0){
    if( life->age_last(t) >= life->mgtfee_age)

```

```

        mgtfee_rate = life->mgtfee_age_after / 100.;
    }
    if (life->mgtfee_acc>0.0 && t >0) {
        if(life->mgtfee_acc_after>=0){
            if (life->units_b(t)>= life->mgtfee_acc * local_surv )
                mgtfee_rate = life->mgtfee_acc_after / 100. ;
            else {
                if (life->mgtfee_from_senior >0.0 && ((t + life->elapsed_months) >= life-
>mgtfee_from_senior) && !eq(life->paid_up,"Y"))
                    mgtfee_rate = life-> mgtfee_senior/100.;
            }
        }
        else
            mgtfee_rate = max(life-> mgtfee_orig/100.+life->mgtfee_acc_after/100.*xint((life-
>units_b(t)/local_surv)/life->mgtfee_acc),life->mgtfee_floor/100.);
    }
    if (life->mgtfee_from_dthben>0.0){
        mgtfee_rate = life-> mgtfee_orig/100.;
        double mgtfee_discount1 = 0.0;
        double mgtfee_discount2 = 0.0;

        // For Special MgtFee discount of format 2105

        if (life->mgtfee_disc_mth >0) {
            if (t + life->elapsed_months <= life-> mgtfee_disc_mth) // For the first 24 months
zero mgtFee
                return min(mgtfee_rate_Data,life-> mgtfee_disc_after/100.);

            if( life->death_benefit(t) < life->mgtfee_from_dthben* local_surv) // if DeathBen <
500k, mgtfee = 1.05%
                return min(mgtfee_rate_Data,mgtfee_rate);

            mgtfee_discount1 = life-> mgtfee_dthben/100.; //if DeathBen >= 500k, mgtfee = 1.05%
-0.25%

            if(life->mgtfee_senior <= -0.0001){
                mgtfee_discount2 = life->mgtfee_senior/100. // From 120 months 0.01% discount
every year
                    * max(0,xint((t +life->elapsed_months -life-
>mgtfee_from_senior - 1)/12.+1));
            }

            mgtfee_rate = max(mgtfee_rate + mgtfee_discount1 + mgtfee_discount2, life-
>mgtfee_floor/100.); // Mgtfee no less than 0.75%
            return min(mgtfee_rate_Data,mgtfee_rate);
        }
        if(life->mgtfee_from_dthben>0.0){
            mgtfee_discount1 = life-> mgtfee_dthben/100.
                *max(xint(life->death_benefit(t)/(life-
>mgtfee_from_dthben*local_surv)),0);
        }
        if(life->mgtfee_senior <= -0.0001){
            mgtfee_discount2 = life->mgtfee_senior/100.
                * max(0,xint((t +life->elapsed_months -life-
>mgtfee_from_senior - 1)/12.+1));
        }
    }

```

```

    mgtfee_rate = max(mgtfee_rate + mgtfee_discount1 + mgtfee_discount2, life-
>mgtfee_floor/100.);

    if(life->mgtfee_max_dthben > 0.0 && (life->death_benefit(t)>= life-
>mgtfee_max_dthben*local_surv))
        mgtfee_rate = life->mgtfee_floor/100.;
    }

return min(mgtfee_rate_Data,mgtfee_rate);

```

### 6.1.1.2.2 External Functions

<No External Functions Exist>

### 6.1.1.2.3 Temporary Tables

#### 6.1.1.2.3.1 pup\_units\_tt

```

// column c is duration since premium cessation (pup) with 0 being the month of prem. cess.
// row r is months since policy started with r=1 being the first policy month

if (c<0 || r<=1+elapsed_months) // first real row is 2 [+elaps.] which contains units made pup in
month 1 [+elaps.]
    return 0.;

if (c==0) // units made paid up in month r-1-elapsed_months (at end of month; no pup-lapses in same
month)
    return units_b_bef_pup(r - elapsed_months);

if (pup_units_tt(r,c-1) <= 0.0000001)
    return 0.; // to avoid divide by zero

return ( pup_units_tt(r,c-1)
        + pup_units_tt(r,c-1) / pup_units_tt.sum_of_diagonal(r+c-1) // proportion of interest
& management fee applying to this tranche of pup units
        * (int_cred_units_e(r+c-elapsed_months-1)
          - management_fee(r+c-elapsed_months-1)))
        * (1. - life->death_rate(r+c-elapsed_months-1))
        * (1. - lapse_rate_bal(r+c-elapsed_months-1));

```

### 6.1.1.2.4 Scalars

#### 6.1.1.2.4.1 premium\_nb\_sp

```

// Calculate single premium at eom t=0, required to be paid
// such that the calculated account balance = current actual account balance
if (!eq(projection_type,"Rollup"))
    return 0.0;

double temp = 0.0;
if ((paid_up=="Y") && (inlist(unit_type, "Accum_pup,Saving_pup")))
    temp = 1.0;
if ((paid_up=="N") && (inlist(unit_type, "Accum_prem,Saving")))
    temp = 1.0;

return (unit_value_if * benefits_curr * temp) -

```

```
(units_b(0) + int_cred_units_e(0) - management_fee(0)
- death_claims_units(0) - claims_surrender(0)
- surr_charge(0));
```

### 6.1.1.3 life\_cflow

#### 6.1.1.3.1 Columns

##### 6.1.1.3.1.1 blank\_test

```
return 0.0;
```

##### 6.1.1.3.1.2 cashflow\_b\_bef\_ret

```
if (t <= commence_period_w || t > maturity_period_ann)
    return 0.0;
```

```
return income_b(t) - outgo_b_before_ret(t);
```

##### 6.1.1.3.1.3 claims\_lrc\_q1\_pv

```
if (t < commence_period_w || t >= maturity_period_w || eq(paid_up,"C"))
    return 0.0;
```

```
int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));
```

```
return (claims_lrc_q1_pv(t+1) + claims_lrc_q1(t+1))
        * v_month_t[proj_yr];
```

##### 6.1.1.3.1.4 claims\_lrc\_q2\_pv

```
if (t < commence_period_w || t >= maturity_period_w || eq(paid_up,"C"))
    return 0.0;
```

```
int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));
```

```
return (claims_lrc_q2_pv(t+1) + claims_lrc_q2(t+1))
        * v_month_t[proj_yr];
```

##### 6.1.1.3.1.5 claims\_lrc\_q3\_pv

```
if (t < commence_period_w || t >= maturity_period_w || eq(paid_up,"C"))
    return 0.0;
```

```
int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));
```

```
return (claims_lrc_q3_pv(t+1) + claims_lrc_q3(t+1))
        * v_month_t[proj_yr];
```

##### 6.1.1.3.1.6 claims\_lrc\_q4\_pv

```
if (t < commence_period_w || t >= maturity_period_w || eq(paid_up,"C"))
    return 0.0;
```

```
int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

return      (claims_lrc_q4_pv(t+1) + claims_lrc_q4(t+1))
            * v_month_t[proj_yr];
```

#### **6.1.1.3.1.7 claims\_re\_lrc\_q1\_pv**

```
if (t < commence_period_w || t >= maturity_period_w || eq(paid_up,"C"))
    return 0.0;

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

return      (claims_re_lrc_q1_pv(t+1) + claims_re_lrc_q1(t+1))
            * v_month_t[proj_yr];
```

#### **6.1.1.3.1.8 claims\_re\_lrc\_q2\_pv**

```
if (t < commence_period_w || t >= maturity_period_w || eq(paid_up,"C"))
    return 0.0;

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

return      (claims_re_lrc_q2_pv(t+1) + claims_re_lrc_q2(t+1))
            * v_month_t[proj_yr];
```

#### **6.1.1.3.1.9 claims\_re\_lrc\_q3\_pv**

```
if (t < commence_period_w || t >= maturity_period_w || eq(paid_up,"C"))
    return 0.0;

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

return      (claims_re_lrc_q3_pv(t+1) + claims_re_lrc_q3(t+1))
            * v_month_t[proj_yr];
```

#### **6.1.1.3.1.10 claims\_re\_lrc\_q4\_pv**

```
if (t < commence_period_w || t >= maturity_period_w || eq(paid_up,"C"))
    return 0.0;

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

return      (claims_re_lrc_q4_pv(t+1) + claims_re_lrc_q4(t+1))
            * v_month_t[proj_yr];
```

#### **6.1.1.3.1.11 claims\_re\_lrc\_yr2plus\_pv**

```
if (t < commence_period_w || t >= maturity_period_w || eq(paid_up,"C"))
    return 0.0;
```

```

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

return      (claims_re_lrc_yr2plus_pv(t+1) + claims_re_lrc_yr2plus(t+1))
            * v_month_t[proj_yr];

```

#### 6.1.1.3.1.12 expense\_claims\_lrc\_q1\_pv

```

if (t < commence_period_w || t > maturity_period_ann || (exp_claim_perc == 0. && exp_claim_fix ==0.)
|| eq(paid_up,"C"))
    return 0.0;

```

```

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

```

```

return      expense_claims_lrc_q1_pv(t+1) * v_month_t[proj_yr] + expense_claims_lrc_q1(t+1);

```

#### 6.1.1.3.1.13 expense\_claims\_lrc\_q2\_pv

```

if (t < commence_period_w || t > maturity_period_ann || (exp_claim_perc == 0. && exp_claim_fix ==0.)
|| eq(paid_up,"C"))
    return 0.0;

```

```

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

```

```

return      expense_claims_lrc_q2_pv(t+1) * v_month_t[proj_yr] + expense_claims_lrc_q2(t+1);

```

#### 6.1.1.3.1.14 expense\_claims\_lrc\_q3\_pv

```

if (t < commence_period_w || t > maturity_period_ann || (exp_claim_perc == 0. && exp_claim_fix ==0.)
|| eq(paid_up,"C"))
    return 0.0;

```

```

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

```

```

return      expense_claims_lrc_q3_pv(t+1) * v_month_t[proj_yr] + expense_claims_lrc_q3(t+1);

```

#### 6.1.1.3.1.15 expense\_claims\_lrc\_q4\_pv

```

if (t < commence_period_w || t > maturity_period_ann || (exp_claim_perc == 0. && exp_claim_fix ==0.)
|| eq(paid_up,"C"))
    return 0.0;

```

```

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

```

```

return      expense_claims_lrc_q4_pv(t+1) * v_month_t[proj_yr] + expense_claims_lrc_q4(t+1);

```

#### 6.1.1.3.1.16 expense\_claims\_lrc\_yr2plus\_pv

```

if (t < commence_period_w || t > maturity_period_ann || (exp_claim_perc == 0. && exp_claim_fix ==0.)
|| eq(paid_up,"C"))
    return 0.0;

```

```

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

return    expense_claims_lrc_yr2plus_pv(t+1) * v_month_t[proj_yr] +
expense_claims_lrc_yr2plus(t+1);

```

#### 6.1.1.3.1.17 initialise

```

//copied from set_other_variables
// reduce SI if premium not high enough

if (eq(ben_class,"adif") && (paid_up=="N") && (sum_ins_curr > 0.000001)) {
    double rate = prem_rates_extra_tt(xint(age_at_issue+elapsed_months/12.), sex_smoker_code) /
1000.0;
    if (sum_ins_curr >= prem_risk_max/100. * prem_curr / rate) {
        sum_ins_curr = prem_risk_max/100. * prem_curr / rate
            + (1.-pre_risk_max/100.) * prem_curr / prem_freq
            *
sum_ins_basic_tt(xint(age_at_issue),sex_smoker_code)/100.;
        sum_ins_curr = max( sum_ins_curr, 0.0);
        log_strm << "Fixed sum-insured for policy "+pol_number+" is too high for premium, so
reduced to "<< sum_ins_curr << endl;
    }
}

int i = 0.;

if (submodel == "UNIT"){
    if (!eq(surr_chg_set,"default") && !eq(surr_chg_set,"zero")){
//copied from set_accum_fund and set_accum_pup_fund
        for (i = 0; i < 1000; i++){
            accum->surr_chg_perc_units[i] = surr_charge_tt(i,0);
            acc_pup->surr_chg_perc_units[i] = surr_charge_tt(i,0);
            saving->surr_chg_perc_units[i] = surr_charge_tt(i,1);
            saving_pup->surr_chg_perc_units[i] = surr_charge_tt(i,1);
        }
    }
}

// Reduce reserve commission according to the reduced management fee
if (eq(ben_class,"profil") && (mgtfee_age>0.0) && (mgt_fee_fixed>0.0)) {
    for (i = xint(mgtfee_age) - xint(age_last(commence_period_w)); i<116; i++) {
        sm_accum->comm_perc_res[i] = comm_perc_res_a[i] *mgtfee_age_after/ mgt_fee_fixed;
    }
}

int j = min_retirement_age;

ret_prop_array.resize(sm_annuity.size());
ret_prop_array[0] = 1;
for(int i = 1; i < sm_annuity.size(); i++){
    if(age_last(1) + 1 > j+i-1)
        ret_prop_array[i] = 1;
    else
        ret_prop_array[i] = ret_prop_array[i-1] * (1 - sm_annuity[ann_index_map[j+i-
1]]->retirement_prop);
}

```

```
}

```

```
return 0.0;

```

#### 6.1.1.3.1.18 int\_units\_piz\_active

```
if(submodel != "UNIT")
    return NO_AVG;

```

```
if (units_e_piz_active(t-1) <= 0.0)
    return 0.0;

```

```
if (sm_accum->units_b(t) + sm_saving->units_b(t) <= 0.0)
    return 0.0;

```

```
double piz = units_e_piz_active(t-1) + units_e_piz_int_active (t-1);

```

```
//Pup to deduct
double new_pup = 0;

```

```
if (sm_accum->units_e(t-1) + sm_saving->units_e(t-1) > 0)
    new_pup = (
        units_b_bef_pup_acc(t) * (1. - sm_accum-
>surrg_chg_perc_units[t+elapsed_months]/100.)
        + units_b_bef_pup_sav(t) * (1. - sm_saving-
>surrg_chg_perc_units[t+elapsed_months]/100.)
    )
    * piz
    / (sm_accum->units_e(t-1) + sm_saving->units_e(t-1));

```

```
piz = piz - new_pup + alloc_units_piz(t);

```

```
double other_deductions = 0.0;

```

```
if ((sm_accum->units_b_bef(t) + sm_saving->units_b_bef(t)) > 0.0)
    other_deductions = other_deductions

```

```

        +
        (sm_accum->cover_charge(t) + sm_saving-
>cover_charge(t))
        * (piz - new_pup)
        / (sm_accum->units_b_bef(t) + sm_saving-
>units_b_bef(t));
    else{

```

```

        if(alloc_units(t) > 0.0)
            other_deductions = other_deductions
            + (sm_accum->cover_charge(t) + sm_saving-
>cover_charge(t))
            * piz
            / alloc_units(t);

```

```
    }

```

```
piz = piz - other_deductions;

```



```

double temp_inv_rate_m = 0.0;

int proj_yr = xint(proj_year(t));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t));
proj_yr = max(proj_yr, 0);

if (margin_add_asset == "Y" && t == 1 && submodel == "UNIT" && par_nonpar == "P")
    temp_inv_rate_m = asset_shock;
else
    temp_inv_rate_m = inv_rate_mth_t[proj_yr];

double int_cred = piz * temp_inv_rate_m;

double mgt = 0;
if(sm_accum->units_b(t) + sm_saving->units_b(t) != 0)
    mgt = (sm_accum->management_fee(t) + sm_saving->management_fee(t))
          * piz
          / (sm_accum->units_b(t) + sm_saving->units_b(t));

return int_cred - mgt;

```

#### 6.1.1.3.1.19 int\_units\_piz\_pup

```

if(submodel != "UNIT")
    return NO_AVG;

double temp_inv_rate_m = 0.0;

int proj_yr = xint(proj_year(t));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t));
proj_yr = max(proj_yr, 0);

if (margin_add_asset == "Y" && t == 1 && submodel == "UNIT" && par_nonpar == "P")
    temp_inv_rate_m = asset_shock;
else
    temp_inv_rate_m = inv_rate_mth_t[proj_yr];

if(paid_up == "Y" && paid_up_input=="N"){

    if(units_b(t) <= 0.0)
        return 0.0;

    double piz = units_e_piz_pup(t-1) + units_e_piz_int_pup(t-1);

    double new_bonus = 0;
    if(units_e(t-1) != 0)
        new_bonus = units_bon(t-1)
                    * pup_rate_bal_dep(t-1)
                    * surv_per_ret(t-1)
                    * piz
                    / units_e(t-1);

    piz = piz + new_bonus;

```

```

double int_cred = piz * temp_inv_rate_m;

double mgt = 0;
if(units_b(t) != 0)
    mgt = (management_fees_fixed_active(t) + management_fees_var_active(t))
          * piz
          / units_b(t);

return int_cred - mgt;
}

if (sm_acc_pup->units_b(t) + sm_saving_pup->units_b(t) <= 0.0)
    return 0.0;

double piz = units_e_piz_pup(t-1) + units_e_piz_int_pup(t-1);

//Pup to deduct
double new_pup = 0.0;
double new_bonus = 0.0;

if (sm_accum->units_e(t-1) + sm_saving->units_e(t-1) > 0.0){
    if (paid_up == "N"){
        new_pup = (
            units_b_bef_pup_acc(t) * (1. - sm_accum-
>surrg_chg_perc_units[t+elapsed_months]/100.)
            + units_b_bef_pup_sav(t) * (1. - sm_saving-
>surrg_chg_perc_units[t+elapsed_months]/100.)
            )
            * (units_e_piz_active(t-1) + units_e_piz_int_active(t-1)) /
(sm_accum->units_e(t-1) + sm_saving->units_e(t-1));

        new_bonus = units_bon(t-1) * pup_rate_bal_dep(t-1) * surv_per_ret(t-1);

        new_bonus = new_bonus
            * (units_e_piz_active(t-1) + units_e_piz_int_active(t-1)) /
(sm_accum->units_e(t-1) + sm_saving->units_e(t-1));

        new_pup = new_pup + new_bonus;
    }
}

piz = piz + new_pup;

double int_cred = piz * temp_inv_rate_m;

double mgt = 0;
if(sm_acc_pup->units_b(t) + sm_saving_pup->units_b(t) != 0)
    mgt = (sm_acc_pup->management_fee(t) + sm_saving_pup->management_fee(t))
          * piz
          / (sm_acc_pup->units_b(t) + sm_saving_pup->units_b(t));

return int_cred - mgt;

```

**6.1.1.3.1.20 outgo\_b\_before\_ret**

```
return comm_total(t)
      + expense_total_pre_ret(t)
      + premium_re(t);
```

**6.1.1.3.1.21 res\_ann\_deficiency**

```
if(inlist(submodel,"UNIT,TRAD,ANN") && res_prop_kitzba > 0.0){
  if(mult_age_ind == 1)
    return sm_annuity->res_ann_deficiency(t);
  return sm_annuity[ann_index_map[takeup_age]]->res_ann_deficiency(t);
}
return NO_AVG;
```

**6.1.1.3.1.22 retirement\_age\_lookup**

```
if(mult_age_ind == 1)
  return age_last(t) + 1;
```

```
return 0;
```

**6.1.1.3.1.23 retirement\_prop**

```
if(submodel == "ANN" || submodel == "TERM")
  return 1;
```

```
if(t <= 0)
  return 0; //this is to allow for cases where policy holder enters at the exact age, e.g. 64
and when RI prems are calculated.
```

```
if(t < mat_period_min || t > maturity_period_w)
  return 0.;
```

```
if(mult_age_ind == 1){
  if(retirement_age_lookup(1) > sm_annuity[sm_annuity.size()-1]->takeup_age)
    return 1.;

  if(xint(pol_month(t)) == 12){
    return retirement_rate/100.;
  }
  return 0.;
}
```

```
return 1.;
```

**6.1.1.3.1.24 rider\_perc\_allowed**

```
if (t <= commence_period_w || t > maturity_period_w)
  return NO_AVG;
```

```
if (eq(ben_class,"Adif"))
  return 100.;
```

```
double temp = 0;
```

```
if (t >= 0)
  temp = charge_amount_tt.sum_of_row(t) * surv_act_prm(t-1);
```

```

if (temp<=0.0)
    return 100.;

if (inlist(policy_type,"Managers,Selfemp"))
    if (rider_max_perc/100. * alloc_units(t) * tagmulim_perc/100. < temp)
        return rider_max_perc * alloc_units(t) * tagmulim_perc/100. / temp;
    else
        return 100.;
else // Private
    if (rider_max_perc/100. * premium(t) < temp)
        return rider_max_perc * premium(t) / temp;
    else
        return 100.;

```

#### 6.1.1.3.1.25 riskadj\_gross\_rel\_q1\_pv

```

if (t < commence_period_w || t >= maturity_period_ann || eq(paid_up,"C"))
    return 0.0;

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

return (riskadj_gross_rel_q1_pv(t+1) + riskadj_gross_rel_q1(t+1))
        * v_month_t[proj_yr];

```

#### 6.1.1.3.1.26 riskadj\_gross\_rel\_q2\_pv

```

if (t < commence_period_w || t >= maturity_period_ann || eq(paid_up,"C"))
    return 0.0;

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

return (riskadj_gross_rel_q2_pv(t+1) + riskadj_gross_rel_q2(t+1))
        * v_month_t[proj_yr];

```

#### 6.1.1.3.1.27 riskadj\_gross\_rel\_q3\_pv

```

if (t < commence_period_w || t >= maturity_period_ann || eq(paid_up,"C"))
    return 0.0;

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

return (riskadj_gross_rel_q3_pv(t+1) + riskadj_gross_rel_q3(t+1))
        * v_month_t[proj_yr];

```

#### 6.1.1.3.1.28 riskadj\_gross\_rel\_q4\_pv

```

if (t < commence_period_w || t >= maturity_period_ann || eq(paid_up,"C"))
    return 0.0;

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

```

```
return      (riskadj_gross_rel_q4_pv(t+1) + riskadj_gross_rel_q4(t+1))
            * v_month_t[proj_yr];
```

#### 6.1.1.3.1.29 riskadj\_gross\_rel\_total\_pv

```
if (t < commence_period_w || t >= maturity_period_ann)
    return 0.0;
```

```
int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));
```

```
return      (riskadj_gross_rel_total_pv(t+1) + riskadj_gross_rel_total(t+1))
            * v_month_t[proj_yr];
```

#### 6.1.1.3.1.30 riskadj\_gross\_rel\_yr2plus\_pv

```
if (t < commence_period_w || t >= maturity_period_ann || eq(paid_up,"C"))
    return 0.0;
```

```
int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));
```

```
return      (riskadj_gross_rel_yr2plus_pv(t+1) + riskadj_gross_rel_yr2plus(t+1))
            * v_month_t[proj_yr];
```

#### 6.1.1.3.1.31 riskadj\_re\_rel\_q1\_pv

```
if (t < commence_period_w || t >= maturity_period_ann || eq(paid_up,"C"))
    return 0.0;
```

```
int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));
```

```
return      (riskadj_re_rel_q1_pv(t+1) + riskadj_re_rel_q1(t+1))
            * v_month_t[proj_yr];
```

#### 6.1.1.3.1.32 riskadj\_re\_rel\_q2\_pv

```
if (t < commence_period_w || t >= maturity_period_ann || eq(paid_up,"C"))
    return 0.0;
```

```
int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));
```

```
return      (riskadj_re_rel_q2_pv(t+1) + riskadj_re_rel_q2(t+1))
            * v_month_t[proj_yr];
```

#### 6.1.1.3.1.33 riskadj\_re\_rel\_q3\_pv

```
if (t < commence_period_w || t >= maturity_period_ann || eq(paid_up,"C"))
    return 0.0;
```

```
int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));
```

```

return      (riskadj_re_rel_q3_pv(t+1) + riskadj_re_rel_q3(t+1))
             * v_month_t[proj_yr];

```

#### 6.1.1.3.1.34 riskadj\_re\_rel\_q4\_pv

```

if (t < commence_period_w || t >= maturity_period_ann || eq(paid_up,"C"))
    return 0.0;

```

```

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

```

```

return      (riskadj_re_rel_q4_pv(t+1) + riskadj_re_rel_q4(t+1))
             * v_month_t[proj_yr];

```

#### 6.1.1.3.1.35 riskadj\_re\_rel\_total\_pv

```

if (t < commence_period_w || t >= maturity_period_ann)
    return 0.0;

```

```

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

```

```

return      (riskadj_re_rel_total_pv(t+1) + riskadj_re_rel_total(t+1))
             * v_month_t[proj_yr];

```

#### 6.1.1.3.1.36 riskadj\_re\_rel\_yr2plus\_pv

```

if (t < commence_period_w || t >= maturity_period_ann || eq(paid_up,"C"))
    return 0.0;

```

```

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

```

```

return      (riskadj_re_rel_yr2plus_pv(t+1) + riskadj_re_rel_yr2plus(t+1))
             * v_month_t[proj_yr];

```

#### 6.1.1.3.1.37 surv\_per\_ret

```

if (t < commence_period_w || t > maturity_period_w)
    return NO_AVG;

```

```

if(submodel == "ANN" || submodel == "TERM")
    return 1;

```

```

if(mult_age_ind == 1.){
if(retirement_age_lookup(1) > sm_annuity[sm_annuity.size()-1]->takeup_age)
    return 1.;

```

```

return 1. - retirement_prop(t);
}

```

```

return 1. - retirement_prop(t);

```

**6.1.1.3.1.38      surv\_ret**

```
if (t < commence_period_w || t > maturity_period_w)
    return NO_AVG;
```

```
if (t <= 0 || t > maturity_period_ann)
    return surv_per_ret(t);
```

```
return surv_ret(t-1) * surv_per_ret(t);
```

**6.1.1.3.1.39      units\_at\_mat**

```
// Unit value at the end of the period in which the policy matures.
```

```
double surr = max(0, sm_accum->claims_surrender(t)) +
               max(0, sm_acc_pup->claims_surrender(t))+
               max(0, sm_saving->claims_surrender(t)) +
               max(0, sm_saving_pup->claims_surrender(t));
```

```
double units_at_maturity = max(0.0, life->units_e_bef(t)
                               - life->death_claim_units(t) - surr - life->surr_charge(t));
```

```
// if sv is higher than units (because of persistency bonus) then pay sv at maturity
units_at_maturity = max(units_at_maturity, life->surr_value(t) - surr
                        - life->surr_charge(t));
```

```
// deduct surrender penalty from PUP units (the penalty is a result of the change of premium-paying
policies to PUP policies, and is charged when the PUP units are withdrawn, including on maturity)
```

```
if (eq(life->ben_class,"profil"))
    units_at_maturity = units_at_maturity - acc_pup->surr_penalty_e_bef(t);
```

```
return units_at_maturity;
```

**6.1.1.3.1.40      reserve**

```
if (t <= commence_period_w || t > maturity_period_ann)
    return 0.0;
```

```
if(submodel == "TRAD") {
    if(res_prop_kitzba > 0 && mult_age_ind == 1)
        return trad->reserve(t) + reserve_annuity(t);
```

```
    if(t < maturity_period_w)
        return trad->reserve(t);
```

```
    return reserve_basic(t);
}
```

```
if(submodel == "TERM")
    return term->reserve(t);
```

```
if(submodel == "ANN")
    return reserve_basic(t);
```

```
double multage = sm_accum->units_e(t) * bonus[prem_term]/100.;
```

```
if(mult_age_ind == 1)
    multage = sm_annuity->reserve_bonus_units_e_t(t);
```

```
return reserve_basic(t)
```

```

+ reserve_extra(t)
+ res_ann_deficiency(t)
+ multage; // reserve for bonus held from start

```

#### 6.1.1.3.1.41 reserve\_bef\_ret

```

if (t <= commence_period_w || t > maturity_period_ann)
    return 0.0;

if(submodel == "TRAD")
    return trad->reserve(t);

if(submodel == "ANN")
    return reserve_basic_bef_ret(t);

double multage = 0;
if(prem_term >= 0 && prem_term < 1200)
    multage = sm_accum->units_e(t) * bonus[prem_term]/100.;
if(mult_age_ind == 1)
    multage = sm_annuity->reserve_bonus_units_e_t(t);

return reserve_basic_bef_ret(t)
    + reserve_extra(t)
    + res_ann_deficiency(t)
    + multage; // reserve for bonus held from start

```

#### 6.1.1.3.1.42 reserve\_extra

```

if (submodel=="TERM")
    return NO_AVG;

if (submodel=="TRAD")
    return trad->reserve_extra(t);

if (t < commence_period_w || t >= maturity_period_w)
    return 0.0;

// risk reserve
if (eq(ben_class,"adif"))
    return res_perc_prem[1]/100. * (premium_if_b(t) - prem_freq * alloc_units(t));

return 0.0; // *** no URL in model yet

```

#### 6.1.1.3.1.43 ann\_cost\_pv

```

if (t < commence_period_w || t >= maturity_period_w)
    return 0.0;

if(!inlist(submodel, "UNIT"))
    return NO_AVG;

if (t == maturity_period_w-1)
    return res_ann_deficiency(t);

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

return ann_cost_pv(t+1)* v_month_t[proj_yr];

```



**6.1.1.3.1.44 net\_prem\_def**

```

if (submodel=="TERM")
    return term->net_prem_deficiency_b(t);

return 0.0;

```

**6.1.1.3.1.45 net\_premium\_e**

```

if (submodel=="TERM")
    return term->net_premium_e(t);

if (submodel=="TRAD")
    return trad->net_premium_e(t);

return 0.0;

```

**6.1.1.3.1.46 res\_np\_deficiency**

```

double res_temp =0.0; //reduced basic reserve;
//Use zeroised negative basic reserve to reduce NP deficiency reserve

if (submodel=="TERM") {
    if (zeroise_res=="Y")
        res_temp = min(0.0,term->reserve_basic(t) );

    return max(0.0,res_temp+term->res_np_deficiency(t));
}

if (submodel=="TRAD")
    return 0.0;

return 0.0;

```

**6.1.1.3.1.47 reserve\_annuity**

```

if(res_prop_kitzba > 0.0){
    if(mult_age_ind == 1)
        return sm_annuity->reserve_basic(t);
    return sm_annuity[ann_index_map[takeup_age]]->reserve_basic(t);
}
return NO_AVG;

```

**6.1.1.3.1.48 reserve\_basic**

```

if (t <= commence_period_w)
    return 0.0;

if (submodel=="TRAD")
    if(mult_age_ind ==1)
        return trad->reserve_basic(t) + reserve_annuity(t);

if(t < mat_period_min){
    if (submodel=="TERM") {
        if (zeroise_res=="Y")
            return max(0.0,term->reserve_basic(t) + reserve_claims(t));

        return term->reserve_basic(t)+ reserve_claims(t);
    }
}

```

```

        if (submodel=="TRAD")
            return trad->reserve_basic(t);

return units_e(t);
}

if(mult_age_ind == 1)
    return units_e(t) + reserve_annuity(t);

return reserve_annuity(t);

```

#### **6.1.1.3.1.49 reserve\_basic\_bef\_ret**

```

if (t <= commence_period_w)
    return 0.0;

if (submodel=="TRAD")
    if(mult_age_ind ==1)
        return trad->reserve_basic(t);

if(t < mat_period_min)
    return units_e(t);

if(mult_age_ind == 1)
    return units_e(t);

return 0;

```

#### **6.1.1.3.1.50 reserve\_basic\_gt\_su**

```

if(res_prop_kitzba > 0.0){
    if(mult_age_ind == 1)
        return sm_annuity->res_basic_gt_su(t);
    return sm_annuity[ann_index_map[takeup_age]]->res_basic_gt_su(t);
}
return NO_AVG;

```

#### **6.1.1.3.1.51 reserve\_claims**

```

if (t <= commence_period_w || !eq(submodel , "TERM") )
    return 0.0;

if(t < maturity_period_w) {
    return term->reserve_basic_claims(t);
} //end if TERM

return NO_AVG;

```

#### **6.1.1.3.1.52 reserve\_claims\_retent**

```

if (t <= commence_period_w || !eq(submodel , "TERM") )
    return 0.0;

if(t < maturity_period_w) {
    return term->reserve_basic_claims(t) * (1 - re_ratio_w);
} //end if TERM

```

```
return NO_AVG;
```

#### 6.1.1.3.1.53 reserve\_risk\_premium

```
if (submodel=="TERM")
    return term->reserve_risk_premium(t);
```

```
if (submodel=="TRAD")
    return trad->reserve_risk_premium(t);
```

```
return 0.0;
```

#### 6.1.1.3.1.54 ber\_retire\_rm

```
if (t < commence_period_w || t > maturity_period_w)
    return 0.0;
```

```
double temp = 0.0;
```

```
if (v_rm_cumm(t) > 0.0)
    temp = v_rm_cumm(t+12) / v_rm_cumm(t); //Discounting over year
```

```
return (ber_retire_rm(t+12) + be_retire(t+12) )
    * temp;
```

#### 6.1.1.3.1.55 capital\_at\_risk

```
if (t < life->commence_period_w || t >= life->maturity_period_w)
    return 0.0;
```

```
double retention_rate = 1.0;
```

```
if (claims_total(t) > 0)
    retention_rate = 1. - claims_re(t) / claims_total(t);
```

```
return max(sum_insured_if_e(t) * retention_rate + cashflow_pv(t), 0);
```

#### 6.1.1.3.1.56 capital\_at\_risk\_rm

```
if (t < life->commence_period_w || t >= life->maturity_period_w)
    return 0.0;
```

```
double temp = 0.0;
```

```
if (v_rm_cumm(t) > 0.0)
    temp = v_rm_cumm(t+12) / v_rm_cumm(t); //Discounting over year
```

```
return (capital_at_risk_rm(t+12) + capital_at_risk(t+12))
    * temp;
```

#### 6.1.1.3.1.57 claim\_cost\_pv\_rm

```
if (t < commence_period_w || t > maturity_period_w)
    return 0.0;
```

```
double temp = 0.0;
```

```
if (v_rm_cumm(t) > 0.0)
    temp = v_rm_cumm(t+12) / v_rm_cumm(t); //Discounting over year
```

```

return      (claim_cost_pv_rm(t+12) + claim_cost_pv(t+12))
            * temp;

```

#### 6.1.1.3.1.58 claim\_cost\_re\_pv\_rm

```

if (t < commence_period_w || t > maturity_period_w || reinsurance=="N" || eq(re_type,"NONE"))
    return 0.0;

```

```

double temp = 0.0;

```

```

if (v_rm_cumm(t) > 0.0)
    temp = v_rm_cumm(t+12) / v_rm_cumm(t); //Discounting over year

```

```

return      (claim_cost_re_pv_rm(t+12) + claim_cost_re_pv(t+12))
            * temp;

```

#### 6.1.1.3.1.59 claims\_annuity\_pv\_rm

```

if (t < commence_period_w || t >= t_high-12)
    return 0.0;

```

```

double temp = 0.0;

```

```

if (v_rm_cumm(t) > 0.0)
    temp = v_rm_cumm(t+12) / v_rm_cumm(t); //Discounting over year

```

```

return      (claims_annuity_pv_rm(t+12) + claims_annuity_pv(t+12))
            * temp;

```

#### 6.1.1.3.1.60 claims\_death\_pv\_rm

```

if (t < commence_period_w || t > maturity_period_w)
    return 0.0;

```

```

double temp = 0.0;

```

```

if (v_rm_cumm(t) > 0.0)
    temp = v_rm_cumm(t+12) / v_rm_cumm(t); //Discounting over year

```

```

return      (claims_death_pv_rm(t+12) + claims_death_pv(t+12))
            * temp;

```

#### 6.1.1.3.1.61 claims\_disability\_pv\_rm

```

if (t < commence_period_w || t > maturity_period_w)
    return 0.0;

```

```

double temp = 0.0;

```

```

if (v_rm_cumm(t) > 0.0)
    temp = v_rm_cumm(t+12) / v_rm_cumm(t); //Discounting over year

```

```

return      (claims_disability_pv_rm(t+12) + claims_disability_pv(t+12))
            * temp;

```

#### 6.1.1.3.1.62 expense\_pv\_rm

```

if (t < commence_period_w || t > maturity_period_ann)

```

```

        return 0.0;

double temp = 0.0;

if (v_rm_cumm(t) > 0.0)
    temp = v_rm_cumm(t+12) / v_rm_cumm(t); //Discounting over year

return    (expense_pv_rm(t+12) + expense_pv(t+12))
          * temp;

```

#### 6.1.1.3.1.63 inv\_income\_chetz\_pv\_rm

```

if (t < commence_period_w || t > maturity_period_ann)
    return 0.0;

double temp = 0.0;

if (v_rm_cumm(t) > 0.0)
    temp = v_rm_cumm(t+12) / v_rm_cumm(t); //Discounting over year

return    (inv_income_chetz_pv_rm(t+12) + investment_income_chetz_pv(t+12))
          * temp;

```

#### 6.1.1.3.1.64 profit\_book\_vif\_pv\_pos\_rm

```

if (t < commence_period_w || t > maturity_period_w)
    return 0.0;

double temp = 0.0;

if (v_rm_cumm(t) > 0.0)
    temp = v_rm_cumm(t+12) / v_rm_cumm(t); //Discounting over year

return    (profit_book_vif_pv_pos_rm(t+12) + profit_book_vif_pv_pos(t+12))
          * temp;

```

#### 6.1.1.3.1.65 rein\_claims\_pv\_rm

```

if (t < commence_period_w || t > maturity_period_w)
    return 0.0;

double temp = 0.0;

if (v_rm_cumm(t) > 0.0)
    temp = v_rm_cumm(t+12) / v_rm_cumm(t); //Discounting over year

return    (rein_claims_pv_rm(t+12) + rein_claims_pv(t+12))
          * temp;

```

#### 6.1.1.3.1.66 v\_rm\_cumm

```

if (t < commence_period_w || t > maturity_period_ann)
    return 0.0;

if (t <= 0)
    return 1.;

int proj_yr = xint(proj_year(t));
if(eq(projection_type_int, "Rollup"))

```

```

    proj_yr = xint(proj_year_rollup(t));

return v_rm_cumm(t-1)
    * v_month_t_rm[proj_yr];

```

#### 6.1.1.3.1.67 bonus\_shimur

```

if (inlist(submodel,"TERM,ANN,TRAD"))
    return NO_AVG;

```

```

return units_bon(t); //Adif only

```

#### 6.1.1.3.1.68 cashflow\_b\_post\_ret

```

if (t < commence_period_w || t > maturity_period_ann)
    return 0.0;

```

```

if (submodel == "TERM")
    return 0.0;

```

```

return sm_annuity->cashflow_b_post_ret(t);

```

#### 6.1.1.3.1.69 cashflow\_pv\_active

```

if (t < commence_period_w || t > maturity_period_ann)
    return 0.0;

```

```

//if (mult_age_ind !=1)
//    return 0.0;

```

```

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

```

```

return (cashflow_pv_active(t+1) + cashflow_e(t+1))* v_month_t[proj_yr]
    + cashflow_b_bef_ret(t+1);

```

#### 6.1.1.3.1.70 cashflow\_pv\_active\_chetz

```

if (t < commence_period_w || t > maturity_period_ann || free_inv_prop_t[1] >= 1.)
    return 0.0;

```

```

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

```

```

return cashflow_pv_res_active(t)*max_chetz + cashflow_pv_ifrs_active(t)*(1-max_chetz);

```

#### 6.1.1.3.1.71 cashflow\_pv\_active\_e

```

if (t < commence_period_w || t > maturity_period_ann)
    return 0.0;

```

```

//if (mult_age_ind !=1)
//    return 0.0;

```

```

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

```

```
return (cashflow_pv_active_e(t+1) + cashflow_e(t+1) + cashflow_b_bef_ret(t+1))* v_month_t[proj_yr];
```

#### 6.1.1.3.1.72 cashflow\_pv\_deferred

```
if (t < commence_period_w || t > maturity_period_ann)
    return 0.0;

if (mult_age_ind !=1)
    return 0.0;

double new_ret = 0.0;

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

if(ann_index_map.count(retirement_age_lookup(t)) != 0){

    if(t == sm_annuity[ann_index_map[retirement_age_lookup(t)]]->maturity_period_w)
        new_ret = sm_annuity[ann_index_map[retirement_age_lookup(t)]]->cashflow_pv(t);

}

return cashflow_pv_deferred(t+1) * v_month_t[proj_yr] + new_ret;
```

#### 6.1.1.3.1.73 cashflow\_pv\_deferred\_chetz

```
if (t < commence_period_w || t > maturity_period_ann)
    return 0.0;

if (mult_age_ind !=1)
    return 0.0;

return cashflow_pv_deferred_chetz_ifrs(t) + cashflow_pv_deferred_chetz_res(t);
```

#### 6.1.1.3.1.74 cashflow\_pv\_deferred\_chetz\_ifrs

```
if (t < commence_period_w || t > maturity_period_ann)
    return 0.0;

if (mult_age_ind !=1)
    return 0.0;

double new_ret = 0.0;

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

if(ann_index_map.count(retirement_age_lookup(t)) != 0){

    if(t == sm_annuity[ann_index_map[retirement_age_lookup(t)]]->maturity_period_w)
        new_ret = sm_annuity[ann_index_map[retirement_age_lookup(t)]]->
>cashflow_pv_ifrs(t)*(1-max_chetz);

}

return cashflow_pv_deferred_chetz_ifrs(t+1) * v_month_t_ifrs[proj_yr] + new_ret;
```

**6.1.1.3.1.75 cashflow\_pv\_deferred\_chetz\_res**

```

if (t < commence_period_w || t > maturity_period_ann)
    return 0.0;

if (mult_age_ind !=1)
    return 0.0;

double new_ret = 0.0;

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

if(ann_index_map.count(retirement_age_lookup(t)) != 0){

    if(t == sm_annuity[ann_index_map[retirement_age_lookup(t)]]->maturity_period_w)
        new_ret = sm_annuity[ann_index_map[retirement_age_lookup(t)]]->
>cashflow_pv_res(t)*max_chetz;

}

return cashflow_pv_deferred_chetz_res(t+1) * v_month_t_int_res + new_ret;

```

**6.1.1.3.1.76 cashflow\_pv\_deferred\_e**

```

if (t < commence_period_w || t > maturity_period_ann)
    return 0.0;

if (mult_age_ind !=1)
    return 0.0;

double new_ret = 0.0;

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

if(ann_index_map.count(retirement_age_lookup(t)) != 0){

    if(t == sm_annuity[ann_index_map[retirement_age_lookup(t)]]->maturity_period_w)
        new_ret = sm_annuity[ann_index_map[retirement_age_lookup(t)]]->cashflow_pv_e(t);

}

return (cashflow_pv_deferred_e(t+1)) * v_month_t[proj_yr] + new_ret;

```

**6.1.1.3.1.77 cashflow\_pv\_ifrs**

```

if (t < commence_period_w || t > maturity_period_ann || free_inv_prop_t[1] >= 1.)
    return 0.0;

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

if ( t > maturity_period_w)

```



```

return (cashflow_pv_ifrs(t+1) + cashflow_e(t+1))* ann_v_month_t_ifrs[proj_yr]
      + cashflow_b(t+1);

```

```

else

```

```

return (cashflow_pv_ifrs(t+1) + cashflow_e(t+1))* v_month_t_ifrs[proj_yr]
      + cashflow_b(t+1);

```

#### 6.1.1.3.1.78 cashflow\_pv\_ifrs\_active

```

if (t < commence_period_w || t > maturity_period_ann || free_inv_prop_t[1] >= 1.)
  return 0.0;

```

```

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
  proj_yr = xint(proj_year_rollup(t+1));

```

```

return (cashflow_pv_ifrs_active(t+1) + cashflow_e(t+1))* v_month_t_ifrs[proj_yr]
      + cashflow_b_bef_ret(t+1);

```

#### 6.1.1.3.1.79 cashflow\_pv\_res

```

if (t < commence_period_w || t > maturity_period_ann || free_inv_prop_t[1] >= 1.)
  return 0.0;

```

```

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
  proj_yr = xint(proj_year_rollup(t+1));

```

```

return (cashflow_pv_res(t+1) + cashflow_e(t+1))* v_month_t_int_res
      + cashflow_b(t+1);

```

#### 6.1.1.3.1.80 cashflow\_pv\_res\_active

```

if (t < commence_period_w || t > maturity_period_ann || free_inv_prop_t[1] >= 1.)
  return 0.0;

```

```

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
  proj_yr = xint(proj_year_rollup(t+1));

```

```

return (cashflow_pv_res_active(t+1) + cashflow_e(t+1))* v_month_t_int_res
      + cashflow_b_bef_ret(t+1);

```

#### 6.1.1.3.1.81 cashflow\_re\_pv

```

if (t < commence_period_w || t >= maturity_period_ann)
  return 0.0;

```

```

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

if (t > maturity_period_w)
    return (cashflow_re_pv(t+1) + cashflow_re_e(t+1)) * ann_v_month_t[proj_yr] +
    cashflow_re_b(t+1);
else
    return (cashflow_re_pv(t+1) + cashflow_re_e(t+1)) * v_month_t[proj_yr] + cashflow_re_b(t+1);

```

#### 6.1.1.3.1.82 claims\_annuity\_gt

```

if(t <= mat_period_min || t > maturity_period_ann || eq(submodel, "TERM"))
    return NO_AVG;

```

```

return max(claims_annuity(t) - claims_annuity_nogt(t), 0);

```

#### 6.1.1.3.1.83 claims\_insurance

```

if(submodel == "TRAD")
    return trad->claims_death(t);

if(submodel == "TERM")
    return term->claims_total(t);

return death_claim_si(t); //For adif/profil

```

#### 6.1.1.3.1.84 comm\_hekef\_net

```

return comm_hekef(t) - comm_clawback(t);

```

#### 6.1.1.3.1.85 comm\_profit

```

if (t <= commence_period_w || t > maturity_period_w)
    return 0.0;

if (comm_prof==0)
    return 0.0;

if (t + elapsed_months <= (comm_renewal_year-1)*12)
    return 0.0;

return
    premium_gross(t)
    * comm_prof/ 100.
    * (1+vat/100.);

```

#### 6.1.1.3.1.86 comm\_reg

```

return comm_nihul(t) +
    comm_regular(t) +
    comm_renewal(t) +
    comm_reg_riders_out(t);

```

#### 6.1.1.3.1.87 coverage\_units

```

double DF = 1. ;

if (cu_discounted == "Y")
    DF = discount_factor_acc(t);

```

```

if(eq(submodel, "TERM")){
return profit_weighting * service_units(t) * DF;
}

return (profit_weighting * service_units(t) + claims_annuity_nogt(t)) * DF;

```

#### 6.1.1.3.1.88 coverage\_units\_re

```

if (cu_discounted == "Y")
    return profit_weighting_re * service_units(t) * discount_factor_acc(t);

return profit_weighting_re * service_units(t);

```

#### 6.1.1.3.1.89 expense\_clm

```

return expense_ren_perc_ann(t) + expense_claims(t);

```

#### 6.1.1.3.1.90 expense\_init

```

return expense_initial_fix(t) + expense_initial_perc(t);

```

#### 6.1.1.3.1.91 expense\_pv\_active

```

if (t < commence_period_w || t >= maturity_period_ann)
    return 0.0;

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

return expense_pv_active(t+1) * v_month_t[proj_yr]
    + expense_total_pre_ret(t+1);

```

#### 6.1.1.3.1.92 expense\_pv\_active\_no\_inv

```

if (t < commence_period_w || t >= maturity_period_ann)
    return 0.0;

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

return expense_pv_active_no_inv(t+1) * v_month_t[proj_yr]
    + expense_total_pre_ret_no_inv(t+1);

```

#### 6.1.1.3.1.93 expense\_pv\_deferred

```

if (t < commence_period_w || t >= maturity_period_ann || mult_age_ind !=1)
    return 0.0;

double new_ret = 0.0;

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

if(ann_index_map.count(retirement_age_lookup(t)) != 0){

    if(t == sm_annuity[ann_index_map[retirement_age_lookup(t)]]->maturity_period_w)

```

```

        new_ret = sm_annuity[ann_index_map[retirement_age_lookup(t)]]-
>expense_ren_perc_post_ret_pv(t);
    }

```

```

return expense_pv_deferred(t+1) * v_month_t[proj_yr] + new_ret;

```

#### 6.1.1.3.1.94 expense\_ren

```

return expense_ren_perc(t) - expense_ren_perc_ann(t)
    + expense_ren_charge(t)
    + expense_ren_fix(t)
    + comm_supervisor(t)
    - comm_claw_spv(t);

```

#### 6.1.1.3.1.95 expense\_var\_pv\_active

```

if (t < commence_period_w || t >= maturity_period_ann || mult_age_ind !=1)
    return 0.0;

```

```

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

```

```

return expense_var_pv_active(t+1) * v_month_t[proj_yr]
    + expense_initial_perc(t+1)
    + expense_ren_perc_bef_ret(t+1)
    + expense_ren_charge(t+1);

```

#### 6.1.1.3.1.96 fvui

```

if (t < commence_period_w || t > maturity_period_ann)
    return 0.0;

```

```

if (eq(savings_pol_prod_code, "Y"))

```

```

return claims_annuity_pv(t) + claims_maturity_pv(t) + claims_surrender_pv(t) - prem_savings_pv(t) +
management_fee_pv(t);

```

```

return 0.0;

```

#### 6.1.1.3.1.97 int\_cred

```

if(submodel == "ANN" || submodel=="TERM")
    return NO_AVG;

```

```

if(submodel == "UNIT")
    return interest_units_e(t);

```

```

//Klasi

```

```

if(!eq(par_nonpar, "N")) //Participating
    return trad->int_cred(t) + trad->int_cred_pup(t) + trad->int_cred_mat(t); //Already
calculated for bonus

```

```

//Guaranteed rate

```

```

if (t < mat_period_original) //When there is reserve
    return

```

```

        trad->int_rate_res_mthly //Guaranteed rate
        * (trad->reserve_basic_prem_if(t-1)
            * surv_per_act_bal_bef_ret(t) //Surviving active reserve

```

```

        +
        trad->reserve_basic_pup(t)
        * (1.- trad->death_rate(t))
        * (1. - lapse_rate_pup_bal(t))//Surviving pup reserve
    );

//Guaranteed after maturity period
return trad->int_post_mat(t)
    *
    (trad->surr_value(t-1)
    * surv_act_post_ret(t) //Surviving active surrender value
    + trad->surr_value_pup(t)
    * surv_pup_post_ret(t)
    );

```

#### 6.1.1.3.1.98 investment\_income\_chetz\_pv\_deferred

```

if (t < commence_period_w || t >= maturity_period_ann || mult_age_ind !=1 || free_inv_prop_t[1] >=
1.)
    return 0.0;

double new_ret = 0.0;

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

if(ann_index_map.count(retirement_age_lookup(t)) != 0){
    if(t == sm_annuity[ann_index_map[retirement_age_lookup(t)]]->maturity_period_w)
        new_ret = sm_annuity[ann_index_map[retirement_age_lookup(t)]]-
>investment_income_chetz_pv(t);
}

return investment_income_chetz_pv_deferred(t+1) * v_month_t[proj_yr] + new_ret;

```

#### 6.1.1.3.1.99 investment\_income\_chetz\_pv\_inpay

```

if (t < commence_period_w || t > maturity_period_ann || mult_age_ind != 1 || free_inv_prop_t[1] >=
1.)
    return 0.0;

double ann_in_pay = 0;

for (int i=0; i < sm_annuity.size(); i++){
    if(t > sm_annuity[i]->maturity_period_w)
        ann_in_pay = ann_in_pay + sm_annuity[i]->investment_income_chetz_pv(t);
}

return ann_in_pay;

```

#### 6.1.1.3.1.100 investment\_income\_pv\_active

```

if (t < commence_period_w || t >= maturity_period_ann)
    return 0.0;

```

```
int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));
```

```
return (investment_income_pv_active(t+1)
        + investment_income_bef_ret(t+1) )
        * v_month_t[proj_yr];
```

#### 6.1.1.3.1.101 investment\_income\_pv\_deferred

```
if (t < commence_period_w || t >= maturity_period_ann || mult_age_ind !=1)
    return 0.0;
```

```
double new_ret = 0.0;
```

```
int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));
```

```
if(ann_index_map.count(retirement_age_lookup(t)) != 0){
```

```
    if(t == sm_annuity[ann_index_map[retirement_age_lookup(t)]]->maturity_period_w)
        new_ret = sm_annuity[ann_index_map[retirement_age_lookup(t)]]-
>investment_income_pv(t);
```

```
}
```

```
return investment_income_pv_deferred(t+1) * v_month_t[proj_yr] + new_ret;
```

#### 6.1.1.3.1.102 mgt\_fees\_prem

```
if (!eq(ben_class, "profil"))
    return 0.0;
```

```
return max(premium_gross(t) - alloc_units(t), 0);
```

#### 6.1.1.3.1.103 outgo\_pv\_active

```
if (t < commence_period_w || t >= maturity_period_ann || mult_age_ind !=1)
    return 0.0;
```

```
int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));
```

```
return (outgo_pv_active(t+1)
        + outgo_e(t+1) )
        * v_month_t[proj_yr]
        + outgo_b_before_ret(t+1);
```

#### 6.1.1.3.1.104 outgo\_pv\_deferred

```
if (t < commence_period_w || t >= maturity_period_ann || mult_age_ind !=1)
    return 0.0;
```

```
double new_ret = 0.0;
```

```
int proj_yr = xint(proj_year(t+1));
```

```

if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

if(ann_index_map.count(retirement_age_lookup(t)) != 0){
    if(t == sm_annuity[ann_index_map[retirement_age_lookup(t)]]->maturity_period_w)
        new_ret = sm_annuity[ann_index_map[retirement_age_lookup(t)]]-
>outgo_b_post_ret_pv(t);
}

return outgo_pv_deferred(t+1) * v_month_t[proj_yr] + new_ret;

```

#### 6.1.1.3.1.105 prem\_insurance

```

if(submodel == "ANN")
    return NO_AVG;

if (submodel=="TERM")
    return premium_gross(t);

if (submodel=="TRAD")
    return 0.0;

if (eq(ben_class, "profil"))
    return cover_charge(t);

if (eq(ben_class, "adif"))
    return premium_gross(t) - alloc_units(t);

return 0.0; //Should not get here

```

#### 6.1.1.3.1.106 prem\_savings

```

if(submodel == "ANN" || submodel=="TERM")
    return NO_AVG;

if (submodel=="TRAD")
    return premium_gross(t);

if (eq(ben_class, "profil"))
    return premium_gross(t) - cover_charge(t);

if (eq(ben_class, "adif"))
    return alloc_units(t);

return 0.0; //Should not get here

```

#### 6.1.1.3.1.107 profit\_book\_pv\_active

```

if (t < commence_period_w || t >= maturity_period_ann || mult_age_ind !=1)
    return 0.0;

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

return (profit_book_pv_active(t+1)

```

```

+ profit_book_bef_ret(t+1) )
* v_month_t[proj_yr];

```

#### 6.1.1.3.1.108 profit\_book\_pv\_deferred

```

if (t < commence_period_w || t >= maturity_period_ann || mult_age_ind !=1)
    return 0.0;

double new_ret = 0.0;

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

if(ann_index_map.count(retirement_age_lookup(t)) != 0){

    if(t == sm_annuity[ann_index_map[retirement_age_lookup(t)]]->maturity_period_w)
        new_ret = sm_annuity[ann_index_map[retirement_age_lookup(t)]]->profit_book_pv(t);
}

return profit_book_pv_deferred(t+1) * v_month_t[proj_yr] + new_ret;

```

#### 6.1.1.3.1.109 profit\_book\_vif\_gross\_pv\_active

```

if (t < commence_period_w || t >= maturity_period_ann || mult_age_ind !=1)
    return 0.0;

double vif_gross_pre_ret = cashflow_e(t+1)

+ cashflow_b_bef_ret(t+1)
+ investment_income_bef_ret(t+1)
- reserve_increase_bef_ret(t+1)
+ cashflow_re_b(t+1)
+ cashflow_re_e(t+1);

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

return (profit_book_vif_gross_pv_active(t+1)
+ vif_gross_pre_ret)
* v_month_t[proj_yr];

```

#### 6.1.1.3.1.110 profit\_book\_vif\_pv\_active

```

if (t < commence_period_w || t >= maturity_period_ann || mult_age_ind !=1)
    return 0.0;

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

return (profit_book_vif_pv_active(t+1)
+ profit_book_vif_bef_ret(t+1) )
* v_month_t[proj_yr];

```

#### 6.1.1.3.1.111 profit\_book\_vif\_pv\_deferred

```

if (t < commence_period_w || t >= maturity_period_ann || mult_age_ind !=1)
    return 0.0;

```



```

double new_ret = 0.0;

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

if(ann_index_map.count(retirement_age_lookup(t)) != 0){
    if(t == sm_annuity[ann_index_map[retirement_age_lookup(t)]]->maturity_period_w)
        new_ret = sm_annuity[ann_index_map[retirement_age_lookup(t)]]-
>profit_book_vif_post_ret_pv(t);
}

return profit_book_vif_pv_deferred(t+1) * v_month_t[proj_yr] + new_ret;

```

#### 6.1.1.3.1.112 profit\_gross\_vif\_pv\_active

```

if (t < commence_period_w || t >= maturity_period_ann || mult_age_ind !=1)
    return 0.0;

double vif_gross_pre_ret = (cashflow_e(t+1)
                            + cashflow_b_bef_ret(t+1)
                            + investment_income_bef_ret(t+1)
                            - reserve_increase_bef_ret(t+1)
                            + cashflow_re_b(t+1)
                            + cashflow_re_e(t+1) )
                            * (1 - tax_rate/ 100.);

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

return (profit_gross_vif_pv_active(t+1)
        + vif_gross_pre_ret)
        * v_month_t[proj_yr];

```

#### 6.1.1.3.1.113 profit\_net\_vif\_pv\_active

```

if (t < commence_period_w || t >= maturity_period_ann || mult_age_ind !=1)
    return 0.0;

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

return (profit_net_vif_pv_active(t+1)
        + profit_book_vif_bef_ret(t+1)* (1- tax_rate/ 100.) )
        * v_month_t[proj_yr];

```

#### 6.1.1.3.1.114 profit\_net\_vif\_pv\_deferred

```

if (t < commence_period_w || t >= maturity_period_ann || mult_age_ind !=1)
    return 0.0;

double new_ret = 0.0;

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))

```

```

    proj_yr = xint(proj_year_rollup(t+1));

    if(ann_index_map.count(retirement_age_lookup(t)) != 0){
        if(t == sm_annuity[ann_index_map[retirement_age_lookup(t)]]->maturity_period_w)
            new_ret = sm_annuity[ann_index_map[retirement_age_lookup(t)]]->profit_net_vif_post_ret_pv(t);
    }

    return profit_net_vif_pv_deferred(t+1) * v_month_t[proj_yr] + new_ret;

```

#### 6.1.1.3.1.115 reserve\_increase\_pv\_active

```

if (t < commence_period_w || t >= maturity_period_ann)
    return 0.0;

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

return (reserve_increase_pv_active(t+1)
        + reserve_increase_bef_ret(t+1) )
        * v_month_t[proj_yr];

```

#### 6.1.1.3.1.116 reserve\_increase\_pv\_deferred

```

if (t < commence_period_w || t >= maturity_period_ann || mult_age_ind !=1)
    return 0.0;

double new_ret = 0.0;

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

if(ann_index_map.count(retirement_age_lookup(t)) != 0){
    if(t == sm_annuity[ann_index_map[retirement_age_lookup(t)]]->maturity_period_w)
        new_ret = sm_annuity[ann_index_map[retirement_age_lookup(t)]]->reserve_increase_pv(t);
}

return reserve_increase_pv_deferred(t+1) * v_month_t[proj_yr] + new_ret;

```

#### 6.1.1.3.1.117 reserve\_pv

```

if (t < commence_period_w || t >= maturity_period_ann)
    return 0.0;

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

return (reserve_pv(t+1) + reserve(t+1))
        * v_month_t[proj_yr];

```

**6.1.1.3.1.118 rid\_cashflow\_pv**

```

if (!eq(ben_class, "profil") || riders_count_w <= 0)
    return 0.0;

if (t <= commence_period_w || t > maturity_period_w)
    return 0.0;

double cf_b = 0;
double cf_e = 0;
xstring key_temp;

//For profil, all reinsurance is from riders

cf_b = cover_charge(t+1)
      + comm_re(t+1)
      + comm_re_prof(t+1) //No reinsurance reserve for UNIT
      - premium_re(t+1);

double comm_hek = 0.0;

if ((elapsed_months + elapsed_months_extra) <= 12){
    if ( eq(paid_up, "Y"))
        comm_hek = cover_charge(1) * comm_hekef_pc_rider /100;
    else {
        double comm_temp_pc = comm_hekef_pc_rider/100.;
        double temp_surv = 1.;
        if (surv_act_prm(1)> 0.00001)
            temp_surv = surv_act_prm(1);
        comm_hek = (cover_charge(1)*surv_act_prm(1 - elapsed_months -
elapsed_months_extra)/temp_surv)*prem_freq * comm_temp_pc;
    }
}

double comm_claw = 0.0;

if (t+1+elapsed_months-1+elapsed_months_extra <=180){
    if(eq(paid_up, "Y"))
    {
        double temp = surv_pup_prm(t-1) * lapse_rate_pup_prm(t);

        if (temp != 0)
            comm_claw = comm_claw_prpn_hekef[t+elapsed_months-1+elapsed_months_extra]
                        / 100.
                        * comm_hek
                        * temp / surv_prm(commence_period_w);
    }
    else{
        double temp = surv_act_prm(t-1) * (lapse_rate_act_prm_dep(t) + pup_rate_prm_dep(t));

        if (temp != 0)

```

```

        comm_claw = comm_claw_prpn_hekef[t+elapsed_months-1+elapsed_months_extra] /
100.
        * comm_hek
        * temp / surv_prm(commence_period_w);
    }
}

cf_b = cf_b - comm_hek - comm_reg_riders_out(t+1) + comm_claw;

double exp_tot = expense_ren_charge(t+1);

if (t+1 + elapsed_months +elapsed_months_extra == 1) {

    double profil_risk_init_exps = 0.0;

    double margin = 0.;
    if(margin_add=="Y")
        margin = margin_exp_ini_pc;

    for (int i=0; i < riders_count_w; i++) {
        rider_tarif_row_key=xstring(tarif_rider[i]);
        prod_code_rider = rider_tarif_tbl;
        key_temp =prod_assumpt_rider_exp_tbl;
        exp_row_key=key_temp+"_"+company+"_"+pol_type_expenses;
        exp_col_key="I_PREM";

        profil_risk_init_exps = profil_risk_init_exps +
            (sm_riders[i]->prem_cover/12.) * 12 *
            exp_initial_extra_perc_charge[i] / 100. * surv_act_prm(t);
    }

    exp_tot = exp_tot + profil_risk_init_exps * (1. + margin/100.);
}

cf_b = cf_b - exp_tot;

cf_e = claims_re(t+1) - death_claim_si(t+1);

int proj_yr = xint(proj_year(t+2));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+2));

return (rid_cashflow_pv(t+1) + cf_e) * v_month_t[proj_yr] + cf_b;

```

#### 6.1.1.3.1.119 service\_units

```

if (t <= commence_period_w || t > life->maturity_period_ann )
    return 0.0;

if(eq(paid_up,"C"))
    return 0.0;

if(eq(ben_class, "phi"))
    return term->sum_insured_if_b(t) * serv_units_dur;

```

```

if(eq(submodel, "TERM")){ //Risk
    if(eq(ben_class, "dd") || eq(ben_class, "accdis") || eq(ben_class, "tpd") || eq(ben_class,
"l1tc"))
        return term->sum_insured_if_b_no_dec(t) + term->sum_insured_if_b_2_no_dec(t);
    else
        return term->sum_insured_if_b_no_dth(t);
}

if(mult_age_ind !=1){
    if( t >= maturity_period_w)
        return reserve_basic_gt_su(t); // currently not relevant as all policies that are not
multi-age, res_kiz equal zero therefore no claims annuity
    else
        return surr_value(t);

} //Not multi-age retirement

return surr_value(t) + reserve_basic_gt_su(t); //Only claims currently in payment

```

#### 6.1.1.3.1.120 service\_units\_pv

```

if (t < commence_period_w || t > maturity_period_ann)
    return 0.0;

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

return (service_units_pv(t+1) + service_units(t+1))
        * v_month_t[proj_yr];

```

#### 6.1.1.3.1.121 units\_for\_takeup

```

if(t > maturity_period_ann)
    return NO_AVG;

if(!inlist(submodel,"UNIT,TRAD") || res_prop_kitzba <= 0.0)
    return NO_AVG;

if(mult_age_ind == 1)
    return sm_annuity->units_for_takeup(t) ;

return sm_annuity[ann_index_map[takeup_age]]->units_for_takeup(t);

```

#### 6.1.1.3.1.122 income\_b

```

return premium_gross(t)
    + comm_re(t)
    + comm_re_prof(t);

```

#### 6.1.1.3.1.123 income\_e

```

if (eq(projection_type,"Rollup") && t==0)
    return claims_re(t) + premium_nb_sp;

```

```
return claims_re(t);
```

#### 6.1.1.3.1.124 income\_pv

```
if (t < commence_period_w || t >= maturity_period_ann)
    return 0.0;

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

if ( t > maturity_period_w)
    return (income_pv(t+1) + income_e(t+1)) * ann_v_month_t[proj_yr] + income_b(t+1);
else
    return (income_pv(t+1) + income_e(t+1)) * v_month_t[proj_yr] + income_b(t+1);
```

#### 6.1.1.3.1.125 charges\_premium

```
if (!eq(submodel,"UNIT"))
    return 0.0;

if (t <= commence_period_w || t > maturity_period_w)
    return 0.0;

return premium_gross(t) - alloc_units(t);
```

#### 6.1.1.3.1.126 charges\_premium\_pv

```
if (t < commence_period_w || t > maturity_period_w)
    return 0.0;

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

return charges_premium_pv(t+1)* v_month_t[proj_yr]
    + charges_premium(t+1);
```

#### 6.1.1.3.1.127 cover\_charge

```
if(submodel != "UNIT")
    return NO_AVG;

if (t <= 0 || t > maturity_period_w) // commence_period_w
    return 0.0;

if (!eq(ben_class,"profil"))
    return 0.;

if (t<0)
    return 0;

return rider_perc_allowed(t) / 100.
    * charge_amount_tt.sum_of_row(t) * surv_act_prm(t-1);
```

#### 6.1.1.3.1.128 cover\_charge\_pv

```
if (t < commence_period_w || t > maturity_period_w)
    return 0.0;
```

```

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

return    cover_charge_pv(t+1)* v_month_t[proj_yr]
    + cover_charge(t+1);

```

#### 6.1.1.3.1.129 management\_fee\_pv

```

if (t < commence_period_w || t > maturity_period_ann)
    return 0.0;

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

if(mult_age_ind != 1)
    return    (management_fee_pv(t+1) + management_fees(t+1)) * v_month_t[proj_yr];

//multi-age retirement
return  manage_fees_fixe_active_pv(t)
    + manage_fees_var_active_pv(t)
    + manage_fees_fixed_ann_pv_def(t)
    + manage_fees_fixed_ann_pv_ip(t)
    + manage_fees_var_ann_pv_def(t)
    + manage_fees_var_ann_pv_ip(t);

```

#### 6.1.1.3.1.130 management\_fees

```

return  management_fees_fixed_active(t)
    + management_fees_fixed_ann(t)
    + management_fees_var_active(t)
    + management_fees_var_ann(t);

```

#### 6.1.1.3.1.131 mgt\_var\_no\_bor

```

if (t <= 0 || t > maturity_period_w || mgt_fee_variable == 0 || submodel != "UNIT" || par_nonpar !=
"P")
    return 0.0;

```

```

if (net_interest_rate(t) > 0.0)
return  net_interest_rate(t)
    * (sm_accum->units_b(t) + sm_saving->units_b(t) )
    * mgt_fee_variable/100.;

return 0.0;

```

#### 6.1.1.3.1.132 mgt\_var\_no\_bor\_pup

```

if (t <= 0 || t > maturity_period_w || mgt_fee_variable == 0 || submodel != "UNIT" || par_nonpar !=
"P")
    return 0.0;

```

```

if (net_interest_rate(t) > 0.0)

```

```

return net_interest_rate(t)
    * (sm_acc_pup->units_b(t) + sm_saving_pup->units_b(t) )
    * mgt_fee_variable/100.;

return 0.0;

```

#### 6.1.1.3.1.133 surr\_charge

```

return sm_accum->surr_charge(t) +
    sm_acc_pup->surr_charge(t)+
    sm_saving->surr_charge(t) +
    sm_saving_pup->surr_charge(t);

```

#### 6.1.1.3.1.134 investment\_income

```

if (t <= commence_period_w || t > maturity_period_ann)
    return 0.0;

int proj_yr = xint(proj_year(t));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t));

double temp_inv_rate = 0.0;
double temp_ann_inv_rate = 0.0;
double invinc_chetz = 0.0;
double Cal_yr_mth=100*cal_year(t) +cal_month(t);

if (margin_add_asset == "Y" && t == 1 && submodel != "TERM" && par_nonpar == "P"){
    temp_inv_rate = asset_shock;
    temp_ann_inv_rate = asset_shock;
}
else{
    if (free_inv_prop_t[0] < 1.0 && chetz_be_ind == "Y" && Cal_yr_mth>=chetz_be_ind_yrs) {
        temp_inv_rate = inv_rate_rf_mth_t[proj_yr];
        temp_ann_inv_rate = ann_inv_rate_rf_mth_t[proj_yr];
        if(mult_age_ind == 1)
            invinc_chetz = investment_income_chetz_bef_ret(t);
        else
            invinc_chetz = investment_income_chetz(t);
    }
    else {
        temp_inv_rate = inv_rate_mth_t[proj_yr];
        temp_ann_inv_rate = ann_inv_rate_mth_t[proj_yr];
    }
}

if(mult_age_ind == 1)
    return sm_annuity->investment_income(t) + temp_inv_rate * (reserve_bef_ret(t-1) +
cashflow_b_bef_ret(t)) + invinc_chetz;

if(t > maturity_period_w)
    return temp_ann_inv_rate* (reserve(t-1) + cashflow_b(t)) + invinc_chetz;
else
    return temp_inv_rate
        * (reserve(t-1) + cashflow_b(t)) + invinc_chetz;

```



**6.1.1.3.1.135 investment\_income\_bef\_ret**

```

if (t <= commence_period_w || t > maturity_period_ann || eq(paid_up,"G"))
    return 0.0;

int proj_yr = xint(proj_year(t));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t));

double temp_inv_rate_m = 0.0;
double invinc_chetz = 0.0;
double Cal_yr_mth=100*cal_year(t) +cal_month(t);

if (margin_add_asset == "Y" && t == 1 && submodel != "TERM" && par_nonpar == "P")
    temp_inv_rate_m = asset_shock;
else
    if (free_inv_prop_t[0] < 1.0 && chetz_be_ind == "Y" && Cal_yr_mth>=chetz_be_ind_yrs) {
        temp_inv_rate_m = inv_rate_rf_mth_t[proj_yr];
        invinc_chetz = investment_income_chetz_bef_ret(t); }
    else temp_inv_rate_m = inv_rate_mth_t[proj_yr];

if(mult_age_ind == 1)
    return temp_inv_rate_m * (reserve_bef_ret(t-1) + cashflow_b_bef_ret(t)) + invinc_chetz;

return investment_income(t);

//return 0.0; //Unconditional return

```

**6.1.1.3.1.136 investment\_income\_chetz**

```

if (t <= commence_period_w || t > maturity_period_ann || free_inv_prop_t[1] >= 1.)
    return 0.0;

int proj_yr = xint(proj_year(t));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t));

double temp_inv_rate = 0.0;
double temp_ann_inv_rate = 0.0;

if (margin_add_asset == "Y" && t == 1 && submodel != "TERM" && par_nonpar == "P"){
    temp_inv_rate = asset_shock;
    temp_ann_inv_rate = asset_shock;
}
else{
    temp_inv_rate = inv_rate_mth_t[proj_yr];
    temp_ann_inv_rate = ann_inv_rate_mth_t[proj_yr];
}

if (proj_yr == 0 && eq(projection_type_int, "Rollup") && dump_vars == "Y"){
    double free = free_inv_prop_t[cal_year(t) - valn_year];
    log_strm<<"Inv rate at time "<<t<<": "<<temp_inv_rate<<endl;
}

```

```

log_strm<<"Rollup rate at time "<<t<<": "<<inv_rate_rollup<<endl;
log_strm<<"Free rate at time "<<t<<": "<<inv_rate_rf_mth_t[proj_yr]<<endl;
log_strm<<"Guaranteed rate at time "<<t<<": "<<invinc<<endl;
log_strm<<"Free inv rate at time "<<t<<": "<<free<<endl;
log_strm<<"Free inv rate by proj_year at time "<<t<<": "<<free_inv_prop_t[proj_yr]<<endl;

}

double temp_reserve_pre=0;
double temp_reserve_all=0;
double FORCE_CALC=0;
double Cal_yr_mth=100*cal_year(t) +cal_month(t);

if (chetz_be_ind == "Y" && Cal_yr_mth>=chetz_be_ind_yrs){
    temp_reserve_all=-cashflow_pv_chetz(t-1);
    temp_reserve_pre=-(cashflow_pv_active_chetz(t-1)+cashflow_pv_deferred_chetz(t-1) -
riskadj_gross(t-1));}
    else{
        temp_reserve_all=reserve(t-1) + cashflow_b(t);
        temp_reserve_pre=reserve_bef_ret(t-1) + cashflow_b_bef_ret(t);
        FORCE_CALC=-cashflow_pv_chetz(t-1);
    }

if(mult_age_ind == 1)
    return sm_annuity->investment_income_chetz(t) + (temp_inv_rate -inv_rate_rf_mth_t[proj_yr])
* (temp_reserve_pre);

if ( t > maturity_period_w)
    return (temp_ann_inv_rate -ann_inv_rate_rf_mth_t[proj_yr]) * (temp_reserve_all);
return (temp_inv_rate -inv_rate_rf_mth_t[proj_yr]) * (temp_reserve_all);

```

#### 6.1.1.3.1.137 units\_e\_piz\_int\_active

// מצטברת תשואה כולל - פיצויים צבירה על תקופתי תשואה

```

if (t <= commence_period_w || t >= maturity_period_ann)
    return 0.0;

if(submodel != "UNIT" || eq(ben_class, "profil") || par_nonpar == "N" || paid_up == "Y")
    return NO_AVG;

if (t == 0)
    return 0.0;

double piz = units_e_piz_int_active(t-1);

//Pup to deduct
double new_pup = 0;

if (sm_accum->units_e(t-1) + sm_saving->units_e(t-1) > 0)
    new_pup = (
        units_b_bef_pup_acc(t) * (1. - sm_accum-
>surr_chg_perc_units[t+elapsed_months]/100.)
        + units_b_bef_pup_sav(t) * (1. - sm_saving-
>surr_chg_perc_units[t+elapsed_months]/100.)
    )
    * piz

```

```

        / (sm_accum->units_e(t-1) + sm_saving->units_e(t-1));

piz = piz - new_pup + int_units_piz_active(t);

//Decrements
double decrements = 0.0;

if ((sm_accum->units_e_bef(t) + sm_saving->units_e_bef(t)) > 0)
    decrements = (sm_accum->death_claims_units(t) + sm_saving->death_claims_units(t)
        + sm_accum->claims_surrender(t) + sm_saving->
>claims_surrender(t)
        - units_bon(t) * lapse_rate_act_bal(t)
        + sm_accum->surr_charge(t) + sm_saving->surr_charge(t)
        )
    *
    piz
    / (sm_accum->units_e_bef(t) + sm_saving->units_e_bef(t));

piz = piz - decrements;

return min(    piz * surv_per_ret(t),
              units_e(t)); //Final cannot be greater than units

```

#### 6.1.1.3.1.138 units\_e\_piz\_int\_pup

```

if (t <= commence_period_w || t >= maturity_period_ann)
    return 0.0;

if(submodel != "UNIT" || eq(ben_class, "profil") || par_nonpar == "N" || paid_up == "Y")
    return NO_AVG;

if (t == 0)
    return 0.0;

double piz = units_e_piz_int_pup(t-1);
double new_pup = 0.0;

if (sm_accum->units_e(t-1) + sm_saving->units_e(t-1) > 0)
    new_pup = (
        units_b_bef_pup_acc(t) * (1. - sm_accum->
>surr_chg_perc_units[t+elapsed_months]/100.)
        + units_b_bef_pup_sav(t) * (1. - sm_saving->
>surr_chg_perc_units[t+elapsed_months]/100.)
        )
    * units_e_piz_int_active(t-1)
    / (sm_accum->units_e(t-1) + sm_saving->units_e(t-1));

piz = piz + new_pup + int_units_piz_pup(t);

//Decrements
double decrements = 0.0;

if (units_e_bef(t) > 0)
    decrements = (sm_accum->death_claims_units(t) + sm_saving->death_claims_units(t)
        + sm_acc_pup->death_claims_units(t) + sm_saving_pup->
>death_claims_units(t)

```

```

+ sm_accum->claims_surrender(t) + sm_saving-
>claims_surrender(t)
+ sm_acc_pup->claims_surrender(t) + sm_saving_pup-
>claims_surrender(t)
- units_bon(t) * lapse_rate_act_bal(t)
+ surr_charge(t)
)
*
piz
/ units_e_bef(t);

```

```
piz = piz - decrements;
```

```
return min( piz * surv_per_ret(t),
            units_e(t)); //Final cannot be greater than units
```

#### 6.1.1.3.1.139 be\_reserve

```

if (t <= commence_period_w || t > maturity_period_ann)
    return NO_AVG;

double Investment=0.0;
if(prodcode_par_nonpar == "0" || paid_up == "C")
    Investment=investment_income_chetz_pv(t);

return - cashflow_pv(t) - Investment;

```

#### 6.1.1.3.1.140 discount\_factor\_acc

```

if (t < commence_period_w || t > maturity_period_ann)
    return 0.0;

if (t <= 0) return 1.;

int proj_yr = xint(proj_year(t));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t));

if ( t > maturity_period_w)
    return discount_factor_acc(t-1)* ann_v_month_t[proj_yr];

return discount_factor_acc(t-1)* v_month_t[proj_yr];

```

#### 6.1.1.3.1.141 investment\_income\_chetz\_bef\_ret

```

if (t <= commence_period_w || t > maturity_period_ann)
    return 0.0;

int proj_yr = xint(proj_year(t));

```

```

if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t));

double temp_reserve=0;
double Cal_yr_mth=100*cal_year(t) +cal_month(t);

if (chetz_be_ind == "Y" && Cal_yr_mth>=chetz_be_ind_yrs)
    temp_reserve=-cashflow_pv_active_chetz(t-1)-cashflow_pv_deferred_chetz(t-1)+(riskadj_gross(t-1));
    else
        temp_reserve=reserve_bef_ret(t-1) + cashflow_b_bef_ret(t);

if(mult_age_ind == 1)
    return (inv_rate_mth_t[proj_yr] -inv_rate_rf_mth_t[proj_yr]) * (temp_reserve);

return 0.0; //Unconditional return

```

#### 6.1.1.3.1.142 investment\_income\_chetz\_pv

```

if (t < commence_period_w || t > maturity_period_ann || free_inv_prop_t[1] >= 1.)
    return 0.0;

if (mult_age_ind != 1) {

    int proj_yr = xint(proj_year(t+1));
    if(eq(projection_type_int, "Rollup"))
        proj_yr = xint(proj_year_rollup(t+1));

    if (t >= maturity_period_w)
        return (investment_income_chetz_pv(t+1) + investment_income_chetz(t+1))*
ann_v_month_t[proj_yr];

    return (investment_income_chetz_pv(t+1) + investment_income_chetz(t+1))*
v_month_t[proj_yr];

}

return investment_income_chetz_pv_active(t)
    + investment_income_chetz_pv_deferred(t)
    + investment_income_chetz_pv_inpay(t);

```

#### 6.1.1.3.1.143 investment\_income\_chetz\_pv\_active

```

if (t < commence_period_w || t >= maturity_period_ann || mult_age_ind !=1 || free_inv_prop_t[1] >= 1.)
    return 0.0;

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

return (investment_income_chetz_pv_active(t+1)
    + investment_income_chetz_bef_ret(t+1) )
    * v_month_t[proj_yr];

```

**6.1.1.3.1.144 investment\_income\_pv**

```

if (t < commence_period_w || t > maturity_period_ann)
    return 0.0;

if(mult_age_ind != 1){

    int proj_yr = xint(proj_year(t+1));
    if(eq(projection_type_int, "Rollup"))
        proj_yr = xint(proj_year_rollup(t+1));

    if (t >= maturity_period_w)
        return (investment_income_pv(t+1) + investment_income(t+1))*
ann_v_month_t[proj_yr];

    return (investment_income_pv(t+1) + investment_income(t+1))* v_month_t[proj_yr];

}

return investment_income_pv_active(t)
    + investment_income_pv_deferred(t)
    + investment_income_pv_inpay(t);

```

**6.1.1.3.1.145 pol\_fee**

```

if(submodel == "TRAD")
    return trad->pol_fee(t);

if(submodel == "TERM")
    return term->pol_fee(t);

if (t <= commence_period_w || t > maturity_period_w || premium_if_b(t)==0.)
    return 0.0;

// The policy fee is deducted at the same frequency as the premiums are payable.
double freq = prem_freq;
if (prem_term == 1)
    freq = 1.;

return policy_fee_if * policies_curr * surv_act_prm(t-1) / freq;

```

**6.1.1.3.1.146 pol\_fee\_pv**

```

if (t < commence_period_w || t >= maturity_period_w)
    return 0.0;

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

return pol_fee_pv(t+1)* v_month_t[proj_yr]
    + pol_fee(t+1);

```

**6.1.1.3.1.147 prem\_savings\_pv**

```

if (t < commence_period_w || t >= maturity_period_w)
    return 0.0;

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

return    prem_savings_pv(t+1)* v_month_t[proj_yr]
        + prem_savings(t+1);

```

**6.1.1.3.1.148 premium**

```

if(submodel == "ANN")
    return NO_AVG;

if(submodel == "TRAD")
    return trad->premium(t);

if(submodel == "TERM")
    return term->premium(t);

// Single premium for paid-up policies
if (t == commence_period_w +1 && (paid_up=="Y"))
    return (prem_curr_if) * benefits_curr;

if (prem_term <= 1 && t + elapsed_months == 1) // single premium
    return premium_if_b(t);

if (t <= commence_period_w || t + elapsed_months > prem_term)
    return 0.0;

if (fmod(xint(pol_month(t-1)), xint(12. / prem_freq))!=0)
    return 0.0; //not a premium due date

double SI = sum_insured(t)*surv_act_prm(t-1); // fixed SI
double prem_tot = premium_if_b(t)/prem_freq; // total premium
if (SI <=0. || !eq(ben_class,"adif"))
    return prem_tot;

double temp=0.0; // premium for Adif, remainder buys pure risk (Sapir) to reach fixed SI
double a = sm_accum->allocation_rate(t); // basic allocation rate
double t2 = prem_rates_extra_tt(xint(age_last(t)),sex_smoker_code)/prem_freq; // tarif for extra adif risk
double t1 = 0;
if (sum_ins_basic_tt(xint(age_last(t)),sex_smoker_code) != 0)
    t1 = (100.-alloc_rate[1]) / sum_ins_basic_tt(xint(age_last(t)),sex_smoker_code); // tarif
for basic sum insured
if (basic_perc(t)<1.)
    temp = prem_tot;
else if (t2 != 0 && t1 != 0){
    if((1-a)/t1 - 1/t2 != 0)
        temp = (SI-sm_accum->units_b_bef(t) - sm_saving->units_b_bef(t)-prem_tot/t2)
                / ((1.-a)/t1 - 1./t2); // reduced premium for basic Adif, remainder is
in prem_extra to buy extra risk
}
if (temp > prem_tot) // i.e. no extra sum insured required
    return prem_tot;

```

```

if (temp < 0) { // i.e. premium not enough to buy sum insured required (should not occur, since
fixed SI limited in sum_insured formula)
    //Fix premium to zero since it causes negative premium when lapse rate 100% for Qis5
purpose.
    return 0;
}
return temp;

```

#### 6.1.1.3.1.149 premium\_disc

```

if(submodel == "TERM")
    return term->premium_disc(t);

return NO_AVG;

```

#### 6.1.1.3.1.150 premium\_disc\_pv

```

if (t < commence_period_w || t > maturity_period_w)
    return 0.0;

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

return    premium_disc_pv(t+1)* v_month_t[proj_yr]
    + premium_disc(t+1);

```

#### 6.1.1.3.1.151 premium\_disc\_shimur

```

if(submodel == "TERM")
    return term->premium_disc(t) - term->premium_disc_no_shimur(t);

return NO_AVG;

```

#### 6.1.1.3.1.152 premium\_disc\_shimur\_pv

```

if (t < commence_period_w || t > maturity_period_w)
    return 0.0;

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

return    premium_disc_shimur_pv(t+1)* v_month_t[proj_yr]
    + premium_disc_shimur(t+1);

```

#### 6.1.1.3.1.153 premium\_extra

```

if(submodel == "ANN")
    return NO_AVG;

if (t <= commence_period_w || t > maturity_period_w)
    return 0.0;

if (!eq(ben_class,"adif"))
    return 0.;

return premium_if_b(t)/prem_freq - premium(t);

```



**6.1.1.3.1.154 premium\_gross**

```
if (t < commence_period_w || t > maturity_period_w)
    return 0.0;

if(submodel == "ANN")
    return NO_AVG;

if (submodel=="TERM")
    return term->premium_gross(t);

if (submodel=="TRAD")
    return trad->premium_gross(t);

if (fmod(xint(pol_month(t-1)), xint(12. / prem_freq))!=0)
    return 0.0; //not a premium due date

return premium(t) + pol_fee(t) + premium_extra(t);
```

**6.1.1.3.1.155 premium\_gross\_fix**

```
if (t <= commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if (eq(ben_class,"ltc"))

    return life->prem_insurance(t);

else

    return 0.0;
```

**6.1.1.3.1.156 premium\_gross\_var**

```
if (t <= commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if (eq(ben_class,"ltc"))

    return 0.0;

else

    return life->prem_insurance(t);
```

**6.1.1.3.1.157 premium\_pv**

```
if (t < commence_period_w || t >= maturity_period_w)
    return 0.0;

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));
```

```
return    premium_pv(t+1)* v_month_t[proj_yr]
        + premium_gross(t+1);
```

#### 6.1.1.3.1.158 total\_bor\_acc\_pv

```
if (t < commence_period_w || t >= maturity_period_w)
    return 0.0;
```

```
int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));
```

```
return    total_bor_acc_pv(t+1)* v_month_t[proj_yr]
        + bor_acc(t+1) + bor_acc_pup(t+1);
```

#### 6.1.1.3.1.159 total\_bor\_return\_pv

```
if (t < commence_period_w || t >= maturity_period_w)
    return 0.0;
```

```
int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));
```

```
return    total_bor_return_pv(t+1)* v_month_t[proj_yr]
        + bor_return(t+1) + bor_return_pup(t+1);
```

#### 6.1.1.3.1.160 outgo\_b

```
return comm_total(t)
    + claims_annuity(t)    // Annuities paying in the beginning of the period
    + exp_total(t)
    + premium_re(t);
```

#### 6.1.1.3.1.161 outgo\_e

```
return claims_total(t)
    -claims_annuity(t)    // Annuities paying in the beginning of the period
    +interest_re(t);
```

#### 6.1.1.3.1.162 outgo\_pv

```
if (t < commence_period_w || t >= maturity_period_ann)
    return 0.0;
```

```
if (mult_age_ind != 1){
```

```
    int proj_yr = xint(proj_year(t+1));
    if(eq(projection_type_int, "Rollup"))
        proj_yr = xint(proj_year_rollup(t+1));
```

```
    if ( t > maturity_period_w)
        return (outgo_pv(t+1) + outgo_e(t+1)) * ann_v_month_t[proj_yr] + outgo_b(t+1);
    else
        return (outgo_pv(t+1) + outgo_e(t+1)) * v_month_t[proj_yr] + outgo_b(t+1);
}
```

```
return outgo_pv_active(t)
    + outgo_pv_deferred(t)
```

```
+ outgo_pv_inpay(t);
```

#### 6.1.1.3.1.163 claim\_cost

```
if (t <= commence_period_w || !eq(submodel, "TERM") || t > maturity_period_w || t- term->t_start >
maturity_period_w || eq(paid_up,"C"))
    return 0.0;
```

```
if (eq(ben_class, "phi") && use_phi_claims_cf == "Y" && t- term->t_start >=0 )
    return term->claims_inpay_pv(t-term->t_start,0);
```

```
return NO_AVG;
```

#### 6.1.1.3.1.164 claim\_cost\_pv

```
if (t < commence_period_w || t > maturity_period_w)
    return 0.0;
```

```
int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));
```

```
return (claim_cost_pv(t+1) + claim_cost(t+1))
        * v_month_t[proj_yr];
```

#### 6.1.1.3.1.165 claim\_cost\_re

```
if (t <= commence_period_w || !eq(submodel, "TERM") || t > maturity_period_w || t- term->t_start >
maturity_period_w || reinsurance=="N" || eq(re_type,"NONE") || eq(paid_up,"C"))
    return 0.0;
```

```
if (eq(ben_class, "phi") && use_phi_claims_cf == "Y" && t- term->t_start >=0 )
    return claim_cost(t)*re_ratio_w;
```

```
return NO_AVG;
```

#### 6.1.1.3.1.166 claim\_cost\_re\_pv

```
if (t < commence_period_w || t > maturity_period_w || reinsurance=="N" || eq(re_type,"NONE"))
    return 0.0;
```

```
int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));
```

```
return (claim_cost_re_pv(t+1) + claim_cost_re(t+1))
        * v_month_t[proj_yr];
```

#### 6.1.1.3.1.167 claims\_annuity

```
if(t<= mat_period_min || t >= t_high)
    return NO_AVG;
if(inlist(submodel,"UNIT,TRAD,ANN") && res_prop_kitzba > 0.0){
    if(mult_age_ind == 1)
        return sm_annuity->pmt_total(t);
```

```

        return sm_annuity[ann_index_map[takeup_age]]->pmt_total(t);
    }
    return NO_AVG;

```

#### 6.1.1.3.1.168 claims\_annuity\_nogt

```

if(t<= mat_period_min || t >= t_high)
    return NO_AVG;

if(inlist(submodel,"UNIT,TRAD,ANN") && res_prop_kitzba > 0.0){
    if(mult_age_ind == 1)
        return sm_annuity->pmt_total_nogt(t);
    return sm_annuity[ann_index_map[takeup_age]]->pmt_total_nogt(t);
}
return NO_AVG;

```

#### 6.1.1.3.1.169 claims\_annuity\_nogt\_pv

```

if (t < commence_period_w || t >= t_high)
    return 0.0;

if(mult_age_ind !=1){

    int proj_yr = xint(proj_year(t+1));
    if(eq(projection_type_int, "Rollup"))
        proj_yr = xint(proj_year_rollup(t+1));

    if( t >= maturity_period_w)
        return claims_annuity_nogt_pv(t+1)* ann_v_month_t[proj_yr] +
claims_annuity_nogt(t+1);
    else
        return claims_annuity_nogt_pv(t+1)* v_month_t[proj_yr] +
claims_annuity_nogt(t+1);
}

return claims_annuity_nogt_pv_deferred(t) + claims_annuity_nogt_pv_inpay(t);

```

#### 6.1.1.3.1.170 claims\_annuity\_nogt\_pv\_deferred

```

if (t < commence_period_w || t >= t_high || mult_age_ind !=1)
    return 0.0;

double new_ret = 0.0;

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

if(ann_index_map.count(retirement_age_lookup(t)) != 0){

    if(t == sm_annuity[ann_index_map[retirement_age_lookup(t)]]->maturity_period_w)
        new_ret = sm_annuity[ann_index_map[retirement_age_lookup(t)]]-
>claims_annuity_nogt_pv(t);
}

return claims_annuity_nogt_pv_deferred(t+1) * v_month_t[proj_yr] + new_ret;

```

**6.1.1.3.1.171 claims\_annuity\_pv**

```

if (t < commence_period_w || t >= t_high)
    return 0.0;

if(mult_age_ind !=1){

    int proj_yr = xint(proj_year(t+1));
    if(eq(projection_type_int, "Rollup"))
        proj_yr = xint(proj_year_rollup(t+1));

    if( t >= maturity_period_w)
        return claims_annuity_pv(t+1)* ann_v_month_t[proj_yr] + claims_annuity(t+1);
    else
        return claims_annuity_pv(t+1)* v_month_t[proj_yr] + claims_annuity(t+1);
}

return claims_annuity_pv_deferred(t) + claims_annuity_pv_inpay(t);

```

**6.1.1.3.1.172 claims\_annuity\_pv\_deferred**

```

if (t < commence_period_w || t >= t_high || mult_age_ind !=1)
    return 0.0;

double new_ret = 0.0;

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

if(ann_index_map.count(retirement_age_lookup(t)) != 0){

    if(t == sm_annuity[ann_index_map[retirement_age_lookup(t)]]->maturity_period_w)
        new_ret = sm_annuity[ann_index_map[retirement_age_lookup(t)]]->claims_annuity_pv(t);
}

return claims_annuity_pv_deferred(t+1) * v_month_t[proj_yr] + new_ret;

```

**6.1.1.3.1.173 claims\_death**

```

if(submodel == "TRAD")
    return trad->claims_death(t);

if(submodel == "TERM"){
    if (inlist(ben_class, "accdth,dth,fib,mortg"))
        return term->claims_total(t);
    return 0;
}

return death_claim_si(t);

```

**6.1.1.3.1.174 claims\_death\_pv**

```

if (t < commence_period_w || t > maturity_period_w)
    return 0.0;

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

```

```

return      (claims_death_pv(t+1) + claims_death(t+1))
             * v_month_t[proj_yr];

```

#### 6.1.1.3.1.175 claims\_disability

```

if(submodel == "TRAD")
    return 0;

if(submodel == "TERM"){
    if (inlist(ben_class, "accdth,dth,fib,mortg"))
        return 0;
    return term->claims_total(t);
}

return 0.0;

```

#### 6.1.1.3.1.176 claims\_disability\_pv

```

if (t < commence_period_w || t > maturity_period_w)
    return 0.0;

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

return      (claims_disability_pv(t+1) + claims_disability(t+1))
             * v_month_t[proj_yr];

```

#### 6.1.1.3.1.177 claims\_lrc\_q1

```

if (eq(paid_up, "C"))
    return 0.0;

if(ben_class == "phi" && use_phi_claims_cf == "Y")
    return term->claims_inpay_q1(t);

if(ben_class == "phi" || prod_code == "ltc-shil")

    if (t<4 && t>0)
        return term->claims_total(t);

return 0.0;

```

#### 6.1.1.3.1.178 claims\_lrc\_q2

```

if (eq(paid_up, "C"))
    return 0.0;

if(ben_class == "phi" && use_phi_claims_cf == "Y")
    return term->claims_inpay_q2(t);

if(ben_class == "phi" || prod_code == "ltc-shil")

    if (t<7 && t>3)
        return term->claims_total(t);

return 0.0;

```

**6.1.1.3.1.179 claims\_lrc\_q3**

```
if (eq(paid_up, "C"))
    return 0.0;

if(ben_class == "phi" && use_phi_claims_cf == "Y")
    return term->claims_inpay_q3(t);

if(ben_class == "phi" || prod_code == "ltc-shil")

    if (t<10 && t>6)
        return term->claims_total(t);

return 0.0;
```

**6.1.1.3.1.180 claims\_lrc\_q4**

```
if (eq(paid_up, "C"))
    return 0.0;

if(ben_class == "phi" && use_phi_claims_cf == "Y")
    return term->claims_inpay_q4(t);

if(ben_class == "phi" || prod_code == "ltc-shil")

    if (t<13 && t>9)
        return term->claims_total(t);

return 0.0;
```

**6.1.1.3.1.181 claims\_lrc\_yr2plus**

```
if (eq(paid_up, "C"))
    return 0.0;

if(ben_class == "phi" && use_phi_claims_cf == "Y")
    return term->claims_inpay_other(t);

if(ben_class == "phi" || prod_code == "ltc-shil")

    if (t>12)
        return term->claims_total(t);

return 0.0;
```

**6.1.1.3.1.182 claims\_maturity**

```
if (t <= life->commence_period_w || t > life-> maturity_period_ann)
    return NO_AVG;

if(inlist(submodel,"TERM,ANN"))
    return 0.0;

if(submodel == "TRAD") {
    return trad->claims_maturity(t) * retirement_prop(t) * (1.-
sm_annuity[ann_index_map[retirement_age_lookup(t)]]->ann_takeup_rate(t-1));
}

if (t < mat_period_min || t > maturity_period_w)
```

```

    return 0.0;

if (t == maturity_period_w)
    return units_at_mat(t) * (1.- sm_annuity[ann_index_map[retirement_age_lookup(t)]]-
>ann_takeup_rate(t-1));

return units_at_mat(t) * retirement_prop(t) * (1.-
sm_annuity[ann_index_map[retirement_age_lookup(t)]]->ann_takeup_rate(t-1));

```

#### 6.1.1.3.1.183 claims\_maturity\_pv

```

if (t < commence_period_w || t > maturity_period_w)
    return 0.0;

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

return (claims_maturity_pv(t+1) + claims_maturity(t+1))
        * v_month_t[proj_yr];

```

#### 6.1.1.3.1.184 claims\_maturity\_ret

```

if (t <= life->commence_period_w || t > life-> maturity_period_ann)
    return NO_AVG;

if(inlist(submodel,"TERM,ANN"))
    return 0.0;

if(submodel == "TRAD") {
    return trad->claims_maturity(t) * retirement_prop(t) ;
}

if (t < mat_period_min || t > maturity_period_w)
    return 0.0;

if (t == maturity_period_w)
    return units_at_mat(t) ;

return units_at_mat(t) * retirement_prop(t);

```

#### 6.1.1.3.1.185 claims\_maturity\_ret\_pv

```

if (t < commence_period_w || t > maturity_period_w)
    return 0.0;

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

return (claims_maturity_ret_pv(t+1) + claims_maturity_ret(t+1))
        * v_month_t[proj_yr];

```

#### 6.1.1.3.1.186 claims\_pv

```

if (t < commence_period_w || t > maturity_period_ann)
    return 0.0;

if (mult_age_ind != 1){

```



```
int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

if (t >= maturity_period_w)
    return (claims_pv(t+1) + claims_total(t+1) - claims_annuity(t+1))*
ann_v_month_t[proj_yr] + claims_annuity(t+1);
else
    return (claims_pv(t+1) + claims_total(t+1) - claims_annuity(t+1))*
v_month_t[proj_yr] + claims_annuity(t+1);
}

return claims_annuity_pv(t)
    + claims_pv_not_annuity (t);
```

#### **6.1.1.3.1.187 claims\_re\_lrc\_q1**

```
if (eq(paid_up, "C"))
    return 0.0;

if(ben_class == "phi" || prod_code == "ltc-shil")
    return claims_lrc_q1(t) * re_ratio_w;

return 0.0;
```

#### **6.1.1.3.1.188 claims\_re\_lrc\_q2**

```
if (eq(paid_up, "C"))
    return 0.0;

if(ben_class == "phi" || prod_code == "ltc-shil")
    return claims_lrc_q2(t) * re_ratio_w;

return 0.0;
```

#### **6.1.1.3.1.189 claims\_re\_lrc\_q3**

```
if (eq(paid_up, "C"))
    return 0.0;

if(ben_class == "phi" || prod_code == "ltc-shil")
    return claims_lrc_q3(t) * re_ratio_w;

return 0.0;
```

#### **6.1.1.3.1.190 claims\_re\_lrc\_q4**

```
if (eq(paid_up, "C"))
    return 0.0;

if(ben_class == "phi" || prod_code == "ltc-shil")
    return claims_lrc_q4(t) * re_ratio_w;

return 0.0;
```

**6.1.1.3.1.191 claims\_re\_lrc\_yr2plus**

```

if (eq(paid_up, "C"))
    return 0.0;

if(ben_class == "phi" || prod_code == "ltc-shil")
    return claims_lrc_yr2plus(t) * re_ratio_w;;
return 0.0;

```

**6.1.1.3.1.192 claims\_surrender**

```

if(submodel == "TRAD")
    return trad->claims_surrender(t);

if(submodel == "TERM")
    return 0.0;

// force calc of surrender charge in main model
double temp1 = surr_charge(t);

// force calc of surrender value and penalties in the main model
double temp2 = surr_penalty_e_bef(t);

// Assume a zero surrender value is paid for each distinct unit type
// if there are negative units for that unit type.

return      max(0, sm_accum->claims_surrender(t)) +
             max(0, sm_acc_pup->claims_surrender(t))+
             max(0, sm_saving->claims_surrender(t)) +
             max(0, sm_saving_pup->claims_surrender(t)) +
             death_claim_units(t);

```

**6.1.1.3.1.193 claims\_surrender\_pv**

```

if (t < commence_period_w || t > maturity_period_w)
    return 0.0;

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

return      (claims_surrender_pv(t+1) + claims_surrender(t+1))
             * v_month_t[proj_yr];

```

**6.1.1.3.1.194 claims\_total**

```

return claims_death(t) + claims_disability(t) + claims_surrender(t)
       + claims_maturity(t) + claims_annuity(t);

```

**6.1.1.3.1.195 death\_benefit**

```

if (!eq(ben_class,"profil"))
    return NO_AVG;

if (t <= commence_period_w || t > maturity_period_w)
    return 0.0;

if (t>0)
    return sum_insured(t) + units_b(t);

```

```
return 0.0;
```

#### 6.1.1.3.1.196 death\_claim\_si

```
if (t <= commence_period_w || t > maturity_period_w)
    return 0.0;

if (eq(ben_class,"adif"))
    return sum_at_risk_claim(t)* death_rate(t);
else
    if (t>0)
        return rider_perc_allowed(t) / 100.
            * claim_amount_tt.sum_of_row(t) * surv_act_prm(t-1);
    else
        return 0.0;
```

#### 6.1.1.3.1.197 death\_claim\_units

```
// Assume a zero unit value if the unit value is negative
if(submodel == "TRAD")
    return (trad->surr_value(t)+trad->surr_value_pup(t))*trad->death_rate(t);

double temp=0.0;

temp = max(0, sm_accum->death_claims_units(t)) +
        max(0, sm_acc_pup->death_claims_units(t))+
        max(0, sm_saving->death_claims_units(t)) +
        max(0, sm_saving_pup->death_claims_units(t)) +
        units_bon(t) * death_rate(t);

return temp;
```

#### 6.1.1.3.1.198 expense\_claims\_lrc\_q1

```
if (t <= commence_period_w || t > maturity_period_ann||(exp_claim_perc == 0. && exp_claim_fix ==0.)
|| eq(paid_up,"C"))
    return 0.0;

if(ben_class == "phi" && use_phi_claims_cf == "Y")
{
    double clm_perc = exp_claim_perc/100.
        * claims_lrc_q1 (t)
        * expense_inflation(t);

    double clm_fix = exp_claim_fix
        * term->claims_rate_per_q1(t)
        * expense_inflation(t);

    double margin = 0.;
    if(margin_add=="Y")
        margin = margin_exp_ren_pc;

    double result = (clm_perc + clm_fix) * (1 + margin/100.);

    return result;
}
```

```
if(ben_class == "phi" || prod_code == "ltc-shil")
```

```
    if (t<4 && t>0)
        return expense_claims(t);
```

```
return 0.0;
```

#### **6.1.1.3.1.199 expense\_claims\_lrc\_q2**

```
if (t <= commence_period_w || t > maturity_period_ann||(exp_claim_perc == 0. && exp_claim_fix ==0.)
|| eq(paid_up,"C"))
    return 0.0;
```

```
if(ben_class == "phi" && use_phi_claims_cf == "Y")
{
```

```
    double clm_perc = exp_claim_perc/100.
        * claims_lrc_q2 (t)
        * expense_inflation(t);
```

```
    double clm_fix = exp_claim_fix
        * term->claims_rate_per_q2(t)
        * expense_inflation(t);
```

```
    double margin = 0.;
    if(margin_add=="Y")
        margin = margin_exp_ren_pc;
```

```
    double result = (clm_perc + clm_fix) * (1 + margin/100.);
```

```
    return result;
```

```
}
```

```
if(ben_class == "phi" || prod_code == "ltc-shil")
```

```
    if (t<7 && t>3)
        return expense_claims(t);
```

```
return 0.0;
```

#### **6.1.1.3.1.200 expense\_claims\_lrc\_q3**

```
if (t <= commence_period_w || t > maturity_period_ann||(exp_claim_perc == 0. && exp_claim_fix ==0.)
|| eq(paid_up,"C"))
    return 0.0;
```

```
if(ben_class == "phi" && use_phi_claims_cf == "Y")
{
```

```
    double clm_perc = exp_claim_perc/100.
        * claims_lrc_q3 (t)
        * expense_inflation(t);
```

```
    double clm_fix = exp_claim_fix
```

```

        * term->claims_rate_per_q3(t)
        * expense_inflation(t);

double margin = 0.;
if(margin_add=="Y")
    margin = margin_exp_ren_pc;

double result = (clm_perc + clm_fix) * (1 + margin/100.);

return result;
}

if(ben_class == "phi" || prod_code == "ltc-shil")

    if (t<10 && t>6)
        return expense_claims(t);

return 0.0;

```

#### 6.1.1.3.1.201 expense\_claims\_lrc\_q4

```

if (t <= commence_period_w || t > maturity_period_ann || (exp_claim_perc == 0. && exp_claim_fix ==0.)
|| eq(paid_up,"C"))
    return 0.0;

if(ben_class == "phi" && use_phi_claims_cf == "Y")
{

    double clm_perc = exp_claim_perc/100.
        * claims_lrc_q4 (t)
        * expense_inflation(t);

    double clm_fix = exp_claim_fix
        * term->claims_rate_per_q4(t)
        * expense_inflation(t);

    double margin = 0.;
    if(margin_add=="Y")
        margin = margin_exp_ren_pc;

    double result = (clm_perc + clm_fix) * (1 + margin/100.);

    return result;
}

if(ben_class == "phi" || prod_code == "ltc-shil")

    if (t<13 && t>9)
        return expense_claims(t);

return 0.0;

```

**6.1.1.3.1.202 expense\_claims\_lrc\_yr2plus**

```

if (t <= commence_period_w || t > maturity_period_ann || (exp_claim_perc == 0. && exp_claim_fix ==0.)
|| eq(paid_up,"C"))
    return 0.0;

if(ben_class == "phi" && use_phi_claims_cf == "Y")
{
    double clm_perc = exp_claim_perc/100.
        * claims_lrc_yr2plus (t)
        * expense_inflation(t);

    double clm_fix = exp_claim_fix
        * term->claims_rate_per_other(t)
        * expense_inflation(t);

    double margin = 0.;
    if(margin_add=="Y")
        margin = margin_exp_ren_pc;

    double result = (clm_perc + clm_fix) * (1 + margin/100.);

    return result;
}

if(ben_class == "phi" || prod_code == "ltc-shil")
    if (t>12)
        return expense_claims(t);

return 0.0;

```

**6.1.1.3.1.203 nogt\_annpv**

```

if (t < commence_period_w || t >= t_high)
    return 0.0;

return claims_annuity_nogt_pv(t);

```

**6.1.1.3.1.204 surr\_penalty\_e\_bef**

```

return sm_accum->surr_penalty_e_bef(t) +
    sm_acc_pup->surr_penalty_e_bef(t)+
    sm_saving->surr_penalty_e_bef(t) +
    sm_saving_pup->surr_penalty_e_bef(t);

```

**6.1.1.3.1.205 surr\_value**

```

if (inlist(submodel,"TERM,ANN"))
    return NO_AVG;

if (submodel=="TRAD")
    return trad->surr_value(t)+ trad->surr_value_pup(t);

```

```

return (sm_accum->surr_value(t) +
        sm_acc_pup->surr_value(t) +
        sm_saving->surr_value(t) +
        sm_saving_pup->surr_value(t));

```

#### 6.1.1.3.1.206 comm\_hekef

// Commission is Paid on total initial annualized premium (including loadings and excluding discounts)

```

if (t <= commence_period_w || t > maturity_period_w || eq(paid_up,"G"))
    return 0.0;

```

```

double comex1 =0. ; // payment at policy commencement
double comm_addition_rider =0. ;

```

```

if ( eq(paid_up, "Y")){

```

```

    if(t + elapsed_months + elapsed_months_extra == 1){

```

```

        if(eq(prod_code,"prof-fin")) //Paid-up Ogen
            comex1 = prem_curr_if
                    * comm_hekef_pc / 100.;

```

```

        if (eq(comm_set_temp, "finance") || eq(prod_code, "prof13-see")) //Ofek and niud
(dinami should only be for niud part)

```

```

            comex1 = resinforce
                    * comm_hekef_pc_res /100.;

```

```

        if(eq(submodel,"UNIT") && (elapsed_months + elapsed_months_extra) < 12){

```

```

            comm_addition_rider = cover_charge(1) * (comm_hekef_pc_rider -
comm_hekef_pc)/100;

```

```

        }

```

```

        return (comex1 + comm_addition_rider)*(1+vat/100.);

```

```

    }

```

```

    return 0.0;

```

```

}

```

```

if (t + elapsed_months + elapsed_months_extra == 1){
    comex1 = premium_gross(t)*prem_freq
            *comm_hekef_pc /100.;

```

```

    if(eq(submodel,"UNIT") && (elapsed_months + elapsed_months_extra) <= 12){

```

```

        double comm_temp_pc = (comm_hekef_pc_rider - comm_hekef_pc)/100.;

```

```

        double temp_surv = 1.;

```

```

        if (surv_act_prm(1)> 0.00001)

```

```

            temp_surv = surv_act_prm(1);

```

```

        comm_addition_rider = (cover_charge(1)*surv_act_prm(t)/temp_surv)*prem_freq *

```

```

comm_temp_pc;

```

```

    }

```

```

}

```

```
return (comex1 + comm_addition_rider)*(1+vat/100.);
```

#### 6.1.1.3.1.207 comm\_nihul

```
if (t <= commence_period_w || t > maturity_period_w || inlist(paid_up,"Y,G"))
    return 0.0;

if ( (prod_yr_w >= 2004) && !eq(submodel,"TRAD") ) { // for policies sold from 2004 commission
paid on policy fee etc.
    return ( premium(t) * basic_perc(t) + pol_fee(t) - premium_disc(t) )
        * comm_nihul_rate[xint(pol_year(t))] /100.
        * (1.+vat/100.) ;
    }
else
    return 0.0;
```

#### 6.1.1.3.1.208 comm\_nihul\_pv

```
if (t < commence_period_w || t >= maturity_period_w)
    return 0.0;

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

return comm_nihul_pv(t+1) * v_month_t[proj_yr]
    + comm_nihul(t+1);
```

#### 6.1.1.3.1.209 comm\_not\_res\_pv

```
if (t < commence_period_w || t >= maturity_period_w)
    return 0.0;

return comm_pv(t) - comm_reserve_pv(t);
```

#### 6.1.1.3.1.210 comm\_prize

```
// Commission is Paid on total initial annualized premium (including loadings and excluding
discounts)

if (t <= commence_period_w || t > maturity_period_w )
    return 0.0;

double comex1 =0. ; // payment at policy commencement

if ( eq(paid_up, "Y")){

    if(eq(comm_set_temp, "finance")){

        if (t + elapsed_months + elapsed_months_extra == 1)
            comex1 = resinforce
                * comm_prizes_pc_res /100.;

        //No riders on Ofek
        return comex1*(1+vat/100.);

    }
    else
        return 0.0;
```



```

}

if (t + elapsed_months + elapsed_months_extra == 1)
    comex1 = premium_gross(t)*prem_freq
            * comm_prizes_pc /100.;

return comex1*(1+vat/100.);

```

#### 6.1.1.3.1.211 comm\_pv

```

if (t < commence_period_w || t >= maturity_period_w)
    return 0.0;

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

return      comm_pv(t+1)* v_month_t[proj_yr]
            + comm_total(t+1);

```

#### 6.1.1.3.1.212 comm\_reg\_riders\_out\_pv

```

if (t < commence_period_w || t >= maturity_period_w)
    return 0.0;

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

return      comm_reg_riders_out_pv(t+1)* v_month_t[proj_yr]
            + comm_reg_riders_out(t+1);

```

#### 6.1.1.3.1.213 comm\_regular

```

if (t <= commence_period_w || t > maturity_period_w || paid_up=="Y")
    return 0.0;

if ( (prod_yr_w >= 2004) && !eq(submodel,"TRAD") )    { // for policies sold from 2004 commission
paid on policy fee etc.
    return ( ( premium(t) * basic_perc(t) + pol_fee(t) - premium_disc(t) )
            * comm_regular_pc[xint(pol_year(t))] /100. )
            * (1.+vat/100.) ;
}
else {
    if (eq(submodel,"UNIT"))
        return ( sm_accum->comm_regular(t) + sm_saving->comm_regular(t)+
                premium_extra(t)*comm_regular_pc[xint(pol_year(t))] / 100. )
                * (1.+vat/100.);
    else
        return premium(t) / (1.+tat_shnatiut_rate/100.)
                * comm_regular_pc[xint(pol_year(t))] / 100.
                * (1.+vat/100.);
}
}

```

#### 6.1.1.3.1.214 comm\_renewal

```

if (t <= commence_period_w || t > maturity_period_w)
    return 0.0;

```

```

if (t + elapsed_months <= (comm_renewal_year-1)*12)
    return 0.0;

double comm_ren_perc_prem_temp=comm_ren_perc_prem;
double amla_hishtatfut_dnp_temp = amla_hishtatfut_dnp/ 100.;

if(ben_class == "mortg" && origidate >= 200704 && channel != "Banks" && xint(pol_year(t) +
round(elapsed_months_extra/12.,0)) >=16)
    comm_ren_perc_prem_temp=min(comm_ren_perc_prem_mrtg,comm_ren_perc_prem);

if(comm_ren_perc_prem_temp>0 || comm_ren_perc_sav>0)
    amla_hishtatfut_dnp_temp=0.0;

if (submodel=="UNIT")
    return ((sm_accum->comm_renewal(t) + sm_saving->comm_renewal(t) +
            (premium_extra(t) + pol_fee(t))
            * comm_ren_perc_prem_temp / 100.) + amla_hishtatfut_dnp_temp* charges_premium(t))
            * (1+vat/100.);
else
    return      premium_gross(t)
            * comm_ren_perc_prem_temp/ 100.
            * (1+vat/100.);

```

#### 6.1.1.3.1.215 comm\_renewal\_pv

```

if (t < commence_period_w || t >= maturity_period_w)
    return 0.0;

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

return      comm_renewal_pv(t+1)* v_month_t[proj_yr]
            + comm_renewal(t+1) + comm_profit(t+1);

```

#### 6.1.1.3.1.216 comm\_reserve

```

if (submodel=="UNIT")
    return (sm_accum->comm_reserve(t) +
            sm_acc_pup->comm_reserve(t)+
            sm_saving->comm_reserve(t) +
            sm_saving_pup->comm_reserve(t));
    // no need to add VAT as Res Comm Perc already includes VAT
else
    return 0.0;

```

#### 6.1.1.3.1.217 comm\_reserve\_pv

```

if (t < commence_period_w || t >= maturity_period_w)
    return 0.0;

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

return      comm_reserve_pv(t+1)* v_month_t[proj_yr]
            + comm_reserve(t+1);

```

**6.1.1.3.1.218 comm\_total**

```
if (t <= commence_period_w || t > maturity_period_w)
    return 0.0;
```

```
return comm_nihul(t) +
    comm_regular(t) +
    comm_renewal(t) +
    comm_reserve(t) -
    comm_clawback(t) +
    comm_prize(t) +
    comm_hekef(t) +
    comm_profit(t) +
    comm_reg_riders_out(t);
```

**6.1.1.3.1.219 profit\_re\_pv**

```
if (t < commence_period_w || t >= maturity_period_w)
    return 0.0;
```

```
double prof = premium_re(t+1)
    - claims_re(t+1)
    - comm_re(t+1)
    - comm_re_prof(t+1);
```

```
int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));
```

```
return profit_re_pv(t+1)* v_month_t[proj_yr] + prof;
```

**6.1.1.3.1.220 comm\_clawback**

/\*The amount of this cashflow in period t is equal to the accumulated commission amount, in respect of currently inforce policies, at the end of period t-1 multiplied by the total of lapsing policies and policies becoming pup in period t-1. Initial commission is clawed back only after the policy has been in force for greater than one period.  
\*/

```
if (t-1 <= commence_period_w)
    return 0;
```

```
// Clawback is not calculated once a policy has become paid-up.
if (inlist(paid_up,"G"))
    return 0;
```

```
if(eq(paid_up, "Y"))
{
    double temp = surv_pup_prm(t-2) * lapse_rate_pup_prm(t-1);

    if (t+elapsed_months-1+elapsed_months_extra <=180 && surv_prm(commence_period_w) != 0)
        return comm_claw_prpn_hekef[t+elapsed_months-1+elapsed_months_extra] / 100
            * comm_hekef_cum(t-1)
            * temp / surv_prm(commence_period_w);

    return 0;
```

```

}

// temp = policies lapsing or becoming paid-up in period
double temp = surv_act_prm(t-2) * (lapse_rate_act_prm_dep(t-1) + pup_rate_prm_dep(t-1));

double result1 = 0.0; //hekef commission

if (t+elapsed_months-1+elapsed_months_extra <=180){

    result1 = comm_claw_prpn_hekef[t+elapsed_months-1+elapsed_months_extra] / 100.*
              comm_hekef_cum(t-1);
}

if (surv_prm(commence_period_w) != 0)
    return result1
        * temp / surv_prm(commence_period_w);

return 0;

```

#### 6.1.1.3.1.221 comm\_hekef\_cum

```

if (t <= commence_period_w || t > maturity_period_w || inlist(paid_up,"G"))
    return 0.0;

if(eq(paid_up, "Y")){

    if (surv_pup_prm(t-1) < 0.0000001)
        return comm_hekef_cum(t-1);

    return comm_hekef_cum(t-1)
        + comm_hekef(t) * surv_prm(commence_period_w) / surv_pup_prm(t-1);

}

if(surv_act_prm(t-1)<0.0000001) return comm_hekef_cum(t-1);

return comm_hekef_cum(t-1)
    + comm_hekef(t) * surv_prm(commence_period_w) / surv_act_prm(t-1);

```

#### 6.1.1.3.1.222 exp\_total

```

return expense_initial_fix(t)
    + expense_initial_perc(t)
    + expense_ren_fix(t)
    + expense_ren_perc(t)
    + expense_investment(t)
    + expense_claims(t)
    + expense_ren_charge(t)
    + comm_supervisor(t)
    - comm_claw_spv(t);

```

#### 6.1.1.3.1.223 expense\_inflation

```

if (t < 1 || t > maturity_period_ann)
    return 1.0;

```

```
return expense_inflation(t-1) * (1.0 + exp_inflation_mthly);
```

#### 6.1.1.3.1.224 expense\_pv

```
if (t < commence_period_w || t > maturity_period_ann)
    return 0.0;

if(mult_age_ind != 1){

    int proj_yr = xint(proj_year(t+1));
    if(eq(projection_type_int, "Rollup"))
        proj_yr = xint(proj_year_rollup(t+1));

    if ( t > maturity_period_w)
        return expense_pv(t+1) * ann_v_month_t[proj_yr] + exp_total(t+1);
    else
        return expense_pv(t+1) * v_month_t[proj_yr] + exp_total(t+1);
}
return expense_pv_active(t)
    + expense_pv_deferred(t)
    + expense_pv_inpay(t);
```

#### 6.1.1.3.1.225 expense\_total\_pre\_ret

```
return expense_initial_fix(t)
    + expense_initial_perc(t)
    + expense_ren_fix(t)
    + expense_ren_perc_bef_ret(t)
    + expense_claims(t)
    + expense_ren_charge(t)
    + comm_supervisor(t)
    - comm_claw_spv(t);
```

#### 6.1.1.3.1.226 expense\_total\_pre\_ret\_no\_inv

```
return expense_initial_fix(t)
    + expense_initial_perc(t)
    + expense_ren_fix(t)
    + expense_ren_perc_bef_ret_no_inv(t)
    + expense_claims(t)
    + expense_ren_charge(t)
    + comm_supervisor(t)
    - comm_claw_spv(t);
```

#### 6.1.1.3.1.227 expense\_var\_pv

```
if (t < commence_period_w || t > maturity_period_ann)
    return 0.0;

if(mult_age_ind != 1){

    int proj_yr = xint(proj_year(t+1));
    if(eq(projection_type_int, "Rollup"))
        proj_yr = xint(proj_year_rollup(t+1));

    if ( t > maturity_period_w){
        return expense_var_pv(t+1)* ann_v_month_t[proj_yr]
```

```

        + expense_initial_perc(t+1)
        + expense_ren_perc(t+1)
        + expense_ren_charge(t+1);
    }
    return expense_var_pv(t+1)* v_month_t[proj_yr]
        + expense_initial_perc(t+1)
        + expense_ren_perc(t+1)
        + expense_ren_charge(t+1);
}

return expense_pv_deferred(t)
    + expense_pv_inpay(t)
    + expense_var_pv_active(t);

```

#### 6.1.1.3.1.228 expense\_init\_fix\_cvr

```

if (t <= commence_period_w || t > maturity_period_w)
    return 0.0;

if (t + elapsed_months+elapsed_months_extra == 1) {

    double result1 = exp_init_fix_cov * benefits_curr * surv_cnt(0);

    //add expenses for profil riders
    double result2 = 0;
    if (eq(ben_class,"profil")){
        for (int i=0; i < riders_count_w; i++) {
            result2 = result2 + exp_initial_fix_rider[i];
        }
    }

    double margin = 0.;
    if(margin_add=="Y")
        margin = margin_exp_ini_fix;

    return (result1 + result2)
        * (1+margin/100.);
}
return 0.0;

```

#### 6.1.1.3.1.229 expense\_initial\_fix

```

if (t <= commence_period_w || t > maturity_period_w)
    return 0.0;

return expense_initial_fix_pol(t) + expense_init_fix_cvr(t);

```

#### 6.1.1.3.1.230 expense\_initial\_fix\_pol

```

if (t <= commence_period_w || t > maturity_period_w)
    return 0.0;

if (t + elapsed_months+elapsed_months_extra == 1) {

    double margin = 0.;
    if(margin_add=="Y")

```

```

        margin = margin_exp_ini_fix;

    return exp_init_fix
        * policies_b(t)
        * (1. + margin/100.);
}
return 0.0;

```

#### 6.1.1.3.1.231 expense\_initial\_perc

```

if (t <= commence_period_w || t > maturity_period_w || paid_up == "Y")
    return 0.0;

xstring key_temp;

if (t + elapsed_months + elapsed_months_extra == 1) {

    double margin = 0.;
    if(margin_add=="Y")
        margin = margin_exp_ini_pc;

    double profil_risk_init_exps = 0.0;
    if (eq(ben_class,"profil")) {
        double risk_extra_init_perc = 0.0;

        for (int i=0; i < riders_count_w; i++) {
            rider_tarif_row_key=xstring(tarif_rider[i]);
            prod_code_rider = rider_tarif_tbl;
            key_temp =prod_assumpt_rider_exp_tbl;
            exp_row_lookup=key_temp+"_"+company+"_"+pol_type_expenses;
            risk_extra_init_perc = i_prem;
            risk_extra_init_perc = risk_extra_init_perc - exp_init_perc_prem;

            profil_risk_init_exps = profil_risk_init_exps +
                (sm_riders[i]->prem_cover/12.) * 12 *
                exp_initial_extra_perc_charge[i] / 100. * surv_act_prm(t-1);

        } // end for
    } // end if (eq(ben_class,"profil"))

    if (margin_add_discount == "Y" && eq(submodel, "TERM"))
        return (exp_init_perc_prem / 100. * term->prem_gross_no_scen(t)* prem_freq
//Do not apply expenses to discount scenario
        + profil_risk_init_exps ) * (1. + margin/100.);

    return (exp_init_perc_prem / 100. * premium_gross(t) * prem_freq
        + profil_risk_init_exps ) * (1. + margin/100.);
} // end if t = month 1 of policy

return 0.0;

```

#### 6.1.1.3.1.232 expense\_claims

```

return expense_clm_perc(t) + expense_clms_fix(t);

```

#### 6.1.1.3.1.233 expense\_claims\_pv

```

if (t < commence_period_w || t > maturity_period_w)
    return 0.0;

```

```

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

return expense_claims_pv(t+1)* v_month_t[proj_yr]+ expense_claims(t+1);

```

#### 6.1.1.3.1.234 expense\_clm\_perc

```

if (t <= commence_period_w || t > maturity_period_ann || exp_claim_perc == 0.)
    return 0.0;

double result = exp_claim_perc/100.
                * (claims_total(t-1)-claims_annuity(t-1))
                * expense_inflation(t);

double margin = 0.;
if(margin_add=="Y")
    margin = margin_exp_ren_pc;

return result * (1 + margin/100.);

```

#### 6.1.1.3.1.235 expense\_clms\_fix

```

if (t <= commence_period_w || t > maturity_period_ann || exp_claim_fix == 0.)
    return 0.0;

double result = exp_claim_fix
                * claims_rate_per(t)
                * expense_inflation(t);

double margin = 0.;
if(margin_add=="Y")
    margin = margin_exp_ren_pc;

return result * (1 + margin/100.);

```

#### 6.1.1.3.1.236 expense\_pv\_ann

```

if (t < commence_period_w || t >= maturity_period_ann)
    return 0.0;

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

if ( t > maturity_period_w)
    return expense_pv_ann(t+1) * ann_v_month_t[proj_yr]+ expense_ren_perc_ann(t+1);
else
    return expense_pv_ann(t+1) * v_month_t[proj_yr]+ expense_ren_perc_ann(t+1);

```

#### 6.1.1.3.1.237 expense\_ren\_fix\_pv

```

if (t <= commence_period_w || t > maturity_period_ann)
    return 0.0;

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

```



```
return expense_ren_fix_pv(t+1)* v_month_t[proj_yr]+ expense_ren_fix(t+1);
```

#### 6.1.1.3.1.238 expense\_ren\_perc\_pv

```
if (t < commence_period_w || t > maturity_period_ann)
    return 0.0;
```

```
int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));
```

```
return expense_ren_perc_pv(t+1)* v_month_t[proj_yr]+ expense_ren_perc(t+1);
```

#### 6.1.1.3.1.239 comm\_reg\_riders\_out

```
if (submodel != "UNIT")
    return NO_AVG;
```

```
if (t <= commence_period_w || t > maturity_period_w || !eq(ben_class,"profil"))
    return 0.0;
```

```
double result = 0.0;
double extra_rider_comm = 0.0;
```

```
/** for some reason negative t values are called (covercharges not working ? for t<0)
```

```
if (t<0)
    return 0;
```

```
for (int i=0; i < riders_count_w; i++) {
```

```
    if ( (tarif_rider[i] > 0) && sm_riders[i]->prm_in_ppn <=0.5 ) { // rider premium is "out"
```

```
        if (proj_year(t)<7.0)
```

```
            extra_rider_comm = max((sm_riders[i]->amla_1_6 + sm_riders[i]-
            >aml_ni_1_6)/100.
```

```
                -comm_regular_pc[xint(pol_year(t))] / 100.
```

```
                -comm_nihul_rate[xint(pol_year(t))] / 100. , 0.0);
```

```
        else
```

```
            extra_rider_comm = max((sm_riders[i]->amla_7 + sm_riders[i]->amla_ni_7)/100.
```

```
                -comm_regular_pc[xint(pol_year(t))] / 100.
```

```
                -comm_nihul_rate[xint(pol_year(t))] / 100. , 0.0);
```

```
        if (accum->allocation_rate(t)>0.0)
```

```
            result = result + charge_amount_tt(t,i) * rider_perc_allowed(t) / 100. /
```

```
(accum->allocation_rate(t))
```

```
            * extra_rider_comm * surv_act_prm(t-1);
```

```
        else
```

```
            result = result + charge_amount_tt(t,i) * rider_perc_allowed(t) / 100.
```

```
            * extra_rider_comm * surv_act_prm(t-1);
```

```
    }
```

```
}
```

```
return result;
```

**6.1.1.3.1.240 expense\_ren\_charge**

```

if (submodel != "UNIT")
    return NO_AVG;

if (t <= commence_period_w || t > maturity_period_w)
    return 0.0;

if (!eq(ben_class,"profil"))
    return 0.;

double result = 0.0;

if (t<0)
    return 0;

if (eq(company, "hasne")){//expenses for Hasne remain unchanged from model v32

    for (int i=0; i < riders_count_w; i++) {
        result = result + charge_amount_tt(t,i) * rider_perc_allowed(t) / 100. *
exp_extra_perc_charge[i] / 100. * surv_act_prm(t-1);    // t=r ? ***
    }
}
else

{
//Expenses for clal

    for (int i=0; i < riders_count_w; i++) {
        result = result + charge_amount_tt(1,i) * rider_perc_allowed(t) / 100. *
exp_extra_perc_charge[i] / 100. * surv_act_prm(t-1);
    }

}

return result;

```

**6.1.1.3.1.241 expense\_ren\_fix**

```

if (t <= commence_period_w || t > maturity_period_ann)
    return 0.0;

return expense_ren_fix_cvr(t)
    + expense_ren_fix_pol(t)
    + expense_ren_fix_pup(t);

```

**6.1.1.3.1.242 expense\_ren\_fix\_cvr**

```

if (t <= commence_period_w || t > maturity_period_ann)
    return 0.0;

if (paid_up=="C")
    return 0.0;

double temp = surv_act_cnt(t-1);

if (eq(ben_class,"dd") && surv_cnt(t-1)>0.)
    temp = (term->surv_2(t-1)+surv_cnt(t-1)); // to allow for expenses for secondary lives

```

```

double result1 = exp_ren_fix_cov / 12.
    * benefits_curr
    * expense_inflation(t)
    * temp;

//add expenses for profil riders
double result2 = 0;
if (eq(ben_class,"profil")){
    for (int i=0; i < riders_count_w; i++) {
        result2 = result2
            + exp_ren_fix_rider[i] * expense_inflation(t) * temp / 12.;
    }
}

double margin = 0.;
if(margin_add=="Y")
    margin = margin_exp_ren_fix;

return (result1 + result2) * (1 + margin/100.);

```

#### 6.1.1.3.1.243 expense\_ren\_fix\_pol

```

if (t <= commence_period_w || t > maturity_period_ann)
    return 0.0;

if (paid_up=="C")
    return 0.0;

double temp = surv_act_cnt(t-1);

if (eq(ben_class,"dd") && surv_cnt(t-1)>0.)
    temp = (term->surv_2(t-1)+surv_cnt(t-1)); // to calculate expenses for secondary lives

double result = exp_ren_fix / 12.
    * policies_curr
    * expense_inflation(t)
    * temp;

double margin = 0.;
if(margin_add=="Y")
    margin = margin_exp_ren_fix;

return result * (1 + margin/100.);

```

#### 6.1.1.3.1.244 expense\_ren\_fix\_pup

```

if (t <= commence_period_w || t > maturity_period_ann)
    return 0.0;

if (paid_up=="C")
    return 0.0;

double result = exp_pup_fix / 12.
    * expense_inflation(t)
    * policies_curr
    * surv_pup_cnt(t-1);

double margin = 0.;

```

```
if(margin_add=="Y")
    margin = margin_exp_ren_fix;
```

```
return result * (1 + margin / 100.);
```

#### 6.1.1.3.1.245 expense\_ren\_perc

```
if (t <= commence_period_w || t > maturity_period_ann)
    return 0.0;
```

```
// % of premium expenses are assumed to be
// payable at the same frequency as the premium
```

```
double result = 0.0;
```

```
double index = expense_inflation(t);
```

```
if (eq(submodel, "TERM")) {
    double temp = surv_act_prm(t-1);
```

```
    // ltc pups should not have expenses if ((eq(life->ben_class,"dd") || eq(life-
>ben_class,"ltc")) && surv(t-1)>0.)
```

```
    if (eq( ben_class,"dd") && surv_prm(t-1)>0.)
```

```
        temp = (term->surv_2(t-1)+surv_prm(t-1)); // to calculate expenses for secondary
lives
```

```
    result = (exp_ren_perc_prem /100. * prem_curr_if/12. * temp
        + exp_ren_perc_annuity/100. * claims_annuity(t))*index;
```

```
}
else
{
```

```
    result = (exp_ren_perc_prem / 100. * premium_gross(t)
        + exp_ren_perc_annuity/100. * claims_annuity(t))*index;
```

```
}
```

```
double margin = 0.;
```

```
if(margin_add=="Y")
    margin = margin_exp_ren_pc;
```

```
return result * (1 + margin / 100.);
```

#### 6.1.1.3.1.246 expense\_ren\_perc\_ann

```
if (t <= commence_period_w || t > maturity_period_ann)
    return 0.0;
```

```
double result = 0.0;
```

```
result = exp_ren_perc_annuity/100. * claims_annuity(t) * expense_inflation(t);
```

```
double margin = 0.;
```

```
if(margin_add=="Y")
    margin = margin_exp_ren_pc;
```

```
return result * (1 + margin / 100.);
```

**6.1.1.3.1.247 expense\_ren\_perc\_bef\_ret**

```

if (t <= commence_period_w || t > maturity_period_w)
    return 0.0;

// % of premium expenses are assumed to be
// payable at the same frequency as the premium

double result = 0.0;

double index = expense_inflation(t);

if (eq(company, "hasne")){

if (margin_add_discount == "Y" && eq(submodel, "TERM"))
    result = (exp_ren_perc_prem / 100. * term->prem_gross_no_scen(t)
              + exp_ren_res / 1200. * reserve_bef_ret(t) *
free_inv_prop_t[proj_year(t)])*index;

result = (exp_ren_perc_prem / 100. * premium_gross(t)
          //+ exp_ren_res / 1200. * reserve(t) * free_inv_prop_t[proj_year(t)]
          + exp_ren_res / 1200. * reserve_bef_ret(t) *
free_inv_prop_t[proj_year(t)])*index;
}
else
{

    double temp = surv_act_prm(t-1);

    // ltc pups should not have expenses if ((eq(life->ben_class,"dd") || eq(life-
>ben_class,"ltc")) && surv(t-1)>0.)
    if (eq( ben_class,"dd") && surv_prm(t-1)>0.)
        temp = (term->surv_2(t-1)+surv_prm(t-1)); // to calculate expenses for secondary
lives

    result = (exp_ren_perc_prem /100. * prem_curr_if/12. * temp
              + exp_ren_res / 1200. * reserve_bef_ret(t) *
free_inv_prop_t[proj_year(t)])*index;
}

double margin = 0.;
if(margin_add=="Y")
    margin = margin_exp_ren_pc;

return result * (1 + margin / 100.);

```

**6.1.1.3.1.248 expense\_ren\_perc\_bef\_ret\_no\_inv**

```

if (t <= commence_period_w || t > maturity_period_w)
    return 0.0;

// % of premium expenses are assumed to be
// payable at the same frequency as the premium

double result = 0.0;

double index = expense_inflation(t);

if (eq(company, "hasne")){

```

```

if (margin_add_discount == "Y" && eq(submodel, "TERM"))
    result = (exp_ren_perc_prem / 100. * term->prem_gross_no_scen(t))*index;

result = (exp_ren_perc_prem / 100. * premium_gross(t))*index;
}
else
{
    double temp = surv_act_prm(t-1);

    // ltc pups should not have expenses if ((eq(life->ben_class,"dd") || eq(life-
>ben_class,"ltc")) && surv(t-1)>0.)
    if (eq( ben_class,"dd") && surv_prm(t-1)>0.)
        temp = (term->surv_2(t-1)+surv_prm(t-1)); // to calculate expenses for secondary
lives

    result = (exp_ren_perc_prem /100. * prem_curr_if/12. * temp)*index;
}

double margin = 0.;
if(margin_add=="Y")
    margin = margin_exp_ren_pc;

return result * (1 + margin / 100.);

```

#### 6.1.1.3.1.249 comm\_claw\_spv

```

/*The amount of this cashflow in period t is equal to the accumulated commission
amount, in respect of currently inforce policies, at the end of period t-1 multiplied
by the total of lapsing policies and policies becoming pup in period t-1.
Initial commission is clawed back only after the policy has been in force
for greater than one period.
*/

if (t-1 <= commence_period_w)
    return 0.0;

if (prem_term == 1) // avoid clawback when single premium policy "gets paid-up"
    return 0.0;

// Clawback is not calculated once a policy has become paid-up.
if (inlist(paid_up,"Y,G"))
    return 0.0;

if (amala_pikuach_0 == 0.0 && amala_pikuach_1 == 0.0)
    return 0.0;

// temp = policies lapsing or becoming paid-up in period
double temp = surv_act_prm(t-2) * (lapse_rate_act_prm_dep(t-1) + pup_rate_prm_dep(t-1));

double result = 0.0;

if (t+elapsed_months-1 +elapsed_months_extra<= 72)
    result = comm_claw_prpn_spv[t+elapsed_months+elapsed_months_extra-1] / 100.* comm_spv_cum(t-
1);

if (surv_prm(commence_period_w) == 0)

```

```
return 0;
```

```
return result * temp / surv_prm(commence_period_w);
```

#### 6.1.1.3.1.250 comm\_spv\_cum

```
if (t <= commence_period_w || t > maturity_period_w)
    return 0.0;
```

```
if(surv_act_prm(t-1)<0.0000001)
    return comm_spv_cum(t-1);
```

```
return comm_spv_cum(t-1)
    + comm_supervisor(t) * surv_prm(commence_period_w) / surv_act_prm(t-1);
```

#### 6.1.1.3.1.251 comm\_supervisor

```
if (t <= commence_period_w || t > maturity_period_w)
    return 0.0;
```

```
int extra_yrs = xint(elapsed_months_extra/12.+0.5);
```

```
double prem_base = pol_fee(t) + premium(t);
```

```
double temp = 1.0; // used to make commission in 1st year all paid in 1st month
```

```
if (eq(submodel,"TRAD")) {
    prem_base= premium(t)/(1.+ tat_shnatiut_rate/100.);
    temp=0.0;      // for TRAD supervisor commission is only paid in year 1
}
```

```
if (xint(pol_year(t)+extra_yrs)==1 && prod_yr_w >= 2004) {
    if (xint(pol_month(t))==1)
        temp = prem_freq;
    else
        temp = 0.;
} //end if (xint(
```

```
return temp * (prem_base + premium_extra(t))
    * basic_perc(t)*comm_spvisor[xint(pol_year(t))+extra_yrs]/ 100.;
```

#### 6.1.1.3.1.252 reserve\_increase

```
if (t <= commence_period_w || t > maturity_period_ann)
    return 0.0;
```

```
if (t==1)      // add in initial reserve difference
    return reserve(t) - reserve(t-1) + reserve_opening_difference ;
```

```
return reserve(t) - reserve(t-1);
```

#### 6.1.1.3.1.253 reserve\_increase\_bef\_ret

```
if (t <= commence_period_w || t > maturity_period_ann || eq(paid_up,"G"))
    return 0.0;
```

```
if(mult_age_ind == 1){
```

```

if(t == 1){
    if(ann_index_map.count(retirement_age_lookup(t)) != 0)
        if(t == sm_annuity[ann_index_map[retirement_age_lookup(t)]]->maturity_period_w)
            if(sm_annuity[ann_index_map[retirement_age_lookup(t)]]->reserve_basic(t) !=
0)
                return sm_annuity[ann_index_map[retirement_age_lookup(t)]]-
>reserve_basic(t) + reserve_bef_ret(t) - reserve_bef_ret(t-1) + reserve_opening_difference;
            return reserve_bef_ret(t) - reserve_bef_ret(t-1) + reserve_opening_difference;// add in
initial reserve difference
        }

```

```

if(ann_index_map.count(retirement_age_lookup(t)) != 0)
    if(t == sm_annuity[ann_index_map[retirement_age_lookup(t)]]->maturity_period_w)
        if(sm_annuity[ann_index_map[retirement_age_lookup(t)]]->reserve_basic(t) != 0)
            return sm_annuity[ann_index_map[retirement_age_lookup(t)]]->reserve_basic(t)
+ reserve_bef_ret(t) - reserve_bef_ret(t-1);

```

```

return reserve_bef_ret(t) - reserve_bef_ret(t-1);}

```

```

return reserve_increase(t);

```

#### 6.1.1.3.1.254 reserve\_increase\_pv

```

if (t <= commence_period_w || t > maturity_period_ann)
    return 0.0;

```

```

if (mult_age_ind != 1){

    int proj_yr = xint(proj_year(t+1));
    if(eq(projection_type_int, "Rollup"))
        proj_yr = xint(proj_year_rollup(t+1));

    if ( t >= maturity_period_w)
        return (reserve_increase_pv(t+1) + reserve_increase(t+1))*
ann_v_month_t[proj_yr];
    return (reserve_increase_pv(t+1) + reserve_increase(t+1))* v_month_t[proj_yr];
}

```

```

return reserve_increase_pv_active(t)
    + reserve_increase_pv_deferred(t)
    + reserve_increase_pv_inpay(t);

```

#### 6.1.1.3.1.255 reserve\_total\_increase\_pv

```

if (t < commence_period_w || t > maturity_period_ann)
    return 0.0;

```

```

return reserve_increase_pv(t) - reserve_re_increase_pv(t);

```

#### 6.1.1.3.1.256 cashflow

```

if (t <= commence_period_w || t > maturity_period_ann)

```



```
return 0.0;
```

```
return cashflow_b(t) + cashflow_e(t);
```

#### **6.1.1.3.1.257 cashflow\_b**

```
if (t <= commence_period_w || t > maturity_period_ann)
    return 0.0;
```

```
return income_b(t) - outgo_b(t);
```

#### **6.1.1.3.1.258 cashflow\_e**

```
if (t <= commence_period_w || t > maturity_period_ann)
    return 0.0;
```

```
return income_e(t) - outgo_e(t);
```

#### **6.1.1.3.1.259 cashflow\_profit**

```
if (t <= commence_period_w || t > maturity_period_ann)
    return 0.0;
```

```
return cashflow_b(t) + cashflow_e(t);
```

#### **6.1.1.3.1.260 cashflow\_profit\_bef\_ret**

```
if (t <= commence_period_w || t > maturity_period_ann)
    return 0.0;
```

```
return cashflow_b_bef_ret(t) + cashflow_e(t);
```

#### **6.1.1.3.1.261 profit\_book\_active\_vif**

```
if (t <= commence_period_w || t > maturity_period_ann || eq(paid_up,"G"))
    return 0.0;
```

```
if(mult_age_ind == 1)
    return profit_book_vif_bef_ret(t);
```

```
return (cashflow(t) + investment_income(t)
        - reserve_increase(t) + reserve_re_increase(t));
```

#### **6.1.1.3.1.262 profit\_book\_bef\_ret**

```
if (t <= commence_period_w || t > maturity_period_ann)
    return 0.0;
```

```
if (mult_age_ind!= 1)
    return 0.0;
```

```
return cashflow_b_bef_ret(t)
        + cashflow_e(t)
        + investment_income_bef_ret(t)
        - reserve_increase_bef_ret(t)
        + reserve_re_increase(t)
        + dac_increase(t);
```

#### **6.1.1.3.1.263 profit\_book\_vif**

```
if (t <= commence_period_w || t > maturity_period_ann)
```

```

    return 0.0;

    if(mult_age_ind == 1)
        return profit_book_vif_post_ret(t) + profit_book_vif_bef_ret(t);

    return (cashflow(t) + investment_income(t)
            - reserve_increase(t) + reserve_re_increase(t) );

```

#### 6.1.1.3.1.264 profit\_book\_vif\_bef\_ret

```

if (t <= commence_period_w || t > maturity_period_ann)
    return 0.0;

return (cashflow_profit_bef_ret(t) + investment_income_bef_ret(t)
        - reserve_increase_bef_ret(t) + reserve_re_increase(t));

```

#### 6.1.1.3.1.265 profit\_book\_vif\_gross

```

if (t <= commence_period_w || t > maturity_period_ann)
    return 0.0;

return cashflow(t)
    + investment_income(t)
    - reserve_increase(t)
    + cashflow_re_b(t)
    + cashflow_re_e(t);

```

#### 6.1.1.3.1.266 profit\_book\_vif\_gross\_pv

```

if (t < commence_period_w || t >= maturity_period_ann)
    return 0.0;

if (mult_age_ind != 1){

    int proj_yr = xint(proj_year(t+1));
    if(eq(projection_type_int, "Rollup"))
        proj_yr = xint(proj_year_rollup(t+1));

    if ( t >= maturity_period_w)
        return (profit_book_vif_gross(t+1) + profit_book_vif_gross_pv(t+1)) *
ann_v_month_t[proj_yr];

    return (profit_book_vif_gross(t+1) + profit_book_vif_gross_pv(t+1)) * v_month_t[proj_yr];
}

return profit_book_vif_gross_pv_active(t)
    + profit_book_vif_pv_deferred(t)
    + profit_book_vif_pv_inpay(t);

```

#### 6.1.1.3.1.267 profit\_book\_vif\_post\_ret

```

if (t <= commence_period_w || t > maturity_period_ann)
    return 0.0;

return sm_annuity->profit_book_vif_post_ret(t);

```

**6.1.1.3.1.268 profit\_book\_vif\_pv\_pos**

```

if (t < commence_period_w || t >= maturity_period_ann || inlist(paid_up, "C,G"))
    return 0.0;

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

if (mult_age_ind !=1){

    if (t >= maturity_period_w)
        return 0.0;

    return max(profit_book_vif_pv(t) - res_ann_deficiency(t), 0.0);

}

double new_ret = 0.0;

if(ann_index_map.count(retirement_age_lookup(t)) != 0){

    if(t == sm_annuity[ann_index_map[retirement_age_lookup(t)]]->maturity_period_w)
        new_ret = sm_annuity[ann_index_map[retirement_age_lookup(t)]]-
>profit_book_vif_post_ret_pv(t);

}

return max(profit_book_vif_pv_active(t) + profit_book_vif_pv_deferred(t) - new_ret -
res_ann_deficiency(t), 0.0);

```

**6.1.1.3.1.269 profit\_gross\_vif**

```

if (t <= commence_period_w || t > maturity_period_ann)
    return 0.0;

return profit_book_vif_gross(t)
    * (1 - tax_rate/ 100.);

```

**6.1.1.3.1.270 profit\_gross\_vif\_pv**

```

if (t < commence_period_w || t >= maturity_period_ann)
    return 0.0;

if (mult_age_ind != 1){

    int proj_yr = xint(proj_year(t+1));
    if(eq(projection_type_int, "Rollup"))
        proj_yr = xint(proj_year_rollup(t+1));

    if ( t >= maturity_period_w)
        return (profit_gross_vif(t+1) + profit_gross_vif_pv(t+1))* ann_v_month_t[proj_yr];
    return (profit_gross_vif(t+1) + profit_gross_vif_pv(t+1))* v_month_t[proj_yr];

}

return profit_gross_vif_pv_active(t)
    + profit_net_vif_pv_deferred(t)
    + profit_net_vif_pv_inpay(t);

```

**6.1.1.3.1.271 profit\_net\_vif**

```
if (t <= commence_period_w || t > maturity_period_ann)
    return 0.0;
```

```
//check why need to add these
double temp = bonus_shimur(t);
temp = cal_month(t);
temp = cal_year(t);
temp = cashflow_profit(t);
temp = claims_annuity_gt(t);
temp = claims_annuity_nogt(t);
temp = claims_insurance(t);
temp = comm_profit(t);
temp = comm_reg(t);
temp = coverage_units(t);
temp = coverage_units_re(t);
temp = expense_clm(t);
temp = expense_init(t);
temp = expense_ren(t);
temp = int_cred(t);
temp = management_fees_fixed_ann(t);
temp = management_fees_var_ann(t);
temp = mgt_fees_prem(t);
temp = prem_insurance(t);
temp = prem_savings(t);
temp = units_for_takeup(t);
temp = premium_gross_fix(t);
temp = premium_gross_var(t);
temp = claims_lrc_q1(t);
temp = claims_lrc_q2(t);
temp = claims_lrc_q3(t);
temp = claims_lrc_q4(t);
temp = claims_lrc_yr2plus(t);
temp = claims_re_lrc_q1(t);
temp = claims_re_lrc_q2(t);
temp = claims_re_lrc_q3(t);
temp = claims_re_lrc_q4(t);
temp = claims_re_lrc_yr2plus(t);
temp = expense_claims_lrc_q1(t);
temp = expense_claims_lrc_q2(t);
temp = expense_claims_lrc_q3(t);
temp = expense_claims_lrc_q4(t);
temp = expense_claims_lrc_yr2plus(t);
temp = claims_maturity_ret(t);
temp = expense_total_pre_ret_no_inv(t);
temp = expense_ren_perc_bef_ret_no_inv(t);
temp = expense_investment_post_ret(t);
temp = expense_investment_bef_ret(t);
temp = term->surv_2(t);
temp = term->surv_2_no_dec(t);
temp = cashflow_b_post_ret(t);
temp = units_b_bef(t);
```

```

temp = units_for_takeup(t);
temp = comm_hekef_net(t);
temp = riskadj_gross_rel_q1(t);
temp = riskadj_gross_rel_q2(t);
temp = riskadj_gross_rel_q3(t);
temp = riskadj_gross_rel_q4(t);
temp = riskadj_gross_rel_yr2plus(t);
temp = riskadj_re_rel_q1(t);
temp = riskadj_re_rel_q2(t);
temp = riskadj_re_rel_q3(t);
temp = riskadj_re_rel_q4(t);
temp = riskadj_re_rel_yr2plus(t);
temp = surv_cnt(t);
temp = reserve_claims_retent(t);
temp = claims_retent(t);
temp = sum_insured_occ_retent(t);
temp = sum_insured_occ_gross(t);
temp = management_fee_variable(t);
temp = units_b(t);
temp = reserve_manual;
temp = riskadj_net(t);
temp = policy_surr(t);

```

```

return (profit_book_vif(t))
      * (1- tax_rate/ 100.);

```

#### 6.1.1.3.1.272 profit\_vif\_net\_bef\_ret

```

if (t <= commence_period_w || t > maturity_period_ann)
    return 0.0;

```

```

return (profit_book_vif_bef_ret(t))
      * (1- tax_rate / 100.);

```

#### 6.1.1.3.1.273 comm\_dac

```

if(submodel == "ANN")
    return NO_AVG;

```

```

if (t <= commence_period_w || t > mat_period_original)
    return 0.0;

```

```

// theoretical renewal commission, even before comm_renewal_year
double renew_reduc=0.0;

```

```

renew_reduc = premium_gross(t) *
    ( (comm_regular_pc[15] + comm_ren_perc_prem +comm_nihul_rate[16])* basic_perc(t) +
      comm_ren_perc_sav * (1. - basic_perc(t)) ) /100.
    * (1+vat/100.);

```

```

return max(0, comm_nihul(t) + comm_regular(t)
    - renew_reduc + comm_prize(t));

```

#### 6.1.1.3.1.274 dac\_book

```

if(paid_up=="Y" && t>0)
    return NO_AVG;

```

```

if (eq(dac_type_temp,"il_dac"))
    return dac_il_book(t);

if (eq(dac_type_temp,"zillmer")){
    if (zillmer_book(0) > 0.)
        return zillmer_book(t) * dac_tax_inforce/zillmer_book(0);
    else if (prem_term_original > elapsed_months)
        return max(0. , dac_tax_inforce * (prem_term_original - elapsed_months - t) /
(prem_term_original - elapsed_months) );
    }

return 0.0;

```

#### 6.1.1.3.1.275 dac\_il\_book

```

if (t <= commence_period_w || t + elapsed_months + elapsed_months_extra >= dac_amort_per ||
!eq(dac_type_temp,"il_dac"))
    return 0.0;

// If rollup projection then adjust DAC until valuation date
if (t<=0 && eq(projection_type,"Rollup")) {
    if (surv_act_prm(t-1)<0.000001)
        return 0.0;
    return (dac_il_book(t-1) + (exp_dac(t)+comm_dac(t))*dac_book_adj_factor/100. )
        * surv_act_prm(t)/surv_act_prm(t-1)
        * (1.-1./(dac_amort_per - elapsed_months - elapsed_months_extra - t+1));
}

// If start of projection then use starting DAC balance
if (t==0 && eq(projection_type,"Valn"))
    return dac_book_inforce;

if (surv_act_prm(t-1)<0.000001)
    return 0.0;

return (dac_il_book(t-1) + exp_dac(t)+comm_dac(t))* surv_act_prm(t)/surv_act_prm(t-1)
    * (1.-1./(dac_amort_per - elapsed_months - elapsed_months_extra - t+1));

```

#### 6.1.1.3.1.276 dac\_il\_tax

```

if (t <= commence_period_w || t >= mat_period_original || surv_act_prm(t) < 0.000001)
    return 0.0;

double result = 0.0;
double dac_in_month = 0.0;

// option : "immediate 1 year amortisation" for expenses in 1st cal.year

for (int i = 0; i <= min(t + elapsed_months + elapsed_months_extra - 1,
                        dac_amort_per_tax-12+xint(cal_month(t))-1); i++) {
    if(surv_act_prm(t-i-1)<0.000001) dac_in_month = 0;
    else {
        // expense paid during the 1st cal. year
        if (xint(cal_year(t-i))==xint(cal_year(commence_period_w+1))) {

            // dac calculated during the 1st cal year

```

```

        if(xint(cal_year(t))==xint(cal_year(commence_period_w+1)) && surv_act_prm(t-
i-1) != 0 && dac_amort_per_tax != 0)
            dac_in_month = (exp_dac(t-i)+comm_dac(t-i))*surv_act_prm(t)/surv_act_prm(t-
i-1)
            *( dac_amort_per_tax - 12) / (dac_amort_per_tax);

        else if (surv_act_prm(t-i-1) != 0 && dac_amort_per_tax != 0 &&
dac_amort_per_tax-12 != 0)// dac calc. after the 1st cal year
            dac_in_month = (exp_dac(t-i)+comm_dac(t-i))*surv_act_prm(t)/surv_act_prm(t-
i-1)
            *( dac_amort_per_tax - 12) / (dac_amort_per_tax)
            *( 1. - (12*(xint(cal_year(t))-
xint(cal_year(commence_period_w+1))-1)
            +cal_month(t)) /(dac_amort_per_tax-12));
    }
    else if (surv_act_prm(t-i-1) != 0 && dac_amort_per_tax + 1 - xint(cal_month(t-i)) !=
0)// expense paid after the 1st cal. year
        dac_in_month = (exp_dac(t-i)+comm_dac(t-i))*surv_act_prm(t)/surv_act_prm(t-i-
1)
        *(dac_amort_per_tax + 1 - xint(cal_month(t-i))-(i+1))/
        (dac_amort_per_tax + 1 - xint(cal_month(t-i)));
    }
    result += dac_in_month;
}

return result;

```

#### 6.1.1.3.1.277 dac\_increase

```

if (t <= commence_period_w || t > mat_period_original)
    return 0.0;

```

```

return dac_book(t) - dac_book(t-1);

```

#### 6.1.1.3.1.278 dac\_tax

```

if(paid_up=="Y")
    return NO_AVG;

if (eq(dac_type_temp,"il_dac")) {
    if (surv_act_prm(0)<0.000001)
        return 0.0;

    if (eq(projection_type,"Valn")) {
        if(dac_il_tax(0)>SMALL_DOUBLE && elapsed_months_extra==0) // if elapsed_months_extra
> 0 then dac-tax formula does not work because commencement period is based on tarif-date
            return dac_il_tax(t) * dac_tax_inforce/dac_il_tax(0);
        else
            return dac_tax_inforce * max(24-t,0)/24. *surv_act_prm(t)/surv_act_prm(0);
    }
    else { // rollup
        return dac_il_tax(t) *dac_tax_adj_factor/100. ;
    }
}

if (eq(dac_type_temp,"zillmer")) {
    if(zillmer_tax(0)>SMALL_DOUBLE)

```

```

        return zillmer_tax(t) * dac_tax_inforce/zillmer_tax(0);
    else {
        if (surv_act_prm(0)<0.000001)
            return 0.0;

        return dac_tax_inforce * max(36-t,0)/36. *surv_act_prm(t)/surv_act_prm(0);
    }

return 0.0;

```

#### 6.1.1.3.1.279 dac\_tax\_increase

```

if (t <= commence_period_w || t > mat_period_original)
    return 0.0;

return dac_tax(t) - dac_tax(t-1);

```

#### 6.1.1.3.1.280 exp\_dac

```

if (t <= commence_period_w || t > mat_period_original)
    return 0.0;

return (expense_initial_fix(t) + expense_initial_perc(t)) * exp_dac_perc / 100.
    + comm_supervisor(t);

```

#### 6.1.1.3.1.281 zillmer\_book

```

if (t <= commence_period_w || t >= mat_period_original)
    return 0.0;

if (submodel == "TERM")
    return NO_AVG;

if (submodel=="TRAD")
    return trad->zillmer_book(t);

if (eq(dac_type_temp,"zillmer")){

    if(xint(pol_year(t))<=10) {
        row_char = policy_type+"_g";
        double rate = zillmer_pr_tbl;
        return rate /100.* 12
            * premium(t)* min(1.,basic_perc(t));
    }
    else
        return 0.0;
}

return 0.0;

```

#### 6.1.1.3.1.282 zillmer\_tax

```

if (submodel=="TERM")
    return NO_AVG;

if (submodel=="TRAD")
    return trad->zillmer_tax(t);

if (t <= commence_period_w || t >= maturity_period_w)

```



```

    return 0.0;

if (eq(dac_type_temp,"zillmer")){

    if(xint(pol_year(t))<=10) {
        row_char = policy_type+"_t";
        double rate = zillmer_pr_tbl;
        return rate /100.* 12
            * premium(t)* min(1.,basic_perc(t));
    }
    else
        return 0.0;
}
return 0.0;

```

#### 6.1.1.3.1.283      **bor\_acc**

```

if (t < 0 || t > maturity_period_w)
    return 0.0;

if (mgt_fee_variable == 0 || submodel != "UNIT" || par_nonpar != "P")
    return 0;

if(t==0 && mgt_deficit_perc < 0. && paid_up == "N"){

    //Initial bor as % of reserve

    return units_e(t) * mgt_deficit_perc *(-1.) * mgt_fee_variable/100.; //Bor is managed as
    positive number

}

if (t== 0)
    return 0.0;

if(har_acc(t) > 0)
    return 0; //No bor if there is har

double bor = 0.0;

bor = bor_acc(t-1)
    * (1. - decrement_rate_unit(t-1));

if (net_interest_rate(t) < 0.0){

    double new_bor = - (sm_accum->units_b(t) + sm_saving->units_b(t)) * net_interest_rate(t) *
    mgt_fee_variable/100.;

    if (har_return(t) > 0)
        new_bor = max(new_bor - har_return(t), 0);

    bor = bor
        + new_bor; //Addition to bor

```

```

}

if (net_interest_rate(t) > 0.0)
    bor = bor + bor_return(t); //Bor returned

//Add bor passed to pup

//Remove bor passed to pup

    bor = bor
        - (
            bor_acc(t-1)
            * pup_rate_bal_dep(t-1)
            * surv_per_ret(t-1)
        );

return max(bor, 0.0); //Shouldn't really be possible for bor to be negative, but just in case

```

#### 6.1.1.3.1.284 bor\_acc\_pup

```

if (t < 0 || t > maturity_period_w)
    return 0.0;

if (mgt_fee_variable == 0 || submodel != "UNIT" || par_nonpar != "P")
    return 0;

if(t==0 && mgt_deficit_perc < 0. && paid_up == "Y"){

    //Initial bor as % of reserve

    return units_e(t) * mgt_deficit_perc * (-1.) * mgt_fee_variable/100.; //Bor is managed as
positive number

}

if (t== 0)
    return 0.0;

if(har_acc_pup(t) > 0)
    return 0; //No bor if there is har

double bor = 0.0;

bor = bor_acc_pup(t-1)
    * (1. - decrement_rate_unit_pup(t-1));

if (net_interest_rate(t) < 0.0){

```

```

        double new_bor = - (sm_acc_pup->units_b(t) + sm_saving_pup->units_b(t)) *
net_interest_rate(t) * mgt_fee_variable/100.;

        if (har_return_pup(t) > 0)
            new_bor = max(new_bor - har_return_pup(t), 0);

        bor = bor
            + new_bor; //Addition to bor

    }

    if (net_interest_rate(t) > 0.0)
        bor = bor + bor_return_pup(t); //Bor returned

//Add bor passed to pup

    if (t == 1 && gross_up_historic=="N")
        bor = bor + (bor_acc(t-1) * pup_rate_bal_dep(t) * surv_per_ret(t-1));

    else
        bor = bor + (bor_acc(t-1) * pup_rate_bal_dep(t-1) * surv_per_ret(t-1));

    return max(bor, 0.0); //Shouldn't really be possible for bor to be negative, but just in case

```

#### 6.1.1.3.1.285 bor\_har\_retire

```

if (t <= commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if (submodel == "TERM" || par_nonpar == "N")
    return 0.0;

if (submodel == "ANN"){
    if (t==0)
        return mgt_deficit_perc * resinforce * mgt_fee_variable; //Is this correct? Probably
it doesn't matter as long as it's consistent

        return 0.0; //No new annuities coming in after t=0
    }

if (t == 0)
    return 0.0;

if (retirement_prop(t-1) == 0)
    return 0.0; //Not a retirement period

if (submodel == "TRAD"){

    double bor = (trad->bor_acc(t-1) * (1. - lapse_rate_act_bal(t-1))
                  + trad->bor_acc_pup(t-1) * (1. - lapse_rate_pup_bal(t-1))
                  + trad->bor_acc_mat(t-1)
                  )
                  * (1. - death_rate(t-1))

```

```

        * retirement_prop(t-1)
        * (-1.);

double har = 0.0;

if (cal_month(t) != 1)
    har = (trad->har_acc(t-1) * (1. - lapse_rate_act_bal(t-1))
          + trad->har_acc_pup(t-1) * (1. - lapse_rate_pup_bal(t-1))
          + trad->har_acc_mat(t-1)
          )
          * (1. - death_rate(t-1))
          * retirement_prop(t-1);

return bor + har;
}

```

```

double bor = (bor_acc(t-1) * (1 - lapse_rate_act_bal(t-1))
              + bor_acc_pup(t-1) * (1 - lapse_rate_pup_bal(t-1))
              )
              * (1 - death_rate(t-1))
              * retirement_prop(t-1)
              * (-1.); //Bor is accumulated as a positive

```

```
double har = 0.0;
```

```

if (cal_month(t) != 1) //If it is January, it won't pass to annuity anyway
    har = (
        har_acc(t-1) * (1 - lapse_rate_act_bal(t-1))
        + har_acc_pup(t-1) * (1 - lapse_rate_pup_bal(t-1))
        )
        * (1 - death_rate(t-1))
        * retirement_prop(t-1) ;

```

```
return bor + har;
```

#### 6.1.1.3.1.286 bor\_return

```

if (t <= 0 || t > maturity_period_w || submodel != "UNIT" || par_nonpar != "P")
    return 0.0;

```

```
double mgt_fee_pos = 0.0; //Management fees available for return
```

```

if (net_interest_rate(t) < 0.0 || bor_acc(t-1) == 0.0)
    return 0.0;

```

```
mgt_fee_pos = -mgt_var_no_bor(t); //Management fees from current month available to repay bor
```

```

return max(mgt_fee_pos,
            bor_acc(t-1) * (-1.) * (1.- decrement_rate_unit(t-1))); //Cannot return more
than outstanding bor

```

**6.1.1.3.1.287 bor\_return\_pup**

```

if (t <= 0 || t > maturity_period_w || submodel != "UNIT" || par_nonpar != "P")
    return 0.0;

double mgt_fee_pos = 0.0; //Management fees available for return

if (net_interest_rate(t) < 0.0 || bor_acc_pup(t-1) == 0.0)
    return 0.0;

mgt_fee_pos = -mgt_var_no_bor_pup(t); //Management fees from current month available to repay bor

return max(mgt_fee_pos-new_pup_har_ret(t),
           bor_acc_pup(t-1) * (-1.) * (1.- decrement_rate_unit_pup(t-1))); //Cannot
return more than outstanding bor

```

**6.1.1.3.1.288 har\_acc**

```

if (t < 0 || t > maturity_period_w || mgt_fee_variable == 0 || submodel != "UNIT" || par_nonpar !=
"P")
    return 0.0;

if(t==0 && paid_up == "N"){

    if(life->cal_month(t) == 12. || life->mgt_deficit_perc < 0)
        return 0.0; //If year-end, no accumulation. If there is bor, no accumulation

    return (sm_accum->units_e(t) + sm_saving->units_e(t) ) * life->mgt_deficit_perc *
mgt_fee_variable/100.;

}

double har = 0;

if(cal_month(t) > 1)
    har = har_acc(t-1)
        * (1.-decrement_rate_unit(t-1));

har = har + management_fee_variable(t); //management fees paid this month

//har = har - har_return(t); //Deduct management fees returned

//Remove har passed to pup

    if(cal_month(t) > 1)
        har = har
            - har_acc(t-1)
            * pup_rate_bal_dep(t-1)
            * surv_per_ret(t-1);

return max(har, 0);

```

**6.1.1.3.1.289 har\_acc\_pup**

```

if (t < 0 || t > maturity_period_w || mgt_fee_variable == 0 || submodel != "UNIT" || par_nonpar !=
"P" || bor_acc_pup(t-1)>0)
    return 0.0;

```

```

if(t==0 && paid_up == "Y"){
    if(life->cal_month(t) == 12. || life->mgt_deficit_perc < 0)
        return 0.0; //If year-end, no accumulation. If there is bor, no accumulation

    return (sm_acc_pup->units_e(t) + sm_saving_pup->units_e(t) ) * life->mgt_deficit_perc *
mgt_fee_variable/100.;
}

if(t == 1 && paid_up == "N"){
    if (bor_acc(t-1) * pup_rate_bal_dep(t-1) * surv_per_ret(t-1) > 0)
        return 0.0;
}

double har = 0;

if(cal_month(t) > 1)
    har = har_acc_pup(t-1)
        * (1.-decrement_rate_unit_pup(t-1));

har = har + management_fee_variable_pup(t); //management fees paid this month

//Remove har passed to pup

if(cal_month(t) > 1)
    har = har
        + har_acc(t-1)
        * pup_rate_bal_dep(t-1)
        * surv_per_ret(t-1);

return max(har, 0);

```

#### 6.1.1.3.1.290 har\_return

```

if (t <= 0 || t > maturity_period_w || mgt_fee_variable == 0 || submodel != "UNIT" || par_nonpar !=
"P")
    return 0.0;

if (har_acc(t-1) == 0.0)
    return 0; //Nothing collected to return

if (net_interest_rate(t) > 0.0)
    return 0.0; //No need to return if interest is positive

if (cal_month(t) == 1)
    return 0.0; //Do not carry over to new year

double har_ret = net_interest_rate(t) * mgt_fee_variable/100.
                * (sm_accum->units_b(t) + sm_saving->units_b(t))
                * (-1.); // Management fees that should be returned

```

```

return min(har_ret,
            har_acc (t-1)
            * (1 - decrement_rate_unit(t-1))); //Cannot return more than accumulated har
for that year

```

#### 6.1.1.3.1.291 har\_return\_pup

```

if (t <= 0 || t > maturity_period_w || mgt_fee_variable == 0 || submodel != "UNIT" || par_nonpar !=
"P")
    return 0.0;

```

```

if (har_acc_pup(t-1) == 0.0)
    return 0; //Nothing collected to return

```

```

if (net_interest_rate(t) > 0.0)
    return 0.0; //No need to return if interest is positive

```

```

if (cal_month(t) == 1)
    return 0.0; //Do not carry over to new year

```

```

double har_ret = net_interest_rate(t) * mgt_fee_variable/100.
                * (sm_acc_pup->units_b(t) + sm_saving_pup->units_b(t))
                * (-1.); // Management fees that should be returned

```

```

return min(har_ret,
            har_acc_pup (t-1)
            * (1 - decrement_rate_unit_pup(t-1))); //Cannot return more than accumulated
har for that year

```

#### 6.1.1.3.1.292 manage\_fees\_fixe\_active\_pv

```

if (t < commence_period_w || t >= maturity_period_ann)
    return 0.0;

```

```

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

```

```

return (management_fees_fixed_active(t+1)
        + manage_fees_fixe_active_pv(t+1) )
        * v_month_t[proj_yr];

```

#### 6.1.1.3.1.293 manage\_fees\_fixed\_ann\_pv

```

return manage_fees_fixed_ann_pv_def(t)
        + manage_fees_fixed_ann_pv_ip(t);

```

#### 6.1.1.3.1.294 manage\_fees\_fixed\_ann\_pv\_def

```

if (t < commence_period_w || t > maturity_period_ann || mult_age_ind !=1)
    return 0.0;

```

```

double new_ret = 0.0;

```

```

int proj_yr = xint(proj_year(t+1));

```

```

if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

if(ann_index_map.count(retirement_age_lookup(t)) != 0){
    if(t == sm_annuity[ann_index_map[retirement_age_lookup(t)]]->maturity_period_w)
        new_ret = sm_annuity[ann_index_map[retirement_age_lookup(t)]]->mgt_fee_fixed_pv(t);
}

return manage_fees_fixed_ann_pv_def(t+1) * v_month_t[proj_yr] + new_ret;

```

#### 6.1.1.3.1.295 manage\_fees\_fixed\_ann\_pv\_ip

```

if (t < commence_period_w || t > maturity_period_ann)
    return 0.0;

if(eq(paid_up, "G"))
    return sm_annuity->mgt_fee_fixed_pv(t);

if( mult_age_ind != 1)
    return 0.0;

double ann_in_pay = 0.0;

for (int i=0; i < sm_annuity.size(); i++){
    if(t > sm_annuity[i]->maturity_period_w)
        ann_in_pay = ann_in_pay + sm_annuity[i]->mgt_fee_fixed_pv(t);
}

return ann_in_pay;

```

#### 6.1.1.3.1.296 manage\_fees\_var\_active\_pv

```

if (t < commence_period_w || t >= maturity_period_ann)
    return 0.0;

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

return (management_fees_var_active(t+1)
        + manage_fees_var_active_pv(t+1) )
        * v_month_t[proj_yr];

```

#### 6.1.1.3.1.297 manage\_fees\_var\_ann\_pv

```

return manage_fees_var_ann_pv_def(t)
        + manage_fees_var_ann_pv_ip(t);

```

#### 6.1.1.3.1.298 manage\_fees\_var\_ann\_pv\_def

```

if (t < commence_period_w || t > maturity_period_ann || mult_age_ind !=1)
    return 0.0;

double new_ret = 0.0;

int proj_yr = xint(proj_year(t+1));

```



```

if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

if(ann_index_map.count(retirement_age_lookup(t)) != 0){
    if(t == sm_annuity[ann_index_map[retirement_age_lookup(t)]]->maturity_period_w)
        new_ret = sm_annuity[ann_index_map[retirement_age_lookup(t)]]->mgt_fee_var_pv(t);
}

return manage_fees_var_ann_pv_def(t+1) * v_month_t[proj_yr] + new_ret;

```

#### 6.1.1.3.1.299 manage\_fees\_var\_ann\_pv\_ip

```

if (t < commence_period_w || t > maturity_period_ann )
    return 0.0;

if(eq(paid_up, "G"))
    return sm_annuity->mgt_fee_var_pv(t);

if(mult_age_ind != 1)
    return 0.0;

double ann_in_pay = 0;

for (int i=0; i < sm_annuity.size(); i++){
    if(t > sm_annuity[i]->maturity_period_w)
        ann_in_pay = ann_in_pay + sm_annuity[i]->mgt_fee_var_pv(t);
}

return ann_in_pay;

```

#### 6.1.1.3.1.300 management\_fee\_variable

```

if (t <= 0 || t > maturity_period_w || mgt_fee_variable == 0 || submodel != "UNIT" || par_nonpar !=
"P")
    return 0.0;

double mgt_var = mgt_var_no_bor(t);

//if (net_interest_rate(t) > 0.0)
//    mgt_var = net_interest_rate(t)
//        * (sm_accum->units_b(t) + sm_saving->units_b(t) )
//        * mgt_fee_variable/100.; // Management fee (assuming no adjustment)

mgt_var = mgt_var + bor_return(t); //Deduct bor to be returned

mgt_var = max(mgt_var, 0);

mgt_var = mgt_var - har_return(t); //Can be negative

if (abs(mgt_var) < SMALL_DOUBLE)
    return 0.0; //Remove small rounding issues

return mgt_var;

```

**6.1.1.3.1.301 management\_fee\_variable\_pup**

```

if (t <= 0 || t > maturity_period_w || mgt_fee_variable == 0 || submodel != "UNIT" || par_nonpar != "P")
    return 0.0;

if(t == 1 && paid_up == "N"){
    if (bor_acc(t-1) * pup_rate_bal_dep(t-1) * surv_per_ret(t-1) > 0)
        return 0.0;
}

double mgt_var = mgt_var_no_bor_pup(t);

//if (net_interest_rate(t) > 0.0)
//    mgt_var = net_interest_rate(t)
//        * (sm_acc_pup->units_b(t) + sm_saving_pup->units_b(t) )
//        * mgt_fee_variable/100.; // Management fee (assuming no adjustment)

mgt_var = mgt_var + bor_return_pup(t); //Deduct bor to be returned

mgt_var = max(mgt_var, 0);

mgt_var = mgt_var - har_return_pup(t); //Can be negative

if (abs(mgt_var) < SMALL_DOUBLE)
    return 0.0; //Remove small rounding issues

return mgt_var;

```

**6.1.1.3.1.302 management\_fees\_fixed\_active**

```

if (t <= commence_period_w || t > maturity_period_w)
    return 0.0;

if (submodel == "UNIT")
    return sm_accum->management_fee_fixed(t)
        + sm_acc_pup->management_fee_fixed(t)
        + sm_saving->management_fee_fixed(t)
        + sm_saving_pup->management_fee_fixed(t);

if (submodel == "TRAD")
    return trad->mgt_fee_fix(t)
        + trad->mgt_fee_fix_mat(t)
        + trad->mgt_fee_fix_pup(t);

return 0.0;

```

**6.1.1.3.1.303 management\_fees\_fixed\_ann**

```

if (t < commence_period_w || t > maturity_period_ann)
    return 0.0;

double mgt_fee = 0.0;

for (int i=0; i < sm_annuity.size(); i++){

```

```

        if(t > sm_annuity[i]->maturity_period_w)
            mgt_fee = mgt_fee + sm_annuity[i]->mgt_fee_fixed_dth(t)
                                + sm_annuity[i]->mgt_fee_fixed_gtd(t)
                                + sm_annuity[i]->mgt_fee_fixed_jl1(t)
                                + sm_annuity[i]->mgt_fee_fixed_jl2(t)
                                + sm_annuity[i]->mgt_fee_fixed_nogt(t)
                                ;
    }

    return mgt_fee;

```

#### 6.1.1.3.1.304 management\_fees\_var\_active

```

if (t <= commence_period_w || t > maturity_period_w)
    return 0.0;

if (submodel == "UNIT")
    return sm_accum->management_fee_variable(t)
           + sm_acc_pup->management_fee_variable(t)
           + sm_saving->management_fee_variable(t)
           + sm_saving_pup->management_fee_variable(t);

if (submodel == "TRAD")
    return trad->mgt_fee_var(t)
           + trad->mgt_fee_var_mat(t)
           + trad->mgt_fee_var_pup(t);

return 0.0;

```

#### 6.1.1.3.1.305 management\_fees\_var\_ann

```

if (t < commence_period_w || t > maturity_period_ann)
    return 0.0;

double mgt_fee = 0.0;

for (int i=0; i < sm_annuity.size(); i++){

    if(t > sm_annuity[i]->maturity_period_w)
        mgt_fee = mgt_fee + sm_annuity[i]->mgt_fee_var_dth(t)
                        + sm_annuity[i]->mgt_fee_var_gtd(t)
                        + sm_annuity[i]->mgt_fee_var_jl1(t)
                        + sm_annuity[i]->mgt_fee_var_jl2(t)
                        + sm_annuity[i]->mgt_fee_var_nogt(t)
                        ;

}

return mgt_fee;

```

**6.1.1.3.1.306 net\_interest\_rate**

```

if (t <= commence_period_w || t >= maturity_period_w || submodel != "UNIT" || mgt_fee_variable == 0
|| par_nonpar == "N")
    return 0.0;

double temp_inv_rate_m = 0.0;

int proj_yr = xint(proj_year(t));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t));
proj_yr = max(proj_yr, 0);

if (margin_add_asset == "Y" && t == 1 && par_nonpar == "P")
    temp_inv_rate_m = asset_shock;
else
    temp_inv_rate_m = inv_rate_mth_t[proj_yr];

return (1+temp_inv_rate_m
        * (1- mgt_fee_fixed/1200.)
        -1.);

```

**6.1.1.3.1.307 new\_pup\_har\_ret**

```

if (t < 0 || t > maturity_period_w || mgt_fee_variable == 0 || submodel != "UNIT" || par_nonpar !=
"P")
    return 0.0;

double har = 0;

    if(cal_month(t) > 1)
        har = har_acc(t-1)
            * pup_rate_bal_dep(t-1)
            * surv_per_ret(t-1);

return max(har, 0);

```

**6.1.1.3.1.308 cashflow\_pv\_inpay**

```

if (t < commence_period_w || t > maturity_period_ann)
    return 0.0;

if (mult_age_ind != 1)
    return 0.0;

double ann_in_pay = 0;

for (int i=0; i < sm_annuity.size(); i++){

    if(t > sm_annuity[i]->maturity_period_w)
        ann_in_pay = ann_in_pay + sm_annuity[i]->cashflow_pv(t);

}

return ann_in_pay;

```

**6.1.1.3.1.309 cashflow\_pv\_inpay\_chetz**

```
if (t < commence_period_w || t > maturity_period_ann)
    return 0.0;

if (mult_age_ind != 1)
    return 0.0;

double ann_in_pay = 0;

for (int i=0; i < sm_annuity.size(); i++){

    if(t > sm_annuity[i]->maturity_period_w)
        ann_in_pay = ann_in_pay + sm_annuity[i]->cashflow_pv_ifrs(t)*(1-max_chetz) +
sm_annuity[i]->cashflow_pv_res(t)*max_chetz;

}

return ann_in_pay;
```

**6.1.1.3.1.310 cashflow\_pv\_inpay\_e**

```
if (t < commence_period_w || t > maturity_period_ann)
    return 0.0;

if (mult_age_ind != 1)
    return 0.0;

double ann_in_pay = 0;

for (int i=0; i < sm_annuity.size(); i++){

    if(t > sm_annuity[i]->maturity_period_w)
        ann_in_pay = ann_in_pay + sm_annuity[i]->cashflow_pv_e(t);

}

return ann_in_pay;
```

**6.1.1.3.1.311 claims\_annuity\_nogt\_pv\_inpay**

```
if (t < commence_period_w || t >= t_high || mult_age_ind != 1)
    return 0.0;

double ann_in_pay = 0;

for (int i=0; i < sm_annuity.size(); i++){

    if(t > sm_annuity[i]->maturity_period_w)
        ann_in_pay = ann_in_pay + sm_annuity[i]->claims_annuity_nogt_pv(t);

}

return ann_in_pay;
```

**6.1.1.3.1.312 claims\_annuity\_pv\_inpay**

```
if (t < commence_period_w || t >= t_high || mult_age_ind != 1)
    return 0.0;
```

```
double ann_in_pay = 0;

for (int i=0; i < sm_annuity.size(); i++){

    if(t > sm_annuity[i]->maturity_period_w)
        ann_in_pay = ann_in_pay + sm_annuity[i]->claims_annuity_pv(t);

}

return ann_in_pay;
```

#### 6.1.1.3.1.313 claims\_lrc\_yr2plus\_pv

```
if (t < commence_period_w || t >= maturity_period_w || eq(paid_up,"C"))
    return 0.0;

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

return (claims_lrc_yr2plus_pv(t+1) + claims_lrc_yr2plus(t+1))
        * v_month_t[proj_yr];
```

#### 6.1.1.3.1.314 claims\_pv\_not\_annuity

```
if (t < commence_period_w || t >= t_high || mult_age_ind != 1)
    return 0.0;

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

return (claims_pv_not_annuity(t+1)
        + claims_total(t+1) - claims_annuity(t+1))
        * v_month_t[proj_yr];
```

#### 6.1.1.3.1.315 claims\_rate\_per

```
if(t<=commence_period_w || t > maturity_period_w || eq(paid_up, "G"))
    return 0.0;

if (submodel == "TERM")
    return term->claims_rate_per(t);

if (submodel == "TRAD")
    return trad->claims_rate_per(t);

double maturity_rate = 0.0;
double surr_rate = 0.0;
double dth_rate = 0.0;

//Maturity
if (mult_age_ind == 1){

    maturity_rate = surv_cnt_bef_ret(t) * retirement_prop(t) * benefits_curr;

}
else
{
    //Not part of multi-age retirement
}
```

```

        if (t == maturity_period_w)
            maturity_rate = (surv_act_cnt(t-1) + surv_pup_cnt(t-1)) * benefits_curr;
    }

    //surv_val is the inforce item after the surrenders have occurred
    if(surv_per_cnt(t) > 0.0)
        surr_rate = (surv_act_cnt(t-1) * lapse_rate_act_cnt_dep(t)
                    + surv_pup_cnt(t-1) * lapse_rate_pup_cnt_dep(t))
                    * benefits_curr;

    //Death
    if(surv_cnt(t-1) > 0.0)
        dth_rate = surv_cnt(t-1) * benefits_curr * death_rate(t);

    return maturity_rate + surr_rate + dth_rate;

```

#### 6.1.1.3.1.316 expense\_pv\_inpay

```

if (t < commence_period_w || t > maturity_period_ann || mult_age_ind != 1)
    return 0.0;

double ann_in_pay = 0;

for (int i=0; i < sm_annuity.size(); i++){

    if(t > sm_annuity[i]->maturity_period_w)
        ann_in_pay = ann_in_pay + sm_annuity[i]->expense_ren_perc_post_ret_pv(t);
}

return ann_in_pay;

```

#### 6.1.1.3.1.317 investment\_income\_pv\_inpay

```

if (t < commence_period_w || t > maturity_period_ann || mult_age_ind != 1 )
    return 0.0;

double ann_in_pay = 0;

for (int i=0; i < sm_annuity.size(); i++){

    if(t > sm_annuity[i]->maturity_period_w)
        ann_in_pay = ann_in_pay + sm_annuity[i]->investment_income_pv(t);
}

return ann_in_pay;

```

#### 6.1.1.3.1.318 outgo\_pv\_inpay

```

if (t < commence_period_w || t > maturity_period_ann || mult_age_ind != 1)
    return 0.0;

double ann_in_pay = 0;

for (int i=0; i < sm_annuity.size(); i++){

    if(t > sm_annuity[i]->maturity_period_w)

```

```
        ann_in_pay = ann_in_pay + sm_annuity[i]->outgo_b_post_ret_pv(t);
    }

    return ann_in_pay;
```

#### **6.1.1.3.1.319 profit\_book\_pv\_inpay**

```
if (t < commence_period_w || t > maturity_period_ann || mult_age_ind != 1 )
    return 0.0;

double ann_in_pay = 0;

for (int i=0; i < sm_annuity.size(); i++){

    if(t > sm_annuity[i]->maturity_period_w)
        ann_in_pay = ann_in_pay + sm_annuity[i]->profit_book_pv(t);
}

return ann_in_pay;
```

#### **6.1.1.3.1.320 profit\_book\_vif\_pv\_inpay**

```
if (t < commence_period_w || t > maturity_period_ann || mult_age_ind != 1 )
    return 0.0;

double ann_in_pay = 0;

for (int i=0; i < sm_annuity.size(); i++){

    if(t > sm_annuity[i]->maturity_period_w)
        ann_in_pay = ann_in_pay + sm_annuity[i]->profit_book_vif_post_ret_pv(t);
}

return ann_in_pay;
```

#### **6.1.1.3.1.321 profit\_net\_vif\_pv\_inpay**

```
if (t < commence_period_w || t > maturity_period_ann || mult_age_ind != 1 )
    return 0.0;

double ann_in_pay = 0;

for (int i=0; i < sm_annuity.size(); i++){

    if(t > sm_annuity[i]->maturity_period_w)
        ann_in_pay = ann_in_pay + sm_annuity[i]->profit_net_vif_post_ret_pv(t);
}

return ann_in_pay;
```

#### **6.1.1.3.1.322 reserve\_increase\_pv\_inpay**

```
if (t < commence_period_w || t > maturity_period_ann || mult_age_ind != 1 )
    return 0.0;

double ann_in_pay = 0;

for (int i=0; i < sm_annuity.size(); i++){
```



```

        if(t > sm_annuity[i]->maturity_period_w)
            ann_in_pay = ann_in_pay + sm_annuity[i]->reserve_increase_pv(t);
    }

    return ann_in_pay;

```

#### 6.1.1.3.1.323 comm\_clawback\_pv

```

if (t < commence_period_w || t >= maturity_period_w)
    return 0.0;

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

return comm_clawback_pv (t+1) * v_month_t[proj_yr]
    + comm_clawback(t+1);

```

#### 6.1.1.3.1.324 comm\_reg\_pv

```

if (t < commence_period_w || t >= maturity_period_w)
    return 0.0;

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

return comm_reg_pv(t+1) * v_month_t[proj_yr]
    + comm_regular(t+1);

```

#### 6.1.1.3.1.325 decrement\_rate\_unit

```

if (t <= commence_period_w || t > maturity_period_w || submodel != "UNIT" || mgt_fee_variable == 0)
    return 0.0;

return 1.-
    (1.-death_rate(t))
    *(1.-prem_termination_prop(t)) //Money exits active fund at premium termination rate
    *(1.-lapse_rate_act_bal(t));

```

#### 6.1.1.3.1.326 decrement\_rate\_unit\_pup

```

if (t <= commence_period_w || t > maturity_period_w || submodel != "UNIT" || mgt_fee_variable == 0)
    return 0.0;

return 1.-
    (1.-death_rate(t))
    *(1.-retirement_prop(t)) //Money exits active fund at retirement rate
    *(1.-lapse_rate_pup_bal(t));

```

#### 6.1.1.3.1.327 duration\_denominator

```

if (t < 0 || t > maturity_period_ann)
    return 0.0;

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

return (duration_denominator(t+1)

```

```

+ surv_cnt(t+1) )
* v_month_t[proj_yr];

```

#### 6.1.1.3.1.328 duration\_numerator

```

if (t < 0 || t > maturity_period_ann)
    return 0.0;

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

return (duration_numerator(t+1)
        + (t+1) * surv_cnt(t+1) )
        * v_month_t[proj_yr];

```

#### 6.1.1.3.1.329 surv\_act\_bal

```

if (t < commence_period_w || t >= maturity_period_w)
    return NO_AVG;

if (submodel=="TERM")
    return surv_cnt(t);

if (inlist(paid_up,"Y,C,G"))
    return NO_AVG;

// The proportion of full premium paying policies
// remaining in force at the end of the period.

if (surv_per_act_bal(t)<0.0) // if all decrements are greater than 1 (e.g. because of forced lapse)
    return NO_AVG;

if (decrements_apply == "N")
    return 1.0;

if (t > 0 && fabs(surv_act_bal(t-1)) < .0000001)
    return NO_AVG;

if (t == 0)
    return 1.0;

if (t > 0)
    return surv_act_bal(t-1) * surv_per_act_bal(t);

if( t < 0){
    if (gross_up_historic=="Y") {
        if (surv_per_act_bal(t+1) == 0)
            return 0;
        else
            return surv_act_bal(t+1) / surv_per_act_bal(t+1); }
    else
        return surv_act_bal(t+1);
}

return 0.0; //Unconditional return

```

**6.1.1.3.1.330    surv\_act\_bal\_bef\_ret**

```

if (t < commence_period_w || t > maturity_period_w || paid_up != "N" || submodel != "TRAD")
    return NO_AVG;

if (surv_per_act_bal(t)<0.0) // if all decrements are greater than 1 (e.g. because of forced lapse)
    return NO_AVG;

if (t == 0 || decrements_apply == "N")
    return 1.0;

if (t > 0 && surv_act_bal_bef_ret(t-1) < .0000001)
    return NO_AVG;

if (t > 0)
    return surv_act_bal(t-1) * surv_per_act_bal_bef_ret(t);

if( t < 0){
    if (gross_up_historic=="Y") {
        if (surv_per_act_bal_bef_ret(t+1) == 0)
            return 0;
        else
            return surv_act_bal_bef_ret(t+1) / surv_per_act_bal_bef_ret(t+1); }
    else
        return surv_act_bal_bef_ret(t+1);
}

return 0.0;

```

**6.1.1.3.1.331    surv\_act\_cnt**

```

if (t < commence_period_w || t >= maturity_period_w)
    return NO_AVG;

if (submodel=="TERM")
    return surv_cnt(t);

if (inlist(paid_up,"Y,C,G"))
    return NO_AVG;

// The proportion of full premium paying policies
// remaining in force at the end of the period.

if (surv_per_act_cnt(t)<0.0) // if all decrements are greater than 1 (e.g. because of forced lapse)
    return NO_AVG;

if (decrements_apply == "N")
    return 1.0;

if (t > 0 && fabs(surv_act_cnt(t-1)) < .0000001)
    return NO_AVG;

if (t == 0)
    return 1.0;

if (t > 0)
    return surv_act_cnt(t-1) * surv_per_act_cnt(t);

```

```

if( t < 0){
    if (gross_up_historic=="Y") {
        if (surv_per_act_cnt(t+1) == 0)
            return 0;
        else
            return surv_act_cnt(t+1) / surv_per_act_cnt(t+1); }
    else
        return surv_act_cnt(t+1);
}

return 0.0; //Unconditional return

```

#### 6.1.1.3.1.332 **surv\_act\_cnt\_bef\_ret**

```

if (t < commence_period_w || t > maturity_period_w || paid_up != "N" )
    return NO_AVG;

if (surv_per_act_cnt(t)<0.0) // if all decrements are greater than 1 (e.g. because of forced lapse)
    return NO_AVG;

if (decrements_apply == "N")
    return 1.0;

if (t > 0 && surv_act_cnt_bef_ret(t-1) < .0000001)
    return NO_AVG;

if (t == 0)
    return 1.0;

if (t > 0)
    return surv_act_cnt(t-1) * surv_per_act_cnt_bef_ret(t);

if( t < 0){
    if (gross_up_historic=="Y") {
        if (surv_per_act_cnt_bef_ret(t+1) == 0)
            return 0;
        else
            return surv_act_cnt_bef_ret(t+1) / surv_per_act_cnt_bef_ret(t+1); }
    else
        return surv_act_cnt_bef_ret(t+1);
}

return 0.0;

```

#### 6.1.1.3.1.333 **surv\_act\_post\_ret**

```

if (t < commence_period_w || t > maturity_period_w)
    return NO_AVG;

if(t == mat_period_original)
    return 1.;

return (1. - death_rate(t)) * (surv_per_ret(t-1));

```

#### 6.1.1.3.1.334 **surv\_act\_prm**

```

if (t < commence_period_w || t >= maturity_period_w)
    return NO_AVG;

```

```

if (submodel=="TERM")
    return surv_prm(t);

if (inlist(paid_up,"Y,C,G"))
    return NO_AVG;

// The proportion of full premium paying policies
// remaining in force at the end of the period.

if (surv_per_act_prm(t)<0.0) // if all decrements are greater than 1 (e.g. because of forced lapse)
    return NO_AVG;

if (decrements_apply == "N")
    return 1.0;

if (t > 0 && fabs(surv_act_prm(t-1)) < .0000001)
    return NO_AVG;

if (t == 0)
    return 1.0;

if (t > 0)
    return surv_act_prm(t-1) * surv_per_act_prm(t);

if( t < 0){
    if (gross_up_historic=="Y") {
        if (surv_per_act_prm(t+1) == 0)
            return 0;
        else
            return surv_act_prm(t+1) / surv_per_act_prm(t+1); }
    else
        return surv_act_prm(t+1);
}

return 0.0; //Unconditional return

```

#### 6.1.1.3.1.335 surv\_act\_prm\_bef\_ret

```

if (t < commence_period_w || t > maturity_period_w || paid_up != "N" || submodel != "TRAD")
    return NO_AVG;

if (surv_per_act_prm(t)<0.0) // if all decrements are greater than 1 (e.g. because of forced lapse)
    return NO_AVG;

if (decrements_apply == "N")
    return 1.0;

if (t > 0 && surv_act_prm_bef_ret(t-1) < .0000001)
    return NO_AVG;

if (t == 0)
    return 1.0;

if (t > 0)
    return surv_act_prm(t-1) * surv_per_act_prm_bef_ret(t);

```

```

if( t < 0){
    if (gross_up_historic=="Y") {
        if (surv_per_act_prm_bef_ret(t+1) == 0)
            return 0;
        else
            return surv_act_prm_bef_ret(t+1) / surv_per_act_prm_bef_ret(t+1); }
    else
        return surv_act_prm_bef_ret(t+1);
}

return 0.0;

```

#### 6.1.1.3.1.336 **surv\_bal**

```

if (t < commence_period_w || t > maturity_period_w)
    return NO_AVG;

if (submodel=="TERM")
    return term->surv(t);

if (decrements_apply == "N")
    return 1.0;

if (t > 0 ) {
    if (fabs(surv_bal(t-1)) < .0000001)
        // No surv in previous period
        return NO_AVG;
}

// In the experience model.
return surv_act_bal(t) + surv_pup_bal(t);

```

#### 6.1.1.3.1.337 **surv\_bal\_bef\_ret**

```

if (t < commence_period_w || t > maturity_period_w || submodel != "TRAD")
    return NO_AVG;

if (decrements_apply == "N")
    return 1.0;

if (t > 0 && surv_bal_bef_ret(t-1) < .0000001)
    return NO_AVG;

return surv_act_bal_bef_ret(t) + surv_pup_bal_bef_ret(t);

```

#### 6.1.1.3.1.338 **surv\_cnt**

```

if (t < commence_period_w || t > maturity_period_w)
    return NO_AVG;

if (submodel=="TERM")
    return term->surv(t);

if (decrements_apply == "N")
    return 1.0;

if (t > 0 ) {
    if (fabs(surv_cnt(t-1)) < .0000001)
        // No surv in previous period

```

```

        return NO_AVG;
    }

    // In the experience model.
    return surv_act_cnt(t) + surv_pup_cnt(t);

```

#### 6.1.1.3.1.339 surv\_cnt\_bef\_ret

```

if (t < commence_period_w || t > maturity_period_w)
    return NO_AVG;

if (decrements_apply == "N")
    return 1.0;

if (t > 0 && surv_cnt_bef_ret(t-1) < .0000001)
    return NO_AVG;

return surv_act_cnt_bef_ret(t) + surv_pup_cnt_bef_ret(t);

```

#### 6.1.1.3.1.340 surv\_per\_act\_bal

```

// Assume decrements are distributed uniformly within
// period of the projection.

if (t <= commence_period_w || t >= maturity_period_w)
    return NO_AVG;

if (decrements_apply == "N")
    return 1.0;

if (t > 0 && surv_bal(t-1) < .0000001)
    return NO_AVG;

if (t <= 0 && (gross_up_historic=="N"))
    return 1.0;

if (eq(submodel,"UNIT"))
    if (prod_code == "sav-r")
        if(t >= mat_period_original)
            return 0.;

if (t <= 0)
    return (1. - death_rate(t))
        * (1. - lapse_rate_act_bal(t) );

if ((death_ben_w=="N") && (submodel=="TERM"))
    return lapse_rate_act_bal(t) * (1. - death_rate(t)) * (1. - term->decrem_rate(t));

// dependant/indep. ... Consistent with claims paid? ***
return (1. - death_rate(t))
    * (1. - lapse_rate_act_bal(t) - pup_rate_bal(t)) // creates circular reference ?
    * (1. - prem_termination_prop(t)); // creates circular reference ?

```

#### 6.1.1.3.1.341 surv\_per\_act\_bal\_bef\_ret

```

if (t <= commence_period_w || t > maturity_period_w || submodel != "TRAD")
    return NO_AVG;

if (decrements_apply == "N")

```

```

        return 1.0;

if (t > 0 && surv_bal(t-1) < .0000001)
    return NO_AVG;

if (t <= 0 && (gross_up_historic=="N"))
    return 1.0;

if (t <= 0)
    return (1. - death_rate(t))
           * (1. - lapse_rate_act_bal(t) ) ;

return (1. - death_rate(t))
       * (1. - lapse_rate_act_bal(t) - pup_rate_bal(t));

```

#### 6.1.1.3.1.342 surv\_per\_act\_cnt

```

// Assume decrements are distributed uniformly within
// period of the projection.

if (t <= commence_period_w || t >= maturity_period_w)
    return NO_AVG;

if (decrements_apply == "N")
    return 1.0;

if (t > 0 && surv_cnt(t-1) < .0000001)
    return NO_AVG;

if (t <= 0 && (gross_up_historic=="N"))
    return 1.0;

if (eq(submodel,"UNIT"))
    if (prod_code == "sav-r")
        if(t >= mat_period_original)
            return 0.;

if (t <= 0)
    return (1. - death_rate(t))
           * (1. - lapse_rate_act_cnt(t) ) ;

if ((death_ben_w=="N") && (submodel=="TERM"))
    return lapse_rate_act_cnt(t) * (1. - death_rate(t)) * (1. - term->decrem_rate(t));

// dependant/indep. ... Consistent with claims paid? ***
return (1. - death_rate(t))
       * (1. - lapse_rate_act_cnt(t) - pup_rate_cnt(t)) // creates circular reference ?
       * (1. - prem_termination_prop(t)); // creates circular reference ?

```

#### 6.1.1.3.1.343 surv\_per\_act\_cnt\_bef\_ret

```

if (t <= commence_period_w || t > maturity_period_w)
    return NO_AVG;

if (decrements_apply == "N")
    return 1.0;

if (t > 0 && surv_cnt(t-1) < .0000001)

```



```

        return NO_AVG;

if (t <= 0 && (gross_up_historic=="N"))
    return 1.0;

if (t <= 0)
    return (1. - death_rate(t))
           * (1. - lapse_rate_act_cnt(t) ) ;

return (1. - death_rate(t))
       * (1. - lapse_rate_act_cnt(t) - pup_rate_cnt(t));

```

#### 6.1.1.3.1.344 surv\_per\_act\_prm

```

// Assume decrements are distributed uniformly within
// period of the projection.

if (t <= commence_period_w || t >= maturity_period_w)
    return NO_AVG;

if (decrements_apply == "N")
    return 1.0;

if (t > 0 && surv_prm(t-1) < .0000001)
    return NO_AVG;

if (t <= 0 && (gross_up_historic=="N"))
    return 1.0;

if (eq(submodel,"UNIT"))
    if (prod_code == "sav-r")
        if(t >= mat_period_original)
            return 0.;

if (t <= 0)
    return (1. - death_rate(t))
           * (1. - lapse_rate_act_prm(t) ) ;

if ((death_ben_w=="N") && (submodel=="TERM"))
    return lapse_rate_act_prm(t) * (1. - death_rate(t)) * (1. - term->decrem_rate(t));

// dependant/indep. ... Consistent with claims paid? ***
return (1. - death_rate(t))
       * (1. - lapse_rate_act_prm(t) - pup_rate_prm(t)) // creates circular reference ?
       * (1. - prem_termination_prop(t)); // creates circular reference ?

```

#### 6.1.1.3.1.345 surv\_per\_act\_prm\_bef\_ret

```

if (t <= commence_period_w || t > maturity_period_w || submodel != "TRAD")
    return NO_AVG;

if (decrements_apply == "N")
    return 1.0;

if (t > 0 && surv_prm(t-1) < .0000001)
    return NO_AVG;

if (t <= 0 && (gross_up_historic=="N"))

```

```

        return 1.0;

    if (t <= 0)
        return (1. - death_rate(t))
            * (1. - lapse_rate_act_prm(t) ) ;

    return (1. - death_rate(t))
        * (1. - lapse_rate_act_prm(t) - pup_rate_prm(t));

```

#### **6.1.1.3.1.346      surv\_per\_bal**

```

    if (t <= commence_period_w || t > maturity_period_w)
        return NO_AVG;

    if (decrements_apply == "N")
        return 1.0;

    if (t > 0 && surv_bal(t-1) < .0000001)
        return NO_AVG;

    if ((t <= 0 && gross_up_historic=="N") || surv_bal(t-1)==0)
        return 1.0;

    if (surv_bal(t-1) == 0)
        return 0;

    return surv_bal(t)/surv_bal(t-1);

```

#### **6.1.1.3.1.347      surv\_per\_bal\_bef\_ret**

```

    if (t <= commence_period_w || t > maturity_period_w || submodel != "TRAD")
        return NO_AVG;

    if (decrements_apply == "N")
        return 1.0;

    if (t > 0 && surv_bal_bef_ret(t-1) < .0000001)
        return NO_AVG;

    if (t <= 0 && (gross_up_historic=="N"))
        return 1.0;

    if(surv_bal(t-1) == 0)
        return 0.;

    return surv_bal_bef_ret(t)/surv_bal(t-1);

```

#### **6.1.1.3.1.348      surv\_per\_cnt**

```

    if (t <= commence_period_w || t > maturity_period_w)
        return NO_AVG;

    if (decrements_apply == "N")
        return 1.0;

    if (t > 0 && surv_cnt(t-1) < .0000001)
        return NO_AVG;

    if ((t <= 0 && gross_up_historic=="N") || surv_cnt(t-1)==0)

```

```

        return 1.0;

    if (surv_cnt(t-1) == 0)
        return 0;

    return surv_cnt(t)/surv_cnt(t-1);

```

#### 6.1.1.3.1.349 surv\_prm\_bef\_ret

```

if (t <= commence_period_w || t > maturity_period_w || submodel != "TRAD")
    return NO_AVG;

if (decrements_apply == "N")
    return 1.0;

if (t > 0 && surv_prm_bef_ret(t-1) < .0000001)
    return NO_AVG;

if (t <= 0 && (gross_up_historic=="N"))
    return 1.0;

if(surv_prm(t-1) == 0)
    return 0.;

return surv_prm_bef_ret(t)/surv_prm(t-1);

```

#### 6.1.1.3.1.350 surv\_prm

```

if (t < commence_period_w || t > maturity_period_w)
    return NO_AVG;

if (submodel=="TERM")
    return term->surv(t);

if (decrements_apply == "N")
    return 1.0;

if (t > 0 ) {
    if (fabs(surv_prm(t-1)) < .0000001)
        // No surv in previous period
        return NO_AVG;
}

// In the experience model.
return surv_act_prm(t) + surv_pup_prm(t);

```

#### 6.1.1.3.1.351 surv\_prm\_bef\_ret

```

if (t < commence_period_w || t > maturity_period_w || submodel != "TRAD")
    return NO_AVG;

if (decrements_apply == "N")
    return 1.0;

if (t > 0 && surv_prm_bef_ret(t-1) < .0000001)
    return NO_AVG;

return surv_act_prm_bef_ret(t) + surv_pup_prm_bef_ret(t);

```

**6.1.1.3.1.352    surv\_pup\_bal**

```

if (t < commence_period_w || t > maturity_period_w)
    return NO_AVG;

if (decrements_apply == "N")
    return 1.0; // *** this testing feature not handled perfectly (makes 100% prem-paying and
100% pups!)

if (t > 0 && fabs(surv_bal(t-1)) < .0000001)
    return NO_AVG;

if (paid_up=="Y" && t <= 0)
    return 1.0;

if (t == 0 || t == commence_period_w) // and premium paying
    return NO_AVG;

if (t < 0)
    return surv_pup_bal(t+1);

if (eq(ben_class, "gimla") && prem_termination_prop(t) == 1.)
    return surv_pup_bal(t-1) * (1. - death_rate(t)) * (1. - lapse_rate_pup_bal(t)) *
(surv_per_ret(t))
                                + surv_act_bal(t-1)
                                * (1 - death_rate(t))
                                * (surv_per_ret(t));

return surv_pup_bal(t-1) * (1. - death_rate(t)) * (1. - lapse_rate_pup_bal(t)) * (surv_per_ret(t))
                                + surv_act_bal(t-1) * pup_rate_bal_dep(t) * (surv_per_ret(t));

```

**6.1.1.3.1.353    surv\_pup\_bal\_bef\_ret**

```

if (t < commence_period_w || t > maturity_period_w || submodel != "TRAD")
    return NO_AVG;

if (decrements_apply == "N" || (paid_up=="Y" && t <= 0))
    return 1.0;

if ((t > 0 && surv_bal(t-1) < .0000001) || t == 0 || t == commence_period_w)
    return NO_AVG;

if (t < 0)
    return surv_pup_bal(t+1);

return surv_pup_bal(t-1) * (1. - death_rate(t)) * (1. - lapse_rate_pup_bal(t))
                                + surv_act_bal(t-1) * pup_rate_bal_dep(t);

```

**6.1.1.3.1.354    surv\_pup\_cnt**

```

if (t < commence_period_w || t > maturity_period_w)
    return NO_AVG;

if (decrements_apply == "N")
    return 1.0; // *** this testing feature not handled perfectly (makes 100% prem-paying and
100% pups!)

if (t > 0 && fabs(surv_cnt(t-1)) < .0000001)
    return NO_AVG;

```

```

if (paid_up=="Y" && t <= 0)
    return 1.0;

if (t == 0 || t == commence_period_w) // and premium paying
    return NO_AVG;

if (t < 0)
    return surv_pup_cnt(t+1);

if (eq(ben_class, "gimla") && prem_termination_prop(t) == 1.)
    return surv_pup_cnt(t-1) * (1. - death_rate(t)) * (1. - lapse_rate_pup_cnt(t)) *
(surv_per_ret(t))
                                + surv_act_cnt(t-1)
                                * (1 - death_rate(t))
                                * (surv_per_ret(t));

return surv_pup_cnt(t-1) * (1. - death_rate(t)) * (1. - lapse_rate_pup_cnt(t)) * (surv_per_ret(t))
                                + surv_act_cnt(t-1) * pup_rate_cnt_dep(t) * (surv_per_ret(t));

```

#### **6.1.1.3.1.355     surv\_pup\_cnt\_bef\_ret**

```

if (t < commence_period_w || t > maturity_period_w )
    return NO_AVG;

if (decrements_apply == "N")
    return 1.0;

if (t > 0 && surv_cnt(t-1) < .0000001)
    return NO_AVG;

if (paid_up=="Y" && t <= 0)
    return 1.0;

if (t == 0 || t == commence_period_w)
    return NO_AVG;

if (t < 0)
    return surv_pup_cnt(t+1);

return surv_pup_cnt(t-1) * (1. - death_rate(t)) * (1. - lapse_rate_pup_cnt(t))
                                + surv_act_cnt(t-1) * pup_rate_cnt_dep(t);

```

#### **6.1.1.3.1.356     surv\_pup\_post\_ret**

```

if (t < commence_period_w || t > maturity_period_w)
    return NO_AVG;

if(t == mat_period_original)
    return 1.;

return (1. - death_rate(t)) * (surv_per_ret(t-1));

```

#### **6.1.1.3.1.357     surv\_pup\_prm**

```

if (t < commence_period_w || t > maturity_period_w)
    return NO_AVG;

if (decrements_apply == "N")

```

```

    return 1.0; // *** this testing feature not handled perfectly (makes 100% prem-paying and
100% pups!)

if (t > 0 && fabs(surv_prm(t-1)) < .0000001)
    return NO_AVG;

if (paid_up=="Y" && t <= 0)
    return 1.0;

if (t == 0 || t == commence_period_w) // and premium paying
    return NO_AVG;

if (t < 0)
    return surv_pup_prm(t+1);

if (eq(ben_class, "gimla") && prem_termination_prop(t) == 1.)
    return surv_pup_prm(t-1) * (1. - death_rate(t)) * (1. - lapse_rate_pup_prm(t)) *
(surv_per_ret(t))
        + surv_act_prm(t-1)
        * (1 - death_rate(t))
        * (surv_per_ret(t));

return surv_pup_prm(t-1) * (1. - death_rate(t)) * (1. - lapse_rate_pup_prm(t)) * (surv_per_ret(t))
        + surv_act_prm(t-1) * pup_rate_prm_dep(t) * (surv_per_ret(t));

```

#### 6.1.1.3.1.358 surv\_pup\_prm\_bef\_ret

```

if (t < commence_period_w || t > maturity_period_w || submodel != "TRAD")
    return NO_AVG;

if (decrements_apply == "N")
    return 1.0;

if (t > 0 && surv_prm(t-1) < .0000001)
    return NO_AVG;

if (paid_up=="Y" && t <= 0)
    return 1.0;

if (t == 0 || t == commence_period_w)
    return NO_AVG;

if (t < 0)
    return surv_pup_prm(t+1);

return surv_pup_prm(t-1) * (1. - death_rate(t)) * (1. - lapse_rate_pup_prm(t))
        + surv_act_prm(t-1) * pup_rate_prm_dep(t);

```

#### 6.1.1.3.1.359 death\_rate

```

if (t <= commence_period_w || t > maturity_period_w || eq(paid_up,"G"))
    return NO_AVG;

if (t <= 0 && (gross_up_historic=="N"))
    return 0.0;

if (submodel == "TRAD")
    return trad->death_rate(t);

```

```

int pol_yr = max(xint(pol_year(t)),1);

// Assume all lives die at omega age
if (age_last(t) >= omega_age_w)
    return 1.0;

double rate = 0.0;

// set column to allow for selection in mortality table
int col_temp = 0;
if (mort_sel_status=="Y")
    col_temp =min(pol_yr + xint(elapsed_months_extra/12.), select_periods);
else
    col_temp =select_periods;

death_rate_row_key =age_last(t) - col_temp+1;
col_dth = col_temp;
rate = death_rates_tbl;
rate = rate * mort_mult / 100.;

// Only apply the medical/occupational loading if there is a death benefit
if(death_ben_w=="Y")
    rate = rate * (1.+health_occ_perc/100.);

//***** add margin *****
if (margin_add=="Y")
    rate = rate * (1+margin_mort_pc/100);

//Margin for catastrophe
if (margin_add_cat == "Y" && (eq(submodel, "UNIT") || death_ben_w == "Y" || eq(submodel,
"TRAD"))){//Only apply to savings and death risk

    double m_cat = 0;

    if (proj_year(t) == 1)
        m_cat = cat_risk;

    rate = rate + m_cat;
}

//***** convert to monthly *****

rate = max(0.0, min(1.0, rate));

return rate = (1. - pow(1. - rate, 1./12.));

```

#### 6.1.1.3.1.360 lapse\_factor

```

if (t <= commence_period_w || t > maturity_period_w)
    return NO_AVG;

double rate = 0.0;

factor_key = "_" + company;
lapse_factor_y1_row = "Y1" + factor_key;
rate = lapse_factor_y1;

// column not found in lapse_factor_tbl

```

```

if(rate == -99999.) {
factor_key = "";
lapse_factor_y1_row = "Y1"+factor_key;
rate =lapse_factor_y1;
} // end if

if (pol_year_ext(t)>=2 && pol_year_ext(t)<6) {
lapse_factor_y_col = "Y2_5"+factor_key;
rate = lapse_factor_yplus;
}

if (pol_year_ext(t)>=6 ) {
lapse_factor_y_col = "Y6plus"+factor_key;
rate = lapse_factor_yplus;
}

return rate/100.0;

```

#### 6.1.1.3.1.361 lapse\_rate\_act\_bal

```

if (t <= commence_period_w || t > maturity_period_w)
    return NO_AVG;

if (t <= 0 && gross_up_historic=="N")
    return 0.0;

if((mult_age_ind == 1 && age_last(t) >= min_retirement_age) || inlist(paid_up,"Y,C,G"))
    return 0.;

double Mass_rate = 0;

if(lapse_force_month >0 && lapse_force_month == t){
    if(lapse_force_rate == 1.)
        return 0.;
    Mass_rate = lapse_force_rate;
}

double rate = 0.0;

if (inlist(submodel,"TERM"))
    return term->lapse_rate(t);

lapse_type_col_key = "Surrender";
lapse_expos_col_key = "balance";

tarif_spec_row_key= xstring(tarif); //added to avoid mutating lookup term error
rate= lapse_rate_im/ 100.0;

rate = rate * lapse_factor(t) * lapse_factor_proj/100.0;

if(margin_add=="Y")
    rate = rate * (1. + margin_lapses/100.);

rate = 1. - pow((1. - rate), 1./12.);
rate = min(rate * (1-Mass_rate) + Mass_rate,0.999);

return rate;

```



**6.1.1.3.1.362 lapse\_rate\_act\_bal\_dep**

```

if (t <= commence_period_w || t > maturity_period_w)
    return NO_AVG;

if (death_ben_w=="N") {
    if (submodel=="TERM")
        return lapse_rate_act_bal(t) * (1. - death_rate(t)) * (1. - term->decrem_rate(t));
    }
return lapse_rate_act_bal(t) * (1. - death_rate(t));

```

**6.1.1.3.1.363 lapse\_rate\_act\_cnt**

```

if (t <= commence_period_w || t > maturity_period_w)
    return NO_AVG;

if (t <= 0 && gross_up_historic=="N")
    return 0.0;

if((mult_age_ind == 1 && age_last(t) >= min_retirement_age) || inlist(paid_up,"Y,C,G"))
    return 0.;

double Mass_rate = 0;

if(lapse_force_month >0 && lapse_force_month == t){
    if(lapse_force_rate == 1.)
        return 0.;
    Mass_rate = lapse_force_rate;
}

double rate = 0.0;

if (inlist(submodel,"TERM"))
    return term->lapse_rate(t);

lapse_type_col_key = "Surrender";
lapse_expos_col_key = "count";

tarif_spec_row_key= xstring(tarif); //added to avoid mutating lookup term error
rate= lapse_rate_im/ 100.0;

rate = rate * lapse_factor(t) * lapse_factor_proj/100.0 ;

if(margin_add=="Y")
    rate = rate * (1. + margin_lapses/100.);

rate = 1. - pow((1. - rate), 1./12.);
rate = min(rate * (1-Mass_rate) + Mass_rate,0.999);

return rate;

```

**6.1.1.3.1.364 lapse\_rate\_act\_cnt\_dep**

```

if (t <= commence_period_w || t > maturity_period_w)
    return NO_AVG;

```

```

if (death_ben_w=="N") {
  if (submodel=="TERM")
    return lapse_rate_act_cnt(t) * (1. - death_rate(t)) * (1. - term->decrem_rate(t));
}
return lapse_rate_act_cnt(t) * (1. - death_rate(t));

```

#### 6.1.1.3.1.365 lapse\_rate\_act\_prm

```

if (t <= commence_period_w || t > maturity_period_w)
  return NO_AVG;

if (t <= 0 && gross_up_historic=="N")
  return 0.0;

if((mult_age_ind == 1 && age_last(t) >= min_retirement_age) || inlist(paid_up,"Y,C,G"))
  return 0.;

double Mass_rate = 0;

if(lapse_force_month >0 && lapse_force_month == t){
  if(lapse_force_rate == 1.)
    return 0.;
  Mass_rate = lapse_force_rate;
}

double rate = 0.0;

if (inlist(submodel,"TERM"))
  return term->lapse_rate(t);

lapse_type_col_key = "Surrender";
lapse_expos_col_key = "premium";

tarif_spec_row_key= xstring(tarif); //added to avoid mutating lookup term error
rate= lapse_rate_im/ 100.0;

rate = rate * lapse_factor(t) * lapse_factor_proj/100.0;

if(margin_add=="Y")
  rate = rate * (1. + margin_lapses/100.);

rate = 1. - pow((1. - rate), 1./12.);
rate = min(rate * (1-Mass_rate) + Mass_rate,0.999);

return rate;

```

#### 6.1.1.3.1.366 lapse\_rate\_act\_prm\_dep

```

if (t <= commence_period_w || t > maturity_period_w)
  return NO_AVG;

if (death_ben_w=="N") {
  if (submodel=="TERM")
    return lapse_rate_act_prm(t) * (1. - death_rate(t)) * (1. - term->decrem_rate(t));
}
return lapse_rate_act_prm(t) * (1. - death_rate(t));

```

**6.1.1.3.1.367 lapse\_rate\_pup\_bal**

```

if (t <= commence_period_w || t > maturity_period_w)
    return NO_AVG;

if (t <= 0 && (gross_up_historic=="N"))
    return 0.0;

if (t > 0 && fabs(surv_bal(t-1)) < .0000001)
    return 0.0;

if (!inlist(submodel,"UNIT,TRAD")) //only for UNIT & TRAD
    return 0.0;

if(mult_age_ind == 1 && age_last(t) >= min_retirement_age)
    return 0.;

if (eq(submodel,"UNIT") && units_e_bef(t) <= 0.00001 ) // lapse policy if fund is zero or less
    return 1.;

double Mass_rate = 0;

if(lapse_force_month >0 && lapse_force_month == t && lapse_force_rate < 1.)
    Mass_rate = lapse_force_rate;

lapse_type_col_key = "Surrender";
lapse_expos_col_key = "balance";
tarif_spec_row_key= xstring(tarif); //added to avoid mutating lookup term error

double rate = lapse_rate_pup_im / 100.0;

double margin = 0.;
if(margin_add=="Y")
    margin = margin_lapses;

rate = min(0.999, rate * (1 + margin/100.));

rate = 1. - pow((1. - rate), 1./12.);
rate = min(rate * (1-Mass_rate) + Mass_rate,0.999);

return rate;

```

**6.1.1.3.1.368 lapse\_rate\_pup\_bal\_dep**

```

if (t <= commence_period_w || t > maturity_period_w)
    return NO_AVG;

return lapse_rate_pup_bal(t) * (1. - death_rate(t));

```

**6.1.1.3.1.369 lapse\_rate\_pup\_cnt**

```

if (t <= commence_period_w || t > maturity_period_w)
    return NO_AVG;

if (t <= 0 && (gross_up_historic=="N"))
    return 0.0;

if (t > 0 && fabs(surv_cnt(t-1)) < .0000001)
    return 0.0;

```

```

if (!inlist(submodel,"UNIT,TRAD")) //only for UNIT & TRAD
    return 0.0;

if(mult_age_ind == 1 && age_last(t) >= min_retirement_age)
    return 0.;

if (eq(submodel,"UNIT") && units_e_bef(t) <= 0.00001 ) // lapse policy if fund is zero or less
    return 1.;

double Mass_rate = 0;

if(lapse_force_month >0 && lapse_force_month == t && lapse_force_rate < 1.)
    Mass_rate = lapse_force_rate;

lapse_type_col_key = "Surrender";
lapse_expos_col_key = "count";
tarif_spec_row_key= xstring(tarif); //added to avoid mutating lookup term error

double rate = lapse_rate_pup_im / 100.0;

double margin = 0.;
if(margin_add=="Y")
    margin = margin_lapses;

rate = min(0.999, rate * (1 + margin/100.));

rate = 1. - pow((1. - rate), 1./12.);
rate = min(rate * (1-Mass_rate) + Mass_rate,0.999);

return rate;

```

#### **6.1.1.3.1.370 lapse\_rate\_pup\_cnt\_dep**

```

if (t <= commence_period_w || t > maturity_period_w)
    return NO_AVG;

return lapse_rate_pup_cnt(t) * (1. - death_rate(t));

```

#### **6.1.1.3.1.371 lapse\_rate\_pup\_prm**

```

if (t <= commence_period_w || t > maturity_period_w)
    return NO_AVG;

if (t <= 0 && (gross_up_historic=="N"))
    return 0.0;

if (t > 0 && fabs(surv_prm(t-1)) < .0000001)
    return 0.0;

if (!inlist(submodel,"UNIT,TRAD")) //only for UNIT & TRAD
    return 0.0;

if(mult_age_ind == 1 && age_last(t) >= min_retirement_age)
    return 0.;

if (eq(submodel,"UNIT") && units_e_bef(t) <= 0.00001 ) // lapse policy if fund is zero or less
    return 1.;

```

```

double Mass_rate = 0;

if(lapse_force_month >0 && lapse_force_month == t && lapse_force_rate < 1.)
    Mass_rate = lapse_force_rate;

lapse_type_col_key = "Surrender";
lapse_expos_col_key = "premium";
tarif_spec_row_key= xstring(tarif); //added to avoid mutating lookup term error

double rate = lapse_rate_pup_im / 100.0;

double margin = 0.;
if(margin_add=="Y")
    margin = margin_lapses;

rate = min(0.999, rate * (1 + margin/100.));

rate = 1. - pow((1. - rate), 1./12.);
rate = min(rate * (1-Mass_rate) + Mass_rate,0.999);

return rate;

```

#### 6.1.1.3.1.372 lapse\_total\_bal

```

if (t <= commence_period_w || t > maturity_period_w)
    return NO_AVG;

if (t <= 0 && gross_up_historic=="N")
    return 0.0;

if (t > 0 ) {
    if (fabs(surv_bal(t-1)) < .000000001)
        return 0.0;
    if (eq(submodel,"UNIT")) {
        if (units_e_bef(t) <= 0.00001) // lapse policy if fund is zero or less
            return 1.;

        if (premium_if_b(t)<=0.0 || prem_termination_prop(t) == 1.) // lapse policy if
premium for risk rider sold with Managers Meitav higher than total premium (i.e. not enough
premium)
            return 1.;

        if (sm_accum->units_b_bef(t) + sm_accum->alloc_units(t)
            + sm_saving->units_b_bef(t) + sm_saving->alloc_units(t)
            <= cover_charge(t))
            return 1.; // lapse policy if (non paid-up) fund is less than cover
charges

        if (t+elapsed_months >= prem_term && t+elapsed_months < benefit_term)
            return 0.; // when premiums cease then policy becomes paid-up

    } // end if (eq(sub_model,"UNIT"))
} // end if (t > 0 )

double rate = 0.0;

if (eq(paid_up,"Y"))

```

```

rate = lapse_rate_pup_bal(t);

else {

double Annual_pup_rate = 1. - pow((1. - pup_rate_bal(t)), 12.);
double Annual_lapse_rate = 1. - pow((1. - lapse_rate_act_bal(t)), 12.);

rate = Annual_pup_rate + Annual_lapse_rate;

rate = min(0.999 ,rate);

rate = 1. - pow((1. - rate), 1./12.);}

if (t<=0)
    return rate;

//if(lapse_force_month >0 && lapse_force_month == t)
//return lapse_force_rate;

return rate;

```

#### 6.1.1.3.1.373 lapse\_total\_prm

```

if (t <= commence_period_w || t > maturity_period_w)
    return NO_AVG;

if (t <= 0 && gross_up_historic=="N")
    return 0.0;

if (t > 0 ) {
    if (fabs(surv_prm(t-1)) < .000000001)
        return 0.0;
    if (eq(submodel,"UNIT")) {
        if (units_e_bef(t) <= 0.00001) // lapse policy if fund is zero or less
            return 1.;

        if (premium_if_b(t)<=0.0 || prem_termination_prop(t) == 1.) // lapse policy if
premium for risk rider sold with Managers Meitav higher than total premium (i.e. not enough
premium)
            return 1.;

        if (sm_accum->units_b_bef(t) + sm_accum->alloc_units(t)
            + sm_saving->units_b_bef(t) + sm_saving->alloc_units(t)
            <= cover_charge(t))
            return 1.; // lapse policy if (non paid-up) fund is less than cover
charges

        if (t+elapsed_months >= prem_term && t+elapsed_months < benefit_term)
            return 0.; // when premiums cease then policy becomes paid-up

    } // end if (eq(sub_model,"UNIT"))
} // end if (t > 0 )

double rate = 0.0;

```

```

if (eq(paid_up,"Y"))
rate = lapse_rate_pup_prm(t);

else {

double Annual_pup_rate = 1. - pow((1. - pup_rate_prm(t)), 12.);
double Annual_lapse_rate = 1. - pow((1. - lapse_rate_act_prm(t)), 12.);

rate = Annual_pup_rate + Annual_lapse_rate;

rate = min(0.999 ,rate);

rate = 1. - pow((1. - rate), 1./12.);}

if (t<=0)
    return rate;

//if(lapse_force_month >0 && lapse_force_month == t)
//Mass_rate = lapse_force_rate;

return rate;

```

#### 6.1.1.3.1.374 pup\_rate\_bal

```

if (t <= commence_period_w || t > maturity_period_w)
    return NO_AVG;

if (eq(submodel,"UNIT"))
    if (t+elapsed_months >= prem_term && t+elapsed_months < benefit_term)
        return 1.;

if (eq(submodel,"UNIT"))
    if (prod_code == "sav-r")
        if(t >= mat_period_original)
            return 1.;

if(eq(submodel,"TRAD"))
    if(t > mat_period_original)
        return 1.;

if (inlist(submodel,"TERM") || eq(paid_up,"G"))
return 0.;

if (t <= 0 && gross_up_historic=="N")
    return 0.0;

// solvency II scenario purpose - for Lapse Mass no paid-up ( all surrender) , for EPIFP all paid-
up
if(lapse_force_month >0 && lapse_force_month == t){
    if(lapse_force_rate < 1.)
        return 0.;
    return 1.;
}

if (t > 0 ) {
    if (fabs(surv_bal(t-1)) < .000000001)
        return 0.0;
}

```

```

//if (eq(sub_model,"UNIT") && surv_prem(t-1) > 0.) {
if (eq(submodel,"UNIT") ) {
    if (units_e_bef(t) <= 0.00001) // lapse policy if fund is zero or less
        return 1.;

    if (premium_if_b(t)<=0.0 || prem_termination_prop(t) == 1.) // lapse policy if
premium for risk rider sold with Managers Meitav higher than total premium (i.e. not enough
premium)
        return 1.;

    if (sm_accum->units_b_bef(t) + sm_accum->alloc_units(t)
        + sm_saving->units_b_bef(t) + sm_saving->alloc_units(t)
        <= cover_charge(t))
        return 1.; // lapse policy if (non paid-up) fund is less than cover
charges
}
}

if(eq(paid_up,"Y"))
    return 1.;

double rate = 0;

lapse_type_col_key = "PUP";
lapse_expos_col_key = "balance";
tarif_spec_row_key= xstring(tarif); //added to avoid mutating lookup term error

rate= lapse_rate_im/ 100.0;

rate = rate * lapse_factor(t) * lapse_factor_proj/100.0;

if(margin_add=="Y")
    rate = rate * (1. + margin_lapses/100.);

rate = min(rate ,0.999);

return 1. - pow((1. - rate), 1./12.);

```

#### 6.1.1.3.1.375 pup\_rate\_bal\_dep

```

if (t <= commence_period_w || t > maturity_period_w)
    return NO_AVG;

if (eq(submodel,"UNIT"))
    if (t+elapsed_months >= prem_term && t+elapsed_months < benefit_term)
        return 1.;

return pup_rate_bal(t) * (1. - death_rate(t));

```

#### 6.1.1.3.1.376 pup\_rate\_cnt

```

if (t <= commence_period_w || t > maturity_period_w)
    return NO_AVG;

if (eq(submodel,"UNIT"))
    if (t+elapsed_months >= prem_term && t+elapsed_months < benefit_term)
        return 1.;

```



---

```

if (eq(submodel,"UNIT"))
    if (prod_code == "sav-r")
        if(t >= mat_period_original)
            return 1.;

if(eq(submodel,"TRAD"))
    if(t > mat_period_original)
        return 1.;

if (inlist(submodel,"TERM") || eq(paid_up,"G"))
    return 0.;

if (t <= 0 && gross_up_historic=="N")
    return 0.0;

// solvency II scenario purpose - for Lapse Mass no paid-up ( all surrender) , for EPIFP all paid-
up
if(lapse_force_month >0 && lapse_force_month == t){
    if(lapse_force_rate < 1.)
        return 0.;
    return 1.;
}

if (t > 0 ) {
    if (fabs(surv_cnt(t-1)) < .000000001)
        return 0.0;
    //if (eq(sub_model,"UNIT") && surv_prem(t-1) > 0.) {
    if (eq(submodel,"UNIT") ) {
        if (units_e_bef(t) <= 0.00001) // lapse policy if fund is zero or less
            return 1.;

        if (premium_if_b(t)<=0.0 || prem_termination_prop(t) == 1.) // lapse policy if
premium for risk rider sold with Managers Meitav higher than total premium (i.e. not enough
premium)
            return 1.;

        if (sm_accum->units_b_bef(t) + sm_accum->alloc_units(t)
            + sm_saving->units_b_bef(t) + sm_saving->alloc_units(t)
            <= cover_charge(t))
            return 1.; // lapse policy if (non paid-up) fund is less than cover
charges
    }
}

if(eq(paid_up,"Y"))
    return 1.;

double rate = 0;

lapse_type_col_key = "PUP";
lapse_expos_col_key = "count";
tarif_spec_row_key= xstring(tarif); //added to avoid mutating lookup term error

rate= lapse_rate_im/ 100.0;

rate = rate * lapse_factor(t) * lapse_factor_proj/100.0;

```

```

if(margin_add=="Y")
    rate = rate * (1. + margin_lapses/100.);

rate = min(rate ,0.999);

return 1. - pow((1. - rate), 1./12.);

```

#### 6.1.1.3.1.377 pup\_rate\_cnt\_dep

```

if (t <= commence_period_w || t > maturity_period_w)
    return NO_AVG;

if (eq(submodel,"UNIT"))
    if (t+elapsed_months >= prem_term && t+elapsed_months < benefit_term)
        return 1.;

return pup_rate_cnt(t) * (1. - death_rate(t));

```

#### 6.1.1.3.1.378 pup\_rate\_prm

```

if (t <= commence_period_w || t > maturity_period_w)
    return NO_AVG;

if (eq(submodel,"UNIT"))
    if (t+elapsed_months >= prem_term && t+elapsed_months < benefit_term)
        return 1.;

if (eq(submodel,"UNIT"))
    if (prod_code == "sav-r")
        if(t >= mat_period_original)
            return 1.;

if(eq(submodel,"TRAD"))
    if(t > mat_period_original)
        return 1.;

if (inlist(submodel,"TERM") || eq(paid_up,"G"))
    return 0.;

if (t <= 0 && gross_up_historic=="N")
    return 0.0;

// solvency II scenario purpose - for Lapse Mass no paid-up ( all surrender) , for EPIFP all paid-
up
if(lapse_force_month >0 && lapse_force_month == t){
    if(lapse_force_rate < 1.)
        return 0.;
    return 1.;
}

if (t > 0 ) {
    if (fabs(surv_prm(t-1)) < .000000001)
        return 0.0;
    //if (eq(sub_model,"UNIT") && surv_prem(t-1) > 0.) {
    if (eq(submodel,"UNIT") ) {
        if (units_e_bef(t) <= 0.00001) // lapse policy if fund is zero or less
            return 1.;
    }
}

```

```

        if (premium_if_b(t)<=0.0 || prem_termination_prop(t) == 1.) // lapse policy if
premium for risk rider sold with Managers Meitav higher than total premium (i.e. not enough
premium)
            return 1.;

        if (sm_accum->units_b_bef(t) + sm_accum->alloc_units(t)
            + sm_saving->units_b_bef(t) + sm_saving->alloc_units(t)
            <= cover_charge(t))
            return 1.; // lapse policy if (non paid-up) fund is less than cover
charges
    }
}

if(eq(paid_up,"Y"))
    return 1.;

double rate = 0;

lapse_type_col_key = "PUP";
lapse_expos_col_key = "premium";
tarif_spec_row_key= xstring(tarif); //added to avoid mutating lookup term error

rate= lapse_rate_im/ 100.0;

rate = rate * lapse_factor(t) * lapse_factor_proj/100.0;

if(margin_add=="Y")
    rate = rate * (1. + margin_lapses/100.);

rate = min(rate ,0.999);

return 1. - pow((1. - rate), 1./12.);

```

#### 6.1.1.3.1.379 pup\_rate\_prm\_dep

```

if (t <= commence_period_w || t > maturity_period_w)
    return NO_AVG;

if (eq(submodel,"UNIT"))
    if (t+elapsed_months >= prem_term && t+elapsed_months < benefit_term)
        return 1.;

return pup_rate_prm(t) * (1. - death_rate(t));

```

#### 6.1.1.3.1.380 expense\_investment

```

if (t <= commence_period_w || t > maturity_period_ann)
    return 0.0;

double result = exp_ren_res / 1200.
                * reserve(t)
                * free_inv_prop_t[proj_year(t)]
                * expense_inflation(t);

double margin = 0.;
if(margin_add=="Y")
    margin = margin_exp_ren_pc;

```

```
return result * (1 + margin/100.);
```

#### 6.1.1.3.1.381 expense\_investment\_bef\_ret

```
if (t <= commence_period_w || t > maturity_period_w)
    return 0.0;
```

```
double result = exp_ren_res / 1200.
                * reserve_bef_ret(t)
                * free_inv_prop_t[proj_year(t)]
                * expense_inflation(t);
```

```
double margin = 0.;
if(margin_add=="Y")
    margin = margin_exp_ren_pc;
```

```
return result * (1 + margin/100.);
```

#### 6.1.1.3.1.382 expense\_investment\_post\_ret

```
if (t <= commence_period_w || t > maturity_period_ann)
    return 0.0;
```

```
if(inlist(submodel,"UNIT,TRAD,ANN") && res_prop_kitzba > 0.0){
    if(mult_age_ind == 1)
        return sm_annuity->expense_investment_post_ret(t);
    return sm_annuity[ann_index_map[takeup_age]]->expense_investment_post_ret(t);
}
return NO_AVG;
```

#### 6.1.1.3.1.383 expense\_investment\_pv

```
if (t < commence_period_w || t > maturity_period_ann)
    return 0.0;
```

```
if(mult_age_ind != 1){
int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));
```

```
if( t > maturity_period_w)
    return expense_investment_pv(t+1)* ann_v_month_t[proj_yr]+ expense_investment(t+1);
```

```
return expense_investment_pv(t+1)* v_month_t[proj_yr]+ expense_investment(t+1);
```

```
}
```

```
return expense_investment_pv_bef_ret(t) +expense_investment_pv_post_ret(t);
```

#### 6.1.1.3.1.384 expense\_investment\_pv\_bef\_ret

```
if (t < commence_period_w || t >= maturity_period_ann)
    return 0.0;
```

```
int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));
```

```
return expense_investment_pv_bef_ret(t+1)* v_month_t[proj_yr]+ expense_investment_bef_ret(t+1);
```

#### **6.1.1.3.1.385 expense\_investment\_pv\_post\_ret**

```
if (t < commence_period_w || t >= maturity_period_ann)
    return 0.0;
```

```
int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));
```

```
if ( t > maturity_period_w)
    return expense_investment_pv_post_ret(t+1)* ann_v_month_t[proj_yr]+
expense_investment_post_ret(t+1);
else
    return expense_investment_pv_post_ret(t+1)* v_month_t[proj_yr]+
expense_investment_post_ret(t+1);
```

#### **6.1.1.3.1.386 expense\_ren\_charge\_pv**

```
if (t < commence_period_w || t > maturity_period_ann)
    return 0.0;
```

```
int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));
```

```
if( t > maturity_period_w)
    return expense_ren_charge_pv(t+1)* ann_v_month_t[proj_yr]+ expense_ren_charge(t+1);
```

```
return expense_ren_charge_pv(t+1)* v_month_t[proj_yr]+ expense_ren_charge(t+1);
```

#### **6.1.1.3.1.387 benefits\_b\_prm**

```
if (t <= commence_period_w || t > maturity_period_w)
    return 0.0;
```

```
return benefits_curr * surv_prm(t-1);
```

#### **6.1.1.3.1.388 policies\_b**

```
if (t <= commence_period_w || t > maturity_period_ann)
    return 0.0;
```

```
return policies_curr * surv_cnt(t-1);
```

#### **6.1.1.3.1.389 policies\_pup\_b**

```
if (t <= commence_period_w || t > maturity_period_w)
    return 0.0;
```

```
return policies_curr * surv_pup_cnt(t-1);
```

#### **6.1.1.3.1.390 policy\_deaths**

```
if (t <= 0 || t > maturity_period_w)
    return 0.0;
```

```
if (t > 0 && fabs(surv_prm(t-1)) < .0000001)
    return 0.0;
```

```
return policy_deaths(t-1) + policies_b(t) * death_rate(t);
```

#### 6.1.1.3.1.391 policy\_surr

```
if (t <= 0 || t > maturity_period_w)
    return 0.0;
```

```
if (t > 0 && fabs(surv_prm(t-1)) < .0000001)
    return 0.0;
```

```
return policy_surr(t-1) + (policies_b(t)-policies_pup_b(t)) * lapse_rate_act_cnt_dep(t) +
    policies_pup_b(t) * lapse_rate_pup_cnt_dep(t);
```

#### 6.1.1.3.1.392 claims\_retent

```
if (t <= commence_period_w || t > maturity_period_w)
    return 0.0;
```

```
if(submodel == "TERM")
    return term->claims_total(t) * (1 - re_ratio_w);
```

```
return 0.0;
```

#### 6.1.1.3.1.393 sum\_at\_risk\_claim

```
if (t <= commence_period_w || t > maturity_period_w || inlist(paid_up,"Y,C,G"))
    return 0.0;
```

```
double temp=0.0;
```

```
if(eq(ben_class,"adif")) {
double adifsa=0.0; // standard sum insured bought from basic premium

    adifsa = xint(premium(t)/100. * min(1.,basic_perc(t)) //_w /100.
        * sum_ins_basic_tt(xint(age_last(t)),sex_smoker_code));

    temp = xint(max(sum_insured(t) * surv_act_prm(t-1) // total SAR for premium-paying policies
        - (sm_accum->units_e_bef(t) + sm_saving->units_e_bef(t)), 0.0));
    return max(temp,adifsa);
}
```

```
return 0.0;
```

#### 6.1.1.3.1.394 sum\_insured

```
if (t < commence_period_w || t > maturity_period_w)
    return 0.0;
```

```
if(submodel == "TRAD")
    return trad->sum_insured(t);
```

```
if(submodel == "TERM")
    return term->sum_insured(t);
```

```
if ((t+elapsed_months > prem_term && eq(ben_class,"adif") )|| (inlist(paid_up,"Y,G"))))
    return 0.0;
```

```

if (t > 0 && fabs(surv_prm(t-1)) < .0000001)
    return 0.0;

double limit=0.0; // maximum total sum-insured, for Adif, to ensure that premium is adequate to
purchase it.

if (t > 0) {
    if (eq(ben_class,"adif")) {
        if(surv_act_prm(t-1)>0) {
            limit = (sm_accum->units_e(t-1)+sm_saving->units_e(t-1)
                + (1.-prem_risk_max/100.)*premium_if_b(t)/prem_freq
                * (sum_ins_basic_tt(xint(age_last(t)),sex_smoker_code)/100. +
alloc_rate[1]/100.)) // *** assumes allocation rate constant for basic Adif
                /surv_act_prm(t-1);
            if (prem_rates_extra_tt(xint(age_last(t)),sex_smoker_code)>0.0)
                limit = limit +
                (prem_risk_max/100.*premium_if_b(t)/prem_rates_extra_tt(xint(age_last(t)),sex_smoker_code) )
// *** assumes allocation rate constant for basic Adif
                /surv_act_prm(t-1);
        }
        else if (eq(ben_class,"profil")) {
            limit= sum_insured_rider_tt.sum_of_row(t);
        }
        else
            limit = 0.0;
    }
    else
        limit = sum_insured(t-1);

    return min(limit,sum_insured(t-1));
}

if (t == 0) {

    if (eq(ben_class,"profil")) {
        limit= sum_insured_rider_tt.sum_of_row(t);
    }

return max(sum_ins_curr,limit) * benefits_curr;
}

// if (t<0)
if (gross_up_historic=="N")
    if (eq(ben_class,"adif")) {
        if (prem_rates_extra_tt(xint(age_last(t)),sex_smoker_code)>0)
            return
min(prem_risk_max/100.*premium_if_b(1)/prem_rates_extra_tt(xint(age_last(t)),sex_smoker_code),sum_i
nsured(t+1));
        else
            return sum_insured(t+1);
    }

return sum_ins_curr;

```

#### 6.1.1.3.1.395 sum\_insured\_if\_e

```

if (t < commence_period_w || t >= maturity_period_w)

```

```
        return 0.0;

    if(submodel == "TRAD")
        return trad->sum_insured_if_e(t) + trad->sum_insured_if_b_pup(t+1);

    if(submodel == "TERM")
        return term->sum_insured_if_b(t+1);

    return sum_insured(t)*surv_prm(t);
```

#### **6.1.1.3.1.396 sum\_insured\_occ\_gross**

```
if (t < commence_period_w || t >= maturity_period_w)
    return 0.0;

if(submodel == "TERM")
    return sum_insured_if_e(t) * (1 + health_occ_perc/100.);

return 0.0;
```

#### **6.1.1.3.1.397 sum\_insured\_occ\_retent**

```
if (t < commence_period_w || t >= maturity_period_w)
    return 0.0;

if(submodel == "TERM")
    return sum_insured_occ_gross (t) * (1 - re_ratio_w);

return 0.0;
```

#### **6.1.1.3.1.398 alloc\_units**

```
if (t <= commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if (!eq(submodel,"UNIT"))
    return 0.0;

double temp=0.0;

    temp = sm_accum->alloc_units(t) +
           sm_acc_pup->alloc_units(t)+
           sm_saving->alloc_units(t) +
           sm_saving_pup->alloc_units(t);

return temp;
```

#### **6.1.1.3.1.399 interest\_units\_e**

```
if (submodel=="TERM")
    return NO_AVG;

if (submodel=="TRAD")
    return NO_AVG;

double temp=0.0;
```



```

temp = sm_accum->int_cred_units_e(t) +
        sm_acc_pup->int_cred_units_e(t)+
        sm_saving->int_cred_units_e(t) +
        sm_saving_pup->int_cred_units_e(t);

```

```

return temp;

```

#### **6.1.1.3.1.400 units\_b**

```

return sm_accum->units_b(t) +
        sm_acc_pup->units_b(t)+
        sm_saving->units_b(t)+
        sm_saving_pup->units_b(t);

```

#### **6.1.1.3.1.401 units\_b\_active**

```

return sm_accum->units_b(t)
        + sm_saving->units_b(t);

```

#### **6.1.1.3.1.402 units\_b\_bef**

```

return sm_accum->units_b_bef(t) +
        sm_acc_pup->units_b_bef(t)+
        sm_saving->units_b_bef(t)+
        sm_saving_pup->units_b_bef(t);

```

#### **6.1.1.3.1.403 units\_b\_bef\_pup\_acc**

```

if (submodel=="TERM")
    return NO_AVG;

```

```

if (submodel=="TRAD")
    return NO_AVG;

```

```

if (t <= 0 || t > maturity_period_w)
    return 0.0;

```

```

return sm_accum->units_e_bef(t-1) * pup_rate_bal_dep(t-1) * surv_per_ret(t-1); //Pups occurring in
the last month

```

#### **6.1.1.3.1.404 units\_b\_bef\_pup\_sav**

```

if (submodel=="TERM")
    return NO_AVG;

```

```

if (submodel=="TRAD")
    return NO_AVG;

```

```

if (t <= 0 || t > maturity_period_w)
    return 0.0;

```

```

return sm_saving->units_e_bef(t-1) * pup_rate_bal_dep(t-1) * surv_per_ret(t-1); //Pups occurring in
the last month;

```

#### **6.1.1.3.1.405 units\_b\_pup**

```

return sm_acc_pup->units_b(t)
        + sm_saving_pup->units_b(t);

```

**6.1.1.3.1.406 units\_bon**

```

if (t <= commence_period_w || t > maturity_period_w)
    return NO_AVG;

if (inlist(submodel,"TERM,ANN"))
    return NO_AVG;

if (submodel=="TRAD")
    return trad->bonus_if(t)+trad->bonus_if_pup(t);

return sm_accum->units_e_bef(t) * bonus[t+elapsed_months]/100.;

```

**6.1.1.3.1.407 units\_e**

```

if(submodel != "UNIT")
    return NO_AVG;

return sm_accum->units_e(t) +
    sm_acc_pup->units_e(t)+
    sm_saving->units_e(t) +
    sm_saving_pup->units_e(t);

```

**6.1.1.3.1.408 units\_e\_bef**

```

return sm_accum->units_e_bef(t) +
    sm_acc_pup->units_e_bef(t)+
    sm_saving->units_e_bef(t)+
    sm_saving_pup->units_e_bef(t);

```

**6.1.1.3.1.409 units\_e\_hon**

```

if(submodel != "UNIT")
    return NO_AVG;

if(t >= maturity_period_ann)
    return 0.0;

return units_e_hon_active(t)
    + units_e_hon_pup(t);

```

**6.1.1.3.1.410 units\_e\_hon\_active**

```

if (t <= commence_period_w || t >= maturity_period_ann)
    return 0.0;

if(submodel != "UNIT" || (res_kitzba >= resinforce && resinforce > 0.0))
    return NO_AVG;

if (t==0) {
    if (paid_up == "N")
        return max(units_e(t)-res_kitzba * benefits_curr, 0) ;
    else
        return 0.0;
}

if (sm_accum->units_b(t) + sm_saving->units_b(t) <= 0.0)
    return 0.0;

```

```

//Pup to deduct
double new_pup = 0;

if (sm_accum->units_e(t-1) + sm_saving->units_e(t-1) > 0)
    new_pup = (
        units_b_bef_pup_acc(t) * (1. - sm_accum-
>surrg_chg_perc_units[t+elapsed_months]/100.)
        + units_b_bef_pup_sav(t) * (1. - sm_saving-
>surrg_chg_perc_units[t+elapsed_months]/100.)
    )
    * units_e_hon_active(t-1)
    / (sm_accum->units_e(t-1) + sm_saving->units_e(t-1));

//if (t==1) log_strm<<"New pup: "<<new_pup<<endl;

double units_to_add = 0.0;

units_to_add = alloc_units_honi(t);

//if (t==1) log_strm<<"Units to add: "<<units_to_add<<endl;

double temp_inv_rate_m = 0.0;

int proj_yr = xint(proj_year(t));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t));
proj_yr = max(proj_yr, 0);

if (margin_add_asset == "Y" && t == 1 && submodel == "UNIT" && par_nonpar == "P")
    temp_inv_rate_m = asset_shock;
else
    temp_inv_rate_m = inv_rate_mth_t[proj_yr];

double other_deductions = 0;

if ((sm_accum->units_b_bef(t) + sm_saving->units_b_bef(t)) > 0.0)
    other_deductions = (sm_accum->cover_charge(t) + sm_saving->cover_charge(t))
        * (units_e_hon_active(t-1) - new_pup)
        / (sm_accum->units_b_bef(t) + sm_saving-
>units_b_bef(t));
    else
        other_deductions = other_deductions
            + (sm_accum->cover_charge(t) + sm_saving-
>cover_charge(t))
            * (1- res_prop_kitzba);

double int_cred = (units_e_hon_active(t-1) - new_pup + units_to_add - other_deductions) *
temp_inv_rate_m;

other_deductions = other_deductions
    + (sm_accum->management_fee(t) + sm_saving-
>management_fee(t)
    )
    * (units_e_hon_active(t-1) - new_pup +
units_to_add)
    / (sm_accum->units_b(t) + sm_saving-
>units_b(t));

```

```

other_deductions = other_deductions
                    +
                    (sm_accum->death_claims_units(t) + sm_saving-
>death_claims_units(t)
                    + sm_accum->claims_surrender(t) + sm_saving-
>claims_surrender(t)
                    - units_bon(t) * lapse_rate_act_bal(t)
                    + sm_accum->surr_charge(t) + sm_saving->surr_charge(t)
                    )
                    *
                    (units_e_hon_active(t-1) - new_pup + units_to_add + int_cred -
other_deductions)
                    / (sm_accum->units_e_bef(t) + sm_saving->units_e_bef(t));

return min(
    (units_e_hon_active(t-1)
    - new_pup
    + units_to_add
    + int_cred
    - other_deductions
    )
    * surv_per_ret(t),
    units_e(t)); //Final cannot be greater than units

```

#### 6.1.1.3.1.411 units\_e\_hon\_pup

```

if (t <= commence_period_w || t >= maturity_period_ann)
    return 0.0;

if(submodel != "UNIT" || (res_kitzba >= resinforce && resinforce > 0.0))
    return NO_AVG;

if (t==0) {
    //log_strm<<"Paid up: "<<paid_up<<endl;
    if (paid_up == "N")
        return 0.0;
    else
        return max(units_e(t)-res_kitzba * benefits_curr, 0) ;
}

if(paid_up == "Y" && paid_up_input=="N"){ //Adjust for scenario where paid_up = Y but units are in
active - treat all as one unit type

    if(units_b(t) <= 0.0)
        return 0.0;

    if(units_e_bef(t) <= 0.0)
        return 0.0;

    double new_bonus = 0;
    if (units_e(t-1) != 0)
        new_bonus = units_bon(t-1)
            * pup_rate_bal_dep(t-1)
            * surv_per_ret(t-1)
            * units_e_hon_pup(t-1)
            / units_e(t-1);

```

```

double temp_inv_rate_m = 0.0;

int proj_yr = xint(proj_year(t));
if(eq(projection_type_int, "Rollup"))
proj_yr = xint(proj_year_rollup(t));
proj_yr = max(proj_yr, 0);

if (margin_add_asset == "Y" && t == 1 && submodel == "UNIT" && par_nonpar == "P")
temp_inv_rate_m = asset_shock;
else
temp_inv_rate_m = inv_rate_mth_t[proj_yr];

double int_cred = (units_e_hon_pup(t-1) + new_bonus) * temp_inv_rate_m;

double other_deductions = 0;
if(units_b(t) != 0)
other_deductions = ( sm_accum->cover_charge(t) + sm_saving->cover_charge(t)
+ sm_acc_pup->cover_charge(t) +
sm_saving_pup->cover_charge(t)
+ management_fees_fixed_active(t)
+ management_fees_var_active(t)
)
* (units_e_hon_pup(t-1) + new_bonus)
/ units_b(t);

if (units_e_bef(t) != 0 )
other_deductions = other_deductions
+
(sm_accum->death_claims_units(t) + sm_saving-
>death_claims_units(t)
+ sm_acc_pup->death_claims_units(t) +
+ sm_accum->claims_surrender(t) + sm_saving-
>claims_surrender(t)
+ sm_acc_pup->claims_surrender(t) +
sm_saving_pup->claims_surrender(t)
- units_bon(t) * lapse_rate_act_bal(t)
+ surr_charge(t)
)
*
(units_e_hon_pup(t-1) + new_bonus +
alloc_units_honi(t) + int_cred - other_deductions)
/ units_e_bef(t);

return min(
(units_e_hon_pup(t-1)
+ new_bonus
+ alloc_units_honi(t)
+ int_cred
- other_deductions
)
* surv_per_ret(t),
units_e(t));
}

```

---

```

if (sm_acc_pup->units_b(t) + sm_saving_pup->units_b(t) <= 0)
    return 0.0;

if ((sm_acc_pup->units_e_bef(t) + sm_saving_pup->units_e_bef(t)) <= 0.0)
    return 0.0;

//Pup to deduct
double new_pup = 0.0;
double new_bonus = 0.0;

if (sm_accum->units_e(t-1) + sm_saving->units_e(t-1) > 0.0){
    if (paid_up == "N"){
        new_pup = (
            units_b_bef_pup_acc(t) * (1. - sm_accum-
>surrg_chg_perc_units[t+elapsed_months]/100.)
            + units_b_bef_pup_sav(t) * (1. - sm_saving-
>surrg_chg_perc_units[t+elapsed_months]/100.)
            )
            * units_e_hon_active(t-1) / (sm_accum->units_e(t-1) +
sm_saving->units_e(t-1));

        new_bonus = units_bon(t-1) * pup_rate_bal_dep(t-1) * surv_per_ret(t-1);

        new_bonus = new_bonus
            * units_e_hon_active(t-1) / (sm_accum->units_e(t-1) +
sm_saving->units_e(t-1));

        new_pup = new_pup + new_bonus;
    }
}

double temp_inv_rate_m = 0.0;

int proj_yr = xint(proj_year(t));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t));
proj_yr = max(proj_yr, 0);

if (margin_add_asset == "Y" && t == 1 && submodel == "UNIT" && par_nonpar == "P")
    temp_inv_rate_m = asset_shock;
else
    temp_inv_rate_m = inv_rate_mth_t[proj_yr];

double int_cred = (units_e_hon_pup(t-1) + new_pup) * temp_inv_rate_m;

double other_deductions = 0;
if (sm_acc_pup->units_b(t) + sm_saving_pup->units_b(t) != 0)
    other_deductions = (
        sm_acc_pup->cover_charge(t) + sm_saving_pup-
>cover_charge(t)
        + sm_acc_pup->management_fee(t) + sm_saving_pup-
>management_fee(t)
    )
    * (units_e_hon_pup(t-1) + new_pup)

```

```

/ (sm_acc_pup->units_b(t) + sm_saving_pup->units_b(t));
>units_b(t));

if (sm_acc_pup->units_e_bef(t) + sm_saving_pup->units_e_bef(t) != 0)
    other_deductions = other_deductions
        +
        (sm_acc_pup->death_claims_units(t) + sm_saving_pup->death_claims_units(t)
        + sm_acc_pup->claims_surrender(t) + sm_saving_pup->claims_surrender(t)
        + sm_acc_pup->surr_charge(t) + sm_saving_pup->surr_charge(t)
        )
        *
        (units_e_hon_pup(t-1) + new_pup + int_cred - other_deductions)
        / (sm_acc_pup->units_e_bef(t) + sm_saving_pup->units_e_bef(t));

return min(
    (units_e_hon_pup(t-1)
    + new_pup
    + int_cred
    - other_deductions
    )
    * surv_per_ret(t),
    units_e(t)); //Final cannot be greater than
units

```

#### 6.1.1.3.1.412 units\_e\_kiz

```

if(submodel != "UNIT")
    return NO_AVG;

if(t >= maturity_period_ann)
    return 0.0;

return max(units_e(t) - units_e_hon(t), 0);

```

#### 6.1.1.3.1.413 units\_e\_new

```

if(submodel != "UNIT")
    return NO_AVG;

if(t >= maturity_period_ann)
    return 0.0;

return max(0, units_e_kiz(t) - units_e_old(t));

```

#### 6.1.1.3.1.414 units\_e\_newtag

```

if(submodel != "UNIT")
    return NO_AVG;

return units_e(t) - (units_e_prat(t) + units_e_piz(t) + units_e_old(t) + units_e_hon(t));

```

**6.1.1.3.1.415 units\_e\_old**

```

if(submodel != "UNIT")
    return NO_AVG;

return units_e_old_active(t)
    + units_e_old_pup(t);

```

**6.1.1.3.1.416 units\_e\_old\_active**

```

if (t < 0 || t >= maturity_period_ann)
    return 0.0;

if(submodel != "UNIT" || res_kitzba <= 0.0 || paid_up != "N")
    return NO_AVG;

if (t==0) {
    // (note that all the factors below apply to old policies only [with guaranteed annuity
    rates], as new ones have annuitization_rate set to 0)
    if (paid_up == "N")
        return min(res_kitzba * benefits_curr, units_e(t)) * res_prop_kitzba_oldtag ;
    else
        return 0.0;
}

if (sm_accum->units_b(t) + sm_saving->units_b(t) <= 0.0)
    return 0.0;

if (sm_accum->units_e_bef(t) + sm_saving->units_e_bef(t) <=0.0)
    return 0.0;

if (sm_accum->units_b_bef(t) + sm_saving->units_b_bef(t) <=0.0)
    return 0.0;

//Pup to deduct
double new_pup = 0;

if (sm_accum->units_e(t-1) + sm_saving->units_e(t-1) > 0)
    new_pup = (
        units_b_bef_pup_acc(t) * (1. - sm_accum-
>surrg_chg_perc_units[t+elapsed_months]/100.)
        + units_b_bef_pup_sav(t) * (1. - sm_saving-
>surrg_chg_perc_units[t+elapsed_months]/100.)
        )
        * units_e_old_active(t-1)
        / (sm_accum->units_e(t-1) + sm_saving->units_e(t-1));

double temp_inv_rate_m = 0.0;

int proj_yr = xint(proj_year(t));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t));
proj_yr = max(proj_yr, 0);

if (margin_add_asset == "Y" && t == 1 && submodel == "UNIT" && par_nonpar == "P")
    temp_inv_rate_m = asset_shock;
else

```



```

    temp_inv_rate_m = inv_rate_mth_t[proj_yr];

double other_deductions = (sm_accum->cover_charge(t) + sm_saving->cover_charge(t))
                          * (units_e_old_active(t-1) - new_pup)
                          / (sm_accum->units_b_bef(t) + sm_saving->units_b_bef(t));

double int_cred = (units_e_old_active(t-1) - new_pup - other_deductions) * temp_inv_rate_m;

    other_deductions = other_deductions
                      + (sm_accum->management_fee(t) + sm_saving->management_fee(t))
                      * (units_e_old_active(t-1) - new_pup)
                      / (sm_accum->units_b(t) + sm_saving->units_b(t));

    other_deductions = other_deductions
                      + (sm_accum->death_claims_units(t) + sm_saving->death_claims_units(t)
                        + sm_accum->claims_surrender(t) + sm_saving->claims_surrender(t)
                        - units_bon(t) * lapse_rate_act_bal(t)
                        + sm_accum->surr_charge(t) + sm_saving->surr_charge(t)
                        )
                      * (units_e_old_active(t-1) - new_pup + int_cred - other_deductions)
                      / (sm_accum->units_e_bef(t) + sm_saving->units_e_bef(t));

return min(
    (units_e_old_active(t-1)
     - new_pup
     + int_cred
     - other_deductions
    )
    * surv_per_ret(t),
    units_e(t)); //Can't be more than units

```

#### 6.1.1.3.1.417 units\_e\_old\_pup

```

if (t < 0 || t >= maturity_period_ann)
    return 0.0;

if(submodel != "UNIT" || res_kitzba <= 0.0)
    return NO_AVG;

if (t==0) {
    // (note that all the factors below apply to old policies only [with guaranteed annuity
    rates], as new ones have annuitization_rate set to 0)
    if (paid_up == "Y")
        return min(res_kitzba * benefits_curr, units_e(t)) * res_prop_kitzba_oldtag ;
    else
        return 0.0;
}

```

```
if(paid_up == "Y" && paid_up_input=="N"){ //Adjust for scenario where paid_up = Y but units are in
active - treat all as one unit type
```

```

    if(units_b(t) <= 0.0)
        return 0.0;

    if(units_e_bef(t) <= 0.0)
        return 0.0;

    double new_bonus = units_bon(t-1)
                        * pup_rate_bal_dep(t-1)
                        * surv_per_ret(t-1)
                        * units_e_old_pup(t-1)
                        / units_e(t-1);

    double temp_inv_rate_m = 0.0;

    int proj_yr = xint(proj_year(t));
    if(eq(projection_type_int, "Rollup"))
        proj_yr = xint(proj_year_rollup(t));
    proj_yr = max(proj_yr, 0);

    if (margin_add_asset == "Y" && t == 1 && submodel == "UNIT" && par_nonpar == "P")
        temp_inv_rate_m = asset_shock;
    else
        temp_inv_rate_m = inv_rate_mth_t[proj_yr];

    double int_cred = (units_e_old_pup(t-1) + new_bonus) * temp_inv_rate_m;

    double other_deductions = (sm_accum->cover_charge(t) + sm_saving->cover_charge(t)
                                + sm_acc_pup->cover_charge(t) +
sm_saving_pup->cover_charge(t)
                                + management_fees_fixed_active(t)
                                + management_fees_var_active(t)
                                )
                                * (units_e_old_pup(t-1) + new_bonus)
                                / units_b(t);

    other_deductions = other_deductions
                        +
>death_claims_units(t)
                                (sm_accum->death_claims_units(t) + sm_saving-
>death_claims_units(t)
                                + sm_acc_pup->death_claims_units(t) + sm_saving_pup-
>claims_surrender(t)
                                + sm_accum->claims_surrender(t) + sm_saving-
>claims_surrender(t)
                                + sm_acc_pup->claims_surrender(t) + sm_saving_pup-
                                - units_bon(t) * lapse_rate_act_bal(t)
                                + surr_charge(t)
                                )
                                *
                                (units_e_old_pup(t-1) + new_bonus + int_cred -
other_deductions)
                                / units_e_bef(t);
```

```

        return min(
            (units_e_old_pup(t-1)
             + new_bonus
             + int_cred
             - other_deductions
            )
            * surv_per_ret(t),
            units_e(t));
    }

    if (sm_acc_pup->units_b(t) + sm_saving_pup->units_b(t) <= 0.0)
        return 0.0;

    if (sm_acc_pup->units_e_bef(t) + sm_saving_pup->units_e_bef(t) <= 0.0)
        return 0.0;

    //Pup to deduct
    double new_pup = 0.0;
    double new_bonus = 0.0;

    if (sm_accum->units_e(t-1) + sm_saving->units_e(t-1) > 0.0){
        if (paid_up == "N"){
            new_pup = (
                units_b_bef_pup_acc(t) * (1. - sm_accum-
>surrg_chg_perc_units[t+elapsed_months]/100.)
                + units_b_bef_pup_sav(t) * (1. - sm_saving-
>surrg_chg_perc_units[t+elapsed_months]/100.)
            )
            * units_e_old_active(t-1) / (sm_accum->units_e(t-1) +
sm_saving->units_e(t-1));

            new_bonus = units_bon(t-1) * pup_rate_bal_dep(t-1) * surv_per_ret(t-1);

            new_bonus = new_bonus
                * units_e_old_active(t-1) / (sm_accum->units_e(t-1) +
sm_saving->units_e(t-1));

            new_pup = new_pup + new_bonus;
        }
    }

    double temp_inv_rate_m = 0.0;

    int proj_yr = xint(proj_year(t));
    if(eq(projection_type_int, "Rollup"))
        proj_yr = xint(proj_year_rollup(t));
    proj_yr = max(proj_yr, 0);

    if (margin_add_asset == "Y" && t == 1 && submodel == "UNIT" && par_nonpar == "P")
        temp_inv_rate_m = asset_shock;
    else
        temp_inv_rate_m = inv_rate_mth_t[proj_yr];

```

```

double int_cred = (units_e_old_pup(t-1) + new_pup) * temp_inv_rate_m;

double other_deductions = (sm_acc_pup->cover_charge(t) + sm_saving_pup->cover_charge(t) +
                           sm_acc_pup->management_fee(t) + sm_saving_pup-
>management_fee(t)
                           )
                           * (units_e_old_pup(t-1) + new_pup)
                           / (sm_acc_pup->units_b(t) + sm_saving_pup-
>units_b(t));

other_deductions = other_deductions
+
(sm_acc_pup->death_claims_units(t) + sm_saving_pup-
>death_claims_units(t)
+ sm_acc_pup->claims_surrender(t) + sm_saving_pup-
>claims_surrender(t)
//- bonus_if(t) * lapse_rate(t) - not for pup
+ sm_acc_pup->surr_charge(t) + sm_saving_pup->surr_charge(t)
)
*
(units_e_old_pup(t-1) + new_pup + int_cred - other_deductions)
/ (sm_acc_pup->units_e_bef(t) + sm_saving_pup->units_e_bef(t));

return min(
    (units_e_old_pup(t-1)
    + new_pup
    + int_cred
    - other_deductions
    )
    * surv_per_ret(t),
    units_e(t)); //Can't be more than units

```

#### 6.1.1.3.1.418 units\_e\_piz

```

if(submodel != "UNIT")
    return NO_AVG;

double int_piz = 0.0;

if ((units_e_piz_int_active(t) + units_e_piz_int_pup(t)) < 0)
    int_piz = units_e_piz_int_active(t) + units_e_piz_int_pup(t);

return min(units_e_piz_active(t) + units_e_piz_pup(t) + int_piz,
    units_e(t));

```

#### 6.1.1.3.1.419 units\_e\_piz\_active

```

// פעילים פיצויים עבור תשואה ללא צבירה

if (t <= commence_period_w || t >= maturity_period_ann)
    return 0.0;

if(submodel != "UNIT" || units_e_new(t) <= 0)
    return NO_AVG;

```

```

if(t== 0){
    if (paid_up == "N")
        return min(res_kitzba * benefits_curr, units_e(t)) * res_prop_kitzba_piz ;
    else
        return 0.0;
}

if (sm_accum->units_b(t) + sm_saving->units_b(t) <= 0.0)
    return 0.0;

if (sm_accum->units_e_bef(t) + sm_saving->units_e_bef(t) <=0.0)
    return 0.0;

double piz = units_e_piz_active(t-1);

//Pup to deduct
double new_pup = 0;

if (sm_accum->units_e(t-1) + sm_saving->units_e(t-1) > 0)
    new_pup = (
        units_b_bef_pup_acc(t) * (1. - sm_accum-
>surr_chg_perc_units[t+elapsed_months]/100.)
        + units_b_bef_pup_sav(t) * (1. - sm_saving-
>surr_chg_perc_units[t+elapsed_months]/100.)
    )
    * units_e_piz_active(t-1)
    / (sm_accum->units_e(t-1) + sm_saving->units_e(t-1));

piz = piz - new_pup + alloc_units_piz(t);

double other_deductions = 0;

if ((sm_accum->units_b_bef(t) + sm_saving->units_b_bef(t)) > 0.0)
    other_deductions = other_deductions
        +
        (sm_accum->cover_charge(t) + sm_saving-
>cover_charge(t))
        * (units_e_piz_active(t-1) - new_pup)
        / (sm_accum->units_b_bef(t) + sm_saving-
>units_b_bef(t));
    else{
        if(alloc_units_piz(t) > 0.0)
            other_deductions = other_deductions
                + (sm_accum->cover_charge(t) + sm_saving-
>cover_charge(t))
                * piz
                / alloc_units(t);
    }

piz = piz - other_deductions;

```

```

other_deductions = (sm_accum->death_claims_units(t) + sm_saving->death_claims_units(t)
                    + sm_accum->claims_surrender(t) + sm_saving->
>claims_surrender(t)
                    - units_bon(t) * lapse_rate_act_bal(t)
                    + sm_accum->surr_charge(t) + sm_saving->surr_charge(t)
                    )
                    *
                    piz
                    / (sm_accum->units_e_bef(t) + sm_saving->units_e_bef(t));

piz = piz - other_deductions;

return max(    piz * surv_per_ret(t),
              0);

```

#### 6.1.1.3.1.420 units\_e\_piz\_newprems

```

if(submodel != "UNIT")
    return NO_AVG;

if (units_e_piz(t) <= 0 || units_e_bef(t)<= 0)
    return 0.0;

if (eq(policy_type, "private") || paid_up != "N")
    return 0.0;

if(t== 0)
    return 0.0;

//
//    return 0.0;

double piz = units_e_piz_newprems(t-1);
double prem_prop = 0.0;

if (alloc_units(t) > 0)
    prem_prop = alloc_units_piz(t) / alloc_units(t);

piz = piz + alloc_units_piz(t)
        - prem_prop * (cover_charge(t)); //Prem-related charges

double other_deductions = (sm_accum->death_claims_units(t) + sm_saving->death_claims_units(t) +
sm_acc_pup->death_claims_units(t) + sm_saving_pup->death_claims_units(t)
                    + sm_accum->claims_surrender(t) + sm_saving->
>claims_surrender(t) + sm_acc_pup->claims_surrender(t) + sm_saving_pup->claims_surrender(t)
                    - units_bon(t) * lapse_rate_act_bal(t)
                    + surr_charge(t)
                    )
                    *
                    piz
                    / units_e_bef(t); //Other decrements - assume
proportionate to whole of units

piz = piz - other_deductions;

```

```
return min(    piz * surv_per_ret(t),
              units_e_piz(t)); //Final cannot be greater than units piz active
```

#### 6.1.1.3.1.421 units\_e\_piz\_pup

```
if (t < 0 || t >= maturity_period_ann)
    return 0.0;

if(submodel != "UNIT" || units_e_new(t) <= 0)
    return NO_AVG;

if(t== 0){

    if (paid_up != "N")
        return min(units_e(t), res_kitzba * benefits_curr) * res_prop_kitzba_piz ;
    else
        return 0.0;
}

if(paid_up == "Y" && paid_up_input=="N"){ //Adjust for scenario where paid_up = Y but units are in
active - treat all as one unit type

    if(units_b(t) <= 0.0)
        return 0.0;

    if(units_e_bef(t) <= 0.0)
        return 0.0;

    double piz = units_e_piz_pup(t-1);

    double new_bonus = units_bon(t-1)
        * pup_rate_bal_dep(t-1)
        * surv_per_ret(t-1)
        * piz
        / units_e(t-1);

    piz = piz + new_bonus;

    double other_deductions = (
        sm_accum->cover_charge(t) + sm_saving-
>cover_charge(t)
        + sm_acc_pup->cover_charge(t) +
sm_saving_pup->cover_charge(t)
        //+ management_fee(t) Moved to new_tag
        )
        * piz
        / units_b(t);

    piz = piz + alloc_units_piz(t) - other_deductions;

    other_deductions = (sm_accum->death_claims_units(t) + sm_saving->death_claims_units(t)
        + sm_acc_pup->death_claims_units(t) + sm_saving_pup-
>death_claims_units(t)
        + sm_accum->claims_surrender(t) + sm_saving-
>claims_surrender(t)
```

```

+ sm_acc_pup->claims_surrender(t) + sm_saving_pup-
>claims_surrender(t)
- units_bon(t) * lapse_rate_act_bal(t)
+ surr_charge(t)
)
* piz
/ units_e_bef(t);

piz = piz - other_deductions;

return max(    piz * surv_per_ret(t),
                                0);

}

if (sm_acc_pup->units_b(t) + sm_saving_pup->units_b(t) <= 0.0)
    return 0.0;

if (sm_acc_pup->units_e_bef(t) + sm_saving_pup->units_e_bef(t) <=0.0)
    return 0.0;

double piz = units_e_piz_pup(t-1);

//Pup to deduct
double new_pup = 0.0;
double new_bonus = 0.0;

if (sm_accum->units_e(t-1) + sm_saving->units_e(t-1) > 0.0){
    if (paid_up == "N"){
        new_pup = (
                                units_b_bef_pup_acc(t) * (1. - sm_accum-
>surr_chg_perc_units[t+elapsed_months]/100.)
                                + units_b_bef_pup_sav(t) * (1. - sm_saving-
>surr_chg_perc_units[t+elapsed_months]/100.)
                                )
                                * units_e_piz_active(t-1) / (sm_accum->units_e(t-1) +
sm_saving->units_e(t-1));

        new_bonus = units_bon(t-1) * pup_rate_bal_dep(t-1) * surv_per_ret(t-1);

        new_bonus = new_bonus
                                * units_e_piz_active(t-1) / (sm_accum->units_e(t-1) +
sm_saving->units_e(t-1));

        new_pup = new_pup + new_bonus;
    }
}

piz = piz + new_pup;

double other_deductions = (

```



```

sm_acc_pup->cover_charge(t) + sm_saving_pup-
>cover_charge(t)
sm_saving_pup->management_fee(t) moved to new_tag
//+ sm_acc_pup->management_fee(t) +
)
* piz
/ (sm_acc_pup->units_b(t) + sm_saving_pup-
>units_b(t));

piz = piz - other_deductions;

other_deductions = (sm_acc_pup->death_claims_units(t) + sm_saving_pup->death_claims_units(t)
+ sm_acc_pup->claims_surrender(t) + sm_saving_pup-
>claims_surrender(t)
//- bonus_if(t) * lapse_rate(t) Not for pup
+ sm_acc_pup->surr_charge(t) + sm_saving_pup->surr_charge(t)
)
*
piz
/ (sm_acc_pup->units_e_bef(t) + sm_saving_pup->units_e_bef(t));

piz = piz - other_deductions;

return max( piz * surv_per_ret(t),
            0);

```

#### 6.1.1.3.1.422 units\_e\_prat

```

if(submodel != "UNIT")
    return NO_AVG;

return min(units_e_prat_active(t) + units_e_prat_pup(t),
            units_e(t));

```

#### 6.1.1.3.1.423 units\_e\_prat\_active

```

if (t < 0 || t >= maturity_period_ann)
    return 0.0;

if(submodel != "UNIT" || units_e_new(t) <= 0)
    return NO_AVG;

if(t== 0){
    if (paid_up == "N")
        return min(res_kitzba * benefits_curr, units_e(t)) * res_prop_kitzba_prat ;
    else
        return 0.0;
}

if (sm_accum->units_b(t) + sm_saving->units_b(t) <= 0.0)
    return 0.0;

if (sm_accum->units_e_bef(t) + sm_saving->units_e_bef(t) <=0.0)
    return 0.0;

```

```

double prat = units_e_prat_active(t-1);

//Pup to deduct
double new_pup = 0;

if (sm_accum->units_e(t-1) + sm_saving->units_e(t-1) > 0)
    new_pup = (
        units_b_bef_pup_acc(t) * (1. - sm_accum-
>surrg_chg_perc_units[t+elapsed_months]/100.)
        + units_b_bef_pup_sav(t) * (1. - sm_saving-
>surrg_chg_perc_units[t+elapsed_months]/100.)
    )
    * units_e_prat_active(t-1)
    / (sm_accum->units_e(t-1) + sm_saving->units_e(t-1));

prat = prat - new_pup + alloc_units_prat(t);

double temp_inv_rate_m = 0.0;

int proj_yr = xint(proj_year(t));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t));
proj_yr = max(proj_yr, 0);

if (margin_add_asset == "Y" && t == 1 && submodel == "UNIT" && par_nonpar == "P")
    temp_inv_rate_m = asset_shock;
else
    temp_inv_rate_m = inv_rate_mth_t[proj_yr];

double other_deductions = 0.0;

if ((sm_accum->units_b_bef(t) + sm_saving->units_b_bef(t)) > 0.0)
    other_deductions = (sm_accum->cover_charge(t) + sm_saving->cover_charge(t))
        * (units_e_prat_active(t-1) - new_pup)
        / (sm_accum->units_b_bef(t) + sm_saving-
>units_b_bef(t));
    else
        other_deductions = (sm_accum->cover_charge(t) + sm_saving->cover_charge(t))
            * prat
            / (sm_accum->units_b(t) + sm_saving-
>units_b(t));

prat = prat - other_deductions;

double int_cred = prat * temp_inv_rate_m;

other_deductions = (
    //sm_accum->cover_charge(t) + sm_saving-
>cover_charge(t)
    + sm_accum->management_fee(t) + sm_saving-
>management_fee(t)
)
    * prat
    / (sm_accum->units_b(t) + sm_saving-
>units_b(t));

```

```

prat = prat + int_cred - other_deductions;

other_deductions = (sm_accum->death_claims_units(t) + sm_saving->death_claims_units(t)
                    + sm_accum->claims_surrender(t) + sm_saving->
>claims_surrender(t)
                    - units_bon(t) * lapse_rate_act_bal(t)
                    + sm_accum->surr_charge(t) + sm_saving->surr_charge(t)
                    )
                    *
                    prat
                    / (sm_accum->units_e_bef(t) + sm_saving->units_e_bef(t));

prat = prat - other_deductions;

return min(    prat * surv_per_ret(t),
              units_e(t)); //Final cannot be greater than units

```

#### 6.1.1.3.1.424 units\_e\_prat\_newprems

```

if(submodel != "UNIT")
    return NO_AVG;

if (!eq(policy_type, "private") || paid_up != "N" || res_kitzba == 0.0 || units_e_prat(t) <= 0 ||
units_b(t) <= 0 || units_e_bef(t) || t==0)
    return 0.0;

double prat = units_e_prat_newprems(t-1);

prat = prat + alloc_units_prat(t)
        - cover_charge(t);

double temp_inv_rate_m = 0.0;

int proj_yr = xint(proj_year(t));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t));
proj_yr = max(proj_yr, 0);

if (margin_add_asset == "Y" && t == 1 && submodel == "UNIT" && par_nonpar == "P")
    temp_inv_rate_m = asset_shock;
else
    temp_inv_rate_m = inv_rate_mth_t[proj_yr];

double other_deductions = 0.0;

double int_cred = prat * temp_inv_rate_m;

other_deductions = (management_fees_fixed_active(t) + management_fees_var_active(t))
                    * prat
                    / units_b(t);

prat = prat + int_cred - other_deductions;

```

```

other_deductions = (sm_accum->death_claims_units(t) + sm_saving->death_claims_units(t) +
sm_acc_pup->death_claims_units(t) + sm_saving_pup->death_claims_units(t)
                    + sm_accum->claims_surrender(t) + sm_saving-
>claims_surrender(t) + sm_acc_pup->claims_surrender(t) + sm_saving_pup->claims_surrender(t)
                    - units_bon(t) * lapse_rate_act_bal(t)
                    + surr_charge(t)
                    )
                    *
                    prat
                    / units_e_bef(t);

prat = prat - other_deductions;

return min(    prat * surv_per_ret(t),
              units_e_prat(t)); //Final cannot be greater than active prat units

```

#### 6.1.1.3.1.425 units\_e\_prat\_pup

```

if (t < 0 || t >= maturity_period_ann)
    return 0.0;

if(submodel != "UNIT" || units_e_new(t) <= 0)
    return NO_AVG;

if(t== 0){
    if (paid_up != "N")
        return min(res_kitzba * benefits_curr, units_e(t)) * res_prop_kitzba_prat ;
    else
        return 0.0;
}

if(paid_up == "Y" && paid_up_input=="N"){ //Adjust for scenario where paid_up = Y but units are in
active - treat all as one unit type

    if(units_b(t) <= 0.0)
        return 0.0;

    if(units_e_bef(t) <= 0.0)
        return 0.0;

    double prat = units_e_prat_pup(t-1);

    double new_bonus = units_bon(t-1)
                        * pup_rate_bal_dep(t-1)
                        * surv_per_ret(t-1)
                        * prat
                        / units_e(t-1);

    prat = prat + new_bonus;

    double temp_inv_rate_m = 0.0;

    int proj_yr = xint(proj_year(t));
    if(eq(projection_type_int, "Rollup"))
        proj_yr = xint(proj_year_rollup(t));

```

```

proj_yr = max(proj_yr, 0);

if (margin_add_asset == "Y" && t == 1 && submodel == "UNIT" && par_nonpar == "P")
    temp_inv_rate_m = asset_shock;
else
    temp_inv_rate_m = inv_rate_mth_t[proj_yr];

double int_cred = prat * temp_inv_rate_m;

double other_deductions = (
    sm_accum->cover_charge(t) + sm_saving-
>cover_charge(t)
    + sm_acc_pup->cover_charge(t) +
sm_saving_pup->cover_charge(t)
    + management_fees_fixed_active(t)
    + management_fees_var_active(t)
    )
    * prat
    / units_b(t);

prat = prat + alloc_units_prat(t) + int_cred - other_deductions;

other_deductions = (sm_accum->death_claims_units(t) + sm_saving->death_claims_units(t)
    + sm_acc_pup->death_claims_units(t) + sm_saving_pup-
>death_claims_units(t)
    + sm_accum->claims_surrender(t) + sm_saving-
>claims_surrender(t)
    + sm_acc_pup->claims_surrender(t) + sm_saving_pup-
>claims_surrender(t)
    - units_bon(t) * lapse_rate_act_bal(t)
    + surr_charge(t)
    )
    * prat
    / units_e_bef(t);

prat = prat - other_deductions;

return min(    prat * surv_per_ret(t),
    units_e(t)); //Final cannot be greater
than units
}

if (sm_acc_pup->units_b(t) + sm_saving_pup->units_b(t) <= 0.0)
    return 0.0;

if (sm_acc_pup->units_e_bef(t) + sm_saving_pup->units_e_bef(t) <= 0.0)
    return 0.0;

double prat = units_e_prat_pup(t-1);

//Pup to deduct
double new_pup = 0.0;
double new_bonus = 0.0;

if (sm_accum->units_e(t-1) + sm_saving->units_e(t-1) > 0.0){
    if (paid_up == "N"){

```

```

        new_pup = (
            units_b_bef_pup_acc(t) * (1. - sm_accum-
>surrg_chg_perc_units[t+elapsed_months]/100.)
            + units_b_bef_pup_sav(t) * (1. - sm_saving-
>surrg_chg_perc_units[t+elapsed_months]/100.)
        )
        * units_e_prat_active(t-1) / (sm_accum->units_e(t-1) +
sm_saving->units_e(t-1));

        new_bonus = units_bon(t-1) * pup_rate_bal_dep(t-1) * surv_per_ret(t-1);

        new_bonus = new_bonus
            * units_e_prat_active(t-1) / (sm_accum->units_e(t-1) +
sm_saving->units_e(t-1));

        new_pup = new_pup + new_bonus;
    }
}

prat = prat + new_pup;

double temp_inv_rate_m = 0.0;

int proj_yr = xint(proj_year(t));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t));
proj_yr = max(proj_yr, 0);

if (margin_add_asset == "Y" && t == 1 && submodel == "UNIT" && par_nonpar == "P")
    temp_inv_rate_m = asset_shock;
else
    temp_inv_rate_m = inv_rate_mth_t[proj_yr];

double int_cred = prat * temp_inv_rate_m;

double other_deductions = (
    sm_acc_pup->cover_charge(t) + sm_saving_pup-
>cover_charge(t)
    + sm_acc_pup->management_fee(t) + sm_saving_pup-
>management_fee(t)
    )
    * prat
    / (sm_acc_pup->units_b(t) + sm_saving_pup-
>units_b(t));

prat = prat + int_cred - other_deductions;

other_deductions = (sm_acc_pup->death_claims_units(t) + sm_saving_pup->death_claims_units(t)
    + sm_acc_pup->claims_surrender(t) + sm_saving_pup-
>claims_surrender(t)
    + sm_acc_pup->surr_charge(t) + sm_saving_pup->surr_charge(t)
    )

```

```

        *
        prat
        / (sm_acc_pup->units_e_bef(t) + sm_saving_pup->units_e_bef(t));

prat = prat - other_deductions;

return min(    prat * surv_per_ret(t),
             units_e(t)); //Final cannot be greater than units

```

#### 6.1.1.3.1.426 prem\_termination\_prop

```

if(submodel == "ANN" || submodel == "TERM")
    return 1;

if(t <= 0)
    return 0; //this is to allow for cases where policy holder enters at the exact age, e.g. 64
and when RI prems are calculated.

if(t < mat_period_min || t > maturity_period_w)
    return 0.;

if(mult_age_ind == 1.){
    if(retirement_age_lookup(1) > sm_annuity[sm_annuity.size()-1]->takeup_age)
        return 1.;

    if(xint(pol_month(t)) == 12 && paid_up == "N"){
        return prem_termination_rate/100.;
    }
    return 0.;
}

return 1.;

```

#### 6.1.1.3.1.427 mort\_year

```

return year_prod + pol_year(t)-1;

```

#### 6.1.1.3.1.428 basic\_perc

```

if (t < commence_period_w || t > maturity_period_w)
    return NO_AVG;

if (!eq(submodel,"UNIT"))
    return 1.0;

if (!eq(ben_class,"adif") || sum_ins_curr<0.0001)
    return min(1.0,basic_perc_w/100.);

if (t >= 1) {
    double basic_SA = sum_ins_basic_tt(xint(age_last(t)),sex_smoker_code) * surv_ret(t-1);
    if (basic_SA >0) {
        double adifSAR = sum_insured(t)*surv_act_prm(t-1) - sm_accum->units_b_bef(t) -
sm_saving->units_b_bef(t) -
        premium_if_b(t)/prem_freq*accum->allocation_rate(t);
        return max(0.0, min(1.0, adifSAR / basic_SA) );
    }
}

```

```
//else  
return 0.0;
```

#### **6.1.1.3.1.429 age\_last**

```
if (t <= commence_period_w || t > maturity_period_w)  
    return NO_AVG;  
  
if (t == 1 - elapsed_months)  
    return floor(age_at_issue);  
  
if (pol_month(t) == 1)  
    return age_last(t-1) + 1.0;  
  
return age_last(t-1) ;
```

#### **6.1.1.3.1.430 interest\_re\_lrc\_q1**

```
if (t < 1 || t > 3 || t > maturity_period_ann || eq(paid_up,"C"))  
    return NO_AVG;  
  
if(ben_class == "phi" || prod_code == "ltc-shil")  
  
    return interest_re(t);  
  
return 0.0;
```

#### **6.1.1.3.1.431 interest\_re\_lrc\_q2**

```
if (t < 4 || t > 6 || t > maturity_period_ann || eq(paid_up,"C"))  
    return NO_AVG;  
  
if(ben_class == "phi" || prod_code == "ltc-shil")  
  
    return interest_re(t);  
  
return 0.0;
```

#### **6.1.1.3.1.432 interest\_re\_lrc\_q3**

```
if (t < 7 || t > 9 || t > maturity_period_ann || eq(paid_up,"C"))  
    return NO_AVG;  
  
if(ben_class == "phi" || prod_code == "ltc-shil")  
  
    return interest_re(t);  
  
return 0.0;
```

#### **6.1.1.3.1.433 interest\_re\_lrc\_q4**

```
if (t < 10 || t > 12 || t > maturity_period_ann || eq(paid_up,"C"))  
    return NO_AVG;  
  
if(ben_class == "phi" || prod_code == "ltc-shil")  
  
    return interest_re(t);  
  
return 0.0;
```



**6.1.1.3.1.434 interest\_re\_lrc\_yr2plus**

```

if ( t < 13 || t > maturity_period_ann || eq(paid_up,"C"))
    return NO_AVG;

if(ben_class == "phi" || prod_code == "ltc-shil")

    return interest_re(t);

return 0.0;

```

**6.1.1.3.1.435 riskadj\_gross**

```

if (t < 0 || t > maturity_period_ann)
    return NO_AVG;

double dis_component=0, lap_component=0, mort_component=0, long_component=0, exp_component=0,
tu_component=0;

// 1. Disability

if(eq(ben_class, "phi"))
    dis_component = ra_fact_dis_incid_gross / 100 * claim_cost_pv(t) + ra_fact_dis_termi_gross /
100 * claims_disability_pv(t);
else
    dis_component = ra_fact_dis_incid_gross / 100 * claims_disability_pv(t) ;

// 2. Lapse
if(paid_up=="G" || paid_up=="C" || free_inv_prop_t[1]<1) /*We exclude Invest Guaranteed because
of circular reference with Investment income. */
    lap_component = 0.0;
else
    lap_component = ra_fact_lapse_gross / 100 * profit_book_vif_pv_pos(t);

// 3. Mortality
    mort_component = ra_fact_mort_gross / 100 * claims_death_pv(t);

// 4. Longevity
if (eq(savings_pol_prod_code, "Y"))
    long_component = ra_fact_long_gross / 100 * claims_annuity_pv(t);
else
    long_component = ra_fact_long_gross / 100 * claims_disability_pv(t);

// 5. Expenses
exp_component = ra_fact_exp_gross / 100 * expense_pv(t);

// 6. Take up
tu_component = ra_fact_tu_gross / 100 * be_retire(t);

return
    dis_component + lap_component + mort_component + long_component + exp_component +
tu_component;

```

**6.1.1.3.1.436 riskadj\_gross\_rel\_q1**

```
if (t < 1 || t > 3 || t > maturity_period_ann || eq(paid_up,"C"))
    return NO_AVG;

if(ben_class == "phi" || prod_code == "ltc-shil")

    return riskadj_gross_rel_total(t);

return 0.0;
```

**6.1.1.3.1.437 riskadj\_gross\_rel\_q2**

```
if (t < 4 || t > 6 || t > maturity_period_ann || eq(paid_up,"C"))
    return NO_AVG;

if(ben_class == "phi" || prod_code == "ltc-shil")

    return riskadj_gross_rel_total(t);

return 0.0;
```

**6.1.1.3.1.438 riskadj\_gross\_rel\_q3**

```
if (t < 7 || t > 9 || t > maturity_period_ann || eq(paid_up,"C"))
    return NO_AVG;

if(ben_class == "phi" || prod_code == "ltc-shil")

    return riskadj_gross_rel_total(t);

return 0.0;
```

**6.1.1.3.1.439 riskadj\_gross\_rel\_q4**

```
if (t < 10 || t > 12 || t > maturity_period_ann || eq(paid_up,"C"))
    return NO_AVG;

if(ben_class == "phi" || prod_code == "ltc-shil")

    return riskadj_gross_rel_total(t);

return 0.0;
```

**6.1.1.3.1.440 riskadj\_gross\_rel\_total**

```
if (t <= commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if (t == 0)
    return 0;

int proj_yr = xint(proj_year(t));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t));

// end of period

return riskadj_gross(t-1) / v_month_t[proj_yr] - riskadj_gross(t);
```

**6.1.1.3.1.441 riskadj\_gross\_rel\_yr2plus**

```
if ( t < 13 || t > maturity_period_ann || eq(paid_up,"C"))
    return NO_AVG;

if(ben_class == "phi" || prod_code == "ltc-shil")

    return riskadj_gross_rel_total(t);

return 0.0;
```

**6.1.1.3.1.442 riskadj\_net**

```
if (t <= commence_period_w || t > maturity_period_ann)
    return NO_AVG;

return riskadj_gross(t) - riskadj_re(t);
```

**6.1.1.3.1.443 riskadj\_re**

```
if (t < 0 || t > maturity_period_ann)
    return NO_AVG;

double dis_component=0, lap_component=0, mort_component=0, long_component=0, exp_component=0,
tu_component=0;

// 1. Disability

if(eq(ben_class, "phi"))
    dis_component = ra_fact_dis_incid_reins / 100 * claim_cost_re_pv(t) +
ra_fact_dis_termi_reins / 100 * rein_claims_pv(t);
    else
        dis_component = ra_fact_dis_incid_reins / 100 * rein_claims_pv(t) ;

// 2. Lapse
if(paid_up=="G" || paid_up=="C" || free_inv_prop_t[1]<1)
    lap_component = 0.0;
    else
        lap_component = ra_fact_lapse_reins / 100 * cashflow_re_pv(t);

// 3. Mortality
    mort_component = ra_fact_mort_reins / 100 * rein_claims_pv(t);

// 4. Longevity
if (eq(savings_pol_prod_code, "Y"))
    long_component = 0;
    else
        long_component = ra_fact_long_reins / 100 * rein_claims_pv(t);

// 5. Expenses
exp_component = 0;

// 6. Take up
tu_component = 0;
```

```
return
    dis_component + lap_component + mort_component + long_component + exp_component +
    tu_component;
```

#### **6.1.1.3.1.444 riskadj\_re\_rel\_q1**

```
if (t < 1 || t > 3 || t > maturity_period_ann || eq(paid_up,"C"))
    return NO_AVG;
```

```
if(ben_class == "phi" || prod_code == "ltc-shil")
```

```
    return riskadj_re_rel_total(t);
```

```
return 0.0;
```

#### **6.1.1.3.1.445 riskadj\_re\_rel\_q2**

```
if (t < 4 || t > 6 || t > maturity_period_ann || eq(paid_up,"C"))
    return NO_AVG;
```

```
if(ben_class == "phi" || prod_code == "ltc-shil")
```

```
    return riskadj_re_rel_total(t);
```

```
return 0.0;
```

#### **6.1.1.3.1.446 riskadj\_re\_rel\_q3**

```
if (t < 7 || t > 9 || t > maturity_period_ann || eq(paid_up,"C"))
    return NO_AVG;
```

```
if(ben_class == "phi" || prod_code == "ltc-shil")
```

```
    return riskadj_re_rel_total(t);
```

```
return 0.0;
```

#### **6.1.1.3.1.447 riskadj\_re\_rel\_q4**

```
if (t < 10 || t > 12 || t > maturity_period_ann || eq(paid_up,"C"))
    return NO_AVG;
```

```
if(ben_class == "phi" || prod_code == "ltc-shil")
```

```
    return riskadj_re_rel_total(t);
```

```
return 0.0;
```

#### **6.1.1.3.1.448 riskadj\_re\_rel\_total**

```
if (t <= commence_period_w || t > maturity_period_ann)
    return NO_AVG;
```

```
if (t == 0)
    return 0;
```

```
int proj_yr = xint(proj_year(t));
if(eq(projection_type_int, "Rollup"))
```

```

proj_yr = xint(proj_year_rollup(t));

// end of period

return riskadj_re(t-1) / v_month_t[proj_yr] - riskadj_re(t);

```

#### 6.1.1.3.1.449 riskadj\_re\_rel\_yr2plus

```

if ( t < 13 || t > maturity_period_ann || eq(paid_up,"C"))
    return NO_AVG;

if(ben_class == "phi" || prod_code == "ltc-shil")

    return riskadj_re_rel_total(t);

return 0.0;

```

#### 6.1.1.3.1.450 premium\_if\_b

```

if(submodel == "ANN")
    return NO_AVG;

if (submodel=="TERM")
    return term->premium_if_b(t);

if (submodel=="TRAD")
    return trad->premium_if_b(t);

if (t <= commence_period_w || (t + elapsed_months) > prem_term
    || paid_up=="Y" || t > maturity_period_w)
    return 0.0;

return max(premium_if_b_total(t) - premium_if_riders(t)
    + premium_if_riders(0)*surv_act_prm(t-1),0.0);

```

#### 6.1.1.3.1.451 premium\_if\_b\_total

```

if (t <= commence_period_w || (t + elapsed_months) > prem_term
    || inlist(paid_up,"Y,C,G") || t > maturity_period_w)
    return 0.0;

double inc =1.0;
if(xint(pol_month(t))==1 && t>0)
    inc = (1 + premium_inc(t)/100);

double tat_shnatiut = 1.0;
/*if (mod_load_in_prem=="N")
    tat_shnatiut = (1 + life->mod_load_perc/100.);*/

if (t > 1) //prem increase applies on prem curr not on pol fees
    return premium_if_b_total(t-1)*inc * surv_per_act_prm(t-1) ;

if (t == 1)
    return inc* prem_curr * benefits_curr * tat_shnatiut;

// t < 1
if (gross_up_historic=="N" || (surv_per_act_prm(t)<0.000001))
    return premium_if_b_total(t+1);

```

```
else
    return (premium_if_b_total(t+1) / surv_per_act_prm(t));
```

#### 6.1.1.3.1.452 premium\_if\_e

```
if(submodel == "ANN")
    return NO_AVG;

if (submodel=="TERM")
    return term->premium_if_e(t);

if (submodel=="TRAD")
    return trad->premium_if_e(t);

if (t < commence_period_w || (t + elapsed_months) > prem_term
    || paid_up=="Y" || t > maturity_period_w)
    return 0.0;

if (t<0 && t == commence_period_w )
    return 0.0;

return premium_if_b(t+1);
```

#### 6.1.1.3.1.453 premium\_if\_riders

```
if(submodel == "ANN")
    return NO_AVG;

if (t <= commence_period_w || t > maturity_period_w || !eq(ben_class,"Adif") || risk_si<=0.000001
|| surv_prm(t-1)<=0) // no rider
    return 0;

// Calculate premium lookup year
int premium_year=1;
if (prem_lookup_freq_w[25]){
    if (xint(fmod(xint(pol_year(t)),prem_lookup_freq_w[25])) == 0)
        premium_year = prem_lookup_freq_w[25];
    else
        premium_year = xint(fmod(xint(pol_year(t)),prem_lookup_freq_w[25]));
}

// if not a renewal month, return previous premium reduced by survival
if (premium_year!=1 || xint(pol_month(t))!=1)
    return premium_if_riders(t-1) * surv_per_act_prm(t-1);

// else (renewal month), lookup new premium rate
col_char = sex + smoker_stat;
prem_rates_row = xint(age_last(t));
double rate = prem_rates_risk_rider;

return rate / si_unit_w[25]
    * risk_si
    * benefits_b_prm(t) * surv_act_prm(t-1) / surv_prm(t-1)
    * 1.04;
```

#### 6.1.1.3.1.454 alloc\_units\_honi

```
if (t <= commence_period_w || t > maturity_period_ann)
    return NO_AVG;
```

```
if (!eq(submodel,"UNIT"))
    return 0.0;

if (alloc_units(t) <= 0.0)
    return 0.0;

if (!eq(policy_type, "private"))
    return 0.0; //All new units allocated to prat

if (res_prop_kitzba > 0.0) //For prat, policies that have some kiz up to now continue to be
    kitzbati.
    return 0.0;

return alloc_units(t); //Only policies that are private and currently have no kitzbati
```

#### **6.1.1.3.1.455 alloc\_units\_newtag**

```
if (t <= commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if (!eq(submodel,"UNIT"))
    return 0.0;

if (alloc_units(t) <= 0.0)
    return 0.0;

if (eq(policy_type, "private"))
    return 0.0; //All new units allocated to prat

double prop = prem_newtag_prop / 100.;

return alloc_units(t)
    * prop;
```

#### **6.1.1.3.1.456 alloc\_units\_piz**

```
if (t <= commence_period_w || t > maturity_period_ann)
    return NO_AVG;

if (!eq(submodel,"UNIT"))
    return 0.0;

if (alloc_units(t) <= 0.0)
    return 0.0;

if (eq(policy_type, "private"))
    return 0.0; //All new units allocated to prat

double prop = prem_newtag_prop / 100.;

return alloc_units(t)
    * (1. - prop);
```

#### **6.1.1.3.1.457 alloc\_units\_prat**

```
if (t <= commence_period_w || t > maturity_period_ann)
    return NO_AVG;
```

```

if (!eq(submodel,"UNIT"))
    return 0.0;

if (alloc_units(t) <= 0.0)
    return 0.0;

if (!eq(policy_type, "private"))
    return 0.0; //All new units allocated to prat

if (res_prop_kitzba == 0.0) //For prat, policies that have 0 kiz up to now continue to be honi.
    return 0.0;

return alloc_units(t);

```

#### 6.1.1.3.1.458 premium\_inc

```

if (t <= life->commence_period_w || t > life->maturity_period_w)
    return NO_AVG;

```

```

if (eq(submodel,"UNIT")) {
    term_in_profil="N";
    return sal_tbl;
}

```

```

if (eq(submodel,"TERM")){
    return 0.0;
}

```

```

if (eq(submodel,"TRAD")) {
    return atof(prem_inc); //from prod_spec_trad
}

```

```

return 0.0;

```

#### 6.1.1.3.1.459 sum\_ins\_inc

```

if (t <= life->commence_period_w || t > life->maturity_period_w)
    return NO_AVG;

```

```

if (eq(paid_up,"Y")) // if paid-up then no increasing bonus
    return 0.0;

```

```

key_temp = "N";

```

```

if (eq(submodel,"UNIT")) {
    return 0.0;
}

```

```

if (eq(submodel,"TERM")){

    xstring base_code = prod_code_base(0,4);
    if (eq(base_code ,"prof"))
        key_temp = "Y";
}

```



```

        term_in_profil=key_temp;

    return sal_tbl;

}

if (eq(submodel,"TRAD")) {
    return atof(sum_inc); //from prod_spec_trad
}

return 0.0;

```

#### 6.1.1.3.1.460 sum\_ins\_inc\_acc

```

if (t <= life->commence_period_w || t > life->maturity_period_w)
    return NO_AVG;

if (!eq(submodel,"TERM") || !eq(paid_up,"N"))
    return 1.0;

double si_inc_pct = 0.0;

if (t <= 0)
    return 1.0;

if (xint(life->pol_month(t)) == 1 && (t>0))
    si_inc_pct = life->sum_ins_inc(t) / 100.;

return sum_ins_inc_acc(t-1)*(1+si_inc_pct);

```

#### 6.1.1.3.1.461 prem\_disc\_shimur\_rate

```

if (t <= 0 || t > maturity_period_w)
    return 0.0;

double new_rate = 0.;

if (t==1 || pol_month(t) == 1.)
    new_rate = prem_disc_shimur_im ;

return prem_disc_shimur_rate(t-1) + new_rate;

```

#### 6.1.1.3.1.462 be\_retire

```

if(t> maturity_period_ann )
    return NO_AVG;

if(!inlist(submodel,"UNIT,TRAD") || res_prop_kitzba <= 0.0)
    return NO_AVG;

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

if(mult_age_ind == 1)
    return (be_retire(t+1) + sm_annuity->units_for_takeup(t+1))
           * v_month_t[proj_yr] ;

```

```
return (be_retire(t+1) + sm_annuity[ann_index_map[takeup_age]]->units_for_takeup(t+1))
        * v_month_t[proj_yr] ;
```

#### 6.1.1.3.1.463 cashflow\_gross\_pv\_pos

```
if (t < commence_period_w || t > maturity_period_ann)
    return 0.0;
```

```
if (cashflow_pv(t) + cashflow_re_pv(t) > 0)
    return cashflow_pv(t) + cashflow_re_pv(t);
else
    return 0.0;
```

#### 6.1.1.3.1.464 cashflow\_pv

```
if (t < commence_period_w || t > maturity_period_ann)
    return 0.0;
```

```
if(mult_age_ind != 1){

    int proj_yr = xint(proj_year(t+1));
    if(eq(projection_type_int, "Rollup"))
        proj_yr = xint(proj_year_rollup(t+1));

    if ( t > maturity_period_w)
        return (cashflow_pv(t+1) + cashflow_e(t+1)) * ann_v_month_t[proj_yr] +
cashflow_b(t+1);
    else
        return (cashflow_pv(t+1) + cashflow_e(t+1)) * v_month_t[proj_yr] + cashflow_b(t+1);
}

return cashflow_pv_active(t)
        + cashflow_pv_deferred(t)
        + cashflow_pv_inpay(t);
```

#### 6.1.1.3.1.465 cashflow\_pv\_chetz

```
if (t < commence_period_w || t > maturity_period_ann || free_inv_prop_t[1] >= 1.)
    return 0.0;
```

```
if(mult_age_ind != 1){

    int proj_yr = xint(proj_year(t+1));
    if(eq(projection_type_int, "Rollup"))
        proj_yr = xint(proj_year_rollup(t+1));

    return cashflow_pv_ifrs(t)*(1.-max_chetz) + cashflow_pv_res(t)*(max_chetz) - riskadj_gross
(t);
}
```

```
return cashflow_pv_active_chetz(t)
        + cashflow_pv_deferred_chetz(t)
        + cashflow_pv_inpay_chetz(t) - riskadj_gross(t);
```

#### 6.1.1.3.1.466 cashflow\_pv\_e

```
if (t < commence_period_w || t > maturity_period_ann)
    return 0.0;
```

```

int proj_yr = xint(proj_year(t+1));
    if(eq(projection_type_int, "Rollup"))
        proj_yr = xint(proj_year_rollup(t+1));

if(mult_age_ind != 1){

    if ( t > maturity_period_w)
        return (cashflow_pv_e(t+1) + cashflow_e(t+1) + cashflow_b(t+1)) *
ann_v_month_t[proj_yr];
    else
        return (cashflow_pv_e(t+1) + cashflow_e(t+1) + cashflow_b(t+1)) *
v_month_t[proj_yr];
}

return cashflow_pv_active_e(t)
    + cashflow_pv_deferred_e(t)
    + cashflow_pv_inpay_e(t) ;

```

#### 6.1.1.3.1.467 cashflow\_pv\_pos

```

if (t < commence_period_w || t > maturity_period_ann)
    return 0.0;

if (cashflow_pv(t) > 0)
    return cashflow_pv(t);
else
    return 0.0;

```

#### 6.1.1.3.1.468 profit\_bk\_act\_vif\_pv

```

if (t < commence_period_w || t > maturity_period_ann)
    return 0.0;

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

return (profit_book_active_vif(t+1) + profit_bk_act_vif_pv(t+1))
    * v_month_t[proj_yr];

```

#### 6.1.1.3.1.469 profit\_book\_vif\_pv

```

if (t < commence_period_w || t >= maturity_period_ann)
    return 0.0;

if (mult_age_ind != 1){

    int proj_yr = xint(proj_year(t+1));
    if(eq(projection_type_int, "Rollup"))
        proj_yr = xint(proj_year_rollup(t+1));

    if ( t >= maturity_period_w)
        return (profit_book_vif(t+1) + profit_book_vif_pv(t+1)) * ann_v_month_t[proj_yr];

    return (profit_book_vif(t+1) + profit_book_vif_pv(t+1)) * v_month_t[proj_yr];
}

return profit_book_vif_pv_active(t)
    + profit_book_vif_pv_deferred(t)

```

```
+ profit_book_vif_pv_inpay(t);
```

### 6.1.1.3.1.470 profit\_net\_vif\_pv

```
if (t < commence_period_w || t >= maturity_period_ann)
    return 0.0;
```

```
// call additional scalars not calculated otherwise
call_extra_scalars();
```

```
//check why need to add these
double temp = cover_charge_pv(t);
temp = manage_fees_fixed_ann_pv(t);
temp = manage_fees_var_ann_pv(t);
temp = reserve_total_increase_pv(t);
temp = profit_book_vif_pv_pos(t);
temp = expense_ren_perc_pv(t);
temp = expense_ren_fix_pv(t);
temp = cashflow_pv_e(t);
temp = cashflow_pv_active_e(t);
temp = be_reserve(t);
temp = expense_pv_ann(t);
temp = comm_not_res_pv(t);
temp = claims_maturity_ret_pv(t);
temp = expense_investment_pv_post_ret(t);
temp = expense_investment_pv_bef_ret(t);
temp = expense_pv_active_no_inv(t);
temp = fvui(t);
temp = cashflow_pv_pos(t);
temp = cashflow_gross_pv_pos(t);
temp = expense_pv_active(t);
temp = claims_lrc_q1_pv(t);
temp = claims_lrc_q2_pv(t);
temp = claims_lrc_q3_pv(t);
temp = claims_lrc_q4_pv(t);
temp = claims_lrc_yr2plus_pv(t);
temp = rid_cashflow_pv(t);
temp = claim_cost_pv(t);
temp = total_bor_acc_pv(t);
temp = total_bor_return_pv(t);
temp = nogt_annpv(t);
```

```
//*****
```

```
if (mult_age_ind != 1){

    int proj_yr = xint(proj_year(t+1));
    if(eq(projection_type_int, "Rollup"))
        proj_yr = xint(proj_year_rollup(t+1));

    if (t >= maturity_period_w)
        return (profit_net_vif(t+1) + profit_net_vif_pv(t+1))* ann_v_month_t[proj_yr];

    return (profit_net_vif(t+1) + profit_net_vif_pv(t+1))* v_month_t[proj_yr];
}
```

```

return profit_net_vif_pv_active(t)
      + profit_net_vif_pv_deferred(t)
      + profit_net_vif_pv_inpay(t);

```

#### 6.1.1.3.1.471 ret\_prop\_col

```

if(mult_age_ind == 1){

if(retirement_age_lookup(t) < min_retirement_age)
    return 1.;

if(retirement_age_lookup(t) <= sm_annuity[sm_annuity.size()-1]->takeup_age)
    return ret_prop_array[ann_index_map[retirement_age_lookup(t)]];

return ret_prop_array[ann_index_map[sm_annuity.size()-1]];
}

return 1.;

```

#### 6.1.1.3.1.472 cashflow\_re\_b

```

return premium_re(t) - comm_re(t) - comm_re_prof(t);

```

#### 6.1.1.3.1.473 cashflow\_re\_e

```

return interest_re(t) - claims_re(t);

```

#### 6.1.1.3.1.474 claims\_re

```

if (submodel=="TERM")
    return term->claims_re(t);

if (submodel=="TRAD")
    return trad->claims_re(t);

if (t <= commence_period_w || t > maturity_period_w)
    return 0.0;

if (eq(ben_class,"adif")) {

    if (eq(re_type,"NONE"))
        return 0.0;

    return death_claim_si(t) * re_ratio_w;
}

else { // Profil

    double tot = 0.0;

    if (t>0)
        for (int i=0; i < riders_count_w; i++)
            tot = tot + rider_perc_allowed(t)/100.
                * claim_amount_tt(t,i) * surv_act_prm(t-1)
                * (1 - sm_riders[i]->retention); // *** r=t?

    return tot;
}

```

**6.1.1.3.1.475 comm\_re**

```
if(submodel == "ANN")
    return NO_AVG;

if (submodel=="TERM")
    return term->comm_re(t);

if (submodel=="TRAD")
    return trad->comm_re(t);

if (eq(re_type,"NONE"))
    return 0.0;

double res = 0.0; //regular commission

int yr;

if (atof(comm_by_cal)==1)
    yr=xint(cal_duration(t)+1);
else
    yr=xint(pol_year_ext(t));

res = comm_ren_re[yr] / 100. * premium_re(t) ;

return res;
```

**6.1.1.3.1.476 comm\_re\_prof**

```
if(submodel == "ANN")
    return NO_AVG;

if (submodel=="TERM")
    return term->comm_re_prof(t);

if (submodel=="TRAD")
    return trad->comm_re_prof(t);

return max(0,(comm_prof_re / 100. *
              (premium_re(t)
               - claims_re(t)
               - comm_re(t) )));
```

**6.1.1.3.1.477 interest\_re**

```
if (interest_re_calculate=="N")
    return 0.0;

if (submodel=="TERM")
    return term->interest_re(t);

if (submodel=="TRAD")
    return 0.0;

return NO_AVG;
```

**6.1.1.3.1.478 interest\_re\_pv**

```
if (t < commence_period_w || t > maturity_period_ann)
```

```

    return 0.0;

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

if( t > maturity_period_w){
    return      (interest_re_pv(t+1) + interest_re(t+1))
                * ann_v_month_t[proj_yr];
}

return      (interest_re_pv(t+1) + interest_re(t+1))
            * v_month_t[proj_yr];

```

#### 6.1.1.3.1.479 premium\_re

```

if(submodel == "ANN")
    return NO_AVG;

if(submodel == "TRAD")
    return trad->premium_re(t);

if(submodel == "TERM")
    return term->premium_re(t);

if (t <= commence_period_w || t > maturity_period_w)
    return 0.0;

double tot = 0.0;

if (eq(ben_class,"adif")) {

    double prate = 0.0;

    //Premium lookup definitions
    if(eq(re_type,"YRT")) {

        prate = prem_rates_re * (1+ max(health_occ_perc_min,health_occ_perc)/100.)      +
        prem_per_unit_si_re;

        return prate/prem_rate_scale_w *sum_at_risk_claim(t)*( re_ratio_w)/prem_freq;
    }

    if(eq(re_type,"OT"))
        return premium_if_b(t) * (re_ratio_w)/prem_freq ;

    if(eq(re_type,"NONE"))
        return 0.0;

}

// Profil
if (t>0)
    for (int i=0; i < riders_count_w; i++)
        tot = tot + rider_perc_allowed(t)/100.
                * claim_amount_tt(t,i) * surv_act_prm(t-1)
                * (1 - sm_riders[i]->retention)
                * (1.0 + re_cost_pc_rider[i]/100.0); // *** r=t?;

```

```
return tot;
```

#### **6.1.1.3.1.480      rein\_claims\_pv**

```
if (t < commence_period_w || t >= maturity_period_w)
    return 0.0;
```

```
int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));
```

```
return (rein_claims_pv(t+1) + claims_re(t+1))
        * v_month_t[proj_yr];
```

#### **6.1.1.3.1.481      rein\_comm\_pv**

```
if (t < commence_period_w || t >= maturity_period_w)
    return 0.0;
```

```
int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));
```

```
return rein_comm_pv(t+1)* v_month_t[proj_yr]
        + comm_re(t+1) + comm_re_prof(t+1);
```

#### **6.1.1.3.1.482      rein\_prem\_pv**

```
if (t < commence_period_w || t >= maturity_period_w)
    return 0.0;
```

```
int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));
```

```
return rein_prem_pv(t+1)* v_month_t[proj_yr]
        + premium_re(t+1);
```

#### **6.1.1.3.1.483      reserve\_re**

```
if(submodel == "TRAD")
    return 0.0;
```

```
if(submodel == "TERM")
    return term->reserve_re(t);
```

```
return NO_AVG;
```

#### **6.1.1.3.1.484      reserve\_re\_increase**

```
if (reserve_re_increase_calculate=="N")
    return 0.0;
```

```
if(submodel == "TRAD")
    return 0.0;
```

```
if(submodel == "TERM")
    return term->reserve_re_increase(t);
```

```
return NO_AVG;
```



**6.1.1.3.1.485    reserve\_re\_increase\_pv**

```

if (t <= commence_period_w || t > maturity_period_ann)
    return 0.0;

int proj_yr = xint(proj_year(t+1));
if(eq(projection_type_int, "Rollup"))
    proj_yr = xint(proj_year_rollup(t+1));

if ( t >= maturity_period_w)
    return      (reserve_re_increase_pv(t+1) + reserve_re_increase(t+1)) * ann_v_month_t[proj_yr]
;
return      (reserve_re_increase_pv(t+1) + reserve_re_increase(t+1)) * v_month_t[proj_yr] ;

```

**6.1.1.3.1.486    startup**

```

//***** EXTERNS Section *****
start_externs
    extern long max_errors;
    map <int, int> ann_index_map;
    map <int, double> ret_prop_map;
    SmartArray <double> ret_prop_array;
end_externs
// reset MoSes to allow more skipped model-points before cancelling run
max_errors = 50000;

// Check if scenario exists in ESG file
double temp=0.0;
risk_free_row_key=1;

    temp = inv_rate_m; //yield pre
    if (temp == 99999.)
        throw FatalError("yield pre not available in ESG assumption file for specific scenario");

    temp = disc_rate_m; //Discount Factor pre
    if (temp == 99999.)
        throw FatalError("Discount Factor pre not available in ESG assumption file for specific
scenario");

    temp = ann_inv_rate_m;          //yield post
    if (temp == 99999.)
        throw FatalError("yield post not available in ESG assumption file for specific scenario");

    temp = ann_disc_rate_m;        //Discount Factor post
    if (temp == 99999.)
        throw FatalError("Discount Factor post not available in ESG assumption file for specific
scenario");

//Set code variables

ben_class = ben_class_input;
mgt_fee_fixed = mgt_fee_fixed_input;
mgt_fee_variable = mgt_fee_variable_input;
dac_book_inforce = dac_book_inforce_input;
dac_book_adj_factor = dac_book_adj_factor_input;
dac_tax_inforce = dac_tax_inforce_input;
dac_tax_adj_factor = dac_tax_adj_factor_input;
benefit_term = benefit_term_input;

```

```

prem_curr = prem_curr_input;
prem_curr_if = prem_curr_input;
comm_renewal_year = comm_renewal_year_input;
policy_fee_if = policy_fee_input*gorem_mult;
prem_disc_perc = prem_disc_perc_input;
prem_disc_perc_2 = prem_disc_perc_2_input;
prem_disc_month = prem_disc_month_input;
prem_disc_month_2 = prem_disc_month_2_input;
sum_ins_curr = sum_ins_curr_input;
prem_term = prem_term_input;
rein_set = rein_set_input;
paid_up = paid_up_input;
unit_value_accum = unit_value_accum_input * (1+res_adj_factor);
unit_value_savings = unit_value_savings_input * (1+res_adj_factor);
surr_value_if = surr_value_if_input * (1+res_adj_factor);
res_kitzba = res_kitzba_input * (1+res_adj_factor);
resinforce = resinforce_input * (1+res_adj_factor);
riders_count_w = riders_count_w_input;
lapse_force_rate = lapse_force_rate_input;

if (use_tat_shnatiut_assum=="Y") // if tat_shnatiut taken from assumption file
    tat_shnatiut_rate = tat_shnatiut_assum;
else
    tat_shnatiut_rate = tat_shnatiut_input;

if ((policy_fee_if - prem_curr) < 1 && (policy_fee_if - prem_curr) > 0 )
    policy_fee_if=prem_curr;

if (rein_set == "132") //manual edit to data to avoid a new skip - not clear what this rein set is
from the data?
    rein_set = "0";

if(life->submodel == "TRAD"){

if(!eq(ben_class, "gimla") || (eq(ben_class, "gimla") && (life->sm_annuity[life->sm_annuity.size()-
1]->takeup_age < age_at_issue + elapsed_months/12)))

benefit_term = max(benefit_term,elapsed_months+1);

else if(eq(ben_class, "gimla"))

benefit_term = max(benefit_term,elapsed_months);

}

int i = 0;
for (i = 0; i <=100; i++) {

    comm_perc_res_a[i] = comm_perc_res_a_input;
    comm_perc_res_b[i] = comm_perc_res_b_input;
}

//***** Set various variables *****/

if (inlist(xstring(error_code),"0,2,4")) // error code from data file
    error_msg = "error_code_field="+xstring(error_code);

if(!eq(ben_class,"profil"))

```

```
riders_count_w=0;

if((eq(ben_class,"ltc") || eq(prod_code,"phi-mitriya")) && !eq(paid_up,"C"))
    use_phi_claims_cf="N";

prod_yr_w = year_prod;

if (eq(prog_name,"KLASI")) // set if modal loading is included in the premium field for level
premium products
    mod_load_in_prem = "N";
else
    mod_load_in_prem = "Y";

int tmp = ann_series;

death_rates ="CMI00" +sex + smoker_stat;

if (eq(pol_type_annuity_tu_switch,"Current"))
    pol_type_annuity_tu = policy_type;
else
    pol_type_annuity_tu = policy_type_orig;

if (eq(pol_type_comm_hekef_switch,"Current"))
    pol_type_comm_hekef = policy_type;
else
    pol_type_comm_hekef = policy_type_orig;

if (eq(pol_type_expenses_switch,"Current"))
    pol_type_expenses = policy_type;
else
    pol_type_expenses = policy_type_orig;

if (eq(pol_type_lapse_switch,"Current"))
    pol_type_lapse = policy_type;
else
    pol_type_lapse = policy_type_orig;

if (eq(pol_type_lapse_rider_switch,"Current"))
    pol_type_lapse_rider = policy_type;
else
    pol_type_lapse_rider = policy_type_orig;

if (eq(pol_type_phi_incidence_switch,"Current"))
    pol_type_phi_incidence = policy_type;
else
    pol_type_phi_incidence = policy_type_orig;

if (eq(pol_type_sal_inc_switch,"Current"))
    pol_type_sal_inc = policy_type;
else
    pol_type_sal_inc = policy_type_orig;
```

```

if (eq(pol_type_recovery_rates_switch,"Current"))
    pol_type_recovery_rates = policy_type;
else
    pol_type_recovery_rates = policy_type_orig;

set_by_prodcod();
set_from_tables();
set_from_data();

// adjust prem_curr to exclude policy fee
if(eq(submodel,"UNIT") && eq(done_startup_w,"false"))// prem_curr(by benefit) excludes policy
fees(by policy)
    prem_curr = prem_curr - policy_fee_if * policies_curr / benefits_curr;

set_other_variables();

//***** Annuity submodel purpose

set_accum_fund();
set_accum_pup_fund();
if (abs(basic_perc_w - 100.)> 0.00001 || (sum_ins_curr>0 && eq(ben_class,"adif")) ||
    (unit_value_savings>0.0)){
    set_saving();
    set_saving_pup();
}

if(eq(ben_class,"profil"))
    set_profil_rider_variables();

if(paid_up=="Y" )
    exp_ren_perc_prem = 0.;

set_reinsurance();
re_ratio_w = 1 - life->retention_perc;
if (eq(re_type,"NONE"))
    re_ratio_w = 0.0;

if(eq(submodel,"UNIT") && eq(done_startup_w,"false") && eq(paid_up ,"N") && (prem_curr <= 0.0))
    paid_up="Y";

if (eq(done_startup_w,"false"))
    validate_data();

if (!eq(error_msg,"no_error"))    { // this causes all formulae to be zero
    maturity_period_w = -1;
    mat_period_min = -1;
    maturity_period_ann = -1;
    commence_period_w = 1;
}

int j = min_retirement_age;

for(int i = 0; i < sm_annuity.size(); i++){
    if(ann_index_map.count(j + i) == 0){
        ann_index_map[j + i] = i;
    }
}

```

```
sm_annuity[i]->setGroup(xstring(j+i));  
  
}  
  
if (inlist(prod_code,"a72,a75,a80-00honi,a80-01hon,a80-01kitz,rsapir1,rsapir5,asav,sav-r,ariske"))  
    prod_code_adif_extra_prem_temp = prod_code;  
  
return 0.0;
```

#### **6.1.1.3.1.487 cal\_duration**

```
if (t < commence_period_w)  
    return NO_AVG;  
  
return max(cal_year(t) - year_start,0);
```

#### **6.1.1.3.1.488 cal\_month**

```
if (t < commence_period_w)  
    return NO_AVG;  
  
int result = 0;  
  
if (t >= -valn_month)  
    result = fabs(fmod(valn_month + t, 12.));  
else  
    result = 12. - fabs(fmod(valn_month + t, 12.));  
  
if (result == 0.)  
    result = 12;  
return result;
```

#### **6.1.1.3.1.489 cal\_year**

```
if (t < -13)  
    return NO_AVG;  
  
if (t == 0)  
    return valn_year;  
  
if (t < 0){  
    if (valn_month == 3){  
        if (t < -2)  
            return valn_year -1;  
        return valn_year;  
    }  
  
    if (valn_month == 6){  
        if (t < -5)  
            return valn_year -1;  
        return valn_year;  
    }  
  
    if (valn_month == 9){  
        if (t < -8)  
            return valn_year -1;  
        return valn_year;  
    }  
}
```

```
        if (valn_month == 12){
            if (t < -11)
                return valn_year -1;
            return valn_year;
        }
    }

//if vetek is 0 in data then fix it for calculation of cal_year to be 1
int vetek = elapsed_months;
if (vetek == 0.0)
    vetek = 1.0;

if (t == 1 - vetek){
    if (t>-12)
        return valn_year;
    return valn_year -1;
}

if (cal_month(t) == 1)
    return cal_year(t-1) + 1.;
return cal_year(t-1);
```

#### **6.1.1.3.1.490 pol\_month**

```
if (t < commence_period_w)
    return NO_AVG;

if (t == -elapsed_months)
    return 0;

int mth = pol_month(t-1) + 1;

if (mth == 13)
    return 1;
return mth;
```

#### **6.1.1.3.1.491 pol\_year**

```
if (t < commence_period_w)
    return NO_AVG;

if (t == -elapsed_months)
    return 1;

if (pol_month(t) == 1.0 && t > -elapsed_months +11)
    return pol_year(t-1) + 1;

return pol_year(t-1);
```

#### **6.1.1.3.1.492 pol\_year\_ext**

```
if (t < commence_period_w)
    return NO_AVG;

return xint(pol_year(t) + round(elapsed_months_extra/12.,0));
```

**6.1.1.3.1.493 proj\_month**

```
if (t < commence_period_w)
    return NO_AVG;

if (t == 0)
    return NO_AVG;

int result = xint(fmod(t, 12));

if (result == 0) {
    if (t > 0)
        result = 12;
    else
        result = -12;
}
return result;
```

**6.1.1.3.1.494 proj\_year**

```
if (t < 1)
    return NO_AVG;

//if (t > 0)
if (proj_month(t) == 1)
    return proj_year(t-1) + 1.;

return proj_year(t-1);
```

**6.1.1.3.1.495 proj\_year\_rollup**

```
if (t < commence_period_w || !eq(projection_type_int, "Rollup"))
    return NO_AVG;

if(t<= rollup_period)
    return 0;

if (eq(start_int_proj_after_rollup, "N"))
    return proj_year(t);

if(proj_month(t) == rollup_period + 1 || (proj_month(t) == 1. && rollup_period == 12.))
    return proj_year_rollup(t-1) + 1;

return proj_year_rollup(t-1);
```

**6.1.1.3.2 External Functions****6.1.1.3.2.1 call\_extra\_scalars**

```
void call_extra_scalars(void)
{
    double temp = 0.0;

    temp = comm_clawback_pv_start;
    temp = comm_hekef_new;
    temp = comm_init_new;
    temp = comm_nihul_pv_start;
    temp = comm_prizes_new;
    temp = comm_pv_start;
```

```

temp = comm_reg_pv_st;
temp = comm_ren_pv_st;
temp = comm_res_pv_st;
temp = duration;
temp = expense_init_new;
temp = expense_pv_start;
temp = policies_new;
temp = prem_alloc_pv;
temp = prem_discount_py1;
temp = premium_1;
temp = premium_disc_pv_start;
temp = premium_new;
temp = premium_pv_st_date;
temp = profit_net_vif_yr0;
temp = reins_comm1;
temp = res_total_increase1;
temp = reserve_rein_opening;
temp = premium_gross_yr1;
temp = claims_total_yr1;
temp = comm_total_yr1;
temp = expense_total_yr1;
temp = claims_re_yr1;
temp = premium_re_yr1;
temp = comm_re_yr1;
temp = comm_re_prof_yr1;
temp = comm_reg_riders_out_pv_st;
temp = charges_premium_pv_st;
temp = management_fee_pv_st;
temp = proj_task_loop_num_scalar;
temp = claims_pv_st;
temp = cashflow_re_pv_st;

xstring tempstr = value_date;
tempstr = portfolio;
tempstr = ktest;
}

```

#### 6.1.1.3.2.2 monthly\_rate

```

double monthly_rate(double annual_rate) {
    return (pow(1. + annual_rate / 100. , 1. / 12.) - 1.);
}

```

#### 6.1.1.3.2.3 set\_accum\_fund

```

void set_accum_fund(void) {

int i=0;

// *****Set variables in submodel:*****
// Accumulation Units for premium paying policies
for (i = 0; i < 16; i++) {
    accum->alloc_rate[i] = alloc_rate[i];
    accum->alloc_rate_period[i] = alloc_rate_period[i];
}

accum->par_nonpar = par_nonpar;
accum->commence_period_w = commence_period_w;

```



```

accum->elapsed_months = elapsed_months;
//sm_accum->isRebaseClone = isRebaseClone;
accum->maturity_period_w = maturity_period_w;
accum->mgt_fee_fixed = mgt_fee_fixed;
accum->mgt_fee_variable = mgt_fee_variable;
accum->paid_up = paid_up;
accum->policies_curr = policies_curr;
accum->benefits_curr = benefits_curr;
accum->decrements_apply = decrements_apply;
accum->prem_freq = prem_freq;
accum->projection_type = projection_type;

for (i = 0; i<116; i++) {
    accum->comm_regular_pc[i] = comm_regular_pc[i];
    sm_accum->comm_perc_res[i] = comm_perc_res_a[i];
}

accum->comm_ren_perc_prem = comm_ren_perc_prem;
accum->comm_renewal_year = comm_renewal_year;

sm_accum->unit_value_if = unit_value_accum-unit_value_savings;

sm_accum->unit_type = "Accum_prem";

// close function
}

```

#### 6.1.1.3.2.4 set\_accum\_pup\_fund

```

void set_accum_pup_fund(void) {

int i=0;

// *****Set variables in submodel:*****

acc_pup->par_nonpar = par_nonpar;
acc_pup->commence_period_w = commence_period_w;
acc_pup->elapsed_months = elapsed_months;
acc_pup->maturity_period_w = maturity_period_w;
acc_pup->mgt_fee_fixed = mgt_fee_fixed;
acc_pup->mgt_fee_variable = mgt_fee_variable;
acc_pup->paid_up = paid_up;
acc_pup->policies_curr = policies_curr;
acc_pup->benefits_curr = benefits_curr;
acc_pup->projection_type = projection_type;
acc_pup->decrements_apply = decrements_apply;

for (i=0; i < 116; i++)
    sm_acc_pup->comm_perc_res[i] = comm_perc_res_a[i];

sm_acc_pup->unit_value_if = unit_value_accum-unit_value_savings;
sm_acc_pup->unit_type = "Accum_pup";

// close function
}

```

### 6.1.1.3.2.5 set\_by\_prodcod

```
void set_by_prodcod (void) {
int i=0;
int j=0;
xstring phi_WP = "3";
double series_end_temp=0.0;

xstring endage;

// *****Set prod_code specific assumption sets *****
comm_set_temp = comm_set; //commission assumption

if(comm_set_temp == "N/A") // prod_code does not exist in lookup
    throw NonFatalError("Product code: " + prod_code + " not in product assumptions table.");
exp_row_lookup =exp_set_pol + "_" + exp_set_cvr + "_" + company;
if(eq(pol_type_expenses, "selfemp"))
    exp_row_lookup = exp_row_lookup + "_managers";
else
    exp_row_lookup = exp_row_lookup + "_" + pol_type_expenses;

if(exp_madad==99999.)
    throw NonFatalError("Error looking up expenses for " + exp_row_lookup+"_"+pol_type_expenses
+ " in expense_tbl.");

decrem_mult_set_temp =decrem_mult_set;
clms_mult_set_temp =clms_mult_set;
clwback_set_temp =clwback_set;
exp_mult_set_temp =exp_mult_set;

alloc_rate_set_temp =alloc_rate_set;

surr_charge_set_temp = policy_type + "_" +surr_chg_set;

tarif_spec_row_key= xstring(tarif);

if (eq(ben_class,"phi")) {
    phi_type =pitzui_shichrur;

    if(phi_type == "N/A")
        phi_type = "P";
    }

//check if riders - for setting lapse rates
if (!eq(prod_code_base,prod_code) && eq(submodel,"TERM"))
    rider_ind=1;

//check if pup - for setting lapse rates
if (eq(paid_up,"Y") && (!eq(submodel,"TERM")))
    pup_ind=1;

// *****Set prod_code specific variables *****

fund_name_temp = xstring(fund_name);
```

```

// Change fund name to read right annuity factors
if(inlist(prod_code,"a72,a80-00honi") && (atoi(fund) < 100 || inlist(fund, "521,523,527")))
    fund_name_temp = xstring(min(atoi(fund_name_temp),50));

if(eq(prod_code, "asav") && inlist(fund, "52,521,523,527"))
    fund_name_temp = xstring(min(atoi(fund_name_temp),50));

// ***** Set fund specific rates *****

if(xint(par_npar)==1)
    par_nonpar = "P";
else
    par_nonpar = "N" ;

if(xint(par_npar_yesodi)==1)
    par_nonpar_yesodi = "P";
else
    par_nonpar_yesodi = "N" ;

int dac_code = dactype;
if (dac_code==0) dac_type_temp = "none";
if (dac_code==1) dac_type_temp = "zillmer";
if (dac_code==2) dac_type_temp = "il_dac";

// Set up risk free investment rate array (by year)
for (i = 0; i <=119; i++){

    risk_free_row_key=i;

    inv_rate_mth_t[i] = inv_rate_m; //yield pre
    inv_rate_mth_t_ifrs[i] = inv_rate_m_ifrs;
    v_month_t[i] =1/(1 + disc_rate_m); //Discount Factor pre

    if(esg_run=="Y" && chilean==0 && eq(paid_up, "G")) { //yield post
        ann_inv_rate_mth_t[i] =inv_rate_m; // use pre yield if non chilean for ESG runs
        ann_inv_rate_mth_t_ifrs[i] =inv_rate_m_ifrs;}
    else
        {ann_inv_rate_mth_t[i] =ann_inv_rate_m;
        ann_inv_rate_mth_t_ifrs[i] =ann_inv_rate_m_ifrs;
        }
    ann_v_month_t[i] = 1/(1 + ann_disc_rate_m);          //Discount Factor post

    v_month_t_rm[i] = 1/(1 + inv_rate_rm_m); // Discount Factor - No VA

    risk_free_row_key = valn_year + i;

}

if(inv_rate_rollup != 0.0 && eq(projection_type_int, "Rollup")) //seems not to be used as overwrite
after
    inv_rate_mth_t[0] = inv_rate_rollup;

```

```

// Check if should use actual investment income for projection year 1
double temp_inv_free = 0.0;

if (eq(ben_class,"phi") && eq(paid_up,"C"))
    temp_col_fund="inv_free_rollup_PHI_C";
else
    temp_col_fund="inv_free_rollup";

temp_inv_free = fund_rates_code_tbl;          //WTW Modified code

if(eq(projection_type_int,"Rollup"))
{
    //Set year 1 investment rate according to fund (Only for participating; for NP, use RFR)
    if(xint(par_npar)==1 || xint(par_npar)==0){ // always 0 or 1?
        inv_rate_mth_t[0] = monthly_rate(temp_inv_free);
        ann_inv_rate_mth_t[0] = monthly_rate(temp_inv_free);
    }
    else
    {
        if (start_int_proj_after_rollup == "N"){

            inv_rate_mth_t[0] = inv_rate_mth_t[1];
            ann_inv_rate_mth_t[0] = ann_inv_rate_mth_t[1];}
    }
}

// adjust the investment return array for the fund's investment income rate
if (eq(savings_pol_prod_code, "Y") || eq(paid_up,"C")) {

double curr_year_prop = 0.0;
double next_year_prop = 0.0;
double rollup_fact= 0.0;
double rollup_month=0.0;
double ny_months=0.0;

if(eq(projection_type_int,"Rollup")) {
    if(valn_month==12)
        rollup_fact=1;
    rollup_month=rollup_period;
}

free_inv_row_key = valn_year+rollup_fact;
free_inv_prop_t[0] = free_inv_ratio_tbl;

ny_months = rollup_month+valn_month*(1-rollup_fact);

for (i = 1; i <=119; i++){
    free_inv_row_key = valn_year+i-1+rollup_fact;
    curr_year_prop = free_inv_ratio_tbl;
    free_inv_row_key = valn_year+i+rollup_fact;
    next_year_prop = free_inv_ratio_tbl;

    free_inv_prop_t[i] = (curr_year_prop*(12-ny_months)+next_year_prop*ny_months)/12;
}

```

```

    }
}
double temp_inv_inc = monthly_rate(invinc);

// only adjust vector for when there are non-free assets in the fund
if (free_inv_prop_t[0] < 1.0) { //current year i.e. valuation year

    v_month_t_int_res = 1. / (1. + monthly_rate(intres));

    for (i=0; i<119 ;i++){
        inv_rate_rf_mth_t[i] = inv_rate_mth_t[i];
        ann_inv_rate_rf_mth_t[i] = ann_inv_rate_mth_t[i];
    } //end for loop
    for (i=0; i<119 ;i++){
        inv_rate_mth_t[i] = inv_rate_mth_t[i]*free_inv_prop_t[i] + temp_inv_inc*(1.-
free_inv_prop_t[i]);
        ann_inv_rate_mth_t[i] = ann_inv_rate_mth_t[i]*free_inv_prop_t[i] + temp_inv_inc*(1.-
free_inv_prop_t[i]);

        v_month_t_ifrs[i] = 1/(1 + inv_rate_mth_t_ifrs[i]);
        ann_v_month_t_ifrs[i] = 1/(1 + ann_inv_rate_mth_t_ifrs[i]);

    } //end for loop
} // end if

if (eq(ben_class,"adif") || eq(submodel,"TRAD")) {
    mgt_fee_variable =var_mgt_fee;
    if (par_nonpar=="P"){
        mgt_fee_fixed =fixed_mgt_fee;
        mgt_fee_fixed_puresav = mgt_fee_fixed;
    }
    else{
// For Non Par, set the management fee to be the credited interest rate
        mgt_fee_fixed =intres;
        mgt_fee_fixed_puresav =intres_puresav;
        if(inlist(life->prod_code,"sav-s,sav-r"))
            mgt_fee_fixed_puresav = mgt_fee_fixed;
        }
    if (eq(submodel,"TRAD"))
        mgt_fee_fixed = fixed_mgt_fee;

}

// Adjust starting dac values
// Check if fund rates table has adjustment factors, otherwise use the input values
double adjfac=0.0;

if (!eq(dac_type_temp,"none")) {
    adjfac =dac_book_fac;

    if(adjfac==2.) //In case lookup failed
        adjfac= dac_book_adj_factor;
    dac_book_inforce = dac_book_inforce * adjfac/100. * benefits_curr;
    dac_book_adj_factor = adjfac;

    if (eq(dac_type_temp,"zillmer")) {

```

```

        adjfac= zillmer_adj_factor;
        dac_tax_inforce = dac_tax_inforce * adjfac/100. * benefits_curr;
    }
    else {
        adjfac =dac_tax_fac;

        if(adjfac==2.) //In case lookup failed
            adjfac= dac_tax_adj_factor;
        dac_tax_inforce = dac_tax_inforce * adjfac/100.* benefits_curr;
        dac_tax_adj_factor = adjfac;
    }
}
else
    dac_book_inforce = 0.0;

// Set capital required percentages
cap_req_perc_premium_temp = cap_req_perc_premium/100.0;
cap_req_perc_reserve_temp =cap_req_perc_reserve/100.0;

// *****Set UNIT prod_code specific variables *****
if (eq(submodel,"UNIT")) {
    xstring prod_code_temp = prod_code;

// lookup maximum percentage of [tagmulim] premium that can be used to pay for risk riders
    rider_max_perc = atof(prod_specs_max);

// lookup SV penalty annual rebate factor for paid-up policies
    pup_sv_charge_rebate_temp = atof(pup_sv_charge_rebate);

// calculate annuity reserves for all savings products
    if (res_kitzba==0 && (elapsed_months>0 || eq(policy_type,"private"))) )
        annuitization_rate = 0.0;

//***** Set allocation rate*****
    if (!eq(alloc_rate_set_temp,"data")) {
        alloc_rate_set_temp = alloc_rate_set_temp + "_" + policy_type;

// Set up alloc_rate and alloc_rate_period array
        alloc_rate[0] = 0.0;

        alloc_rate_period[0] = 0.0;
        for (i=1; i<=15.;i++){
            stri = xstring(i);
            alloc_rate_row = alloc_rate_set_temp + "_p";
            alloc_rate[i] = alloc_rate_stri;
            alloc_rate_row =alloc_rate_set_temp + "_m";
            alloc_rate_period[i] = alloc_rate_stri;
        } //end for loop

    } // end      if (!eq(alloc_rate_set,"data"))...

}      //end UNIT specific settings

// ***** TERM specific settings *****
if (eq(submodel,"TERM")){
    prem_lookup_temp = prem_lookup;

```

```

    prem_lookup_freq_temp = atoi(prem_lookup_freq);

    prem_init_different_temp = prem_init_different;    //premium rate different in first period?

    secondary_prop_continue = atof(dd_prop_cont);
    adjust_prem_and_claims_temp = adjust_prem_and_claims;

//*****set premium rate*****
if (prem_curr <= 0.) {    // deal with covers with zero premium (free covers)
    adjust_prem_and_claims_temp = "N"; // cannot adjust claims and premiums based on the zero
    premium
        if (promil < 0.3 && (prem_lookup_temp=="Y")){    // run YRT cover with zero premium only
            until next renewal
                benefit_term =
premi_lookup_freq_temp*12*(1+xint(elapsed_months/(premi_lookup_freq_temp*12)));
                ben_period_min = benefit_term;
            }
            // for LP covers with zero premium (promil < 0.3 && !premi_lookup=="Y") run until end (with
            the zero premium)
            // if there are discounts (negative values) then these should be zeroised so that the zero
            premium will not be further reduced
            if((premi_disc_perc + premi_disc_perc_2 + premi_disc_dcr1_r) <=-100.){
                double temp_prem_scale = 1000.;
                if (eq(ben_class,"phi"))
                    temp_prem_scale = 100.;
                premi_curr = sum_ins_curr*promil/temp_prem_scale;
            }
        }
    }

if (elapsed_months_extra + elapsed_months > premi_lookup_freq_temp * 12) // assume that the tarif
date was a renewal date, not the start of the cover
    premi_init_different_temp = "N";

// Check which table to use - tarif level or product code level
if (premi_lookup_temp=="Y") {
    if (inlist(ben_class,"phi,ltc")) // for yrt phi, looks up GP rates in table based on
premi_key in tarif_spec
        use_tarif_spec_premis = "Y";

    //Check if tarif exists in the tarif_spec table
    stri=premi_key_start;

    if (stri != "N/A" && !eq(stri,"0")){
        use_tarif_spec_premis = "Y";

        // check premium frequency (term) is >0 , otherwise level

        if(atoi(tarif_spec_lookup_freq)==0) {;
            premi_lookup_temp="N";
            use_tarif_spec_premis = "N";
        }
    }
} // end if (premi_lookup=="Y")

// Set prem rate reference and table
if (use_tarif_spec_premis == "Y"){// for yrt phi, looks up GP rates in table based on premi_key in
tarif_spec

```

```

// define starting point for prem_key
xstring prem_key_build=prem_key_start;
//if (LOOKUP_FAILED) // tarif does not exist in tarif_spec table
if(prem_key_build == "N/A")
    throw NonFatalError("tarif" + xstring(tarif) + " not in tarif spec table.");

// expand starting prem key to include occupation, endage, sex, smoker as necessary

if(eq(premkey_occ,"Y")){

    if(eq(prod_code,"phi-mitriya")){

        if (inlist(occ_key,"1,2"))
            prem_key_build=prem_key_build+occ_key+"_";
        else
            prem_key_build=prem_key_build+"1_";
    }

    else {

        if (inlist(occ_key,"1,2,3"))
            prem_key_build=prem_key_build+occ_key+"_";
        else
            prem_key_build=prem_key_build+"3_";
    }
}

//appends end age from policy to prem_key for 2004 hachnasa btucha tariffs that allow a range
of end ages
// only end ages listed below are permissible, otherwise use default end age = 65

if(eq(premkey_endage,"Y")){

    xstring endage=xstring(age_at_issue + benefit_term/12.);

    if(eq(prod_code,"phi-mitriya")){

        if (inlist(endage,"60,62,64,65,67,68,69,70"))
            prem_key_build=prem_key_build+endage+"_";
        else
            prem_key_build=prem_key_build+"65_";
    }

    else {

        if (inlist(endage,"60,62,64,65,67,70"))
            prem_key_build=prem_key_build+endage+"_";
        else
            prem_key_build=prem_key_build+"65_";
    }
}

if(eq(premkey_sex,"Y"))
    prem_key_build=prem_key_build+sex+"_";
if(eq(premkey_smoker,"Y"))
    prem_key_build=prem_key_build+smoker_stat+"_";

```



```

        if(eq(premkey_insured,"Y"))
            prem_key_build=prem_key_build+xstring(insured)+"_";

        if(eq(prod_code,"rsapir1-2019")){
            if(origidate >= 201909)
                prem_key_build=prem_key_build+"postfix_";
            else
                prem_key_build=prem_key_build+"prefix_";
        }

// check if for specific tarif prem rates gross vary by fund name and prem_profil_type (in/out)
//xstring prem_code_test;
int prem_key_test;
    prem_code_test=prem_key_build+fund_name_temp+"_"+prem_profil_type+"_";
    prem_code_test_temp=prem_key_build+fund_name_temp+"_"+prem_profil_type+"_0";
    prem_key_test =prem_rates_series;

    if(prem_rates_series == -99999){
        prem_code_test=prem_key_build+fund_name_temp+"_";
        prem_code_test_temp=prem_key_build+fund_name_temp+"_0";
        prem_key_test =prem_rates_series;

        if(prem_rates_series == -99999){
            prem_code_test=prem_key_build;
            prem_code_test_temp=prem_key_build+"0";
            prem_key_test =prem_rates_series;

            if(prem_rates_series == -99999){
                throw NonFatalError("premium key " +prem_code_test+"0-
not in prem_rates for policy: "+pol_number +" tarif:"+xstring(tarif));
            }}}
// find appropriate premium series
// conditional loop checks if origidate falls within prem series start and end dates
    j=0;
    prem_key_temp=prem_code_test+xstring(j);

    if (eq(prem_series_year,"SA")) {
        // Find series by SI level

        while ((sum_ins_curr/1000)>=prem_rates_temp_series_end)
        {
            j=j+1;
            prem_key_temp=prem_code_test+xstring(j);
        } // end while loop
    }
    else {
        // Find series by start date

        while (origidate>prem_rates_temp_series_end)
        {
            j=j+1;

            prem_key_temp=prem_code_test+xstring(j);
        } // end while loop
    }
    prem_code=prem_key_temp;
}

if (!(use_tarif_spec_prem == "Y") && prem_lookup_temp=="Y" ){
    row = prod_code+"_"+fund_name_temp+"_"+sex+"_"+smoker_stat;

```

```
col = "Prem_code";
prem_code = prem_code_map_tbl;
}

/***** set claims cost & paid-up values tables *****/
if(eq(ben_class,"phi") || eq(ben_class,"ltc")) {
  claims_cost_key = claims_cost_key_start; // from tarif_spec table
  //append end age for 2004 hachnasa btucha tariffs that allow a range of end ages

  if(eq(prod_code,"phi-mitriya")){
    if (inlist(occ_key,"1,2"))
      claims_cost_key=claims_cost_key+occ_key+"_";
    else
      claims_cost_key=claims_cost_key+"1_";
  }

  if(eq(claimskey_endage,"Y")){
    endage=xstring(age_at_issue + benefit_term/12.);

    if(eq(prod_code,"phi-mitriya")){
      if (inlist(endage,"60,62,64,65,67,68,69,70"))
        claims_cost_key=claims_cost_key+endage+"_";
      else
        claims_cost_key=claims_cost_key+"65_";
    }
    else {
      if (inlist(endage,"60,62,64,65,67,70"))
        claims_cost_key=claims_cost_key+endage+"_";
      else
        claims_cost_key=claims_cost_key+"65_";
    }
  }
}

//append sex
if(eq(claimskey_sex,"Y")){
  if (inlist(sex,"M,F"))
    claims_cost_key=claims_cost_key+sex+"_";
  else
    claims_cost_key=claims_cost_key+"M_";
}

// loop to find generation of claim cost
j=0;
key_temp=claims_cost_key+xstring(j);
int tarif_claims_series_start =atoi(claims_series_year);

//test lookup
series_col_key="Series_End";
series_end_temp = claims_cost_factors_tbl;

if(series_end_temp == 10000000.)
```

```

        throw NonFatalError("key " + key_temp + " not in claims_cost_factors table.");

while (tarif_claims_series_start > claims_cost_factors_tbl) {
    j=j+1;
    key_temp=claims_cost_key+xstring(j);
} // end while loop
claims_cost_key=key_temp;

//      determine claims_cost_multiplier from tarif_spec table
if(eq(claims_factor, "OCC"))
    claims_cost_multiplier = atof(claims_factor_occ)/100.;
else
    claims_cost_multiplier = atof(claims_factor)/100.;

    if ((eq(life->ben_class,"phi") || (eq(life->ben_class,"ltc") && eq(life->paid_up,"C")))) &&
use_phi_claims_cf == "Y") {

    // Determine PHI claims in payment reserves factor
    endage=xstring(round(age_at_issue + benefit_term/12.,0));
    if (!inlist(endage,"60,65,67")){
        if(atof(endage) < 62.)
            endage="60";
        else {
            if(atof(endage) < 66.)
                endage="65";
            else
                endage="67";
        }
    }

    phi_WP =waiting_period_modeled;
    if (!inlist(phi_WP,"3,6"))
        phi_WP="3";

    xstring rate = "2.5";
    if (year_start <= 1992)
        rate = "4";

}

} // end if ben_class is ltc or phi

/*****set decrement and risk rates table*****/
key_temp="risk_rates_" + company;
if (!inlist(ben_class,"dth,mortg,fib")) {
    if (eq(prod_assumpt_key_tbl,"tarif"))// for all phi whose decrement rates vary by
tarif
        decrem_rates_tbl = "phi_decrem_"+ incidencerate_key;
    else
        decrem_rates_tbl = prod_assumpt_key_tbl;
    if (eq(use_uw_date,"N")) {
        double check=0.0;
        //check if key is 1-policy type, 2-company, 3-sex&smoker
        col_char = pol_type_phi_incidence;
    }
}

```

```
        check = decrem_rates_check;

        if (check == -999999){
            col_char = company;
            check = decrem_rates_check;

            if (check == -999999)
                col_char = sex+smoker_stat;
        }
        else col_char = sex+smoker_stat;

        decrem_rate_key = col_char;
    }

// ***** Set NP Reserving Basis *****
int_rate_res =intres;

// set reserve factors
if (eq(res_basis,"Perc_Prem")) {
    for (int i=0; i <= 120; i++){
        res_fac_row_key =i;
        res_perc_prem[i] = reserve_factors_tbl;
    } //next i
}

    if (inlist(ben_class,"dth,mortg,fib"))
        death_ben_w = "Y";
    else
        death_ben_w = "N";

    } //end TERM specific

// ***** TRAD specific settings *****
if (eq(submodel,"TRAD")) {
    prem_lookup_temp = prem_lookup_trad;

    if(prem_lookup_trad=="N/A")
        throw NonFatalError("Product code " + prod_code + " not in prod specs trad table.");
    prem_lookup_freq_temp =atoi(prem_lookup_freq_trad);

    matan_perc_temp =atoi(matan_perc);
    //premium_inc =atof(prem_inc);

    // set old numerical product code to reference sv tables
    prod_code_old = xstring(procdold);

    /*****set sv and puv table*****/
    if(eq(sur_val_method,"sv_table")){
        if (eq(ben_class,"GIMLA")){
            sv_tbl = fund_name_temp + "_" + prod_code_old + "_" + sex;
            puv_tbl = fund_name_temp + "_puv_" + prod_code_old + "_" + sex;
        }
    }
}
```

```

        else {
            sv_tbl = fund_name_temp + "_" + prod_code_old;
            puv_tbl = fund_name_temp + "_puv_" + prod_code_old;
        }
    }

// ***** Set NP Reserving Basis *****
int_rate_res = intres;

death_rates_res = "AMF"+xstring(mort_res);
mort_addn_res = mort_addn;

// ***** Set Zilmer Rate *****
double zill;
    if(eq(ben_class,"GIMLA"))
        zill = 0.0;
    else {
        if(eq(ben_class,"WOL")){
            if(year_start < 1996)
                zill = 1.5;
            else {zill = 2.0;}
        }
        else {zill = 3.0;} // for all other endowment type products (END,YTRON)
    }
    if(year_start < 1987) // adjustment to reflect zilmer reduction in 1986
        zill = zill * 0.5;

zillmer_si_book = zill;
zillmer_si_tax = zill;

    } // end TRAD specific

// close function
}

```

#### 6.1.1.3.2.6 set\_from\_data

```

void set_from_data (void) {

int i=0, yr=1;

// Set up regular initial commission percentage (on premium or reg.comm)
i= max(xint((elapsed_months-1)/12.) , 0);
comm_regular_pc[0+i] = amala_0;
comm_regular_pc[1+i] = amala_1;
comm_regular_pc[2+i] = amala_2;
comm_regular_pc[3+i] = amala_3;
comm_regular_pc[4+i] = amala_4;
comm_regular_pc[5+i] = amala_5;
comm_regular_pc[6+i] = amala_6;
comm_regular_pc[7+i] = amala_7;
comm_regular_pc[8+i] = amala_8;
comm_regular_pc[9+i] = amala_9;
comm_regular_pc[10+i] = amala_10;
comm_regular_pc[11+i] = amala_11;

```

```
        comm_regular_pc[12+i] = amala_12;
        comm_regular_pc[13+i] = amala_13;
        comm_regular_pc[14+i] = amala_14;
        comm_regular_pc[15+i] = amala_15;

    for (yr = 16; yr<115-i; yr++) {
        comm_regular_pc[yr+i] = amala_16;
    }

    // Set up Nihul commission percentage (on premium or reg.comm)
    comm_nihul_rate[0+i] = amala_nihul_0;
    comm_nihul_rate[1+i] = amala_nihul_1;
    comm_nihul_rate[2+i] = amala_nihul_2;
    comm_nihul_rate[3+i] = amala_nihul_3;
    comm_nihul_rate[4+i] = amala_nihul_4;
    comm_nihul_rate[5+i] = amala_nihul_5;
    comm_nihul_rate[6+i] = amala_nihul_6;
    comm_nihul_rate[7+i] = amala_nihul_7;
    comm_nihul_rate[8+i] = amala_nihul_8;
    comm_nihul_rate[9+i] = amala_nihul_9;
    comm_nihul_rate[10+i] = amala_nihul_10;
    comm_nihul_rate[11+i] = amala_nihul_11;
    comm_nihul_rate[12+i] = amala_nihul_12;
    comm_nihul_rate[13+i] = amala_nihul_13;
    comm_nihul_rate[14+i] = amala_nihul_14;
    comm_nihul_rate[15+i] = amala_nihul_15;
    comm_nihul_rate[16+i] = amala_nihul_16;

    for (yr = 17; yr<comm_nihul_rate.size()-1-i; yr++) {
        comm_nihul_rate[yr+i] = amala_nihul_16;
    }

    // Set up supervisor commission percentage

    i= max(xint((elapsed_months-1)/12.) , 0);
    comm_spvisor[0+i] = amala_pikuach_0;
    comm_spvisor[1+i] = amala_pikuach_1;

    for (yr = 2; yr<comm_spvisor.size()-1-i; yr++) {
        comm_spvisor[yr+i] = 0.0;
    }

    // Adjust renewal commission starting year
    comm_renewal_year = xint(elapsed_months/12.) + comm_renewal_year;

    // adjust discount variables

    if (prem_disc_perc < 0) {
        prem_disc_perc = min(-prem_disc_perc,100.0); // discounts in data file are represented
        as a negative percentage
        if (prem_disc_type == 4 || prem_disc_type == 6) // permanent discount
            prem_disc_month = max(1000,prem_disc_month);
    }

    if (prem_disc_perc_2 < 0) {
```

```

        prem_disc_perc_2 = min(-prem_disc_perc_2,100.0); // discounts in data file are represented
as a negative percentage
        if (prem_disc_type_2 == 4 || prem_disc_type_2 == 6) // permanent discount
            prem_disc_month_2 = max(1000,prem_disc_month_2);
    }

    if (prem_disc_step > 0) { //If kod discount =120
    - Decreasing discount

        if (prem_disc_dcr5_m > 0) // Basic discount
            prem_disc_step1_r = min(-prem_disc_dcr5_r,100.0);
rate

        if (prem_disc_dcr4_m > 0)
            prem_disc_step2_r = min(prem_disc_dcr5_r-prem_disc_dcr4_r,100.0); // 2nd step
Additional discount rate

        if (prem_disc_dcr3_m > 0)
            prem_disc_step3_r = min(prem_disc_dcr4_r-prem_disc_dcr3_r,100.0); // 3th step
Additional discount rate

        if (prem_disc_dcr2_m > 0)
            prem_disc_step4_r = min(prem_disc_dcr3_r-prem_disc_dcr2_r,100.0); // 4th step
Additional discount rate

        if (prem_disc_dcr1_m > 0)
            prem_disc_step5_r = min(prem_disc_dcr2_r-prem_disc_dcr1_r,100.0); // 5th step
Additional discount rate

        prem_disc_step1_m = prem_disc_dcr1_m + prem_disc_dcr2_m + prem_disc_dcr3_m +
prem_disc_dcr4_m + prem_disc_dcr5_m; // The 1st step discount months rest
        prem_disc_step2_m = prem_disc_dcr1_m + prem_disc_dcr2_m + prem_disc_dcr3_m +
prem_disc_dcr4_m; // The 2nd step discount months rest
        prem_disc_step3_m = prem_disc_dcr1_m + prem_disc_dcr2_m + prem_disc_dcr3_m; // The 3th step
discount months rest
        prem_disc_step4_m = prem_disc_dcr1_m + prem_disc_dcr2_m; // The 4th step discount months
rest
        prem_disc_step5_m = prem_disc_dcr1_m; // The 5th step discount months rest
    }

// ***** UNIT specific settings *****
if (eq(submodel,"UNIT")) {
    // Set up allocation rate for 585 (for other products this will come from usual table in
set_by_prodcde. See below for 583.)
    if (eq(alloc_rate_set_temp,"data")) {
        alloc_rate[1] = aloc_kafuy;
        //alloc_rate_period[1] = 800;
        alloc_limit = allocation_limit_amount;
        int imp_alok_yr = atoi(imp_manual_alloc_rate_term_dt.substr(5,4));
        int imp_alok_mth_int;

        if (imp_alok_yr == 1900 || imp_alok_yr == 3000)

            alloc_rate_period[1] = 800; else
            {
                xstring imp_alok_mth =
imp_manual_alloc_rate_term_dt.substr(2,3);

```

```

        alloc_rate[2] = product_alloc_rate_percent;

        if(eq(imp_alok_mth, "JAN"))
            imp_alok_mth_int = 1; else {
        if(eq(imp_alok_mth, "FEB"))
            imp_alok_mth_int = 2; else {
        if(eq(imp_alok_mth, "MAR"))
            imp_alok_mth_int = 3; else {
        if(eq(imp_alok_mth, "APR"))
            imp_alok_mth_int = 4; else {
        if(eq(imp_alok_mth, "MAY"))
            imp_alok_mth_int = 5; else {
        if(eq(imp_alok_mth, "JUN"))
            imp_alok_mth_int = 6; else {
        if(eq(imp_alok_mth, "JUL"))
            imp_alok_mth_int = 7; else {
        if(eq(imp_alok_mth, "AUG"))
            imp_alok_mth_int = 7; else {
        if(eq(imp_alok_mth, "SEP"))
            imp_alok_mth_int = 9; else {
        if(eq(imp_alok_mth, "OCT"))
            imp_alok_mth_int = 10; else {
        if(eq(imp_alok_mth, "NOV"))
            imp_alok_mth_int = 11; else {
        if(eq(imp_alok_mth, "DEC"))
            imp_alok_mth_int = 12; else {

        imp_alok_mth_int = 1;
    } } } } } } } } } }

        alloc_rate_period[1] = (imp_alok_yr - valn_year) *12 +
(imp_alok_mth_int - valn_month) + elapsed_months;
        alloc_rate_period[2] = 800;
    }

}

// Set management fee for Profil
if (eq(ben_class,"Profil")) {
    if (mgt_fee_fixed <= 0.0)
        mgt_fee_fixed = max(0.0, mgt_fee_variable);
    if (mgt_fee_variable <= 0.0)
        mgt_fee_variable = max(0.0, mgt_fee_fixed);
    mgt_fee_fixed = (mgt_fee_fixed + mgt_fee_variable)/2.0;
    mgt_fee_fixed_puresav = mgt_fee_fixed;
}

if(!inlist(policy_type,"private,selfemp") || res_kitzba >0.0){
    if(age_at_issue + elapsed_months/12. < sm_annuity[sm_annuity.size()-1]->takeup_age){
        if(age_at_issue + elapsed_months/12. < takeup_age){
            benefit_term_original = min(benefit_term, (takeup_age - age_at_issue)
* 12);
        }
        else {
            benefit_term_original = min(benefit_term, elapsed_months + 18);
        }
        if(dump_vars == "Y"){

```



```

        log_strm<<"Annuity size: "<<sm_annuity.size()<<endl;
        log_strm<<"End age: "<<sm_annuity[sm_annuity.size()-1]-
>takeup_age<<endl;
        age_at_issue) * 12 + 1;
        benefit_term = (sm_annuity[sm_annuity.size()-1]->takeup_age -
        ben_period_min = (min_retirement_age - age_at_issue) * 12;

    }
    else {
        benefit_term = min(benefit_term, elapsed_months + 18);    // make policy
continue for 1.5 year as paid up
        ben_period_min = benefit_term;
        benefit_term_original = benefit_term;
    } // end else
    // Add 18 months for expired policies where benefit_term < elapsed_months
    if (elapsed_months >= benefit_term && eq(prog_name,"ADIF")){ // exclude KLASI
savings policies as SI is fixed according to term
        benefit_term = elapsed_months + 18;
        ben_period_min = benefit_term;
        benefit_term_original = benefit_term;
    }

}

if(inlist(policy_type,"private,selfemp") && res_kitzba <= 0. && !eq(ben_class,"profil")){
    if(benefit_term <= elapsed_months){
        benefit_term = elapsed_months + 18;
        ben_period_min = benefit_term;
    }
    else{
        if(age_at_issue + int(benefit_term/12.) > fix_term_end_age_limit){           //End
age bigger then 80
            if(age_at_issue + elapsed_months/12. < fix_term_curr_age_max){           //
if current age under 70
                benefit_term = min(benefit_term, (fix_term_new_end_age -
age_at_issue) * 12); // End age fixed to 75
                ben_period_min = benefit_term;
            }
            else{
                benefit_term = min(benefit_term, elapsed_months +
fix_term_curr_age_above_max_add_months);
                ben_period_min = benefit_term;
            }
        }
    }
}

} // end UNIT specific

if (inlist(policy_type,"private,selfemp") && res_kitzba <= 0. && eq(ben_class,"profil")) {
    benefit_term = max((omega_age_dec - age_at_issue) * 12,0);
    if(age_at_issue + elapsed_months/12 >= omega_age_dec) //current age bigger than 100
        benefit_term = elapsed_months + 18;

    ben_period_min = benefit_term;
}

// ***** Adjust No Partners and Children *****

```

```

if (eq(submodel,"TERM")) {

    // set health_occ_perc as total medical and occupational loading
    health_occ_perc = min(999.,health_perc + occ_perc);

}    // if (eq(sub_model,"TERM"))

// close function
}

```

#### 6.1.1.3.2.7 set\_from\_tables

```

void set_from_tables (void) {
int i=0, yr=1, cal_yr=1;
double comm=0.0, e_madad=0.0;

//***** Set Expenses *****

e_madad = exp_madad;

if (prem_term <= 1) { // Single premium
    exp_init_fix =i_perpol_sp/e_madad*madad_current ;
    exp_init_perc_prem =i_single;
    exp_init_fix_cov = i_percov_sp / e_madad * madad_current;
    exp_ren_perc_prem = 0.0;
    exp_ren_fix =m_pup/e_madad*madad_current;
    exp_ren_fix_cov = m_percov_sp / e_madad * madad_current;
} //end if
else {
    exp_init_fix =i_perpol/e_madad*madad_current;
    exp_init_fix_cov = i_percov / e_madad * madad_current;
    exp_init_perc_prem = i_prem;
    exp_ren_perc_prem =m_prem;
    exp_ren_fix =m_perpol/e_madad*madad_current;
    exp_ren_fix_cov = m_percov / e_madad * madad_current;
} //end else

exp_claim_perc =m_clms;
exp_pup_fix = m_pup/e_madad*madad_current;
exp_ren_perc_annuity =m_ann_pmt;

exp_ren_res = m_res;

if (!eq(paid_up, "G"))
    exp_claim_fix = m_clms_cov / e_madad * madad_current;
else
    exp_claim_fix = 0.0;

// ***** decrement multipliers *****
mort_mult_col_key=sex + smoker_stat;
mort_mult = mort_mult_tbl;

if(death_ben_w=="N") {
    decem_mult_col_key =sex+smoker_stat;

```

```
decrem_mult_row_key =decrem_mult_set_temp;
decrem_mult =decrem_mult_tbl;
decrem_mult_row_key =decrem_mult_set_temp+"_np";
decrem_mult_res = decrem_mult_tbl;

for (i = 1; i<=112; i++)

    if (i>GetColumnKeyValuesCount(clms_mult_i)-1)
        claims_multiplier[i] = claims_multiplier[i-1];
    else {
        clms_mult_i_col= i;
        claims_multiplier[i] =clms_mult_i;
    }
// Set claims inflation percentage (duration=0)
claim_inflation_perc=clms_mult_infl;
} // end if

// ***** Set lapse Rates*****

if ( lapse_force_month == 1 && lapse_force_rate == 0.){
    col_char = "lapse_rate";
    lapse_force_rate = masslaps_tbl;
}
// ***** Adjust lapse Rates by Agent Lapse Factors*****

double clawback_factor;

claw_fact_set = "clawback_factor_" + clwback_set;

agency_no_lookup = xstring(channel);

clawback_factor = lapse_clawback_factor;

// column not found in lapse_factor_tbl
if(clawback_factor == -99999.) {
    claw_fact_set = "clawback_factor_default";
    clawback_factor = lapse_clawback_factor;

} // end if

//row not found in lapse_factor_tbl
if(clawback_factor == 99999.)
{
    agency_no_lookup = xstring(agency_no);
    clawback_factor = lapse_clawback_factor;

    if (lapse_clawback_factor ==99999.)
    {
        agency_no_lookup = "0";
        clawback_factor = lapse_clawback_factor;
    } // end if

} // end if

if (clawback_factor == -99999.)
    throw NonFatalError("Clawback column " + claw_fact_set + " not found.");
```

```
// ***** Set commission & clawback parameters
//      Check if clawback info is by policy type or not

i=1;
comm_claw_row_key=xstring(i);
clwback_set_temp = clwback_set;
comm_ext_col_key = "SPV_" + clwback_set_temp;
comm = comm_claw_prpn_tbl;
if (comm == 9999.)
    clwback_set_temp = "default";
comm_ext_col_key = "SPV_" + clwback_set_temp;
comm = comm_claw_prpn_tbl;
if (comm == 9999.)
    throw NonFatalError("Error looking up clawback rates for " + comm_set_temp + " in
comm_claw_prpn_tbl.");
for (i=1; i<180; i++){

    comm_claw_row_key=xstring(i);
    comm_ext_col_key="SPV_" + clwback_set_temp;
    comm_claw_prpn_spv[i] = comm_claw_prpn_tbl;

    if(comm_claw_prpn_spv[i]==10000)
        comm_claw_prpn_spv[i] = comm_claw_prpn_spv[i-1];

}

i=1;
comm_claw_row_key=xstring(i);
clwback_set_temp = clwback_set;
comm_ext_col_key = "HEKEF_" + clwback_set_temp;
comm = comm_claw_prpn_tbl;

if (comm == 9999.)
    clwback_set_temp = "default";
comm_ext_col_key = "HEKEF_" + clwback_set_temp;
comm = comm_claw_prpn_tbl;

if (comm == 9999.)
    throw NonFatalError("Error looking up clawback rates for " + comm_set_temp + " in
comm_claw_prpn_tbl.");
for (i=1; i<180; i++){

    comm_claw_row_key=xstring(i);
    comm_ext_col_key="HEKEF_" + clwback_set_temp;
    comm_claw_prpn_hekef[i] = comm_claw_prpn_tbl;

    if(comm_claw_prpn_hekef[i]==10000)
        comm_claw_prpn_hekef[i] = comm_claw_prpn_hekef[i-1];

    if (i > 12 && i <= 24)
        comm_claw_prpn_hekef[i] = comm_claw_prpn_hekef[i] * clawback_factor;

}
```

```
// Set up extra commission percentage (on premium or reg.comm)
// Check whether to read in by agent or product and company level
if (comm_extra_agent_use=="N") { //read by product

    // check to see if table row includes the agent
    temp_comm_set = xstring(agent_no);
    comm_ext_col_key = "HEKEF";
    comm_hekef_pc=comm_extra_tbl;

    //if (LOOKUP_FAILED==0)
    if(comm_hekef_pc!=9999. && comm_hekef_pc!=10000.) // To avoid errors
        temp_comm_set=xstring(agent_no);
    else temp_comm_set = comm_set_temp;

    comm_ext_col_key ="CO_MIN_TERM";
    comm_min_prem_term = comm_extra_tbl;
    comm_ext_col_key="PRIZES";
    comm_prizes_pc = comm_extra_tbl;

    comm_ext_col_key="HEKEF";
    comm_hekef_pc = comm_extra_tbl;

    comm_ext_col_key = "HEKEF_res";
    comm_hekef_pc_res = comm_extra_tbl;
    if(comm_hekef_pc_res == 9999.)
        comm_hekef_pc_res = 0.0;

    comm_ext_col_key = "PRIZES_res";
    comm_prizes_pc_res = comm_extra_tbl;
    if(comm_prizes_pc_res == 9999.)
        comm_prizes_pc_res = 0.0;

    temp_comm_set = "risk";
    comm_ext_col_key = "HEKEF";
    comm_hekef_pc_rider = comm_extra_tbl;          //

} // end read comm_extra at product level

else {      // read comm_extra details from agent level table

    temp_agency_no = xstring(agency_no);

    comm_ext_col_key="CO_MIN_TERM";
    comm_min_prem_term = comm_extra_agent_tbl;

    // set prizes % by agency, channel, policy type & product (commission set)
    comm_ext_col_key="PRIZES";
    comm_prizes_pc = comm_extra_agent_tbl;
    comm_ext_col_key="HEKEF";
    comm_hekef_pc = comm_extra_agent_tbl;

    comm_ext_col_key = "HEKEF_res";
    comm_hekef_pc_res = comm_extra_agent_tbl;
    if (comm_hekef_pc_res == 10000.)
        comm_hekef_pc_res = 0.0;
```

```

    comm_ext_col_key = "PRIZES_res";
    comm_prizes_pc_res = comm_extra_agent_tbl;
    if (comm_prizes_pc_res == 10000.)
        comm_prizes_pc_res = 0.0;

    comm_ext_col_key = "HEKEF";
    comm_set_temp = "risk";
    comm_hekef_pc_rider = comm_extra_agent_tbl;

    comm_set_temp=comm_set;

}      // end read comm_extra details from agent level table

}

```

#### 6.1.1.3.2.8 set\_other\_variables

```

void set_other_variables (void) {

int mth=1, year=1, i=0;
double total=0.0,total_pup=0.0;

// cancel SI for profil
if (eq(ben_class,"profil")&& !eq(paid_up,"G"))
    sum_ins_curr = 0.0;

// set premium term
prem_term = benefit_term; // most products
if(mult_age_ind == 1)
    prem_term_original = benefit_term_original;
else
    prem_term_original = prem_term;

// set Capital requirement as a percentage of DAC-Books
if (dac_cap_apply=="N")
    dac_cap_perc_w = 0.0;
else {
    if (prod_yr_w < 1999)
        dac_cap_perc_w = 0.0;
    if (prod_yr_w >= 1999)
        dac_cap_perc_w = 30.0;
    if (prod_yr_w >= 2004)
        dac_cap_perc_w = 100.0;
}

// ***** Set percentage Bonus Rates *****
bonus[0] = 0.0;
for (i=1; i<=1199.;i++){
    if (i < GetRowKeyMinValue(bonus_tbl) || paid_up=="G")
        bonus[i] = 0.0;
    else {
        if (i > GetRowKeyMaxValue(bonus_tbl))
            bonus[i] = bonus[i-1];
        else {
            bonus_tbl_row = i;
            bonus[i] = bonus_tbl;
        }
    }
}

```

```

    }
}

/***** Calculate variables maturity_period_w, and commence_period_w.*****/

commence_period_w = -elapsed_months;

maturity_period_w = commence_period_w + benefit_term;
mat_period_min = maturity_period_w;
mat_period_original = maturity_period_w;
maturity_period_ann = maturity_period_w;

if(mult_age_ind == 1){
    mat_period_min = commence_period_w + ben_period_min;
    mat_period_original = commence_period_w + benefit_term_original;
}

if (paid_up == "G"){
    maturity_period_w = 0;
    mat_period_min = 0;
    mat_period_original = 0;
    ann_maslul = atoi(rein_set);
    rein_set = "0";
    ann_death = ann_maslul - int(ann_maslul/10)*10;
}

// for a rollup run, allow for extra elapsed months
if (eq(projection_type,"Rollup") && (elapsed_months+elapsed_months_extra<=12))
    commence_period_w = -(elapsed_months+elapsed_months_extra);

if (maturity_period_w >= 12*xint(t_high/12.))
    throw("Benefit Term exceeds projection period, rerun with larger t_high\n");

// ***** Set Interest rates *****

// if discount type = Single, then replace the discount rate vector with the input value

if (eq(ev_discount_rate_type,"Single")) {

    double temp_inv_inc = monthly_rate(invinc);
    double temp_disc_rate = monthly_rate(ev_disc_rate);
    v_month_w = 1. / (1. + temp_disc_rate);

    for (i=0; i<=119;i++){
        v_month_t[i] = v_month_w;
        v_month_t_rm[i] = v_month_w;
        ann_v_month_t[i] = v_month_w;

        inv_rate_rf_mth_t[i] = temp_disc_rate;
        ann_inv_rate_rf_mth_t[i] = temp_disc_rate;

        if (free_inv_prop_t[0] <= 1.0) {
            inv_rate_mth_t[i] = temp_disc_rate*free_inv_prop_t[i] + temp_inv_inc*(1.-
free_inv_prop_t[i]);
            ann_inv_rate_mth_t[i] = temp_disc_rate*free_inv_prop_t[i] + temp_inv_inc*(1.-
free_inv_prop_t[i]);
            v_month_t_ifrs[i] = v_month_w;
            ann_v_month_t_ifrs[i] = v_month_w;

```

```

    }
  }
}

if (eq(ben_class,"profil"))
  mgt_fee_variable = 0.0;

// ***** set premium freq *****
if (prem_term == 1)
  prem_freq = 1;

// ***** set commission variables *****

//commission reduction for short premium terms
double red_comm = 1.;
if(comm_min_prem_term > 0)
  red_comm = min(1.,prem_term_original / comm_min_prem_term);
if(prem_term_original == 1) red_comm = 0.0; // only renewal commission for SP (pure savings)

if (red_comm<1.) {
  for(i = 0; i<116; i++){
    comm_regular_pc[i] = comm_regular_pc[i]*red_comm;
  }
}
//***** miscellaneous variables *****

omega_age_w = omega_age;
// Set DAC amortisation period
if (prem_term_original == 1)
  dac_type_temp = "none";
if(eq(dac_amort_type,"Lifetime"))
  dac_amort_per = prem_term_original;
if(dac_amort_per > prem_term_original)
  dac_amort_per = prem_term_original;

//Set basic_percentage
if(eq(ben_class,"profil"))
  basic_perc_w = 100. - saving_perc;
else {
  if (alloc_rate[1]<100)
    basic_perc_w = 100. * (1 - saving_perc/100.) / (1-alloc_rate[1]/100);
  else
    basic_perc_w = 0.0;
}

if (prem_term == 1) {
  basic_perc_w = 0.0; // single premium policy forced to be pure savings
  if (eq(ben_class,"adif"))
    sum_ins_curr = 0.0; // no fixed SI for SP Adif
}

// Set variables for risk-rider (normally for Sapir sold with Meitav Managers Policy)
if (eq(ben_class,"Adif") && (risk_si>0.000001)) {
  prem_rates_risk = fund_name_temp+"_"+risk_code;

  si_unit_w[25] = prem_rates_risk_1/prem_rates_risk_2;
}

```



```

        prem_lookup_freq_w[25] = atof(prod_spec_risk_code);
    }

    // Set total percentage of initial regular commission
    comm_reg_tot_w = 0.;
    for (i = 0; i<115; i++)
        comm_reg_tot_w = comm_reg_tot_w + comm_regular_pc[i];

    // Adjust commission paid on units to a (limited) percentage of management fee if necessary
    if (comm_perc_res_a[1] > 2.0) // assume that if it is > 2% then it represents x% of management fee
    that is paid to agent
        for (year = 1; year<comm_perc_res_a.size(); year++)
            comm_perc_res_a[year] = min(comm_perc_res_a[year]/100. * mgt_fee_fixed ,
mgt_fee_fixed/(1+vat/100.));
    if (eq(ben_class,"Profil"))
        for (year = 1; year<comm_perc_res_b.size(); year++)
            comm_perc_res_b[year] = comm_perc_res_a[year];

    // Adjust initial reserve balance to remove persistency bonus
    if (eq(ben_class,"Adif")) {
        if (eq(prod_code,"a80-00honi")) // adif 2000 adjust reserve by 5%
            unit_value_accum = (unit_value_accum-unit_value_savings)/1.05 +unit_value_savings;

        if (inlist(prod_code,"a80-01hon,a80-01kitz")) // adif 2001 adjust reserve by 7%
            unit_value_accum = (unit_value_accum-unit_value_savings)/1.07 +unit_value_savings;

        if (eq(prod_code,"asav")) // use surrender value as account balance
            unit_value_accum = surr_value_if;
    }

    // taken from term set_other_variables
    if (eq(submodel,"TERM")){

    if(prem_lookup_temp=="Y"){

        prem_rate_scale_w = base;

        if (prem_rate_scale_w == 0)
            throw NonFatalError("Premium-Rate table scale is zero! Check the premium table for
record num:"+xstring(data_rec_num())+" product code: "+life->prod_code);

    }

    if((prem_lookup_temp=="N") && eq(ben_class,"PHI"))
        prem_rate_scale_w = 100.0;

    // taken from term set_premium_si
    if(!eq(paid_up,"C")){

        // formula to adjust current premium (add modal loading, remove policy fee and add back in
discount)
        // also used to derive discount or derive prem & claims loading
        // does not calculate SI
        tarif_spec_row_key= xstring(tarif);

        //no lookup while level premium
    }

```

```

if(prem_lookup_temp=="N")
    prem_lookup_freq_temp = 0;

double prem_curr_original = prem_curr; // used to compare calculated to original premium
double prem_curr_calculated = prem_curr; // used to compare calculated to original premium
double pol_fee_edit = policy_fee_if; // used for resetting prem discount below

// If premium does not include modal loading, then add it in and also add it to the policy
fee
if (mod_load_in_prem=="N") {
    prem_curr_original = prem_curr_original * (1 + tat_shnatiut_rate/100.);
    pol_fee_edit = pol_fee_edit * (1 + tat_shnatiut_rate/100.);
}

// premium includes the policy fee so remove it
premi_curr_original = premi_curr_original - pol_fee_edit *
policies_curr/max(1.0,benefits_curr);

//Check for zero SI
if (sum_ins_curr<=0) {
    error_msg = "Sum_Insured_<=0"; // this will cause record not to be calculated and
reported in individual output with an error
    sum_ins_curr = 1.0; // required to avoid divide by zero
}

//      Calculate premium according to premium rate tables
//      get premium_rate_w as current tabular tarif
if(prem_lookup_temp=="Y"){

    double last_lookup_age = age_at_issue + prem_lookup_freq_temp *
xint(xint(elapsed_months/12)/premi_lookup_freq_temp);
    premi_rates_si_col= xstring(last_lookup_age);

    if (elapsed_months < premi_lookup_freq_temp *12 &&
(prem_init_different_temp=="Y")){
        premi_rates_si_row = premi_code+"_I";
        premi_rate_w = premi_rates_si;
    }
    else {
        premi_rates_si_row = premi_code;
        premi_rate_w = premi_rates_si;
    }
    // apply premi factor to adjust phi rates for murchav
    if (use_tarif_spec_prem == "Y")
        premi_rate_w = premi_rate_w * atof(prem_factor)/100.;

    if (premi_rate_w <= 0.0) {
        error_msg = "premi_rate_in_table_<=0"; // this will cause record not
to be calculated and reported in individual output with an error
        premi_rate_w = 1.0;} // required to avoid divide by zero

    double endage= atof(xstring(age_at_issue + benefit_term/12.));

    if (endage <= last_lookup_age) {
        error_msg = "Current_Age_>=_EndAge"; }
}

```

```

        //      Calculate premium
        if(prem_lookup_temp=="Y"){
            prem_curr_calculated = (premium_rate_w * sum_ins_curr / prem_rate_scale_w) *
(1. + health_occ_perc/100.)* (1 + tat_shnatiut_rate/100.) ;
        }

        // Add back in the discount on current premium
        // DH: current premium is for month t=1, so need to add +1 to elapsed months before applying
the discount
        if (prem_disc_month > 0) {
            if (prem_disc_perc==100.)
                prem_curr_if = 0.0; // Premium discount
            else
                prem_curr = prem_curr/(1 - prem_disc_perc/100.); // Premium discount
        }

        if (prem_disc_month_2 > 0){
            if (prem_disc_perc_2==100.)
                prem_curr_if = 0.0; // Premium discount
            else
                prem_curr = prem_curr/(1 - prem_disc_perc_2/100.); // Premium discount
        }

        if(prem_disc_step > 0){ // If gradually discounts
            double temp_disc_step_r = min((prem_disc_step1_r + prem_disc_step2_r +
prem_disc_step3_r + prem_disc_step4_r + prem_disc_step5_r)/100. ,1.);
            if (temp_disc_step_r == 1.)
                prem_curr_if = 0.0;
            else
                prem_curr = prem_curr/(1 - temp_disc_step_r); // Reconer Premium before
discount
        }

        // Derive discount by comparing to the standard premium
        if (adjust_prem_and_claims_temp=="P" && ( !eq(ben_class,"mortg") || prem_disc_step == 0)) {
            prem_disc_perc = 100.0 * (1.0 - prem_curr_original / prem_curr_calculated);
            // limit the discount for mortgage business to remove negative discount
            if (eq(ben_class,"mortg"))
                prem_disc_perc = max(prem_disc_perc,0.0);

            prem_disc_month = maturity_period_w;
            prem_curr = prem_curr_calculated; // prem_curr must be the standard (calculated)
premium, even for level premium, and later it is discounted back to the original premium
            prem_curr_changed = "Y";
        }

        if (prem_curr <= 0 && eq(paid_up,"N") && (promil >= 0.3) && (prem_disc_perc +
prem_disc_step1_r) <100.)
            error_msg = "Prem_curr<=_0";

    } // end term set_premium_si

} // end term

} // close function

```

**6.1.1.3.2.9 set\_profil\_rider\_variables**

```

void set_profil_rider_variables (void) {

int i=0, j=0;
xstring temp = "blank";

xstring endage, key_temp;
int prem_key_test=0;
prem_code_rider="";
xstring temp_sex ="M";
claims_cost_key_rider = "";
decrem_rate_key_rider="";

// Populate riders data from data-file
// riders_count_w contains the starting (byte) position of the riders for the current policy

if (riders_count_w > 0) {    // if this model point has riders

    for (i=0; i<sm_riders.size(); i++){

        if (sm_riders[i]->prem_cover_input.substr(1,1) == "#")
            sm_riders[i]->prem_cover = 0;
        else
            sm_riders[i]->prem_cover = atof(sm_riders[i]->prem_cover_input);

        tarif_rider[i] = xint(sm_riders[i]->tarif);
        sum_ins_curr_rider[i] = sm_riders[i]->sum_as;

        benefits_curr_rider[i] = benefits_curr;

        discount_perc_rider[i] = sm_riders[i]->lod_amt_1;
        discount_period_rider[i] = sm_riders[i]->lod_pe_r_1;

        // adjust discount variables
        if (discount_perc_rider[i]<0.0) {
            discount_perc_rider[i] = min(-discount_perc_rider[i],100.0);
            if (discount_period_rider[i]==0) //permanent discount
                discount_period_rider[i]=1000;
            else
                discount_period_rider[i]=discount_period_rider[i] +
elapsed_months;
        }

    }

    riders_count_w = max(0,i); // change riders_count_w to be number of riders
}    //close if (riders_count_w > 0)

    for (i=riders_count_w; i <= 25; i++) {
        tarif_rider[i] = 0; // fill remainder of prod_code array with zeroes
    }

// Populate riders specifications from prod-specs table
for (i=0; i <= riders_count_w; i++) {

    if (tarif_rider[i] > 0) {

```

```

rider_tarif_row_key=xstring(tarif_rider[i]);
tarif_spec_row_key=xstring(tarif_rider[i]);
prod_code_rider = rider_tarif_tbl;

if(prod_code_rider == "N/A")
    throw NonFatalError("Tarif " +xstring(tarif_rider[i]) + " not in rider tarif
map table.");
prod_specs_rider_col="ben_class";
risk_type_w[i] = atoi(prod_specs_rider);

if(prod_specs_rider == "N/A")
    throw NonFatalError("Product code " + prod_code_rider + " not in prod specs
table.");
prod_specs_rider_col="rider_type";
rider_type_w[i] = atoi(prod_specs_rider);
// check for extra SI risk rider

if (rider_type_w[i] == 1 && sm_riders[i]->risk_type==1)
    rider_type_w[i] = 2;
prod_specs_rider_col="si_unit";
si_unit_w[i] =atof(prod_specs_rider);
prod_specs_rider_col="prem_lookup_freq";
prem_lookup_freq_w[i] =atoi(prod_specs_rider);

if (sm_riders[i]->rid_sex == 0)
    temp_sex = "F";

// get extra expense rates on charges
temp =prod_assumpt_rider_exp_tbl;

if(temp == "N/A")
    throw NonFatalError("Product code " + prod_code_rider + " not in
prod_ass table.");
if(eq(pol_type_expenses, "selfemp"))
    exp_row_key=exp_set_pol + "_" + temp+"_"+company+"_managers";
else
    exp_row_key=exp_set_pol + "_" +temp+"_"+company+"_"+pol_type_expenses;

exp_row_lookup = exp_row_key;
exp_extra_perc_charge[i] = m_prem - exp_ren_perc_prem;
exp_initial_extra_perc_charge[i] = i_prem - exp_init_perc_prem;
exp_ren_fix_rider[i] = m_percov / exp_madad * madad_current;
exp_initial_fix_rider[i] = i_percov / exp_madad * madad_current;

if(m_prem==100000.)
    throw NonFatalError("Expense set "
+temp+"_"+company+"_"+pol_type_expenses + " not in expense table.");

// set prem_code for rider
// Check which table to use - tarif level or product code level
if (prem_lookup_freq_w[i]>0) {

    //Check if tarif exists in the tarif_spec table
    key_temp=prem_key_start;
    if (key_temp != "N/A" && !eq(key_temp,"0"))
        use_tarif_spec_prem_rider[i] = 1;

}    // end if (prem_lookup=="Y")

```

```

        if (use_tarif_spec_prem_rider[i] == 1){// for yrt phi, looks up GP rates in table
based on prem_key in tarif_spec

            // define starting point for prem_key
            xstring prem_key_build=prem_key_start;

            if(prem_key_build == "N/A")
                throw NonFatalError("tarif" + xstring(tarif_rider[i]) + " not in tarif
spec table.");

            // expand starting prem key to include endage, sex, smoker as necessary
            //appends end age from policy to prem_key for 2004 hachnasa btucha tariffs
that allow a range of end ages
            // only end ages listed below are permissible, otherwise use default end age
= 65

            if(eq(premkey_endage,"Y")){
                endage=xstring(age_at_issue + benefit_term/12.);
            if (inlist(endage,"60,62,64,65,67,70"))
                prem_key_build=prem_key_build+endage+"_";
            else
                prem_key_build=prem_key_build+"65_";
            }

            if(eq(premkey_sex,"Y"))
                prem_key_build=prem_key_build+temp_sex+"_";
            if(eq(premkey_smoker,"Y"))
                prem_key_build=prem_key_build+smoker_stat+"_";

            // check if for specific tarif prem rates gross vary by fund name and
prem_profil_type (in/out)
            prem_profil_type = "out";

            if (sm_riders[i]->risk_type>=0.5)
                prem_profil_type = "in";

            prem_code_test=prem_key_build+fund_name_temp+"_"+prem_profil_type+"_";
            prem_code_test_temp=prem_key_build+fund_name_temp+"_"+prem_profil_type+"_0";
            prem_key_test =prem_rates_series;

            if(prem_rates_series== -99999) {
                prem_code_test=prem_key_build+fund_name_temp+"_";
                prem_code_test_temp=prem_key_build+fund_name_temp+"_0";
                prem_key_test =prem_rates_series;

                if(prem_rates_series== -99999) {
                    prem_code_test=prem_key_build;
                    prem_code_test_temp=prem_key_build+"0";
                    prem_key_test =prem_rates_series;

                    if(prem_rates_series== -99999) {
                        throw NonFatalError("premium key " +prem_code_test+"0" +
" not in prem_rates.tbl for policy: "+pol_number +", tarif:"+xstring(tarif_rider[i]));
                    }
                }

            // find appropriate premium series

```

```

end dates          // conditional loop checks if origidate falls within prem series start and
                    j=0;
                    key_temp=prem_code_test+xstring(j);

                    if (eq(prem_series_year,"SA")) {
                    // Find series by SI level

                    row_char = key_temp;
                    while ((sum_ins_curr_rider[i]/1000)>=prem_rates_series_end_im)

                        {          j=j+1;
                                key_temp=prem_code_test+xstring(j);
                        } // end while loop
                    }
                    else {
                        // Find series by start date
                        row_char = key_temp;

                        while (origidate>prem_rates_series_end_im)

                            {          j=j+1;
                                    key_temp=prem_code_test+xstring(j);
                            } // end while loop
                    }
                } // end      if (use_tarif_spec_prem_rider[i] = 1)

                if ((use_tarif_spec_prem_rider[i] < 1) && (prem_lookup_freq_w[i]>0) ) {
                    row = prod_code_rider+"_"+fund_name_temp+"_"+temp_sex+"_"+smoker_stat;
                    col = "Prem_code";
                    key_temp = prem_code_map_tbl;

                }

                //Assign prem_code_rider,
                // First , make key_temp 22 characters long and then add it to prem_code_rider
                key_temp.resize(22);
                prem_code_rider += key_temp;

                //      determine premium rate multiplier from tarif_spec table
                prem_rate_multiplier_rider[i] = atoi(prem_factor)/100.;

                if(prem_factor == "N/A") //if tarif not in tarif spec table
                    prem_rate_multiplier_rider[i] = 1.0;

                // set claims cost lookup code for rider
                if(rider_type_w[i]==3) {
                    key_temp = prod_code_rider + "_" + temp_sex + smoker_stat;
                }

                // Assign clailms cost key
                key_temp.resize(22);
                claims_cost_key_rider +=key_temp;

                // Get rider SI increase perc

                sum_ins_incr_rider[i] = sal_rider_tbl;

```

```

        double lapse_rider = 0;

        lapse_type_col_key = "Lapse";
        lapse_expos_col_key = "premium";
        lapse_rider = lapse_rider_profil_dth/100 * lapse_factor_profil_rider/100;

        lapse_rider_profil_dth_array[i] = 1 - pow (1 - lapse_rider ,1./12.);

        key_temp = temp_sex+smoker_stat;

        key_temp.resize(22);
        decrem_rate_key_rider +=key_temp;
    }        //close if (tarif_rider[i] != 0)

}        //close for loop

}        // close function

```

#### 6.1.1.3.2.10 set\_reinsurance

```

void set_reinsurance (void) {

int yr=0;
tarif_spec_row_key= xstring(tarif);

if (eq(ben_class,"profil")) {
    re_type="simple";
    for (yr =0; yr<149; yr++) {
        comm_ren_re[yr+1] = 0.0;
    }
    expense_re_nom_temp = 0.0;
    comm_prof_re = 0.0;
    interest_rein = 0.0;
    re_clm_ret_fix = sum_ins_curr;
    prem_per_unit_si_re = 0.0;

    re_clm_rein_pc = reinsur_simple_perc;
    //if assumption set not found
    if (re_clm_rein_pc == -99999.)
        re_clm_rein_pc = 0.0;
    else
        re_cost_perc =reinsure_simple_cost;

    // Populate riders reinsurance parameters
    for (int i=0; i < riders_count_w; i++) {

        if (tarif_rider[i] != 0) {
            rider_tarif_row_key = xstring(tarif_rider[i]);
            prod_code_rider = rider_tarif_tbl;
            re_cost_pc_rider[i] = reinsur_simple_rider_cost;
            //if assumption set not found
            if(reinsur_simple_rider_cost == -99999.)
                re_cost_pc_rider[i] = 0.0;
        }

    }

}
}

```



```
else {
// find appropriate reins_set series
// conditional loop checks if origidate falls within prem series start and end dates
    int j=0;

    rein_key_temp= rein_set+"_" + xstring(j);

    while (origidate > atoi(rein_series_end_key_temp))
    {
        j=j+1;
        rein_key_temp=rein_set+"_" +xstring(j);
    } // end while loop
    rein_set=rein_key_temp;

    // Set up regular reinsurance commission percentage (on premium re.)
    double comm = 0.0;
    for (yr =0; yr<149; yr++) {
        reinsur_comm_key = "COMM_"+xstring(yr+1);
        comm = atof(reinsur_comm);
        //if no error
        if(comm!= -99999. && comm != 99999.)
            comm_ren_re[yr + 1] = comm;
        //if column (year) not found (end of table)
        if(comm == 99999.)
            comm_ren_re[yr + 1] = comm_ren_re[yr];
        //if assumption set not found
        if(comm == -99999.)
            throw NonFatalError("Error looking up commission from Life Treaty Details
table row: " + rein_set);
    } // end for loop

    expense_re_nom_temp =atof(exp_re_nom);
    comm_prof_re = atof(prof_comm);
    interest_rein = atof(interest);

    re_clm_ret_fix = atof(retention) /
        atof(madad) * madad_current;

    re_clm_rein_pc = atof(quota_share_ppn);
    if (re_clm_rein_pc>1.0)
        re_clm_rein_pc = 100.0 - re_clm_rein_pc;

    re_type =type;
    prem_per_unit_si_re = atof(prem_extra);
    // Set reinsurance YRT prem rate ref

    prem_re_bw = blue_white;
    prem_re_wp = waiting_period_modeled;

    if (eq(re_type,"YRT")) {
        if(inlist(ben_class,"phi,ltc")) {

            prem_re_row_key =reins_key_start; // from tarif_spec table
```

```
        if(eq(prod_code,"phi-mitriya")){

            if (inlist(occ_key, "1,2"))
                prem_re_occ= occ_key;
            else
                prem_re_occ = "1";
        }

        else {

            if (inlist(occ_key, "1,2,3"))
                prem_re_occ= occ_key;
            else
                prem_re_occ = "3";
        }

        //appends end age from policy to claims_cost_key for 2004 hachnasa btucha
tariffs that allow a range of end ages
        // only end ages listed below are permissible, otherwise use default end age
= 65

        xstring endage;
        endage=xstring(age_at_issue + benefit_term/12.);

        if(eq(prod_code,"phi-mitriya")){

            if (inlist(endage,"60,62,64,65,67,68,69,70"))
                prem_re_endage= endage;
            else
                prem_re_endage= "65";
        }

        else {

            if (inlist(endage,"60,62,64,65,67"))
                prem_re_endage= endage;
            else
                prem_re_endage= "65";
        }

        if (inlist(sex,"M,F"))
            prem_re_sex=sex;
        else
            prem_re_sex="M";

    }
else { //other products - not LTC or PHI
    prem_re_row_key= reinsur_kod_tavla;
}
```

```

        // conditional loop checks if origidate falls within prem series start and end dates

    } //end if re_type = YRT

} //end else

if (submodel == "TRAD")
    re_clm_ret_fix = (sum_ins_curr - resinforce) * benefits_curr;

// close function
}

```

#### 6.1.1.3.2.11 set\_saving

```

void set_saving(void) {

    int i=0;

    // *****Set variables in submodel:*****
    // Accumulation Units for premium paying policies
    for (i = 0; i < 16; i++) {
        saving->alloc_rate[i] = 100.;
        saving->alloc_rate_period[i] = 600;
    }

    saving->par_nonpar = par_nonpar;
    saving->commence_period_w = commence_period_w;
    saving->elapsed_months = elapsed_months;
    saving->maturity_period_w = maturity_period_w;
    saving->mgt_fee_fixed = mgt_fee_fixed_puresav;
    saving->mgt_fee_variable = mgt_fee_variable;
    saving->paid_up = paid_up;
    saving->policies_curr = policies_curr;
    saving->benefits_curr = benefits_curr;
    saving->decrements_apply = decrements_apply;
    saving->prem_freq = prem_freq;
    saving->projection_type = projection_type;

    for (i = 0; i<116; i++) {
        saving->comm_regular_pc[i] = 0.0;
        sm_saving->comm_perc_res[i] = comm_perc_res_b[i];
    }

    saving->comm_ren_perc_prem = comm_ren_perc_sav;
    saving->comm_renewal_year = 1; //comm_renewal_yr_sav
    sm_saving->unit_value_if = unit_value_savings;
    sm_saving->unit_type = "Saving";

    // close function
}

```

#### 6.1.1.3.2.12 set\_saving\_pup

```

void set_saving_pup(void) {

```

```

int i=0;

// *****Set variables in submodel:*****

saving_pup->par_nonpar = par_nonpar;
saving_pup->commence_period_w = commence_period_w;
saving_pup->elapsed_months = elapsed_months;
saving_pup->maturity_period_w = maturity_period_w;
saving_pup->mgt_fee_fixed = mgt_fee_fixed_puresav;
saving_pup->mgt_fee_variable = mgt_fee_variable;
saving_pup->paid_up = paid_up;
saving_pup->policies_curr = policies_curr;
saving_pup->benefits_curr = benefits_curr;
saving_pup->projection_type = projection_type;
saving_pup->decrements_apply = decrements_apply;

for (i=0; i < 116; i++)
    sm_saving_pup->comm_perc_res[i] = comm_perc_res_b[i];

sm_saving_pup->unit_value_if = unit_value_savings;
sm_saving_pup->unit_type = "Saving_pup";

// close function
}

```

#### 6.1.1.3.2.13 validate\_data

```

void validate_data(void) {

    if (prem_term > benefit_term)
        error_msg = "prem_term_>_ben_term";

    if (pv_period != 12)
        throw NonFatalError("Template set up for monthly projections. Change the discount period in task.");

    // check minimum age of mortality tables

    int min_age_1 = 0;

    if (age_at_issue < min_age_1)
        throw NonFatalError("Policy number " + pol_number + " : Issue age of life 1 is less than the minimum age of the mortality table");

    if (eq(projection_type,"Valn") && elapsed_months < 0)
        error_msg = "elapsed_months_<_0";

    if ((elapsed_months > benefit_term) && (life->submodel != "TRAD"))
        error_msg = "elapsed_months_>_ben_term";

    if ((sum_ins_curr <= 0.0001) && (life->submodel == "TERM"))
        error_msg = "Sum_Insured_<=_0";
}

```

```
// Ensure frequencies given are factors of 12.
if (!eq(paid_up,"Y") || prem_term > elapsed_months){
  if (prem_freq == 0)
    throw NonFatalError("Premium frequency must be 1, 2, 3, 4, 6, or 12.");

  // Premium frequency must be a multiple of 12 whilst premiums are being paid
  if (12 % prem_freq != 0)
    throw NonFatalError("Premium frequency must be 1, 2, 3, 4, 6, or 12.");
}

if (eq(ben_class,"adif") && eq(paid_up,"N") && (saving_perc < alloc_rate[0]))
  error_msg = "saving_perc_too_low";

if ((life->prem_freq == 0 || eq(life->paid_up,"Y")) && (life->promil > 0.3) && !eq(life->ben_class,"ltc") && (life->submodel == "TERM"))
  life->error_msg = "paid_up";

// close function
}
```

### 6.1.1.3.3 Temporary Tables

#### 6.1.1.3.3.1 charge\_amount\_tt

```
if (r <= commence_period_w || r > maturity_period_w || riders_count_w == 0 || tarif_rider[c]==0)
  return 0.;

// use discount_factor to reduce premium if rider is "out" of profil to offset premium charges on
the rider premium
double discount_factor = 1.0;

if (sm_riders[c]->prm_in_ppn<=0.5) // rider premium is "out"
  discount_factor = accum->allocation_rate(r);

if (prem_lookup_freq_w[c]==0)//level premium rider
  return (sm_riders[c]->prem_cover/12.) * benefits_curr_rider[c] * discount_factor;

if (elapsed_months + r <= xint(discount_period_rider[c]))
  discount_factor = discount_factor * (1. - discount_perc_rider[c]/100.);

// for other products claim rate is an annual rate applied to the sum insured
return sum_insured_rider_tt(r, c)
  * charge_rate_tt(pol_year(r), c) / 12.
  * benefits_curr_rider[c]
  * discount_factor;
```

#### 6.1.1.3.3.2 charge\_rate\_tt

```
// *** may need to adjust charge rate to monthly depending on frequency in table)
// *** problem if r=t because of negative periods
// r = policy year , starting from 1 to term
if (r <= 0 || r > xint(benefit_term/12.))
  return 0;

if (tarif_rider[c]==0 || prem_lookup_freq_w[c]==0)
  return 0;
```

```

int age_adj = xint(age_at_issue)+r-1;
// Assume all lives die at omega age
if (age_adj >= omega_age_w)
    return 1.0;

// ***** Calculate charge rate *****

int premium_year=1;
if (xint(fmod(r,prem_lookup_freq_w[c])) == 0) // *** r=t?
    premium_year = prem_lookup_freq_w[c];
else
    return charge_rate_tt(r-1,c);

xstring row = prem_code_rider;

row = row(c*22,22);

row = row.strip();

charge_rate_tt_row=row;
charge_rate_tt_col=xstring(age_adj);
double rate_c = prem_rates_charge_tt;

rate_c = (rate_c / si_unit_w[c]) * (1.+charge_addition_perc[c]/100.) *
premi_rate_multiplier_rider[c] + charge_addition_absolute[c];

return rate_c;

```

#### 6.1.1.3.3.3 claim\_amount\_tt

```

// *** problem if r=t because of negative periods
if (r <= commence_period_w || r > maturity_period_w)
    return 0.;

if (tarif_rider[c]==0)
    return 0;

return sum_insured_rider_tt(r, c) * claim_rate_tt(pol_year(r),c)
    * benefits_curr_rider[c];

```

#### 6.1.1.3.3.4 claim\_cost\_tt

```

// *** may need to adjust charge rate to monthly depending on frequency in table)
// *** problem if r=t because of negative periods
// r = policy year , starting from 1 to term
if (r <= 0 || r > xint(benefit_term/12.))
    return 1.;

if (tarif_rider[c]==0)
    return 1.;

// for fib products add claims costs to claim rate

if (rider_type_w[c] == 3 && risk_type_w[c] == 1){ //FIB benefit uses claims cost tables with
assumed maturity age of 65
    row = claims_cost_key_rider(c*22,22);
    row = row.strip();
    age_adj = min(max(18,xint(age_at_issue)+r-1),65);
    return claim_cost_factors_tbl;
}

```

```
}
```

```
// for other products claim rate is an annual rate applied to the sum insured
return 1.;
```

#### 6.1.1.3.3.5 claims\_mult\_tt

```
// r = policy year , starting from 1 to term
rider_tarif_row_key = xstring(tarif_rider[c]);
prod_code_rider = rider_tarif_tbl;
double temp =clms_mult_tt;
```

```
//if column (year) not found
if(temp == -99999.)
    return claims_mult_tt(r-1,c);
```

```
return temp;
```

#### 6.1.1.3.3.6 decrement\_tt

```
// *** problem if r=t because of negative periods
if (r <= commence_period_w || r > maturity_period_w)
    return 0.;
```

```
if (tarif_rider[c]==0)
    return 0;
```

```
// for death riders (risk_type=1) the inforce is reduced by normal death rates in the main model
if (risk_type_w[c]==1)
    return 0.;
```

```
// if a previous rider has the same type of risk/decrement, then should not double-decrement:
for (int i=0; i < c; i++) {
    if (risk_type_w[i]==risk_type_w[c])
        return 0.;
```

```
}
```

```
int age_temp = age_adj;
col = xstring(risk_type_w[c]) + "_" + sex + smoker_stat;
age_adj = age_temp; // repeat lookup before table call
double rate_c = claim_rates_tbl;
rate_c = rate_c * (1.+charge_addition_perc[c]/100.);
rate_c = rate_c + charge_addition_absolute[c];
rate_c = rate_c/12.0; // convert to monthly
```

```
return rate_c;
```

#### 6.1.1.3.3.7 prem\_rates\_extra\_tt

```
// rows (r, 0 to 100) are the current age
// columns (c, 0 to 3) are for the sex smoker combinations MN, MS, FN, FS
```

```
// ***** Return charge rate *****
double rate = 0.0;
prod_code_adif_extra_prem = prod_code_adif_extra_prem_temp;
prem_rate_row = r;
prem_rate_col = sex+smoker_stat;
rate = prem_rates_extra_prm;
```

```
//if no error
if(rate != -99999. && rate != 99999.)
    return rate * (1.+health_occ_perc/100.) /1000.;

//if row (age) not found
if(rate == -99999.)
    return 0.0;

//if assumption set not found
if(rate == 99999.)
    throw NonFatalError("Error looking up .... "+ xstring(r) + ", "+sex+smoker_stat+ " , in
prem_rates_extra_tbl");

return 0.0;
```

#### 6.1.1.3.3.8 sum\_ins\_basic\_tt

```
// rows (r, 0 to 100) are the current age
// columns (c, 0 to 3) are for the sex smoker combinations MN, MS, FN, FS
if(submodel == "ANN")
    return 0.0;

double suminsbas=0.0;
suminsbas_row= r;
suminsbas_col=sex+smoker_stat;

suminsbas = suminisba_tbl;

//if no error
if(suminsbas!= 99999. && suminsbas!= -99999.)
    return suminsbas / (1.+(health_occ_perc/350.));

//if row (age) not found
if(suminsbas == -99999.)
    return 0.;

//if column (ses, smoke) not found
if(suminsbas == 99999.)
    throw NonFatalError("Error looking up .. age: "+ xstring(r) + ", for column:"+sex+
smoker_stat+ " in suminisba_tbl.");

return 0.0;
```

#### 6.1.1.3.3.9 surr\_charge\_tt

```
// r row = policy duration in months
// c = unit type, 0=accum, 1=saving
xstring temp;
double surchg=0.0;

if (submodel == "ANN")
    return 0.0;

if (c==0)
    temp = "accum_" + surr_charge_set_temp;
else
    temp = "saving_" + surr_charge_set_temp;

if (surr_chg_set == "default")
```



```

        temp = "saving_private_zero";

    if (r==0)
        return 0.0;

    col_char = temp;
    surchg = surr_chg_tbl;

    if (surchg == -9999) // row (month) not found
        return surr_charge_tt(r-1,c);

    if (surchg == -999999) // assumption set not found
        throw NonFatalError("Error looking up "+temp + " in surr_chg table in fixed tables
file.");

    return surchg; //no error

```

#### 6.1.1.3.3.10 claim\_rate\_tt

```

// *** may need to adjust charge rate to monthly depending on frequency in table)
// *** problem if r=t because of negative periods
// r = policy year , starting from 1 to term
if (r <= 0 || r > xint(benefit_term/12.))
    return 0;

if (tarif_rider[c]==0)
    return 0;

// ***** Calculate adjusted age *****
// Assume rates in mort table are for age last birthday
int age_temp = xint(age_at_issue)+r-1;
// Assume all lives die at omega age *** maybe change this???
if (age_temp >= omega_age_w)
    return 1.0;

double rate_c=0.0;

if (risk_type_w[c] == 0)
    return 0;

// for death covers use normal mortality, converted to annual rate
if (risk_type_w[c] == 1){
    rate_c = death_rate(r*12-elapsed_months) * 12.;
}
else { //other cover, lookup from profil decrement rate table
    // for other products claim rates are from claim_rates_tbl by risk-type
    col = xstring(risk_type_w[c]) + "_" + sex + smoker_stat;
    age_adj = age_temp;
    rate_c = claim_rates_tbl;
    rate_c = rate_c * (1.+charge_addition_perc[c]/100.) + charge_addition_absolute[c];
    if (margin_add=="Y")
        rate_c = rate_c * (1+margin_claims/100);
    rate_c = rate_c * claims_mult_tt(r,c);
}

// claim rate is an annual rate, so convert to monthly

return rate_c / 12.;

```

**6.1.1.3.3.11 sum\_insured\_rider\_tt**

```

// r = t, c=rider no.
if (r < commence_period_w || r > maturity_period_w) // *** r=t?
    return 0.0;

if (tarif_rider[c]==0)
    return 0;

if (r == 0)
    return sum_ins_curr_rider[c];

//Set increase perc on pol ann
double si_inc_rid = 1.0;
if ((xint(pol_month(r)) == 1) && (r > 0))
    si_inc_rid = 1 + sum_ins_incr_rider[c] / 100.;
//Calc Accumulated increase perc
double si_inc_rid_acc= 1.0;
if (sum_ins_incr_rider[c] > 0.0)
    si_inc_rid_acc = pow(1 + sum_ins_incr_rider[c] / 100.,int(r/12));
//
double si_surv_rid = 1.0;
if (lapse_rider_profil_dth_array[c] > 0.0)
    si_surv_rid = 1 - lapse_rider_profil_dth_array[c];

double si_surv_rid_acc = 1.0;
if (lapse_rider_profil_dth_array[c] > 0.0)
    si_surv_rid_acc = pow(1 - lapse_rider_profil_dth_array[c],r);

if (rider_type_w[c] == 1) { // sum at risk decreasing with saving acc.
    // paid up covers do not buy extra insurance
    double savings = sm_accum->units_b_bef(r) + sm_accum->alloc_units(r)
        + sm_saving->units_b_bef(r) + sm_saving->alloc_units(r) ;
    // to get savings per benefit & before lapses
    if (surv_act_prm(r-1)<0.00000001 || benefits_curr == 0)
        return 0.0;
    else {
        savings = savings/surv_act_prm(r-1)/benefits_curr;
        return max(0.0 , sum_insured_rider_tt(0,c)*si_inc_rid_acc - savings)*si_surv_rid_acc;
    }
}

// *** r=t?
}

if (rider_type_w[c]==3 && claim_cost_tt(pol_year(0),c) != 0 ) // for the FIB benefit the cc factor
is in the claims cost table with maturity age =65
    return sum_insured_rider_tt(0,c)*si_inc_rid_acc *si_surv_rid_acc/
claim_cost_tt(pol_year(0),c) * claim_cost_tt(pol_year(r),c) ;

// not fixed sum-insured and does not need extra reduction to inforce of rider - fixed sum at risk
return sum_insured_rider_tt(r-1,c) * si_inc_rid *si_surv_rid;

```

**6.1.1.3.4 Scalars****6.1.1.3.4.1 fund\_type**

```

if(par_npar == 0)
    return "N";

```

```
return "P";
```

#### **6.1.1.3.4.2 min\_retirement\_age**

```
return sm_annuity[0]->takeup_age;
```

#### **6.1.1.3.4.3 mult\_age\_ind**

```
if(submodel == "ANN" || submodel == "TERM" || (submodel == "TRAD" && !eq(ben_class, "gimla")))  
    return 0;
```

```
if(!inlist(policy_type,"private,selfemp") || res_kitzba >0.)  
    return 1;
```

```
return 0;
```

#### **6.1.1.3.4.4 use\_uw\_date**

```
if (inlist(life->decrem_rates_tbl, "adi_08,adb_100"))  
    return "Y";
```

```
else
```

```
    return "N";
```

#### **6.1.1.3.4.5 profit\_net\_vif\_pv12**

```
return profit_net_vif_pv(12);
```

#### **6.1.1.3.4.6 res\_prop\_kitzba**

```
if (eq(submodel,"TERM"))  
    return 0.0;
```

```
if (resinforce >0. && eq(submodel,"TRAD"))  
    return min(res_kitzba / resinforce, 1.0);
```

```
if (resinforce >0. && eq(submodel,"ANN"))  
    return 1.0;
```

```
if (resinforce >0. && eq(submodel,"UNIT")) {  
    if (unit_value_accum == 0.)  
        return 0.;  
    return min(res_kitzba / unit_value_accum, 1.0);}
```

```
if (!eq(policy_type,"private"))  
    return 1.0;
```

```
return 0.0;
```

#### **6.1.1.3.4.7 res\_prop\_kitzba\_newtag**

```
if (eq(submodel,"TERM"))  
    return 0.0;
```

```
if (res_prop_kitzba >0.){
```

```
    res_prop_key = "new_tag";  
    double temp = res_prop_newtag_data;
```

```
    if (temp == -9999.){ //Policy does not appear in TK file

        temp_col_fund=policy_type + "_prop_newtag";
        return max(min(fund_rates_code_tbl/100., 1.), 0);

    }

    else

        return max(min(temp, 1.), 0.);
}

return 0.0;
```

#### **6.1.1.3.4.8      res\_prop\_kitzba\_oldtag**

```
if (eq(submodel,"TERM"))
    return 0.0;

if (res_prop_kitzba >0.){

    res_prop_key = "old_tag";
    double temp = res_prop_old_data;

    if (temp == -9999.){ //Policy does not appear in TK file

        temp_col_fund=policy_type + "_prop_oldtag";
        return max(min(fund_rates_code_tbl/100., 1.), 0);

    }
    else

        return max(min(temp, 1.), 0.);

}

return 0.0;
```

#### **6.1.1.3.4.9      res\_prop\_kitzba\_piz**

```
if (eq(submodel,"TERM"))
    return 0.0;

if (res_prop_kitzba >0.){

    res_prop_key = "piz";
    double temp = res_prop_piz_data;

    if (temp == -9999.) {

        temp_col_fund =policy_type + "_prop_newpiz";
        return max(min(fund_rates_code_tbl/100., 1.), 0);

    }
    else

        return max(min(temp, 1.), 0.);

}

return 0.0;
```

**6.1.1.3.4.10 res\_prop\_kitzba\_prat**

```

if (eq(submodel,"TERM"))
    return 0.0;

if (res_prop_kitzba >0.){

    res_prop_key = "prat";
    double temp = res_prop_prat_data;

    if (temp == -9999.){

        temp_col_fund = policy_type + "_prop_prat";
        return max(min(fund_rates_code_tbl/100., 1.), 0);

    }
    else
        return max(min(temp, 1.), 0.);

}

return 0.0;

```

**6.1.1.3.4.11 res\_total\_increase1**

```

return reserve_increase(1)- reserve_re_increase(1);

```

**6.1.1.3.4.12 resanndef\_atmat**

```

if(!inlist(submodel,"UNIT"))
    return 0;

if(annuitization_rate<=0.00001 || res_prop_kitzba<=0.0) // *** need way to distinguish between
policies with and without guarantees, and with and without kitzva option
    return 0.0;

return res_ann_deficiency(maturity_period_w-1);

```

**6.1.1.3.4.13 reserve\_opening\_difference**

```

double res_IF_fac = benefits_curr;

if (eq(projection_type,"Rollup"))
    return 0.0;

// For Klasi, Adif and Profil, use initial difference between calculated reserve and ResInforce
// Adjust for " * benefits_curr" as lapsed policies will get SV (close to reserve) and for Hasne
covers
if (inlist(submodel,"TRAD,ANN")) {
    return reserve_basic(0) - resinforce * res_IF_fac;}

double multage = 0;
if (eq(submodel,"UNIT")) {
    multage = sm_accum->units_e(0) * bonus[prem_term]/100.;
    if(mult_age_ind == 1)
        multage = sm_annuity->reserve_bonus_units_e_0(1);
    return reserve_basic(0) + multage
        - resinforce * res_IF_fac;
}

```

```
// For Term adjustment depends on reserve type
// For Net Premium reserve, adjust for initial difference, taking into account NP_deficiency
reserve
// Do not adjust for "* benefits_curr" (unless Hasne or LTC) as no surr value for risk covers. If
policy lapses, reserve is released and should be included
// Adjust for "* benefits_curr" (Hasne or LTC) as surr value and so no reserve is released
if (eq(res_basis, "Net_Prem")) {
    if (!eq(company, "hasne") && !eq(ben_class, "ltc"))
        res_IF_fac = 1.0;

    return reserve_basic(0) + res_np_deficiency(0) - resinforce * res_IF_fac;
}
// For dd, LBNR releases some reserve.
if (eq(ben_class, "dd"))
    return reserve_basic(0) * min(benefits_curr - 1.0, 0);

// For Res as ppn Prem, no adjustment is needed (Risk ResInforce is IBNR)
return 0.0;
```

#### 6.1.1.3.4.14 premium\_disc\_pv\_start

```
if (commence_period_w > -12 && commence_period_w <= 0) {
    return premium_disc_pv(commence_period_w);
}
return 0.0;
```

#### 6.1.1.3.4.15 premium\_nb\_sp

```
// Calculate single premium at eom t=0, required to be paid
// such that the calculated account balance = current actual account balance
if (!eq(projection_type, "Rollup"))
    return 0.0;

if (!eq(submodel, "UNIT"))
    return 0.0;

return sm_accum->premium_nb_sp +
    sm_acc_pup->premium_nb_sp +
    sm_saving->premium_nb_sp +
    sm_saving_pup->premium_nb_sp;
```

#### 6.1.1.3.4.16 premium\_pv\_st\_date

```
if (commence_period_w >= -12 && commence_period_w < 0) {
    return premium_pv(commence_period_w);
}
return 0.0;
```

#### 6.1.1.3.4.17 claims\_re\_yr1

```
double temp = 0.0;
int i = 0;

for (i = 1; i <= 12; i++) {
    temp = temp + claims_re(i);
}

return temp;
```

**6.1.1.3.4.18 claims\_total\_yr1**

```
double temp =0.0;
int i =0;

for (i=1; i <= 12; i++) {
    temp = temp + claims_total(i);
}

return temp;
```

**6.1.1.3.4.19 comm\_re\_prof\_yr1**

```
double temp =0.0;
int i =0;

for (i=1; i <= 12; i++) {
    temp = temp + comm_re_prof(i);
}

return temp;
```

**6.1.1.3.4.20 comm\_re\_yr1**

```
double temp =0.0;
int i =0;

for (i=1; i <= 12; i++) {
    temp = temp + comm_re(i);
}

return temp;
```

**6.1.1.3.4.21 comm\_total\_yr1**

```
double temp =0.0;
int i =0;

for (i=1; i <= 12; i++) {
    temp = temp + comm_total(i);
}

return temp;
```

**6.1.1.3.4.22 expense\_total\_yr1**

```
double temp =0.0;
int i =0;

for (i=1; i <= 12; i++) {
    temp = temp + exp_total(i);
}

return temp;
```

**6.1.1.3.4.23 prem\_discount\_py1**

```
double temp = 0.0;
int i=0;
```

```
for (i=elapsed_months+1; i <= elapsed_months+12; i++)
    temp = temp + premium_disc(i);
```

```
return temp;
```

#### **6.1.1.3.4.24 prem\_discount\_py2**

```
double temp = 0.0;
int i=0;
```

```
for (i=elapsed_months+13; i <= elapsed_months+24; i++)
    temp = temp + premium_disc(i);
```

```
return temp;
```

#### **6.1.1.3.4.25 prem\_discount\_py3**

```
double temp = 0.0;
int i=0;
```

```
for (i=elapsed_months+25; i <= elapsed_months+36; i++)
    temp = temp + premium_disc(i);
```

```
return temp;
```

#### **6.1.1.3.4.26 premium\_gross\_yr1**

```
double temp =0.0;
int i =0;
```

```
for (i=1; i <= 12; i++) {
    temp = temp + premium_gross(i);
}
```

```
return temp;
```

#### **6.1.1.3.4.27 premium\_re\_yr1**

```
double temp =0.0;
int i =0;
```

```
for (i=1; i <= 12; i++) {
    temp = temp + premium_re(i);
}
```

```
return temp;
```

#### **6.1.1.3.4.28 profit\_net\_vif\_yr0**

```
double temp =0.0;
int i =0;
```

```
for (i=-11; i <= 0; i++) {
    temp = temp + profit_net_vif(i);
}    //close for loop
```

```
return temp;
```



**6.1.1.3.4.29 comm\_nihul\_pv\_start**

```
if (commence_period_w>=-12 && commence_period_w<=0)
    return comm_nihul_pv(commence_period_w);

return 0.0;
```

**6.1.1.3.4.30 comm\_prizes\_new**

```
if (commence_period_w>=-12 && commence_period_w<=0)
    return comm_prize(commence_period_w+1);

return 0.0;
```

**6.1.1.3.4.31 comm\_pv\_start**

```
if (commence_period_w>=-12 && commence_period_w<=0)
    return comm_pv(commence_period_w);

return 0.0;
```

**6.1.1.3.4.32 exp\_inflation\_mthly**

```
return monthly_rate(infl_rate_expenses);
```

**6.1.1.3.4.33 units\_to\_ann**

```
if (submodel == "TRAD" || submodel == "TERM")
    return 0;

if(mult_age_ind == 1)
    return sm_annuity->initial_annuity_purchase;

return sm_annuity[ann_index_map[takeup_age]]->initial_annuity_purchase;
```

**6.1.1.3.4.34 claims\_pv\_st**

```
if (commence_period_w >=-12 && commence_period_w<=0)
    return claims_pv(commence_period_w);

return 0.0;
```

**6.1.1.3.4.35 comm\_clawback\_pv\_start**

```
if (commence_period_w>=-12 && commence_period_w<=0)
    return comm_clawback_pv(commence_period_w);

return 0.0;
```

**6.1.1.3.4.36 charges\_premium\_pv\_st**

```
if (commence_period_w >=-12 && commence_period_w<=0)
    return charges_premium_pv(commence_period_w);

return 0.0;
```

**6.1.1.3.4.37 comm\_hekef\_new**

```
if (commence_period_w >=-12 && commence_period_w <=0)
    return comm_hekef(commence_period_w+1);
```

```
return 0.0;
```

**6.1.1.3.4.38 comm\_init\_new**

```
if (commence_period_w >=-12 && commence_period_w <=0)
    return comm_prize(commence_period_w+1)+comm_hekef(commence_period_w+1);
```

```
return 0.0;
```

**6.1.1.3.4.39 comm\_reg\_pv\_st**

```
if (commence_period_w >=-12 && commence_period_w <=0)
    return comm_reg_pv(commence_period_w);
```

```
return 0.0;
```

**6.1.1.3.4.40 comm\_reg\_riders\_out\_pv\_st**

```
if (commence_period_w >=-12 && commence_period_w <=0)
    return comm_reg_riders_out_pv(commence_period_w);
```

```
return 0.0;
```

**6.1.1.3.4.41 comm\_ren\_pv\_st**

```
if (commence_period_w >=-12 && commence_period_w <=0)
    return comm_renewal_pv(commence_period_w);
```

```
return 0.0;
```

**6.1.1.3.4.42 comm\_res\_pv\_st**

```
if (commence_period_w >=-12 && commence_period_w <=0)
    return comm_reserve_pv(commence_period_w);
```

```
return 0.0;
```

**6.1.1.3.4.43 management\_fee\_pv\_st**

```
if (commence_period_w >=-12 && commence_period_w <=0)
    return management_fee_pv(commence_period_w);
```

```
return 0.0;
```

**6.1.1.3.4.44 proj\_task\_loop\_num\_scalar**

```
return proj_task_loop_num;
```

**6.1.1.3.4.45 duration**

```
if (duration_denominator(0) > 0)
    return duration_numerator(0) / duration_denominator(0);
```

```
return 0.0;
```

**6.1.1.3.4.46 expense\_init\_new**

```
if (commence_period_w >=-12 && commence_period_w<= 0)
    return expense_initial_fix(commence_period_w + 1)
        + expense_initial_perc(commence_period_w + 1);

return 0.0;
```

**6.1.1.3.4.47 expense\_pv\_start**

```
if (commence_period_w>=-12 && commence_period_w<=0)
    return expense_pv(commence_period_w);

return 0.0;
```

**6.1.1.3.4.48 policies\_new**

```
if (commence_period_w>-12 && commence_period_w<=0)
    return policies_b(commence_period_w+1);

return 0.0;
```

**6.1.1.3.4.49 premium\_new**

```
if(submodel == "ANN")
    return 0.0;

if (commence_period_w>=-12 && commence_period_w<0) {
    if (paid_up=="Y")
        return premium_gross(commence_period_w+1);

    return premium_gross(commence_period_w+1)*prem_freq;
}
return 0.0;
```

**6.1.1.3.4.50 origidate**

```
// to back into origidate of cover define origidt_mths
double origidt_mths = valn_year*12+valn_month-elapsed_months-elapsed_months_extra+1;
// dividing origidt_mths by 12 gives year and number of months as fraction
double result = int(origidt_mths/12)*100+(origidt_mths-int(origidt_mths/12)*12);

return result;
```

**6.1.1.3.4.51 yob**

```
//return valn_year - age_last(1) +1;
return cal_year(1) - age_last(1);
```

**6.1.1.3.4.52 prem\_alloc\_pv**

```
if (!eq(submodel,"UNIT"))
    return 0.0;

if (premium_pv(0)<=0.0)
    return 0.0;

int i=1;
double alloc_pv = 0.0;
```

```

if (eq(ben_class,"profil") && paid_up=="Y")
    return premium_gross(commence_period_w+1);

// Calculate PV of premium allocated to savings
for (i=maturity_period_w; i >0; i--) {

    int proj_yr = xint(proj_year(i+1));
    if(eq(projection_type_int, "Rollup"))
        proj_yr = xint(proj_year_rollup(i+1));

    alloc_pv = alloc_pv * v_month_t[proj_yr]
        + alloc_units(i)
        - cover_charge(i);

}

return    alloc_pv;

```

#### **6.1.1.3.4.53 premium\_1**

```
return    premium(1) + pol_fee(1);
```

#### **6.1.1.3.4.54 reins\_simple\_rider\_row**

```

xstring reins_col = prod_code_rider+"_"+company;

return reins_col;

```

#### **6.1.1.3.4.55 reins\_simple\_row**

```

xstring reins_col = prod_code+"_"+company;

return reins_col;

```

#### **6.1.1.3.4.56 ktest**

```

xstring temp;

if ((fund_group == "P1" || atoi(fund_yesodi) <= 24)
    && inlist(ben_class,"adif,gimla")
    && prod_code != "a80-01hon"
    && prod_code != "a80-01kitz")
    temp = "Y";
else
    temp = "N";

return temp;

```

#### **6.1.1.3.4.57 portfolio**

```
return group;
```

#### **6.1.1.3.4.58 prod\_specs\_max\_perc**

```
return "rider_max_perc_"+policy_type;
```

#### **6.1.1.3.4.59 cashflow\_re\_pv\_st**

```

if (commence_period_w >=-12 && commence_period_w<=0)
    return cashflow_re_pv(commence_period_w);

```

```
return 0.0;
```

#### **6.1.1.3.4.60 reins\_comm1**

```
return comm_re(1) + comm_re_prof(1);
```

#### **6.1.1.3.4.61 reinsur\_clm\_cost**

```
xstring row_reinsur = prod_code+"_"+company;
```

```
return row_reinsur;
```

#### **6.1.1.3.4.62 reinsur\_kodtavla**

```
xstring reinsur_col = "KOD_TAVLA_" + sex + smoker_stat;
```

```
return reinsur_col;
```

#### **6.1.1.3.4.63 reserve\_rein\_opening**

```
if (submodel != "TERM")  
    return 0.0;
```

```
return term->reserve_re(0);
```

#### **6.1.1.3.4.64 reserve\_manual**

```
return resinforce_input * res_adj_factor;
```

#### **6.1.1.3.4.65 datetime\_stamp**

```
return to_string(time(0));
```

#### **6.1.1.3.4.66 sex\_smoker\_code**

```
int temp = 0;
```

```
if (sex=="F")  
    temp = 1;
```

```
if (smoker_stat=="S")  
    return temp * 2 + 1;
```

```
return temp * 2;
```

#### **6.1.1.3.4.67 stamp\_output**

```
//Returns output file name
```

```
//xstring s = output_location();
```

```
//int extension = s.find("~life.csv");
```

```
return output_location();
```

**6.1.1.3.4.68 value\_date**

```
//int y = valn_year - xint(valn_year/100.)*100;

xstring vyear = xstring(valn_year);
xstring vmonth = xstring (valn_month);
if (vmonth.length() == 1)
    vmonth = "0"+vmonth;

return vyear + vmonth + "01";
```

**6.1.1.4 sub\_2\_cflow****6.1.1.4.1 Columns****6.1.1.4.1.1 annuity\_if\_b\_bef\_ret**

```
if (t <= life->commence_period_w || t > life->mat_period_original)
    return 0.0;

if(life->submodel != "TRAD" || life->paid_up=="Y" || ann_factor_pol == 0)
    return 0.0;

return (sum_insured(t-1) * life->surv_act_bal_bef_ret(t-1)) / ann_factor_pol * 100.;
```

**6.1.1.4.1.2 bonus**

```
if (t <= 0 || (life->par_nonpar=="N") || t > life->mat_period_original || eq(life->paid_up,"G"))
    return 0.0;

if(life->submodel != "TRAD")
    return 0.0;

if (life->par_nonpar=="N")//not participating
    return 0;

return bonus_rate(t)
    * ( res_to_bonus(t) + bonus_b(t) )

    + bonus_b(t) * int_rate_res_mthly;
```

**6.1.1.4.1.3 bonus\_b**

```
if ((life->par_nonpar=="N") || t > life->mat_period_original || t <= life->commence_period_w
||eq(life->paid_up,"G") || t < 0)
    return 0.0;

if(life->submodel != "TRAD")
    return 0.0;

if (t == 0){
    if(life->paid_up == "Y")
        return 0.0;
    return life->bonus_inforce;
}
```

```

return bonus_if(t-1)
    * life->surv_per_ret(t-1) //Probability not retired
    * life->surv_per_act_bal_bef_ret(t); //Probability remains premium-paying

```

#### 6.1.1.4.1.4 bonus\_b\_pup

```

if ((life->par_nonpar=="N") || t > life->mat_period_original || t <= life->commence_period_w
||eq(life->paid_up,"G") || t < 0)
    return 0.0;

if(life->submodel != "TRAD")
    return 0.0;

if (t == 0){
    if(life->paid_up == "Y")
        return life->bonus_inforce;
    return 0.0;
}

return bonus_if_pup(t-1)
    * life->surv_per_ret(t-1)
    *(1.- death_rate(t))
    * (1. - life->lapse_rate_pup_bal(t))

    + bonus_if(t-1)
    * life->surv_per_ret(t-1)
    *life->pup_rate_bal(t-1)
    *(1.- death_rate(t))
    * (1. - life->lapse_rate_pup_bal(t));

```

#### 6.1.1.4.1.5 bonus\_if

```

if ((life->par_nonpar=="N") || t > life->mat_period_original || t <= life->commence_period_w
||eq(life->paid_up,"G"))
    return 0.0;

if(life->submodel != "TRAD")
    return 0.0;

if (t < 0)
    return 0.0;

if (t == 0){
    if(life->paid_up == "Y")
        return 0.0;
    return life->bonus_inforce;
}

return bonus_if(t-1) * life->surv_per_ret(t-1) * life->surv_per_act_bal_bef_ret(t) + bonus(t);

```

#### 6.1.1.4.1.6 bonus\_if\_pup

```

if ((life->par_nonpar=="N") || t > life->mat_period_original || t <= life->commence_period_w
||eq(life->paid_up,"G"))
    return 0.0;

if(life->submodel != "TRAD")
    return 0.0;

```

```

if (t < 0 )
    return 0.0;

if (t == 0){
    if(life->paid_up == "Y")
        return life->bonus_inforce;
    return 0.0;
}
return bonus_if_pup(t-1)*(1.- death_rate(t))* (1. - life->lapse_rate_pup_bal(t))*life-
>surv_per_ret(t-1) + bonus_pup(t);

```

#### 6.1.1.4.1.7 bonus\_pup

```

if (t <= 0 || (life->par_nonpar=="N") || t > life->mat_period_original || eq(life->paid_up,"G"))
    return 0.0;

if(life->submodel != "TRAD")
    return 0.0;

return bonus_rate_pup(t)
    * ( res_to_bonus_pup(t) + bonus_b_pup(t) )

    + bonus_b_pup(t) * int_rate_res_mthly;

```

#### 6.1.1.4.1.8 bonus\_rate

```

if ((life->par_nonpar=="N") || t > life->mat_period_original || t <= life->commence_period_w
|| eq(life->paid_up,"G") || t <= 0)
    return 0.0;

if(life->submodel != "TRAD")
    return 0.0;

if((res_to_bonus(t) + bonus_b(t)) == 0)
    return 0.0;

return (int_cred(t)
    - mgt_fee_fix(t)
    - mgt_fee_var(t)
    - int_res_deduct(t)
    )
    /
    (res_to_bonus(t) + bonus_b(t));

```

#### 6.1.1.4.1.9 bonus\_rate\_mat

```

if (t > life->maturity_period_w || (life->par_nonpar=="N") || t <= life->mat_period_original
|| eq(life->paid_up,"G") || !eq(life->ben_class,"GIMLA"))
    return 0.0;

if(life->submodel != "TRAD")
    return 0.0;

if ((surr_value(t-1) + surr_value_pup(t-1)) == 0.)
    return 0.0;

double temp = 0.0;

```



```

if (t == life->mat_period_original + 1 && ann_factor_pol != 0 && puv_factor(t-1) != 0){
    temp = (sv_factor(t-1) * annuity_if_b(t-1)/100. * life->surv_per_act_bal_bef_ret(t-1) +
    bonus_if(t-1))
        * life->surv_act_post_ret(t)
        + ((sum_insured_if_b_pup(t-1)
            *(100./ann_factor_pol)
            *sv_factor(t-1)/puv_factor(t-1)
            *(1-death_rate(t-1))
            + bonus_if_pup(t-1)))
        * life->surv_pup_post_ret(t);
}

else
    temp = surr_value(t-1) * life->surv_act_post_ret(t)
        + surr_value_pup(t-1) * life->surv_pup_post_ret(t);

if (temp == 0.)
    return 0.0;

return (int_cred_mat(t)
        - mgt_fee_fix_mat(t)
        - mgt_fee_var_mat(t)
        )
    / temp;

```

#### 6.1.1.4.1.10 bonus\_rate\_mthly

```

if (t < life->commence_period_w || t > life->maturity_period_w)
    return NO_AVG;

if(life->submodel != "TRAD")
    return 0.0;

int proj_yr = xint(life->proj_year(t));
int proj_yr_up = xint(life->proj_year(t+1));
int proj_yr_dn = xint(life->proj_year(t-1));

if(eq(life->projection_type_int, "Rollup")){
    proj_yr = xint(life->proj_year_rollup(t));
    proj_yr_up = xint(life->proj_year_rollup(t+1));
    proj_yr_dn = xint(life->proj_year_rollup(t-1));
}

if (t>0 && (proj_yr ==proj_yr_dn))
    return bonus_rate_mthly(t-1);

// calculate bonus rate
double rate = (1. + life->inv_rate_mth_t[proj_yr]) *(1.-life->mgt_fee_fixed/1200.)-1;
rate = rate * (1.-life->mgt_fee_variable/100.);

return rate - int_rate_res_mthly;

```

#### 6.1.1.4.1.11 bonus\_rate\_pup

```

if ((life->par_nonpar=="N") || t > life->mat_period_original || t <= life->commence_period_w
||eq(life->paid_up,"G") || t <= 0)
    return 0.0;

```

```

if(life->submodel != "TRAD")
    return 0.0;

if (life->paid_up == "N" && t == 1)
    return bonus_rate(t);

if((res_to_bonus_pup(t) + bonus_b_pup(t)) == 0)
    return 0.0;

return (int_cred_pup(t)
        - mgt_fee_fix_pup(t)
        - mgt_fee_var_pup(t)
        - int_res_deduct_pup(t)
        )
    /
    (res_to_bonus_pup(t) + bonus_b_pup(t));

```

#### 6.1.1.4.1.12 bor\_acc

```

if (t < 0 || (life->par_nonpar=="N") || t > life->mat_period_original || eq(life->paid_up,"G") ||
life->mgt_fee_variable == 0)
    return 0.;

if(life->submodel != "TRAD")
    return 0.0;

if (life->paid_up == "Y")
    return 0.0;

if(t==0) {
    if (life->mgt_deficit_perc < 0.)
        return (life->bonus_inforce + life->resinforce)
            * life->mgt_deficit_perc
            * (-1.)
            * life->mgt_fee_variable/100.;
    return 0.0;
}

if (har_acc(t) > 0)
    return 0.0; //No bor if there is har

double bor = 0.0;

bor = bor_acc(t-1)
    * (1. - death_rate(t-1))
    * (1. - life->lapse_total_bal(t-1))
    * (1. - life->prem_termination_prop(t-1));

if (net_interest_rate(t) < 0.0){
    double new_bor = (-1)
        *(int_cred(t) - mgt_fee_fix(t))
        * life->mgt_fee_variable/100.;

    if (har_return(t) > 0)

```

```

        new_bor = max(new_bor - har_return(t), 0);

    bor = bor
        + new_bor; //Addition to bor

}

if (net_interest_rate(t) > 0.0)
    bor = bor + bor_return(t); //Bor returned

return max(bor, 0.0);

```

#### 6.1.1.4.1.13 bor\_acc\_mat

```

if (t < 0 || (life->par_nonpar=="N") || t > life->maturity_period_w || eq(life->paid_up,"G") ||
life->mgt_fee_variable == 0 || t <= life->mat_period_original || !eq(life->ben_class, "gimla"))
    return 0.;

if(life->submodel != "TRAD")
    return 0.0;

if(t==0) {
    if (life->mgt_deficit_perc < 0.)
        return (life->bonus_inforce + life->resinforce)
            * life->mgt_deficit_perc
            * (-1.)
            * life->mgt_fee_variable/100.;

    return 0.0;

}

if (har_acc_mat(t) > 0)
    return 0.0;

double bor = 0.0;

if (t == life->mat_period_original + 1){

    bor = bor_acc(t-1)
        * (1. - death_rate(t-1))
        * (1. - life->lapse_rate_act_bal(t-1))
        * (1. - life->retirement_prop(t-1))

        + bor_acc_pup(t-1)
            * (1. - death_rate(t-1))
            * (1. - life->lapse_total_bal(t-1))
            * (1. - life->prem_termination_prop(t-1));

}
else
    bor = bor_acc_mat(t-1)
        * (1. - death_rate(t-1))
        * (1. - life->retirement_prop(t-1));

```

```

if (net_interest_rate(t) < 0.0){

    double new_bor = (-1)
                                *(int_cred_mat(t) - mgt_fee_fix_mat(t))
                                * life->mgt_fee_variable/100.;

    if (har_return_mat(t) > 0)
        new_bor = max(new_bor - har_return_mat(t), 0);

    bor = bor
        + new_bor; //Addition to bor

}

if (net_interest_rate(t) > 0.0)
    bor = bor + bor_return_mat(t); //Bor returned

return max(bor, 0.0);

```

#### 6.1.1.4.1.14 bor\_acc\_pup

```

if (t < 0 || (life->par_nonpar=="N") || t > life->mat_period_original || eq(life->paid_up,"G") ||
life->mgt_fee_variable == 0)
    return 0.;

if(life->submodel != "TRAD")
    return 0.0;

if(t==0) {
    if (life->mgt_deficit_perc < 0. && life->paid_up == "Y")
        return (life->bonus_inforce + life->resinforce)
                * life->mgt_deficit_perc
                * (-1.)
                * life->mgt_fee_variable/100.;

    return 0.0;
}

if (har_acc_pup(t) > 0)
    return 0.0; //No bor if there is har

double bor = 0.0;

bor = bor_acc_pup(t-1)
    * (1. - death_rate(t-1))
    * (1. - life->lapse_rate_pup_bal(t-1))
    * (1. - life->retirement_prop(t-1));

bor = bor + bor_acc(t-1)
    * life->surv_per_ret(t-1)
    * life->pup_rate_bal_dep(t-1); //Passed from active

```

```

if (net_interest_rate(t) < 0.0){

    double new_bor = (-1)
                                *(int_cred_pup(t) - mgt_fee_fix_pup(t))
                                * life->mgt_fee_variable/100.;

    if (har_return_pup(t) > 0)
        new_bor = max(new_bor - har_return_pup(t), 0);

    bor = bor
        + new_bor; //Addition to bor

}

if (net_interest_rate(t) > 0.0)
    bor = bor + bor_return_pup(t); //Bor returned

return max(bor, 0.0);

```

#### 6.1.1.4.1.15 bor\_return

```

if ((life->par_nonpar=="N") || t > life->mat_period_original || t <= life->commence_period_w
||eq(life->paid_up,"G") || t <= 0)
    return 0.0;

if(life->submodel != "TRAD")
    return 0.0;

if (life->paid_up == "Y")
    return 0.0;

double mgt_fee_pos = 0.0;

double temp = bor_acc(t-1);

if (net_interest_rate(t) < 0.0 || bor_acc(t-1) == 0.0)
    return 0.0;

mgt_fee_pos = -mgt_fee_var_no_bor(t); //Management fees from current month available to repay bor

return max(mgt_fee_pos,
            bor_acc(t-1)
            * (-1.)
            * (1. - death_rate(t-1))
            * (1. - life->lapse_total_bal(t-1))
            * (1. - life->prem_termination_prop(t-1))
            ); //Cannot return more than outstanding bor

```

#### 6.1.1.4.1.16 bor\_return\_mat

```

if (t < 0 || (life->par_nonpar=="N") || t > life->maturity_period_w ||eq(life->paid_up,"G") ||
life->mgt_fee_variable == 0 || t <= life->mat_period_original || !eq(life->ben_class, "gimla"))
    return 0.;

```

```

if(life->submodel != "TRAD")
    return 0.0;

double mgt_fee_pos = 0.0;

double temp = bor_acc_mat(t-1);

if (net_interest_rate(t) < 0.0 || (bor_acc_mat(t-1) == 0.0 && t != life->mat_period_original + 1))
    return 0.0;
    mgt_fee_pos = -mgt_fee_var_no_bor_mat(t);
//mgt_fee_pos = (int_cred_mat(t) - mgt_fee_fix_mat(t))
//              * life->mgt_fee_variable/100.
//              * (-1.); //Management fees from current month available to repay bor

if(t == life->mat_period_original + 1){

if ((bor_acc(t-1) + bor_acc_pup(t-1)) == 0.)
    return 0.0;

return max(mgt_fee_pos,
            (bor_acc(t-1)
             * (1. - death_rate(t-1))
             * (1. - life->lapse_rate_act_bal(t-1))
             * (1. - life->retirement_prop(t-1))

            + bor_acc_pup(t-1)
             * (1. - death_rate(t-1))
             * (1. - life->lapse_total_bal(t-1))
             * (1. - life->prem_termination_prop(t-1))
             ) * (-1.)
            );

}

return max(mgt_fee_pos,
            bor_acc_mat(t-1)
            * (1. - death_rate(t-1))
            * (1. - life->retirement_prop(t-1))
            * (-1.)
            ); //Cannot return more than outstanding bor

```

#### 6.1.1.4.1.17 bor\_return\_pup

```

if ((life->par_nonpar=="N") || t > life->mat_period_original || t <= life->commence_period_w
||eq(life->paid_up,"G") || t <= 0)
    return 0.0;

if(life->submodel != "TRAD")
    return 0.0;

double mgt_fee_pos = 0.0;

double temp = bor_acc_pup(t-1);

if (net_interest_rate(t) < 0.0 || (bor_acc(t-1) + bor_acc_pup(t-1)) == 0.0)
    return 0.0;

```

```
mgt_fee_pos = -mgt_fee_var_no_bor_pup(t); //Management fees from current month available to repay
bor
```

```
double bor_for_ret = bor_acc_pup(t-1)
    * (1. - death_rate(t-1))
    * (1. - life->lapse_rate_pup_bal(t-1))
    * (1. - life->retirement_prop(t-1))
    + bor_acc(t-1)
    * life->surv_per_ret(t-1)
    * life->pup_rate_bal_dep(t-1);

return max(mgt_fee_pos,
    bor_for_ret *(-1.)
); //Cannot return more than outstanding bor
```

#### 6.1.1.4.1.18 claims\_death

```
if(life->submodel != "TRAD")
    return 0.0;

return max(0,death_claims_si(t)+death_claims_bon(t));
```

#### 6.1.1.4.1.19 claims\_maturity

```
if (t <= life->commence_period_w || t > life->maturity_period_ann)
    return NO_AVG;

if(life->submodel != "TRAD")
    return 0.0;

return max(0,(maturities_si(t)+maturities_bon(t) + claims_ret(t)));
```

#### 6.1.1.4.1.20 claims\_surrender

```
if (t <= life->commence_period_w || t > life->mat_period_original || eq(life->paid_up,"G"))
    return 0.0;

if(life->submodel != "TRAD")
    return 0.0;

if (t == life->mat_period_original) {
    if (life->par_nonpar == "P")
        return max(0,max(bonus_if(t) +sum_insured_if_b(t),surr_value(t)) * life-
>lapse_rate_act_bal_dep(t)
    + max(bonus_if_pup(t)+sum_insured_if_b_pup(t),surr_value_pup(t))*
life->lapse_rate_pup_bal_dep(t));
    return max(0,max(sum_insured_if_b(t),surr_value(t)) * life->lapse_rate_act_bal_dep(t)
    +max(sum_insured_if_b_pup(t),surr_value_pup(t)) * life-
>lapse_rate_pup_bal_dep(t));
}

//surr_val is the inforce item after the surrenders have occurred
if (life->surv_per_bal_bef_ret(t)>0.0)
    return max(0,life->lapse_rate_act_bal_dep(t) * surr_value(t)/life->surv_per_bal_bef_ret(t)
    +life->lapse_rate_pup_bal(t) * surr_value_pup(t));

return 0.0;
```

**6.1.1.4.1.21 death\_claims\_bon**

```

if (t <= life->commence_period_w || t > life->maturity_period_w || eq(life->paid_up,"G"))
    return 0.0;

if(life->submodel != "TRAD")
    return 0.0;

if (life->par_nonpar=="N")
    return 0.0;

return death_rate(t)* life->surv_per_ret(t-1) * (bonus_if(t-1)+bonus_if_pup(t-1));

```

**6.1.1.4.1.22 death\_claims\_si**

```

if (t <= life->commence_period_w || t > life->maturity_period_w ||eq(life->paid_up,"G"))
    return 0.0;

if(life->submodel != "TRAD")
    return 0.0;

double sv = surr_value(t) - bonus_if(t)+surr_value_pup(t) - bonus_if_pup(t);

double ytron_perc=0.0;
if(eq(life->ben_class,"Ytron")){
    ytron_perc = life->matan_perc_temp * life->age_at_issue;
    ytron_perc = max(ytron_perc,life->min_ytron_perc);
    ytron_perc = min(ytron_perc,100.);
    return death_rate(t) * max((sum_insured_if_b(t)+ sum_insured_if_b_pup(t))*
ytron_perc/100.,sv);
}

double v_mth_res = 1/(1 + int_rate_res_mthly); //monthly discount //on

reserving basis
if(eq(life->ben_class,"GIMLA")){
    if (eq(life->prod_code,"gml12") && ann_factor_pol != 0) // gimla bet minimum death benefit
        return death_rate(t) * max(sv, 120. * ((sum_insured_if_b(t)+sum_insured_if_b_pup(t))/
ann_factor_pol * 100.));
    else {
        if(t <= life->gimla_db_period_w && ann_factor_pol != 0)
            return death_rate(t) * 75. * (sum_insured_if_b(t)+sum_insured_if_b_pup(t))
                /ann_factor_pol*100.;

        else if(t <= life->mat_period_original)
            return death_rate(t) * max(sum_insured_if_b(t)+sum_insured_if_b_pup(t),sv) *
pow(v_mth_res,life->mat_period_original - t);

        else
            return death_rate(t) * sv;
    }
}

return death_rate(t) * max(sum_insured_if_b(t)+sum_insured_if_b_pup(t), sv);

```

**6.1.1.4.1.23 har\_acc**

```

if ((life->par_nonpar=="N") || t > life->mat_period_original || t <= life->commence_period_w
||eq(life->paid_up,"G") || t < 0)

```



```

    return 0.0;

if(life->submodel != "TRAD")
    return 0.0;

if (life->paid_up == "Y")
    return 0.0;

if(t==0){

    if(life->cal_month(t) == 12. || life->mgt_deficit_perc < 0)
        return 0.0; //If year-end, no accumulation. If there is bor, no accumulation

    return (life->bonus_inforce + life->resinforce)
        * life->mgt_deficit_perc
        * life->mgt_fee_variable/100.;

}

double har = 0;

if(life->cal_month(t) > 1)
    har = har_acc(t-1)
        * (1. - death_rate(t-1))
        * (1. - life->lapse_total_bal(t-1))
        * (1. - life->prem_termination_prop(t-1));

har = har + mgt_fee_var(t); // Works for both positive and negative

return max(har, 0);

```

#### 6.1.1.4.1.24 har\_acc\_mat

```

if (t < 0 || (life->par_nonpar=="N") || t > life->maturity_period_w || eq(life->paid_up,"G") ||
life->mgt_fee_variable == 0 || t <= life->mat_period_original || !eq(life->ben_class, "gimla"))
    return 0.;

if(life->submodel != "TRAD")
    return 0.0;

if(t==0){

    if(life->cal_month(t) == 12. || life->mgt_deficit_perc < 0)
        return 0.0; //If year-end, no accumulation. If there is bor, no accumulation

    return (life->bonus_inforce + life->resinforce)
        * life->mgt_deficit_perc
        * life->mgt_fee_variable/100.;

}

double har = 0;

if(life->cal_month(t) > 1){

    if (t == life->mat_period_original + 1)

        har = har_acc(t-1)

```

```

        * (1. - death_rate(t-1))
        * (1. - life->lapse_rate_act_bal(t-1))
        * (1. - life->retirement_prop(t-1))

        + har_acc_pup(t-1)
        * (1. - death_rate(t-1))
        * (1. - life->lapse_total_bal(t-1))
        * (1. - life->prem_termination_prop(t-1));

    else

        har = har_acc_mat(t-1)
            * (1. - death_rate(t-1))
            * (1. - life->retirement_prop(t-1));

}

har = har + mgt_fee_var_mat(t); // Works for both positive and negative

return max(har, 0);

```

#### 6.1.1.4.1.25 har\_acc\_pup

```

if ((life->par_nonpar=="N") || t > life->mat_period_original || t <= life->commence_period_w
||eq(life->paid_up,"G") || t < 0)
    return 0.0;

if(life->submodel != "TRAD")
    return 0.0;

if(t==0){

    if(life->cal_month(t) == 12. || life->mgt_deficit_perc < 0 || life->paid_up == "N")
        return 0.0; //If year-end, no accumulation. If there is bor, no accumulation

    return (life->bonus_inforce + life->resinforce)
        * life->mgt_deficit_perc
        * life->mgt_fee_variable/100.;

}

double har = 0;

if(life->cal_month(t) > 1)
    har = har_acc_pup(t-1)
        * (1. - death_rate(t-1))
        * (1. - life->lapse_rate_pup_bal(t-1))
        * (1. - life->retirement_prop(t-1))
        + har_acc(t-1)
        * life->surv_per_ret(t-1)
        * life->pup_rate_bal_dep(t-1); //Passed from active;

har = har + mgt_fee_var_pup(t); // Works for both positive and negative

return max(har, 0);

```

**6.1.1.4.1.26 har\_return**

```

if (t <= 0 || (life->par_nonpar=="N") || t > life->mat_period_original || eq(life->paid_up,"G") ||
life->mgt_fee_variable == 0)
    return 0.0;

if(life->submodel != "TRAD")
    return 0.0;

if (life->paid_up == "Y")
    return 0.0;

if (har_acc(t-1) == 0.0)
    return 0; //Nothing collected to return

if (net_interest_rate(t) > 0.0)
    return 0.0; //No need to return if interest is positive

if (life->cal_month(t) == 1)
    return 0.0; //Do not carry over to new year

double har_ret = life->mgt_fee_variable/100.
                * (int_cred(t) - mgt_fee_fix(t))
                * (-1.); // Management fees that should be returned

return min(har_ret,
           har_acc (t-1)
           * (1. - death_rate(t-1))
           * (1. - life->lapse_total_bal(t-1))
           * (1. - life->prem_termination_prop(t-1))
           );//Cannot return more than accumulated har for that year

```

**6.1.1.4.1.27 har\_return\_mat**

```

if (t < 0 || (life->par_nonpar=="N") || t > life->maturity_period_w || eq(life->paid_up,"G") ||
life->mgt_fee_variable == 0 || t <= life->mat_period_original || !eq(life->ben_class, "gimla"))
    return 0.;

if(life->submodel != "TRAD")
    return 0.0;

if (net_interest_rate(t) > 0.0 || (har_acc_mat(t-1) == 0.0 && t != life->mat_period_original + 1))
    return 0.0;

if (life->cal_month(t) == 1)
    return 0.0; //Do not carry over to new year

double har_ret = life->mgt_fee_variable/100.
                * (int_cred_mat(t) - mgt_fee_fix_mat(t))
                * (-1.); // Management fees that should be returned

if(t == life->mat_period_original + 1){

    if ((har_acc(t-1) + har_acc_pup(t-1)) == 0.)
        return 0.0;

    return min(har_ret,
               (har_acc(t-1)

```

```

        * (1. - death_rate(t-1))
        * (1. - life->lapse_rate_act_bal(t-1))
        * (1. - life->retirement_prop(t-1))

    + har_acc_pup(t-1)
        * (1. - death_rate(t-1))
        * (1. - life->lapse_total_bal(t-1))
        * (1. - life->prem_termination_prop(t-1))
    )
);

}

return min(har_ret,
           har_acc_mat(t-1)
           * (1. - death_rate(t-1))
           * (1. - life->retirement_prop(t-1)));

```

#### 6.1.1.4.1.28 har\_return\_pup

```

if (t <= 0 || (life->par_nonpar=="N") || t > life->mat_period_original || eq(life->paid_up,"G") ||
life->mgt_fee_variable == 0)
    return 0.0;

if(life->submodel != "TRAD")
    return 0.0;

if ((har_acc(t-1) + har_acc_pup(t-1)) == 0.0)
    return 0; //Nothing collected to return

if (net_interest_rate(t) > 0.0)
    return 0.0; //No need to return if interest is positive

if (life->cal_month(t) == 1)
    return 0.0; //Do not carry over to new year

double har_ret = life->mgt_fee_variable/100.
    * (int_cred_pup(t) - mgt_fee_fix_pup(t))
    * (-1.); // Management fees that should be returned

double har_for_ret = har_acc_pup(t-1)
    * (1. - death_rate(t-1))
    * (1. - life->lapse_rate_pup_bal(t-1))
    * (1. - life->retirement_prop(t-1))
    + har_acc(t-1)
        * life->surv_per_ret(t-1)
        * life->pup_rate_bal_dep(t-1);

return min(har_ret,
           har_for_ret
           );//Cannot return more than accumulated har for that year

```

#### 6.1.1.4.1.29 int\_cred

```

if (t <= 0 || (life->par_nonpar=="N") || t > life->mat_period_original || eq(life->paid_up,"G"))

    return 0.0;

```

```

if(life->submodel != "TRAD")
    return 0.0;

int proj_yr = xint(life->proj_year(t));
double temp_inv_rate_m = 0.0;

if(eq(life->projection_type_int, "Rollup"))
    proj_yr = xint(life->proj_year_rollup(t));

proj_yr = max(proj_yr, 0);

if (life->margin_add_asset == "Y" && t == 1 && life->par_nonpar == "P")
    temp_inv_rate_m = life->asset_shock;
else
    temp_inv_rate_m = life->inv_rate_mth_t[proj_yr];

return temp_inv_rate_m
    * (bonus_b(t) + res_to_bonus(t));

```

#### 6.1.1.4.1.30 int\_cred\_mat

```

if (t > life->maturity_period_w || (life->par_nonpar=="N") || t <= life->mat_period_original
||eq(life->paid_up,"G")||!eq(life->ben_class,"GIMLA"))
    return 0.0;

if(life->submodel != "TRAD")
    return 0.0;

int proj_yr = xint(life->proj_year(t));
double temp_inv_rate_m = 0.0;

if(eq(life->projection_type_int, "Rollup"))
    proj_yr = xint(life->proj_year_rollup(t));

proj_yr = max(proj_yr, 0);

if (life->margin_add_asset == "Y" && t == 1 && life->par_nonpar == "P")
    temp_inv_rate_m = life->asset_shock;
else
    temp_inv_rate_m = life->inv_rate_mth_t[proj_yr];

double temp = 0.0;

if (t == life->mat_period_original + 1 && ann_factor_pol != 0 && puv_factor(t-1) != 0)
    temp = (sv_factor(t-1) * annuity_if_b(t-1)/100. * life->surv_per_act_bal_bef_ret(t-1) +
    bonus_if(t-1))
        * life->surv_act_post_ret(t)
        + ((sum_insured_if_b_pup(t-1)
            *(100./ann_factor_pol)
            *sv_factor(t-1)/puv_factor(t-1)
            *(1-death_rate(t-1))
            + bonus_if_pup(t-1)))
            * life->surv_pup_post_ret(t);

else
    temp = surr_value(t-1) * life->surv_act_post_ret(t)

```

```
+ surr_value_pup(t-1) * life->surv_pup_post_ret(t);
```

```
return temp * temp_inv_rate_m;
```

#### 6.1.1.4.1.31 int\_cred\_pup

```
if (t <= 0 || (life->par_nonpar=="N") || t > life->mat_period_original || eq(life->paid_up,"G"))
```

```
    return 0.0;
```

```
if(life->submodel != "TRAD")
```

```
    return 0.0;
```

```
int proj_yr = xint(life->proj_year(t));
```

```
double temp_inv_rate_m = 0.0;
```

```
if(eq(life->projection_type_int, "Rollup"))
```

```
    proj_yr = xint(life->proj_year_rollup(t));
```

```
proj_yr = max(proj_yr, 0);
```

```
if (life->margin_add_asset == "Y" && t == 1 && life->par_nonpar == "P")
```

```
    temp_inv_rate_m = life->asset_shock;
```

```
else
```

```
    temp_inv_rate_m = life->inv_rate_mth_t[proj_yr];
```

```
return temp_inv_rate_m
```

```
    * (bonus_b_pup(t) + res_to_bonus_pup(t));
```

#### 6.1.1.4.1.32 int\_post\_mat

```
if (t <= life->mat_period_original || t > life->maturity_period_w)
```

```
    return 0.;
```

```
if(life->submodel != "TRAD")
```

```
    return 0.0;
```

```
double par_rate = 0.;
```

```
double gteed_rate = int_rate_res_mthly;
```

```
if (life->par_nonpar=="N")
```

```
    return gteed_rate;
```

```
if (t == life->mat_period_original + 1)
```

```
    par_rate = bonus_rate_mat(t) + int_rate_res_mthly;
```

```
else
```

```
    par_rate = bonus_rate_mat(t);
```

```
return par_rate * (1- prop_gteedint_post_maturity) + gteed_rate * prop_gteedint_post_maturity;
```

#### 6.1.1.4.1.33 int\_res\_deduct

```
if (t <= 0 || (life->par_nonpar=="N") || t > life->mat_period_original || eq(life->paid_up,"G"))
```

```
    return 0.0;
```

```
if(life->submodel != "TRAD")
    return 0.0;
```

```
return (res_to_bonus(t) + bonus_b(t) )
      * int_rate_res_mthly;
```

#### 6.1.1.4.1.34 int\_res\_deduct\_pup

```
if (t <= 0 || (life->par_nonpar=="N") || t > life->mat_period_original ||eq(life->paid_up,"G"))
    return 0.0;
```

```
if(life->submodel != "TRAD")
    return 0.0;
```

```
return (res_to_bonus_pup(t) + bonus_b_pup(t) )
      * int_rate_res_mthly;
```

#### 6.1.1.4.1.35 maturities\_bon

```
if(life->submodel != "TRAD")
    return 0.0;
```

```
if (t == life->mat_period_original && life->par_nonpar == "P" && !eq(life->paid_up,"G"))
    return bonus_if(t)+ bonus_if_pup(t);
```

```
return 0.0;
```

#### 6.1.1.4.1.36 maturities\_si

```
if (eq(life->paid_up,"G"))
    return 0.0;
```

```
if(life->submodel != "TRAD")
    return 0.0;
```

```
double sv = surr_value(t) - bonus_if(t) + surr_value_pup(t) - bonus_if_pup(t);
```

```
if (t == life->mat_period_original){
    if(eq(life->ben_class,"GIMLA")){
        return max((sum_insured_if_b(t)+ sum_insured_if_b_pup(t)),sv);
    }

    return max((sum_insured_if_b(t)* life->surv_per_act_bal_bef_ret(t)
                + sum_insured_if_b_pup(t)*(1. - death_rate(t)) * (1. - life-
>lapse_rate_pup_bal(t))) ,sv);
}
```

```
return 0.0;
```

#### 6.1.1.4.1.37 mgt\_fee\_fix

```
if (t <= 0 || (life->par_nonpar=="N") || t > life->mat_period_original ||eq(life->paid_up,"G"))
    return 0.0;
```

```
if(life->submodel != "TRAD")
    return 0.0;
```

```

return life->mgt_fee_fixed/1200.
        * (bonus_b(t)
          + res_to_bonus(t)
          + int_cred(t)
          );

```

#### 6.1.1.4.1.38 mgt\_fee\_fix\_mat

```

if (t > life->maturity_period_w || (life->par_nonpar=="N") || t <= life->mat_period_original
||eq(life->paid_up,"G"))
    return 0.0;

if(life->submodel != "TRAD")
    return 0.0;

double temp = 0.0;

if (t == life->mat_period_original + 1 && ann_factor_pol != 0 && puv_factor(t-1) != 0)
    temp = (sv_factor(t-1) * annuity_if_b(t-1)/100. * life->surv_per_act_bal_bef_ret(t-1) +
bonus_if(t-1))
        * life->surv_act_post_ret(t)
    + ((sum_insured_if_b_pup(t-1)
        *(100./ann_factor_pol)
        *sv_factor(t-1)/puv_factor(t-1)
        *(1-death_rate(t-1))
        + bonus_if_pup(t-1)))
        * life->surv_pup_post_ret(t);

else
    temp = surr_value(t-1) * life->surv_act_post_ret(t)
        + surr_value_pup(t-1) * life->surv_pup_post_ret(t);

return life->mgt_fee_fixed/1200.
        * (temp + int_cred_mat(t));

```

#### 6.1.1.4.1.39 mgt\_fee\_fix\_pup

```

if (t <= 0 || (life->par_nonpar=="N") || t > life->mat_period_original ||eq(life->paid_up,"G"))
    return 0.0;

if(life->submodel != "TRAD")
    return 0.0;

return life->mgt_fee_fixed/1200.
        * (bonus_b_pup(t)
          + res_to_bonus_pup(t)
          + int_cred_pup(t)
          );

```

#### 6.1.1.4.1.40 mgt\_fee\_var

```

if (t <= 0 || (life->par_nonpar=="N") || t > life->mat_period_original ||eq(life->paid_up,"G"))
    return 0.0;

if(life->submodel != "TRAD")
    return 0.0;

```



```

double var = mgt_fee_var_no_bor(t);

//if (net_interest_rate(t) > 0.0)
//    var = life->mgt_fee_variable/100.
//    * (int_cred(t) - mgt_fee_fix(t)); // Management fee (assuming no adjustment)

var = var + bor_return(t);

var = max(var, 0);

var = var - har_return(t);

if (abs(var) < SMALL_DOUBLE)
    return 0.0; //Remove small rounding issues

return var;

```

#### 6.1.1.4.1.41 mgt\_fee\_var\_mat

```

if (t > life->maturity_period_w || (life->par_nonpar=="N") || t <= life->mat_period_original
||eq(life->paid_up,"G") || !eq(life->ben_class, "gimla"))
    return 0.0;

if(life->submodel != "TRAD")
    return 0.0;

double var = mgt_fee_var_no_bor_mat(t);

//if (net_interest_rate(t) > 0.0)
//    var = life->mgt_fee_variable/100.
//    * (int_cred_mat(t) - mgt_fee_fix_mat(t)); // Management fee (assuming no
adjustment)

var = var + bor_return_mat(t);

var = max(var, 0);

var = var - har_return_mat(t);

if (abs(var) < SMALL_DOUBLE)
    return 0.0; //Remove small rounding issues

return var;

```

#### 6.1.1.4.1.42 mgt\_fee\_var\_no\_bor

```

if (t <= 0 || (life->par_nonpar=="N") || t > life->mat_period_original ||eq(life->paid_up,"G"))
    return 0.0;

if(life->submodel != "TRAD")
    return 0.0;

if (net_interest_rate(t) > 0.0)
    return life->mgt_fee_variable/100.
        * (int_cred(t) - mgt_fee_fix(t)); // Management fee (assuming no adjustment)

return 0.0;

```

**6.1.1.4.1.43 mgt\_fee\_var\_no\_bor\_mat**

```

if (t > life->maturity_period_w || (life->par_nonpar=="N") || t <= life->mat_period_original
||eq(life->paid_up,"G") || !eq(life->ben_class, "gimla"))
    return 0.0;

if(life->submodel != "TRAD")
    return 0.0;

if (net_interest_rate(t) > 0.0)
    return life->mgt_fee_variable/100.
        * (int_cred_mat(t) - mgt_fee_fix_mat(t)); // Management fee (assuming no
adjustment)

return 0.0;

```

**6.1.1.4.1.44 mgt\_fee\_var\_no\_bor\_pup**

```

if (t <= 0 || (life->par_nonpar=="N") || t > life->mat_period_original ||eq(life->paid_up,"G"))
    return 0.0;

if(life->submodel != "TRAD")
    return 0.0;

if (net_interest_rate(t) > 0.0)
    return life->mgt_fee_variable/100.
        * (int_cred_pup(t) - mgt_fee_fix_pup(t)); // Management fee (assuming no
adjustment)

return 0.0;

```

**6.1.1.4.1.45 mgt\_fee\_var\_pup**

```

if (t <= 0 || (life->par_nonpar=="N") || t > life->mat_period_original ||eq(life->paid_up,"G"))
    return 0.0;

if(life->submodel != "TRAD")
    return 0.0;

double var = mgt_fee_var_no_bor_pup(t);

//if (net_interest_rate(t) > 0.0)
//    var = life->mgt_fee_variable/100.
//        * (int_cred_pup(t) - mgt_fee_fix_pup(t)); // Management fee (assuming no
adjustment)

var = var + bor_return_pup(t);

var = max(var, 0);

var = var - har_return_pup(t);

if (abs(var) < SMALL_DOUBLE)
    return 0.0; //Remove small rounding issues

return var;

```

**6.1.1.4.1.46 net\_interest\_rate**

```

if (t <= 0 || (life->par_nonpar=="N") || t > life->maturity_period_w || eq(life->paid_up,"G"))
    return 0.0;

if(life->submodel != "TRAD")
    return 0.0;

double temp_inv_rate_m = 0.0;

int proj_yr = xint(life->proj_year(t));
if(eq(life->projection_type_int, "Rollup"))
    proj_yr = xint(life->proj_year_rollup(t));
proj_yr = max(proj_yr, 0);

if (life->margin_add_asset == "Y" && t == 1 && life->par_nonpar == "P")
    temp_inv_rate_m = life->asset_shock;
else
    temp_inv_rate_m = life->inv_rate_mth_t[proj_yr];

return (1+temp_inv_rate_m)
        * (1- life->mgt_fee_fixed/1200.)
        -1.;

```

**6.1.1.4.1.47 puv\_factor**

```

if(life->submodel != "TRAD" || eq(life->prod_code_old,"0"))
    return 0.0;

if (eq(life->sur_val_method, "sv_table")){
    int t_lim = t;
    if(t > life->mat_period_original)
        t_lim = life->mat_period_original;

    if (xint(life->pol_year(t_lim)) > 0){

        life->puv_row_key = xstring(xint(life->age_at_issue))+ "_" + xstring(xint(life->benefit_term/12));
        life->puv_col_key = xstring(life->pol_year(t_lim));
        double temp = life->puv_09_tbl;

        life->puv_row_key = xstring(xint(life->age_at_issue))+ "_" + xstring(xint(life->benefit_term/12));
        life->puv_col_key = xstring(life->pol_year(t_lim) - 1);
        double tempdn = life->puv_09_tbl;

        if (xint(life->pol_month(t_lim)) != 12) // interpolation
            return temp * life->pol_month(t_lim)/12.
                    + tempdn
                    * (1. - life->pol_month(t_lim)/12.);

        return temp;
    }
}

return 0.0; //Unconditional return

```

**6.1.1.4.1.48 res\_to\_bonus**

```

if (t <= 0 || (life->par_nonpar=="N") || t > life->mat_period_original || eq(life->paid_up,"G"))
    return 0.0;

if(life->submodel != "TRAD")
    return 0.0;

if (life->surv_bal_bef_ret(t) > 0.)
    return (reserve_basic_prem_if(t-1)
            - zillmer_book(t-1)*
            life->surv_act_bal_bef_ret(t)/life->surv_bal_bef_ret(t)
            )
            * life->surv_per_act_bal_bef_ret(t);

return 0.0;

```

**6.1.1.4.1.49 res\_to\_bonus\_pup**

```

if (t <= 0 || (life->par_nonpar=="N") || t > life->mat_period_original || eq(life->paid_up,"G"))
    return 0.0;

if(life->submodel != "TRAD")
    return 0.0;

if (life->surv_bal_bef_ret(t) > 0.)
    return (reserve_basic_pup(t-1)
            - zillmer_book(t-1)
            *life->surv_pup_bal_bef_ret(t)/life->surv_bal_bef_ret(t)
            )
            *(1.- death_rate(t))
            * (1. - life->lapse_rate_pup_bal(t));

return 0.0;

```

**6.1.1.4.1.50 surr\_value**

```

if (t <= life->commence_period_w || t > life->maturity_period_w || eq(life->paid_up,"G"))
    return 0.0;

if (life->submodel != "TRAD")
    return 0;

if (life->paid_up=="Y") // paid-up cover in force uses secondary sum-life->insured
    return 0.0;

int t_lim = life->mat_period_original;

if (eq(life->sur_val_method, "sv_table")) {

    if(eq(life->ben_class,"GIMLA")){
        if (t == t_lim)
            return sv_factor(t) * annuity_if_b(t)/100. * life-
>surv_per_act_bal_bef_ret(t) + bonus_if(t);

        else if (t < t_lim){

```

```

        if(life->lapse_force_rate == 1. && life->lapse_force_month == t)
            return sv_factor(t) * annuity_if_b_bef_ret(t) / 100. + bonus_if(t);

        return sv_factor(t) * annuity_if_b_bef_ret(t+1)/100. + bonus_if(t); /*** may
not work with new business layering, may need annuity_if_e(t)

    }
    else{
        if (t == t_lim + 1)
            return ((sv_factor(t_lim) * annuity_if_b(t_lim)/100. * life-
>surv_per_act_bal_bef_ret(t_lim) + bonus_if(t_lim)) * (1 + int_post_mat(t))) * life-
>surv_act_post_ret(t);
            return surr_value(t-1) * (1 + int_post_mat(t)) * life->surv_act_post_ret(t);
        }

    }

    if (t == t_lim)
        return sv_factor(t) * life->sum_ins_curr/1000. * life->benefits_curr* life-
>surv_act_bal_bef_ret(t) + bonus_if(t);

    else if(t < t_lim){
        if(life->lapse_force_rate == 1. && life->lapse_force_month == t)
            return sv_factor(t-1) * life->sum_ins_curr/1000. * life->benefits_curr +
bonus_if(t-1);
        return sv_factor(t) * life->sum_ins_curr/1000. *life->benefits_curr * life-
>surv_act_bal_bef_ret(t) + bonus_if(t);
    }
    else {
        if(t == t_lim + 1)
            return ((sv_factor(t_lim) * life->sum_ins_curr/1000. * life->benefits_curr*
life->surv_act_bal_bef_ret(t_lim) + bonus_if(t_lim)) * (1 + int_post_mat(t))) * life-
>surv_act_post_ret(t);
            return surr_value(t-1) * (1 + int_post_mat(t)) * life->surv_act_post_ret(t);
        }
    }

}

if (t == t_lim)
    return bonus_if(t) + sum_insured_if_b(t);

if (t > t_lim){
    if (t == t_lim + 1)
        return ((bonus_if(t_lim) + sum_insured_if_b(t_lim)) * (1 + int_post_mat(t))) * life-
>surv_act_post_ret(t);
    return surr_value(t-1) * (1 + int_post_mat(t)) * life->surv_act_post_ret(t);
}

if (eq(life->sur_val_method, "perc_res"))
    return life->sur_val_perc[xint(life->pol_year(t))]/100. * reserve_basic(t) + bonus_if(t);

return 0.0;

6.1.1.4.1.51      surr_value_pup

if (t < 0 || t > life->maturity_period_w ||eq(life->paid_up,"G"))
    return 0.0;

//if row doesnt exist in sv tbl then use the sv from data
if (eq(life->sur_val_method, "sv_table") && sv_tbl_check == 99999

```

```

&& (!eq(life->ben_class,"GIMLA") || (life->sm_annuity[life->sm_annuity.size()-1]->takeup_age >=
(life->age_at_issue + life->elapsed_months/12))))){
    if (t==0)
        return life->surr_value_if * life->benefits_curr;
    return surr_value_pup(t-1) * (1 + int_post_mat(t)) * life->surv_pup_post_ret(t);
}

int t_lim = life->mat_period_original;

if (life->submodel != "TRAD" || puv_factor(t) == 0 || puv_factor(t_lim) == 0)
    return 0;

if (eq(life->sur_val_method, "sv_table")) {

    double ann_fac_pol=1.0;
    if(eq(life->ben_class,"GIMLA") && ann_factor_pol != 0)
        ann_fac_pol = 100./ann_factor_pol;

    if (t <= life->mat_period_original)
        return sum_insured_if_b_pup(t)*ann_fac_pol
            *sv_factor(t)/puv_factor(t)
            *(1-death_rate(t))
            + bonus_if_pup(t);

    if (t == life->mat_period_original + 1)
        return ((sum_insured_if_b_pup(t_lim)*ann_fac_pol
            *sv_factor(t_lim)/puv_factor(t_lim)
            *(1-death_rate(t_lim))
            + bonus_if_pup(t_lim)) * (1 + int_post_mat(t))) * life-
>surv_pup_post_ret(t);

    return surr_value_pup(t-1) * (1 + int_post_mat(t)) * life->surv_pup_post_ret(t);
}

if (t == life->mat_period_original)
    return bonus_if_pup(t) + sum_insured_if_b_pup(t);

if (t > life->mat_period_original){
    if (t == life->mat_period_original + 1)
        return ((bonus_if_pup(t_lim) + sum_insured_if_b_pup(t_lim)) * (1 + int_post_mat(t)))
* life->surv_pup_post_ret(t);

    return surr_value_pup(t-1) * (1 + int_post_mat(t)) * life->surv_pup_post_ret(t);
}

if (eq(life->sur_val_method, "perc_res"))
    return life->sur_val_perc[xint(life->pol_year(t))]/100. * reserve_basic_pup(t) +
bonus_if_pup(t);

return 0.0;

```

#### 6.1.1.4.1.52 sv\_factor

```

if (life->submodel != "TRAD" || eq(life->prod_code_old,"0"))
    return 0;

sv_tbl_check = 0;

```

```

if (eq(life->sur_val_method, "sv_table")){
    int t_lim = t;

    if(t > life->mat_period_original){
        t_lim = life->mat_period_original;
    }

    if (xint(life->pol_year(t_lim)) > 0){

        //check row exists in the sv table
        life->sv_row_key = xstring(xint(life->age_at_issue))+ "_" + xstring(xint(life-
>benefit_term/12));
        life->sv_col_key = xstring(life->pol_year(t_lim));
        sv_tbl_check = life->sv_09_tbl_check;

        life->sv_row_key = xstring(xint(life->age_at_issue))+ "_" + xstring(xint(life-
>benefit_term/12));
        life->sv_col_key = xstring(life->pol_year(t_lim));
        double temp = life->sv_09_tbl;

        life->sv_row_key = xstring(xint(life->age_at_issue))+ "_" + xstring(xint(life-
>benefit_term/12));
        life->sv_col_key = xstring(life->pol_year(t_lim)-1);
        double tempdn = life->sv_09_tbl;

        if (xint(life->pol_month(t_lim)) != 12) // interpolation
            return (temp * life->pol_month(t_lim)/12.
                    + tempdn
                    * (1. - life->pol_month(t_lim)/12.)) ;

        return temp;
    }
}

return 0.0;

```

#### 6.1.1.4.1.53 zillmer\_book

```

// this is used for bonus as well, therefore must be calculated even for DAC policies
if (life->submodel != "TRAD")
    return 0;

return life->zillmer_si_book/100.*sum_at_risk_if(t);

```

#### 6.1.1.4.1.54 zillmer\_tax

```

if (life->submodel != "TRAD")
    return 0;

if (eq(life->dac_type_temp,"zillmer"))
    return life->zillmer_si_tax/100. * sum_at_risk_if(t);

return 0;

```

#### 6.1.1.4.1.55 ann\_takeup\_rate

```

if ((t < 0) || (t >= life->maturity_period_w)) return 0.0; // *** recreate

if(life->submodel != "TRAD")
    return 0.0;

```

```

if(life->annuitization_rate<=0.00001 || !(eq(life->ben_class,"GIMLA"))) // *** need way to
distinguish between policies with and without guarantees, and with and without kitzva option
    return 0.0;

if(life-> res_prop_kitzba<=0.0)
    return 0.0;

//Split of current savings balance into prat, oldtag, newtag and piz

double tagold_money = reserve_basic(0) * life->res_prop_kitzba_oldtag * life->surv_bal(t)*pow(1. +
life->int_rate_res/100., t/12.); // old money in-force at time t
double new_money = max(reserve_basic(t) * life->res_prop_kitzba- tagold_money, 0.0); // new money
in-force at time t
double prop_new_money = 0.0;
if (life->res_prop_kitzba_prat + life->res_prop_kitzba_piz +life->res_prop_kitzba_newtag>0.0)
    prop_new_money = life->res_prop_kitzba_prat + life->res_prop_kitzba_piz +life-
>res_prop_kitzba_newtag;

double prat_money = 0.0;
double piz_money = 0.0;
double tagnew_money = 0.0;

if (prop_new_money>0.0) {
    prat_money = life->res_prop_kitzba_prat / prop_new_money * new_money;
    piz_money = life->res_prop_kitzba_piz / prop_new_money * new_money;
    tagnew_money = life->res_prop_kitzba_newtag / prop_new_money * new_money;
}

// reset annuitisation rate
double ann_rate = 0.0;
if (reserve_basic(t)>0.0){
    ann_rate = (prat_money * life->annuity_takeup_prat/100.
                + piz_money * life->annuity_takeup_piz/100.
                + tagold_money * life->annuity_takeup_old/100.
                + tagnew_money * life->annuity_takeup_new_tag/100. ) / reserve_basic(t);

    if (life->margin_add=="Y") {
        ann_rate = (prat_money * min(life->annuity_takeup_prat/100.*(1 + life-
>margin_annuity_takeup/100.), life->annuity_takeup_max)
                    + piz_money * min(life->annuity_takeup_piz/100.*(1 + life-
>margin_annuity_takeup/100.), life->annuity_takeup_max)
                    + tagold_money * min(life->annuity_takeup_old/100.*(1 + life-
>margin_annuity_takeup/100.), life->annuity_takeup_max)
                    + tagnew_money * min(life->annuity_takeup_new_tag/100.*(1 + life-
>margin_annuity_takeup/100.), life->annuity_takeup_max) ) / reserve_basic(t);
    }
    return ann_rate;
}
return 0.0;

```

#### 6.1.1.4.1.56 reserve

```

if (t <= life->commence_period_w || t >= life->maturity_period_w ||eq(life->paid_up,"G"))
    return 0.0;

if(life->submodel != "TRAD")
    return 0.0;

```



```

return max(0,max((surr_value(t)+ surr_value_pup(t)) * life->surv_per_ret(t),
                reserve_basic(t) + (bonus_if(t) +bonus_if_pup(t))* life->surv_per_ret(t) +
reserve_extra(t))
        + life->res_ann_deficiency(t));

```

#### 6.1.1.4.1.57 reserve\_extra

```

if(life->submodel != "TRAD" || eq(life->prod_code_old,"0"))
    return 0.0;

double sv_temp=0.0, prop_spread=0.0;

if (inlist(life->ben_class,"END,WOL")) {

    if (!eq(life->sur_val_method, "sv_table"))
        return 0.0;
    life->sv_row_key = xstring(xint(life->age_at_issue))+ "_" +xstring(xint(life-
>benefit_term/12));
    life->sv_col_key = xstring(xint(life->benefit_term/12.));
    sv_temp =life->sv_09_tbl;

    prop_spread = min(min(life->benefit_term , 12 * 25) , life->pol_year(t)*12. - 12. + life-
>pol_month(t))
                    /min(life->benefit_term , 12 * 25); // *** 25 years until full
bonus obtained not parameterized

    return max((sv_temp/1000. - 1.) * prop_spread * reserve_basic(t),0);
}

return 0.0;

```

#### 6.1.1.4.1.58 ann\_factor\_weighted

```

if (t < life->commence_period_w || t > life->mat_period_original)
    return NO_AVG;

if(life->submodel != "TRAD")
    return 0.0;

if(xint(life->pol_month(t))==12)
    return annuity_factor(t);
else {

    int t_down = (xint(life->pol_year(t))-1)*12 - life->elapsed_months;
    double interpol = life->pol_month(t)/12.;

    if (t_down + 12 >= life->mat_period_original)
        return annuity_factor(t_down) * (1. - interpol); //Up part is 0

    else
        return annuity_factor(t_down) * (1. - interpol) + annuity_factor(t_down+12) *
interpol;
}

```

#### 6.1.1.4.1.59 ann\_factor\_weighted\_int0

```

if (t < life->commence_period_w || t > life->mat_period_original)
    return NO_AVG;

```

```

if(life->submodel != "TRAD")
    return 0.0;

if(xint(life->pol_month(t))==12)
    return annuity_factor_int0(t);
else {

    int t_down = (xint(life->pol_year(t))-1)*12 - life->elapsed_months;
    double interpol = life->pol_month(t)/12.;

    if (t_down + 12 >= life->mat_period_original)
        return annuity_factor_int0(t_down) * (1. - interpol); //Up part is 0

    else
        return annuity_factor_int0(t_down) * (1. - interpol) + annuity_factor_int0(t_down+12)
* interpol;
}

```

#### 6.1.1.4.1.60 annuity\_factor

```

if (t < life->commence_period_w || t >= life->prem_term - life->elapsed_months || eq(life->paid_up
,"Y"))
    return NO_AVG;

if(life->submodel != "TRAD")
    return 0.0;

int age_now = xint(life->age_last(t+1));
int age_end = xint(life->age_at_issue + ceil(life->prem_term/12. * life->prem_freq)/life-
>prem_freq);

if (inlist(life->ben_class,"END,WOL,YTRON,GIMLA"))
    return (res_nx(age_now,0) - res_nx(age_end,0)) / res_dx(age_now,0);

return 0;

```

#### 6.1.1.4.1.61 annuity\_factor\_int0

```

if (t < life->commence_period_w || t >= life->mat_period_original || eq(life->paid_up ,"Y") ||
!eq(life->ben_class, "gimla"))
    return NO_AVG;

if(life->submodel != "TRAD")
    return 0.0;

int age_now = xint(life->age_last(t+1));
int age_end = xint(life->age_at_issue + ceil(life->prem_term/12. * life->prem_freq)/life-
>prem_freq);

return (res_nx(age_now,1) - res_nx(age_end,1)) / res_dx(age_now,1);

```

#### 6.1.1.4.1.62 ass\_factor\_weighted

```

if (t < life->commence_period_w || t > life->mat_period_original)
    return NO_AVG;

```

```

if(life->submodel != "TRAD")
    return 0.0;

if(xint(life->pol_month(t))==12)
    return assurance_factor(t);
else {

    int t_down = (xint(life->pol_year(t))-1)*12 - life->elapsed_months;
    double interpol = life->pol_month(t)/12.;

    if (t_down + 12 >= life->mat_period_original)
        return assurance_factor(t_down) * (1. - interpol) + 1. * interpol;

    else
        return assurance_factor(t_down) * (1. - interpol) + assurance_factor(t_down+12) *
interpol;
}

```

#### 6.1.1.4.1.63 **ass\_factor\_weighted\_int0**

```

if (t < life->commence_period_w || t > life->mat_period_original)
    return NO_AVG;

if(life->submodel != "TRAD")
    return 0.0;

if(xint(life->pol_month(t))==12)
    return assurance_factor_int0(t);
else {

    int t_down = (xint(life->pol_year(t))-1)*12 - life->elapsed_months;
    double interpol = life->pol_month(t)/12.;

    if (t_down + 12 >= life->mat_period_original)
        return assurance_factor_int0(t_down) * (1. - interpol) + 1. * interpol;

    else
        return assurance_factor_int0(t_down) * (1. - interpol) +
assurance_factor_int0(t_down+12) * interpol;
}

```

#### 6.1.1.4.1.64 **assurance\_factor**

```

if (t < life->commence_period_w || t > life->mat_period_original)
    return NO_AVG;

if(life->submodel != "TRAD")
    return 0.0;

int age_now = xint(life->age_last(t+1));
int age_end = xint(life->age_at_issue + life->benefit_term/12.);
// endowment formula
// WOL treated as an endowment because the term is only to age 95, and we assume benefit is paid
then (even if alive)

if (inlist(life->ben_class,"END,WOL"))
    return ( (res_mx(age_now,0) - res_mx(age_end,0))
            * int_rate_res_hy
            + res_dx(age_end,0))

```

```

        / res_dx(age_now,0);

if (eq(life->ben_class,"YTRON"))    {
    double ytron_perc = min(max(life->matan_perc_temp * life->age_at_issue , life-
>min_ytron_perc) , 100.);
    return ( ytron_perc/100. * (res_mx(age_now,0) - res_mx(age_end,0))
            * int_rate_res_hy
            + res_dx(age_end,0))
            / res_dx(age_now,0);
}

// GIMLA death benefit assumed to be 75 or 120 times monthly annuity amount
// ignores case where surrender value exceeds above death benefit - at higher policy durations
if (eq(life->ben_class,"GIMLA"))    {
    double gimla_perc=0.0;

    if(life->lapse_force_rate == 1. && life->paid_up == "N" && t == life->maturity_period_w)
        return 1.;
    // to avoid division by 0 from sum_insured_if_b(life->commence_period_w)
    // write annuity_if_b in terms of sum_insured_if_b which will cancel sum_insured_if_b
    life->gimla_row_key = xint(life->age_at_issue+life->benefit_term/12);
    life->gimla_col_key = life->sex+"_"+life->fund_name_temp;

    gimla_perc = 100. / life->gimla_table;

    if (eq(life->prod_code,"gml12"))
        gimla_perc = 120. * gimla_perc;
    else
        gimla_perc = 75. * gimla_perc;

    gimla_perc = min(gimla_perc, 1.);
    gimla_perc = min(gimla_perc*1.7, 1.);

    if (life->dump_vars == "Y"){

        log_strm<<"T: "<<t<<endl;
        log_strm<<"Gimla perc: "<<gimla_perc<<endl;
        log_strm<<"Half-year int rate: "<<int_rate_res_hy<<endl;
        log_strm<<"Age end: "<<age_end<<endl;
        log_strm<<"Age now: "<<age_now<<endl;

    }

    return ( gimla_perc * (res_mx(age_now,0) - res_mx(age_end,0))
            * int_rate_res_hy
            + res_dx(age_end,0))
            / res_dx(age_now,0);
}

return 0.0; //Unconditional return

```

#### 6.1.1.4.1.65 assurance\_factor\_int0

```

if (t < life->commence_period_w || t > life->mat_period_original || !eq(life->ben_class,"GIMLA"))
    return NO_AVG;

if(life->submodel != "TRAD")
    return 0.0;

```

```

int age_now = xint(life->age_last(t+1));
int age_end = xint(life->age_at_issue + life->benefit_term/12.);

//Only relevant to gimla

// GIMLA death benefit assumed to be 75 or 120 times monthly annuity amount
// ignores case where surrender value exceeds above death benefit - at higher policy durations

double gimla_perc=0.0;

if(life->lapse_force_rate == 1. && life->paid_up == "N" && t == life->maturity_period_w)
    return 1.;
// to avoid division by 0 from sum_insured_if_b(life->commence_period_w)
// write annuity_if_b in terms of sum_insured_if_b which will cancel sum_insured_if_b
life->gimla_row_key = xint(life->age_at_issue+life->benefit_term/12.);
life->gimla_col_key = life->sex+"_"+life->fund_name_temp;

gimla_perc = 100. / life->gimla_table;

    if (eq(life->prod_code,"gml12"))
        gimla_perc = 120. * gimla_perc;
    else
        gimla_perc = 75. * gimla_perc;

gimla_perc = min(gimla_perc, 1.);
gimla_perc = min(gimla_perc*1.7, 1.);

if (life->dump_vars == "Y"){

    log_strm<<"T: "<<t<<endl;
    log_strm<<"Gimla perc: "<<gimla_perc<<endl;
    log_strm<<"Half-year int rate: "<<int_rate_res_hy<<endl;
    log_strm<<"Age end: "<<age_end<<endl;
    log_strm<<"Age now: "<<age_now<<endl;

}

return res_dx(age_end,1)
        / res_dx(age_now,1); //Only chance will survive to maturity

```

#### 6.1.1.4.1.66 net\_premium\_b

```

if(t <= life->commence_period_w || t + life->elapsed_months > life->prem_term)
    return 0.0;

if(life->submodel != "TRAD")
    return 0.0;

if (life->paid_up=="Y")
    return 0.0;

if (t == life->commence_period_w + 1) {

```

```

double net_prem = 0.0;
double temp = 0.0;
temp = sum_insured(t) * assurance_factor(t-1)
      / annuity_factor(t-1);
net_prem = min(life->prem_curr * life->netprem_max / 100. * life->benefits_curr, temp);
return net_prem;
}
return net_premium_b(life->commence_period_w + 1);

```

#### 6.1.1.4.1.67 net\_premium\_e

```

if (t < life->commence_period_w || t >= life->maturity_period_w || (life->paid_up=="Y"))
    return 0.0;

if(life->submodel != "TRAD")
    return 0.0;

return net_premium_b(t+1);

```

#### 6.1.1.4.1.68 res\_basic\_act\_newtag

```

if (t < life->commence_period_w || t >= life->maturity_period_w || life->surv_ret(t) == 0)
    return 0.0;

if(life->submodel != "TRAD")
    return 0.0;

if (t == 0) {
    if (life->paid_up == "N")
        return reserve_basic_prem_if(t) * life->res_prop_kitzba_newtag;
    else
        return 0.0;
}

int t_down = (xint(life->pol_year(t))-1)*12 - life->elapsed_months;

double net_prem = 0;

if (!eq(life->policy_type, "private"))
    net_prem = net_premium_e(t_down) * life->prem_newtag_prop / 100.;

if(life->dump_vars == "Y" && t==1)
{
log_strm<<"Net prem: "<<net_premium_e(t_down)<<endl;
log_strm<<"Net prem for tag: "<<net_prem<<endl;
log_strm<<"SI newtag: "<<sum_insured_newtag<<endl;
log_strm<<"Ax: "<<ass_factor_weighted(t) <<endl;
log_strm<<"ax: "<<ann_factor_weighted(t)<<endl;
log_strm<<"Surv prem<<: "<<life->surv_act_prm(t)<<endl;
}

if (t < life->mat_period_original)
    return (sum_insured_newtag * ass_factor_weighted(t)
            - net_prem * ann_factor_weighted(t))
            * life->surv_act_bal(t)
            + (res_basic_act_piz_int(t) - res_basic_act_piz(t)) // Interest value of piz
            ;

```

```

if (t == life->mat_period_original)
    return surr_value(t)
        * life->surv_per_ret(t)
        * res_prop_mat_newtag;

```

```

//t > mat_orig
return (
    res_basic_act_newtag(t-1)
    * (1 + int_post_mat(t))
    +
    res_basic_act_piz(t-1)
    * int_post_mat(t)
    )
    * (1. - life->death_rate(t))
    * life->surv_per_ret(t);

```

#### 6.1.1.4.1.69 res\_basic\_act\_old

```

if (t < life->commence_period_w || t >= life->maturity_period_w || life->res_prop_kitzba_oldtag <=
0 || life->surv_ret(t) == 0)
    return 0.0;

```

```

if(life->submodel != "TRAD")
    return 0.0;

```

```

if (t == 0){
    if (life->paid_up == "N")
        return reserve_basic_prem_if(t) * life->res_prop_kitzba_oldtag;
    else
        return 0.0;
}

```

```

if (t < life->mat_period_original)
    return sum_insured_oldtag * ass_factor_weighted(t) * life->surv_act_bal(t);

```

```

return surr_value(t)
    * life->surv_per_ret(t)
    * res_prop_mat_oldtag;

```

#### 6.1.1.4.1.70 res\_basic\_act\_piz

```

if (t < life->commence_period_w || t >= life->maturity_period_w || life->surv_ret(t) == 0)
    return 0.0;

```

```

if(life->submodel != "TRAD")
    return 0.0;

```

```

if (t == 0){
    if (life->paid_up == "N")
        return reserve_basic_prem_if(t) * life->res_prop_kitzba_piz;
    else
        return 0.0;
}

```

```

int t_down = (xint(life->pol_year(t))-1)*12 - life->elapsed_months;
double net_prem = 0.0;

```

```

if (!eq(life->policy_type, "private") )
    net_prem = net_premium_e(t_down) * (1. - life->prem_newtag_prop / 100.);

if (t < life->mat_period_original)
    return (sum_insured_piz_int0 * ass_factor_weighted_int0(t)
            - net_prem * ann_factor_weighted_int0(t))
            * life->surv_act_bal(t);

if (t == life->mat_period_original)
    return surr_value(t)
            * life->surv_per_ret(t)
            * res_prop_mat_piz;

//t > mat_original
return res_basic_act_piz(t-1)
        * (1. - life->death_rate(t))
        * life->surv_per_ret(t); //Maturity surrender value decremented

```

#### 6.1.1.4.1.71 res\_basic\_act\_piz\_int

```

if (t < life->commence_period_w || t >= life->maturity_period_w || life->surv_ret(t) == 0)
    return 0.0;

if(life->submodel != "TRAD")
    return 0.0;

if (t == 0){
    if (life->paid_up == "N")
        return reserve_basic_prem_if(t) * life->res_prop_kitzba_piz;
    else
        return 0.0;
}

int t_down = (xint(life->pol_year(t))-1)*12 - life->elapsed_months;
double net_prem = 0.0;

if (!eq(life->policy_type, "private") )
    net_prem = net_premium_e(t_down) * (1. - life->prem_newtag_prop / 100.);

if (t < life->mat_period_original)
    return (sum_insured_piz * ass_factor_weighted(t)
            - net_prem * ann_factor_weighted(t))
            * life->surv_act_bal(t);

return surr_value(t)
        * life->surv_per_ret(t)
        * res_prop_mat_piz;

```

#### 6.1.1.4.1.72 res\_basic\_act\_prat

```

if (t < life->commence_period_w || t >= life->maturity_period_w || life->surv_ret(t) == 0)
    return 0.0;

if(life->submodel != "TRAD")
    return 0.0;

if (t == 0){

```



```

    if (life->paid_up == "N")
        return reserve_basic_prem_if(t) * life->res_prop_kitzba_prat;
    else
        return 0.0;
}

```

```

int t_down = (xint(life->pol_year(t))-1)*12 - life->elapsed_months;
double net_prem = 0.0;

```

```

if (eq(life->policy_type, "private"))
    net_prem = net_premium_e(t_down);

```

```

if (t < life->mat_period_original)
    return (sum_insured_prat * ass_factor_weighted(t)
            - net_prem * ann_factor_weighted(t))
            * life->surv_act_bal(t);

```

```

return surr_value(t)
    * life->surv_per_ret(t)
    * res_prop_mat_prat;

```

#### 6.1.1.4.1.73 res\_basic\_pup\_newtag

```

if (t < life->commence_period_w || t >= life->maturity_period_w || life->surv_ret(t) == 0)
    return 0.0;

```

```

if(life->submodel != "TRAD")
    return 0.0;

```

```

if (t == 0){
    if (life->paid_up == "Y")
        return reserve_basic_pup(t) * life->res_prop_kitzba_newtag;
    else
        return 0.0;
}

```

```

if (sum_insured(0) == 0)
    return 0;

```

```

if (t < life->mat_period_original)
    return sum_insured_if_b_pup(t) * ass_factor_weighted(t)
           * sum_insured_newtag / sum_insured(0)
           + (res_basic_pup_piz_int(t) - res_basic_pup_piz(t)); // Add interest part of
piz

```

```

if (t == life->mat_period_original)
    return surr_value_pup(t)
           * life->surv_per_ret(t)
           * res_prop_mat_newtag;

```

```

//t > mat_orig

```

```

return (
    res_basic_pup_newtag(t-1)
    * (1 + int_post_mat(t))
    +
    res_basic_pup_piz(t-1)
    * int_post_mat(t)
)

```

```

* (1. - life->death_rate(t))
* life->surv_per_ret(t);

```

#### 6.1.1.4.1.74 res\_basic\_pup\_old

```

if (t < life->commence_period_w || t >= life->maturity_period_w || life->res_prop_kitzba_oldtag <=
0 || life->surv_ret(t) == 0)
    return 0.0;

if(life->submodel != "TRAD")
    return 0.0;

if (t == 0){
    if (life->paid_up == "Y")
        return reserve_basic_pup(t) * life->res_prop_kitzba_oldtag;
    else
        return 0.0;
}

if(t== 93  && life->dump_vars == "Y")
{
    log_strm<<"Maturity: "<<life->maturity_period_w<<endl;
    log_strm<<"Maturity original: "<<life->mat_period_original<<endl;
    log_strm<<"Res active before mat: "<<reserve_basic_prem_if(life->mat_period_original-
1)<<endl;
    log_strm<<"Res pup before mat: "<<reserve_basic_pup(life->mat_period_original-1)<<endl;

    log_strm<<"Res prop: "<<res_prop_mat_oldtag<<endl;
    log_strm<<"Surv: "<< life->surv_per_ret(t);
    log_strm<<"Surr: "<<surr_value_pup(t)<<endl;
    log_strm<<"SA pup: "<<sum_insured_if_b_pup(t)<<endl;
    log_strm<<"Ass factor: "<<ass_factor_weighted(t)<<endl;
    log_strm<<"SA old: "<<sum_insured_oldtag<<endl;
    log_strm<<"SA orig: "<<life->sum_ins_curr<<endl;
    log_strm<<"Benefits: "<<life->benefits_curr<<endl;
}

if (life->sum_ins_curr * life->benefits_curr == 0)
    return 0;

if (t < life->mat_period_original)
    return sum_insured_if_b_pup(t) * ass_factor_weighted(t)
        * sum_insured_oldtag / (life->sum_ins_curr * life->benefits_curr);

return surr_value_pup(t)
    * life->surv_per_ret(t)
    * res_prop_mat_oldtag;

```

#### 6.1.1.4.1.75 res\_basic\_pup\_piz

```

if (t < life->commence_period_w || t >= life->maturity_period_w || life->surv_ret(t) == 0)
    return 0;

if(life->submodel != "TRAD")
    return 0;

if (t == 0){
    if (life->paid_up == "Y")
        return reserve_basic_pup(t) * life->res_prop_kitzba_piz;
}

```

```

        else
            return 0;
    }

    if (sum_insured(0) == 0)
        return 0;

    if (t < life->mat_period_original)
        return sum_insured_if_b_pup(t) * ass_factor_weighted_int0(t)
            * sum_insured_piz_int0 / sum_insured(0);

    if (t == life->mat_period_original)
        return surr_value_pup(t)
            * life->surv_per_ret(t)
            * res_prop_mat_piz;

    //t > mat_original
    return res_basic_pup_piz(t-1)
        * (1. - life->death_rate(t))
        * life->surv_per_ret(t); //Maturity surrender value decremented

```

#### 6.1.1.4.1.76 res\_basic\_pup\_piz\_int

```

if (t < life->commence_period_w || t >= life->maturity_period_w || life->surv_ret(t) == 0)
    return 0.0;

if(life->submodel != "TRAD")
    return 0.0;

if (t == 0){
    if (life->paid_up == "Y")
        return reserve_basic_pup(t) * life->res_prop_kitzba_piz;
    else
        return 0.0;
}

if (sum_insured(0) == 0)
    return 0;

if (t < life->mat_period_original)
    return sum_insured_if_b_pup(t) * ass_factor_weighted(t)
        * sum_insured_piz / sum_insured(0);

return surr_value_pup(t)
    * life->surv_per_ret(t)
    * res_prop_mat_piz;

```

#### 6.1.1.4.1.77 res\_basic\_pup\_prat

```

if (t < life->commence_period_w || t >= life->maturity_period_w || life->surv_ret(t) == 0)
    return 0.0;

if(life->submodel != "TRAD")
    return 0.0;

if (t == 0){
    if (life->paid_up == "Y")
        return reserve_basic_pup(t) * life->res_prop_kitzba_prat;
}

```

```

        else
            return 0.0;
    }

    if (sum_insured(0) == 0)
        return 0;

    if (t < life->mat_period_original)
        return sum_insured_if_b_pup(t) * ass_factor_weighted(t)
            * sum_insured_prat / sum_insured(0);

    return surr_value_pup(t)
        * life->surv_per_ret(t)
        * res_prop_mat_prat;

```

#### 6.1.1.4.1.78 reserve\_basic

```

if (t < life->commence_period_w || t >= life->maturity_period_w)
    return 0.0;

if(life->submodel != "TRAD")
    return 0.0;

if(t >= life->mat_period_original)
    return (surr_value(t) + surr_value_pup(t)) * life->surv_per_ret(t);

return reserve_basic_prem_if(t) + reserve_basic_pup(t);

```

#### 6.1.1.4.1.79 reserve\_basic\_prem\_if

```

if (t < life->commence_period_w || t >= life->mat_period_original)
    return 0.0;

if(life->submodel != "TRAD")
    return 0.0;

double reserve_temp = 0.0;
double interpol = life->pol_month(t)/12.;

if(t<=23 && t >= 0 && life->policy_type == "managers" && life->dump_vars == "Y"){

    log_strm<<"T: "<<t<<endl;
    log_strm<<"Surv prem: "<<life->surv_act_prm(t)<<endl;
    log_strm<<"Surv pup: "<<life->surv_pup_prm(t)<<endl;
    log_strm<<"Prem term prop: "<<life->prem_termination_prop(t)<<endl;
    //log_strm<<"Surv prev: "<<life->surv_prem(t-1)<<endl;
    //log_strm<<"Surv prem per: "<<life->surv_per_prem(t)<<endl;
    //log_strm<<"Death rate: "<<life->death_rate(t)<<endl;
    //log_strm<<"Pup rate: "<<life->pup_rate(t)<<endl;
    //log_strm<<"Prem term rate: "<<life->prem_termination_prop(t)<<endl;

}

if (life->paid_up=="Y")
    return 0.0;

int t_down = 0;
double res_up = 0.0;

```

```

if(xint(life->pol_month(t))==12)
    reserve_temp = vsa(t) - vnp(t);
else {
    t_down = (xint(life->pol_year(t))-1)*12 - life->elapsed_months;
    if (t_down+12 < life->mat_period_original)
        res_up = vsa(t_down+12)-vnp(t_down+12);
    else
        res_up = sum_insured(t);
    reserve_temp = (1. - interpol) * (vsa(t_down)-vnp(t_down))
        + interpol * res_up;
}

```

```

reserve_temp = reserve_temp * life->surv_act_bal(t);

```

```

if (life->zeroise_res=="Y")
    reserve_temp = max(0.0, reserve_temp);

```

```

return reserve_temp;

```

#### 6.1.1.4.1.80 reserve\_basic\_pup

```

if (t < life->commence_period_w || t >= life->mat_period_original)
    return 0.0;

```

```

if(life->submodel != "TRAD")
    return 0.0;

```

```

double reserve_temp = 0.0;
double reserve_temp_pup = 0.;
double interpol = life->pol_month(t)/12.;

```

```

int t_down = 0;
double res_up = 0.0;
if (life->paid_up=="Y"){
    if(xint(life->pol_month(t))==12)
        reserve_temp = vsa(t) - vnp(t);
    else {
        t_down = (xint(life->pol_year(t))-1)*12 - life->elapsed_months;
        if (t_down+12 < life->mat_period_original)
            res_up = vsa(t_down+12)-vnp(t_down+12);
        else
            res_up = sum_insured(t);
        reserve_temp = (1. - interpol) * (vsa(t_down)-vnp(t_down))
            + interpol * res_up;
    }
    reserve_temp = reserve_temp * life->surv_pup_bal(t);
    if (life->zeroise_res=="Y")
        reserve_temp = max(0.0, reserve_temp);
}
else{
    if ( (sum_insured(t-1) * life->surv_bal(t-1))>0. || (life->lapse_force_rate == 1. && life->paid_up == "N")) { // to add reserve i.r.o. paid-up benefit

        if(xint(life->pol_month(t))==12)
            reserve_temp_pup = vsa(t);
        else {
            t_down = (xint(life->pol_year(t))-1)*12 - life->elapsed_months;

```

```

        res_up = 0.0;
        if(xint(pol_sub_year(t))== xint(life->prem_lookup_freq_temp))
            res_up = 0.0;
        else
            res_up = vsa(t_down+12);

        if (t_down + 12 >= life->mat_period_original)
            res_up = sum_insured(t);

        reserve_temp_pup = (1. - interpol) * vsa(t_down) + interpol * res_up;
    }
}
if (sum_insured(t) != 0)
    reserve_temp_pup = reserve_temp_pup * life->surv_per_ret(t) *sum_insured_if_b_pup(t)
/ sum_insured(t);
if (life->zeroise_res=="Y")
    reserve_temp_pup = max(0.0, reserve_temp_pup);
reserve_temp = reserve_temp + reserve_temp_pup;
}
return reserve_temp;

```

#### 6.1.1.4.1.81 reserve\_risk\_premium

```

if (t < life->commence_period_w || t >= life->maturity_period_w)
    return 0.0;

if(life->submodel != "TRAD")
    return 0.0;

double qx_res_basis = 0.0;
int age_now = xint(life->age_last(t));
if (res_lx(age_now,0) > 0 )
    qx_res_basis = (1.0 - res_lx(age_now+1,0) / res_lx(age_now,0))/12.0;

if ( death_rate(t) > 0.0 )
    return death_claims_si(t) / death_rate(t) * qx_res_basis;
else
    return 0.0;

```

#### 6.1.1.4.1.82 vnp

```

if (t < life->commence_period_w || t >= life->mat_period_original)
    return 0.0;

if (life->submodel != "TRAD")
    return 0;

return net_premium_e(t) * annuity_factor(t);

```

#### 6.1.1.4.1.83 vsa

```

if (t < life->commence_period_w || t > life->mat_period_original)
    return 0.0;

if (life->submodel != "TRAD")
    return 0;

return sum_insured(t) * assurance_factor(t);

```

**6.1.1.4.1.84 death\_rate**

```

if (t <= life->commence_period_w || t > life->maturity_period_w)
    return NO_AVG;

if(life->submodel != "TRAD")
    return 0.0;

if (t <= 0 && (life->gross_up_historic=="N"))
    return 0.0;

int pol_yr = max(xint(life->pol_year(t)),1);

if (life->age_last(t) < 18.)//assume no mortality for children up to age 18.
    return 0.0;

// Assume all lives die at omega age
if (life->age_last(t) >= life->omega_age_w)
    return 1.0;

double rate = 0.0;

// set column to allow for selection in mortality table
life->col_dth = 0;
if (life->mort_sel_status=="Y")
    life->col_dth = min(pol_yr + xint(life->elapsed_months_extra/12.), life->select_periods);
else
    life->col_dth =life->select_periods;

life->death_rate_row_key =life->age_last(t) - life->col_dth+1;
life->col_dth = life->col_dth; // reset lookup variable to avoid mutating lookup error
rate = life->death_rates_tbl;
rate = rate * life->mort_mult / 100. *(1+life->health_occ_perc/100.);

// ***** Add Margin *****
if (life->margin_add=="Y")
    rate = rate * (1+life->margin_mort_pc/100);

//Margin for catastrophe
if (life->margin_add_cat == "Y" ){//Only apply to savings and death risk

    double m_cat = 0;

    if (life->proj_year(t) == 1)
        m_cat = life->cat_risk;

    rate = rate + m_cat;

}

//***** convert to monthly *****
rate = max(0.0, min(1.0, rate));

return rate = (1. - pow(1. - rate, 1./12.));

```

**6.1.1.4.1.85 pol\_fee**

```

if(t <= life->commence_period_w || (t + life->elapsed_months) > life->prem_term)
    return 0.0;

if(life->submodel != "TRAD")
    return 0.0;

if (life->paid_up=="Y")
    return 0.0;

if (fmod(xint(life->pol_month(t-1)), xint(12. / life->prem_freq))!=0)
    return 0.0; //not a premium due date

return life->policy_fee_if
    * life->policies_curr * life->surv_act_prm(t-1)
    / life->prem_freq
    * (1-life->pol_fee_disc_perc/100.);

```

**6.1.1.4.1.86 premium**

```

if (t <= life->commence_period_w || t + life->elapsed_months > life->prem_term)
    return 0.0;

if(life->submodel != "TRAD")
    return 0.0;

if (fmod(xint(life->pol_month(t-1)), xint(12. / life->prem_freq))!=0)
    return 0.0; //not a premium due date

return premium_if_b(t) / life->prem_freq;

```

**6.1.1.4.1.87 premium\_gross**

```

if (t < life->commence_period_w || t > life->maturity_period_w)
    return NO_AVG;

if (fmod(xint(life->pol_month(t-1)), xint(12. / life->prem_freq))!=0)
    return 0.0; //not a premium due date

if(life->submodel != "TRAD")
    return 0.0;

return premium(t) + pol_fee(t);

```

**6.1.1.4.1.88 claims\_rate\_per**

```

if(t <= life->commence_period_w || t > life->maturity_period_w || eq(life->paid_up, "G"))
    return 0.0;

if(life->submodel != "TRAD")
    return 0.0;

double maturity_rate = 0.0;
double surr_rate = 0.0;
double dth_rate = 0.0;

//Maturity
if (life->mult_age_ind == 1) {

```



```

    maturity_rate = life->surv_cnt_bef_ret(t) * life->retirement_prop(t) * life->benefits_curr;
}
else
{
    if (t == life->maturity_period_w)
        maturity_rate = (life->surv_act_cnt(t-1) + life->surv_pup_cnt(t-1))
                        * life->benefits_curr;
}

//Surrender
if (life->surv_per_cnt(t) > 0.0)
    surr_rate = life->surv_act_cnt(t-1) * life->benefits_curr * life->lapse_rate_act_cnt_dep(t)
               + life->surv_pup_cnt(t-1) * life->lapse_rate_pup_cnt_dep(t);

//Death
if (life->surv_cnt(t-1) > 0.0)
    dth_rate = life->surv_cnt(t-1) * death_rate(t);

return maturity_rate + surr_rate + dth_rate;

```

#### 6.1.1.4.1.89 premium\_if\_b

```

if(t <= life->commence_period_w || (t + life->elapsed_months) > life->prem_term )
    return 0.0;

if(life->submodel != "TRAD" || life->paid_up=="Y")
    return 0.0;

double inc = 0.0;
if ((t>=1 && xint(life->pol_month(t)) == 1. && xint(life->pol_year(t)) > 1) ||
    (t < 1 && xint(life->pol_month(t)) == 12))
    inc = life->prem_curr * life->benefits_curr * life->surv_act_prm(t-1) * life->
    >premium_inc(t)/100.
        / (1 + life->premium_inc(t)/100.* (life->pol_year(1)-1));

if (life->prem_lookup_temp=="N") { //level premium
    if (t == 1)
        return life->prem_curr * life->benefits_curr;
    if (t > 1)
        return premium_if_b(t-1) * life->surv_per_act_prm(t-1) + inc;
    // t<1
    if (life->surv_per_act_prm(t) == 0.) {
        return 0.;}
    return premium_if_b(t+1) / life->surv_per_act_prm(t) - inc;
}

if (t + life->commence_period_w == 1)
    return life->prem_curr * life->benefits_curr;

return premium_if_b(t-1) * life->surv_per_act_prm(t-1);

```

#### 6.1.1.4.1.90 premium\_if\_e

```

if (t < life->commence_period_w || (t + life->elapsed_months) >= life->prem_term)
    return 0.0;

```

```
if(life->submodel != "TRAD")
    return 0.0;
```

```
if (life->paid_up=="Y")
    return 0.0;
```

```
return premium_if_b(t+1);
```

#### **6.1.1.4.1.91 annuity\_if\_b**

```
if (t <= life->commence_period_w || t > life->mat_period_original)
    return 0.0;
```

```
if(life->submodel != "TRAD" || life->paid_up=="Y" || ann_factor_pol == 0)
    return 0.0;
```

```
return (sum_insured(t-1) * life->surv_act_bal(t-1)) / ann_factor_pol * 100.;
```

#### **6.1.1.4.1.92 claims\_ret**

```
if(life->mult_age_ind != 1)
    return 0.;
```

```
if(life->submodel != "TRAD")
    return 0.0;
```

```
if (t < life->mat_period_min || t > life->maturity_period_w)
    return 0.0;
```

```
if (t == life->mat_period_original)
    return 0.;
```

```
return surr_value(t) + surr_value_pup(t);
```

#### **6.1.1.4.1.93 sum\_at\_risk\_if**

```
if (t <= life->commence_period_w || t + life->elapsed_months >= life->ben_term_max)
    return 0.0;
```

```
if (life->submodel != "TRAD")
    return 0;
```

```
// Derive sum assured for death from death claims si formula
```

```
double deathSI = 0.0;
```

```
if (death_rate(t) > 0.0)
```

```
    deathSI = death_claims_si(t) / death_rate(t) * life->surv_per_ret(t);
```

```
else
```

```
    deathSI = sum_insured_if_e(t)+sum_insured_if_b_pup(t+1);
```

```
return max(deathSI- reserve_basic(t), 0.);
```

#### **6.1.1.4.1.94 sum\_insured**

```
if (t < life->commence_period_w || t > life->mat_period_original)
    return 0.0;
```

```
if (life->submodel != "TRAD")
    return 0;
```

```

double si_inc_fix = 0.0;
if ((xint(life->pol_month(t)) == 12) && (t > 0))
    si_inc_fix = life->sum_ins_inc(t) / 100. * life->sum_ins_curr;
if ((xint(life->pol_month(t)) == 11) && (t < 0))
    si_inc_fix = life->sum_ins_inc(t) / 100. * life->sum_ins_curr;

if (t == 0)
    return life->sum_ins_curr * life->benefits_curr *
        (1+life->sum_ins_inc(t)/100.*max(life->pol_year(1)-1,0));
if (t > 0)
    return sum_insured(t-1) + si_inc_fix * life->benefits_curr;
// t<0
return sum_insured(t+1) - si_inc_fix * life->benefits_curr;

```

#### 6.1.1.4.1.95 sum\_insured\_if\_b

```

if (t <= life->commence_period_w || t > life->mat_period_original)
    return 0.0;

if (life->submodel != "TRAD")
    return 0;

if (life->paid_up=="Y") // paid-up cover in force uses secondary sum-life->insured
    return 0.0;

return sum_insured(t-1) * life->surv_act_prm(t-1);

```

#### 6.1.1.4.1.96 sum\_insured\_if\_b\_pup

```

if (t <= life->commence_period_w || t > life->mat_period_original || eq(life->paid_up,"G"))
    return 0.0;

if (life->submodel != "TRAD")
    return 0;

if (t <= 1){
    if (life->paid_up=="Y") // paid-up cover in force
        return sum_insured(t);
    else
        return 0.;
}

if (eq(life->sur_val_method, "sv_table")) {
    double PUV = 0.0;
    double ann_fac_pol=1.0;
    if(eq(life->ben_class,"GIMLA"))
        ann_fac_pol = ann_factor_pol/100.;

    if(sv_factor(t-1)>0.0){
        if (life->lapse_total_prm(t-1) == 1.0) //if everyone lapses assume with end of last
months surrender value
            PUV = (surr_value(t-1)- bonus_if(t-1))* life->surv_per_ret(t-1) *ann_fac_pol * (1 -
death_rate(t-1))
                * puv_factor(t-1)/sv_factor(t-1);
        else{
//surr_val is the inforce item after the surrenders have occurred
            if (life->surv_per_prm_bef_ret(t-1)>0.0)
                PUV = (surr_value(t-1)- bonus_if(t-1))*ann_fac_pol/life-
>surv_per_prm_bef_ret(t-1)*life->surv_per_ret(t-1) * puv_factor(t-1)/sv_factor(t-1);

```

```

    }
}

return sum_insured_if_b_pup(t-1)
    * (1. - death_rate(t-1))
    * (1. - life->lapse_rate_pup_prm(t-1)) * life->surv_per_ret(t-1)
+ PUV // addition for new silukim
    * life->pup_rate_prm(t-1);

}

return 0.0;

```

#### 6.1.1.4.1.97 sum\_insured\_if\_e

```

if (t < life->commence_period_w || t >= life->mat_period_original)
    return 0.0;

```

```

if (life->submodel != "TRAD")
    return 0;

```

```

return sum_insured_if_b(t+1);

```

#### 6.1.1.4.1.98 claims\_re

```

if (t <= life->commence_period_w || t > life->mat_period_original || eq(life->re_type,"NONE"))
    return 0.0;

```

```

if(life->submodel != "TRAD")
    return 0.0;

```

```

if (eq(life->re_type,"simple"))
    return death_rate(t) * sum_at_risk_if(t-1) * life->re_clm_rein_pc/100.;

```

```

//else other re_type
return death_rate(t) * max(sum_at_risk_if(t-1) - life->re_clm_ret_fix* life->surv_bal(t-1),0.);

```

#### 6.1.1.4.1.99 comm\_re

```

if (t <= life->commence_period_w || t > life->maturity_period_w || eq(life->re_type,"NONE"))
    return 0.0;

```

```

if(life->submodel != "TRAD")
    return 0.0;

```

```

int yr;

```

```

if (atof(life->comm_by_cal)==1)
    yr=xint(life->cal_duration(t)+1);
else
    yr=xint(life->pol_year_ext(t));

```

```

if (yr >= 2)
    return life->comm_ren_re[yr] / 100. * premium_re(t) ;

```

```

return 0.0;

```

**6.1.1.4.1.100 comm\_re\_prof**

```

if (t <= life->commence_period_w || t > life->maturity_period_w)
  return 0.0;

if(life->submodel != "TRAD")
  return 0.0;

return max(0,life->comm_prof_re / 100. * profit_re(t)) ;

```

**6.1.1.4.1.101 exp\_re\_nom**

```

if (t <= life->commence_period_w || t > life->maturity_period_w || life->reinsurance=="N" ||
eq(life->re_type,"NONE"))
  return 0.0;

if(life->submodel != "TRAD")
  return 0.0;

return life->expense_re_nom_temp / 100. * premium_re(t);

```

**6.1.1.4.1.102 premium\_if\_b\_re**

```

if(t <= life->commence_period_w || t > life->mat_period_original)
  return 0.0;

if(life->submodel != "TRAD" || life->reinsurance=="N" || eq(life->re_type,"NONE"))
  return 0.0;

if(eq(life->re_type,"simple"))
  return claims_re(t)*(1.+life->re_cost_perc/100.) * life->prem_freq;

if (eq(life->re_type,"OT")) {
  return premium_if_b(t)* sum_insured_re(t-1)/sum_insured(t-1);
}

// re_type = YRT
double prate = 0.0;

if (xint(life->pol_month(t)) == 1){

  prate = life->prem_rates_re * (1+ max(life->health_occ_perc_min,life->health_occ_perc)/100.)
+ life->prem_per_unit_si_re;

  prate = prate / 1000.0 * max(sum_at_risk_if(t-1) - life->re_clm_ret_fix* life->surv_bal(t-
1),0.);

  return prate;
}

return premium_if_b_re(t-1) * life->surv_per_bal(t-1);

```

**6.1.1.4.1.103 premium\_re**

```

if (t <= life->commence_period_w || t > life->maturity_period_w)
  return 0.0;

if(life->submodel != "TRAD")
  return 0.0;

```

```

if (t < -13)
  return 0.0;

if (fmod(xint(life->pol_month(t-1)), xint(12. / life->prem_freq))!=0)
  return 0.0; //not a premium due date

return premium_if_b_re(t) / life->prem_freq;

```

#### 6.1.1.4.1.104 profit\_re

```

if (t <= life->commence_period_w || t > life->maturity_period_w || t<-120)
  return 0.0;

if(life->submodel != "TRAD")
  return 0.0;

if (t==0)
  return 0.0;

return premium_re(t)
  - claims_re(t)
  - comm_re(t)
  - exp_re_nom(t);

```

#### 6.1.1.4.1.105 sum\_insured\_re

```

if (t < life->commence_period_w || t > life->mat_period_original)
  return 0.0;

if (life->submodel != "TRAD")
  return 0;

if(life->reinsurance=="N")
  return 0;

double sum_ins_re = sum_insured(t) - life->benefits_curr * life->re_clm_ret_fix;

return max(sum_ins_re,0);

```

#### 6.1.1.4.1.106 startup

```

// do not continue with startup if record not for this submodel
start_extrns
  extern map <int, int> ann_index_map;
end_extrns

if (life->submodel != "TRAD")
  return 0;

set_other_variables();
set_premium_si();

if (eq(life->done_startup_w,"false")){// Dump variables
  validate_data();
  life->done_startup_w = "true";
}

if (!eq(life->error_msg,"no_error")){ // this causes all formulae to be zero

```

```

    life->maturity_period_w = -1;
    life->mat_period_min = -1;
    life->commence_period_w = 1;
}

```

```
return 0.0;
```

#### **6.1.1.4.1.107 pol\_sub\_year**

```

if (t < life->commence_period_w || t > life->maturity_period_w)
    return NO_AVG;

if(life->submodel != "TRAD")
    return 0.0;

if(life->prem_lookup_freq_temp == 0)//level premium
    return life->pol_year(t);

if (xint(fmod(xint(life->pol_year(t)),life->prem_lookup_freq_temp)) == 0)//YRT or stepped premium
    return life->prem_lookup_freq_temp;

return fmod(xint(life->pol_year(t)),life->prem_lookup_freq_temp);

```

### **6.1.1.4.2 External Functions**

#### **6.1.1.4.2.1 monthly\_rate**

```

double monthly_rate(double annual_rate) {
    return (pow(1. + annual_rate / 100. , 1. / 12.) - 1.);
}

```

#### **6.1.1.4.2.2 set\_other\_variables**

```

void set_other_variables (void) {

    int mth=0, year=0, i=0;
    double total=0.0,mult=0.0;

    // calculate benefit term for Whole of Life
    if (eq(life->ben_class, "WOL"))
        life->benefit_term = (95 - xint(life->age_at_issue)) * 12; // to correct limiting of term in
conversion
    // set premium term

        if (eq(life->ben_class,"WOL")&& atoi(life->prem_age)>0)
            life->prem_term = (atoi(life->prem_age) - xint(life->age_at_issue)) * 12;
        else
            life->prem_term = life->benefit_term;

    // set Capital requirement as a percentage of DAC-Books
    if (life->dac_cap_apply=="N")
        life->dac_cap_perc_w = 0.0;
    else {
        if (life->prod_yr_w < 1999)
            life->dac_cap_perc_w = 0.0;
        if (life->prod_yr_w >= 1999)
            life->dac_cap_perc_w = 30.0;
    }
}

```

```

        if (life->prod_yr_w >= 2004)
            life->dac_cap_perc_w = 100.0;
    }

    life->ben_term_max = life->benefit_term;
    life->ben_period_min = life->benefit_term;

    if(life->mult_age_ind == 1){
        life->ben_term_max = max(life->benefit_term, (life->sm_annuity[life-
>sm_annuity.size()-1]->takeup_age - life->age_at_issue) * 12);

        life->ben_period_min = min(life->benefit_term, (life->min_retirement_age - life-
>age_at_issue) * 12);
    }

    life->maturity_period_w = life->commence_period_w + life->ben_term_max;
    life->mat_period_min = life->commence_period_w + life->ben_period_min;
    life->mat_period_original = life->commence_period_w + life->benefit_term;
    life->matan_period_w = life->commence_period_w + life->matan_term;
    life->gimla_db_period_w = life->commence_period_w + (xint(life->benefit_term/18) - 3) * 12;

    if (life->prod_code == "whlf" || life->prod_code == "whlp")
        life->maturity_period_ann = life->maturity_period_w;

    if (life->maturity_period_w >= 12*xint(t_high/12.))
        throw("Benefit Term exceeds projection period, rerun with larger t_high\n");

    // ***** Set Interest rates *****

    life->v_month_w = 1. / (1. + monthly_rate(life->ev_disc_rate));

    // if discount type = Single, then replace the discount rate vector with the input value
    if (eq(life->ev_discount_rate_type,"Single")) {
        for (i=0; i<=119;i++){
            life->v_month_t[i] = life->v_month_w;
        } //end for loop
    } // end if

    // ***** set commission variables *****
    //commission reduction for short premium terms

    mult = 1.;
    if(life->comm_min_prem_term > 0){
        mult = life->prem_term/life->comm_min_prem_term;
        mult = min(1.,mult);

        for(i = 0; i<116; i++){
            life->comm_regular_pc[i] = life->comm_regular_pc[i]*mult;
        }
    }

    // Set total percentage of initial regular commission
    double comm_tot =0.;
    for (i = 0; i<115; i++)
        comm_tot = comm_tot+life->comm_regular_pc[i];

    life->comm_reg_tot_w = comm_tot;

```



```

// Set DAC amortisation period
if(eq(life->dac_amort_type,"Lifetime"))
    life->dac_amort_per = life->prem_term;
if(life->dac_amort_per > life->prem_term)
    life->dac_amort_per = life->prem_term;

// set old numerical product code to reference sv tables
life->prod_code_old = xstring(life->prodcold);

life->fund_name_temp = xstring(life->fund_name);
// Change fund name to read right annuity factors
if(inlist(life->prod_code,"a72,a80-00honi") && (atoi(life->fund) < 100 || inlist(life->fund,
"521,523,527"))))
    life->fund_name_temp = xstring(min(atoi(life->fund_name_temp),50));

if(eq(life->prod_code, "asav") && inlist(life->fund, "52,521,523,527"))
    life->fund_name_temp = xstring(min(atoi(life->fund_name_temp),50));

/*****set sv and puv table*****/
if(eq(life->sur_val_method,"sv_table")){
    if (eq(life->ben_class,"GIMLA")){
        life->sv_tbl = life->fund_name_temp + "_" + life->prod_code_old + "_" + life->sex;
        life->puv_tbl = life->fund_name_temp + "_puv_" + life->prod_code_old + "_" + life-
>sex;
    }
    else {
        life->sv_tbl = life->fund_name_temp + "_" + life->prod_code_old;
        life->puv_tbl = life->fund_name_temp + "_puv_" + life->prod_code_old;
    }
}

// close function
}

```

#### 6.1.1.4.2.3 set\_premium\_si

```

void set_premium_si (void) { // current premium (+loadings) or sum insured calculation

//no lookup while level premium
if (life->prem_lookup_temp=="N")
    life->prem_lookup_freq_temp = 0;

if (life->premium_rate_w <= 0)
    throw NonFatalError("Premium Rate is zero or negative for policy "+life->pol_number+". Check
premium table for relevant age-term.");

if ((life->policy_fee_if > life->prem_curr) && eq(life->paid_up , "N")) // by definition trad
premium comes from the file and includes policy fee and must therefore be at least as high
    life->error_msg = "Policy fee exceeds gross premium"; // skips record with error message

if (eq(life->done_startup_w,"false")) { // for NB layering do not repeat these adjustments because
they are retained from 1 layer to the next
    if (life->mod_load_in_prem=="Y")//take off modal loading
        life->prem_curr = life->prem_curr / (1 + life->tat_shnatiut_rate/100.);
    //take off policy fee

```

```

        life->prem_curr = life->prem_curr - life->policy_fee_if* life->policies_curr/life-
>benefits_curr;

        life->prem_curr = life->prem_curr / (1 + life->health_occ_perc/100.);    // take off %
health/occ loading to premium
    }

//reload premium
life->prem_curr = life->prem_curr * (1 + life->tat_shnatiut_rate/100.) * (1 + life-
>health_occ_perc/100.);

//add modal loading on policy fee
if (eq(life->done_startup_w,"false")) // for NB layering do not repeat these adjustments because
they are retained from 1 layer to the next
    life->policy_fee_if = life->policy_fee_if * (1+ life->tat_shnatiut_rate/100.); /*** Need to
check if p.f. includes modal loading when it comes from file

// close function
}

```

#### 6.1.1.4.2.4 validate\_data

```

void validate_data(void) {

    if (life->prem_term > life->benefit_term)
        life->error_msg = "prem_term_>_ben_term";

    if (pv_period != 12)
        throw NonFatalError("Template set up for monthly projections. Change the discount period in
the projection task.");

    // check minimum age of mortality tables
    int min_age_1 = -3.0; //GetRowKeyMinValue(life->death_rates_tbl)
    if (eq(life->mort_sel_status,"N"))
        min_age_1 =max(0, min_age_1 + life->select_periods - 1);

    if (life->age_at_issue < min_age_1)
        throw NonFatalError ("Policy number " + life->pol_number + ": Issue age of life 1 is less
than the minimum age of the mortality table");

    if (eq(life->projection_type,"Valn") && life->elapsed_months < 0)
        life->error_msg = "elapsed_months_<_0";

    // Ensure frequencies given are factors of 12.
    if ( !inlist(xstring(life->prem_freq),"1,2,3,4,6,12") ||
        (life->prem_freq == 0 && !eq(life->paid_up,"Y")) ||
        (life->prem_freq == 0 && life->prem_term > life->elapsed_months) )
        throw NonFatalError("Premium frequency must be 1, 2, 3, 4, 6, or 12.");

    if (!inlist(life->ben_class,"WOL,END,YTRON,GIMLA"))
        throw NonFatalError("Unknown life->ben_class for policy number: " + life->pol_number);

    if (life->sum_ins_curr <= 0)
        life->error_msg = "Sum_Insured_<=_0";

    // close function
}

```

### 6.1.1.4.3 Temporary Tables

#### 6.1.1.4.3.1 res\_cx

```
// Commutation Function Cx = v^(x+1) * lx - l(x+1)
// r = current age in years

if (r == life->omega_age_w) {

    if (c == 0)
        return res_dx(r,0)/(1.+life->int_rate_res/100.);
    else
        return res_dx(r,1);

}

double d = res_lx(r,0) - res_lx(r+1,0); //deaths aged r

if (life->int_rate_res == 0.0 || c == 1)
    return d;

return d * res_vx(r+1,0);
```

#### 6.1.1.4.3.2 res\_dx

```
// Commutation Function Dx Yearly Dx = lx * v^x
// r = current age in years

if (r > life->omega_age_w) //life->omega_age_w from underlying table
    return 0.0;

return res_lx(r, 0) * res_vx(r,c); //Survival rates have only 1 column, vx has 2 because of 0
interest rate
```

#### 6.1.1.4.3.3 res\_lx

```
// Commutation Function lx

if (r <= 0)
    return 100.0;

if (r > life->omega_age_w) // omega age allows for table adjustment
    return 0.0;
life->row_num = r-1;
double q = life->death_rates_res_tbl;
    q = q +life->mort_addn_res/1000.;

return res_lx(r-1, 0) * (1. - q);
```

#### 6.1.1.4.3.4 res\_mx

```
if (r>life->omega_age_w)
    return 0.0;

if (r==life->omega_age_w)
    return res_cx(r,c);

return res_cx(r,c) + res_mx(r+1,c);
```

**6.1.1.4.3.5 res\_nx**

```
//Nx
```

```
if (r >= life->omega_age_w)
    return res_dx(r, c);
```

```
return res_nx(r+1, c) + res_dx(r, c);
```

**6.1.1.4.3.6 res\_vx**

```
// Commutation Function vx = v^(x)
```

```
// r = current age in years
```

```
if (r <= 0) // cannot look up a zero or negative starting age
    return 1.0;
```

```
if (c==1)//0 interest rate
    return 1.0;
```

```
return res_vx(r-1,0) /(1.+life->int_rate_res/100.);
```

**6.1.1.4.4 Scalars****6.1.1.4.4.1 interest\_rein\_mthly**

```
return monthly_rate(life->interest_rein);
```

**6.1.1.4.4.2 int\_rate\_res\_hy**

```
return pow(1. + life->int_rate_res/100., 0.5);
```

**6.1.1.4.4.3 int\_rate\_res\_mthly**

```
return pow(1. + life->int_rate_res/100., 1/12.)-1;
```

**6.1.1.4.4.4 res\_prop\_mat\_newtag**

```
if (!eq(life->ben_class, "gimla"))
    return 0.0;
```

```
if(life->dump_vars == "Y"){
```

```
    log_strm<<"Elaps_mths: "<<life->elapsed_months<<endl;
    log_strm<<"Benefit term: "<<life->benefit_term<<endl;
    log_strm<<"Res prop orig: "<<life->res_prop_kitzba_newtag<<endl;
    log_strm<<"Res basic active: "<<reserve_basic_prem_if(life->mat_period_original-1)<<endl;
    log_strm<<"Res basic pup: "<<reserve_basic_pup(life->mat_period_original-1)<<endl;
    log_strm<<"Res new active: "<<res_basic_act_newtag (life->mat_period_original-1)<<endl;
    log_strm<<"Res new pup: "<<res_basic_pup_newtag (life->mat_period_original-1)<<endl;
```

```
}
```

```
if(life->elapsed_months >= life->benefit_term)
    return life->res_prop_kitzba_newtag;
```

```
if ((reserve_basic_prem_if(life->mat_period_original-1) + reserve_basic_pup(life-
>mat_period_original-1)) != 0)
    return (res_basic_act_newtag (life->mat_period_original-1)
            + res_basic_pup_newtag (life->mat_period_original-1))
```

```

        / (reserve_basic_prem_if(life->mat_period_original-1) +
reserve_basic_pup(life->mat_period_original-1));

return 0.0;

```

#### 6.1.1.4.4.5 res\_prop\_mat\_oldtag

```

if (!eq(life->ben_class, "gimla"))
    return 0.0;

if(life->elapsed_months >= life->benefit_term)
    return life->res_prop_kitzba_oldtag;

if ((reserve_basic_prem_if(life->mat_period_original-1) + reserve_basic_pup(life-
>mat_period_original-1)) != 0)
    return (res_basic_act_old (life->mat_period_original-1)
            + res_basic_pup_old (life->mat_period_original-1))
            / (reserve_basic_prem_if(life->mat_period_original-1) +
reserve_basic_pup(life->mat_period_original-1));

return 0.0;

```

#### 6.1.1.4.4.6 res\_prop\_mat\_piz

```

if (!eq(life->ben_class, "gimla"))
    return 0.0;

if(life->elapsed_months >= life->benefit_term)
    return life->res_prop_kitzba_piz;

if ((reserve_basic_prem_if(life->mat_period_original-1) + reserve_basic_pup(life-
>mat_period_original-1)) != 0)
    return (res_basic_act_piz (life->mat_period_original-1)
            + res_basic_pup_piz (life->mat_period_original-1))
            / (reserve_basic_prem_if(life->mat_period_original-1) +
reserve_basic_pup(life->mat_period_original-1));

return 0.0;

```

#### 6.1.1.4.4.7 res\_prop\_mat\_prat

```

if (!eq(life->ben_class, "gimla"))
    return 0.0;

if(life->elapsed_months >= life->benefit_term)
    return life->res_prop_kitzba_prat;

if ((reserve_basic_prem_if(life->mat_period_original-1) + reserve_basic_pup(life-
>mat_period_original-1)) != 0)
    return (res_basic_act_prat (life->mat_period_original-1)
            + res_basic_pup_prat (life->mat_period_original-1))
            / (reserve_basic_prem_if(life->mat_period_original-1) +
reserve_basic_pup(life->mat_period_original-1));

return 0.0;

```

#### 6.1.1.4.4.8 ann\_factor\_pol

```

double sv09 = 0;

```

```

if (xint(life->pol_year(life->mat_period_original)) > 0 && !eq(life->prod_code_old,"0")){
  life->sv_col_key = xstring(min(xint(life->benefit_term/12),life->pol_year(life-
>mat_period_original)));
  life->sv_row_key = xstring(xint(life->age_at_issue))+ "_" + xstring(xint(life-
>benefit_term/12));
  sv09 = life->sv_09_tbl;
}

```

```
return sv09;
```

#### 6.1.1.4.4.9 sum\_insured\_newmoney

```

if ((life->res_prop_kitzba_newtag + life->res_prop_kitzba_piz + life->res_prop_kitzba_prat) <= 0 ||
!eq(life->ben_class, "gimla") || life->mat_period_original <= 0 )
  return 0.0;

```

```
return max(0, sum_insured(0) - sum_insured_oldtag);
```

#### 6.1.1.4.4.10 sum\_insured\_newtag

```

if (!eq(life->ben_class, "gimla") || life->mat_period_original <= 0 )
  return 0.0;

```

```
double res_open = reserve_basic_prem_if(0) + reserve_basic_pup(0);
```

```
double res_tag = res_open * life->res_prop_kitzba_newtag;
```

```
if(ass_factor_weighted(0) == 0)
```

```

  return sum_insured_newmoney
    * life->res_prop_kitzba_newtag
    / (life->res_prop_kitzba_newtag + life->res_prop_kitzba_piz + life-
>res_prop_kitzba_prat);

```

```

if(eq(life->policy_type, "private") || life->paid_up == "Y") //Case where tag receives no premiums
  return res_tag / ass_factor_weighted(0);

```

```
//Case where tag receives prems
```

```

return (res_tag + ann_factor_weighted(0) * net_premium_e(1) * life->prem_newtag_prop / 100.)
  / ass_factor_weighted(0);

```

#### 6.1.1.4.4.11 sum\_insured\_oldtag

```

if (life->res_prop_kitzba_oldtag <= 0 || !eq(life->ben_class, "gimla") || life->mat_period_original
<= 0 )
  return 0.0;

```

```
// Note: at present there is no SA increase for gimla, and so this element is ignored. Need to
adjust model if this changes
```

```
double res_open = reserve_basic_prem_if(0) + reserve_basic_pup(0);
```

```
double res_old = res_open * life->res_prop_kitzba_oldtag;
```

```
//Old money assumes no new prems - treat as paid_up
```

```
double ass_factor = ass_factor_weighted(0);
```

```
if (ass_factor > 0)
  return res_old / ass_factor;
```

```
return 0.0;
```

#### 6.1.1.4.4.12 sum\_insured\_piz

```
if (!eq(life->ben_class, "gimla") || life->mat_period_original <= 0 )
  return 0.0;
```

```
double res_open = reserve_basic_prem_if(0) + reserve_basic_pup(0);
```

```
double res_piz = res_open * life->res_prop_kitzba_piz;
```

```
if(ass_factor_weighted(0) == 0)
```

```
  return sum_insured_newmoney
    * life->res_prop_kitzba_piz
    / (life->res_prop_kitzba_newtag + life->res_prop_kitzba_piz + life-
>res_prop_kitzba_prat);
```

```
if(eq(life->policy_type, "private") || life->paid_up == "Y") //Case where piz receives no premiums
  return res_piz / ass_factor_weighted(0);
```

```
//Case where piz receives prems
```

```
return (res_piz + ann_factor_weighted(0) * net_premium_e(1) * (1. - life->prem_newtag_prop / 100.))
  / ass_factor_weighted(0);
```

#### 6.1.1.4.4.13 sum\_insured\_piz\_int0

```
if (!eq(life->ben_class, "gimla") || life->mat_period_original <= 0 )
  return 0.0;
```

```
double res_open = reserve_basic_prem_if(0) + reserve_basic_pup(0);
```

```
double res_piz = res_open * life->res_prop_kitzba_piz;
```

```
if(ass_factor_weighted_int0(0) == 0)
```

```
  return sum_insured_newmoney
    * life->res_prop_kitzba_piz
    / (life->res_prop_kitzba_newtag + life->res_prop_kitzba_piz + life-
>res_prop_kitzba_prat);
```

```
if(eq(life->policy_type, "private") || life->paid_up == "Y") //Case where piz receives no premiums
  return res_piz / ass_factor_weighted_int0(0);
```

```
//Case where piz receives prems
```

```
return (res_piz + ann_factor_weighted_int0(0) * net_premium_e(1) * (1. - life->prem_newtag_prop /
100.))
  / ass_factor_weighted_int0(0);
```

**6.1.1.4.4.14 sum\_insured\_prat**

```

if (!eq(life->ben_class, "gimla") || life->mat_period_original <= 0 )
  return 0.0;

double res_open = reserve_basic_prem_if(0) + reserve_basic_pup(0);

double res_prat = res_open * life->res_prop_kitzba_prat;

if (ass_factor_weighted(0) == 0)
  return sum_insured_newmoney
    * life->res_prop_kitzba_prat
    / (life->res_prop_kitzba_newtag + life->res_prop_kitzba_piz + life-
>res_prop_kitzba_prat);

if(eq(life->policy_type, "private") && life->paid_up == "N"){ //Case where prat receives premiums

  return (res_prat + net_premium_e(1) * ann_factor_weighted(0))
    / ass_factor_weighted(0);

}

//Case where prat receives no premiums

return res_prat / ass_factor_weighted(0);

```

**6.1.1.5 sub\_array****6.1.1.5.1 Columns****6.1.1.5.1.1 initial\_formula**

```

return 0.0;

```

**6.1.1.5.2 External Functions**

<No External Functions Exist>

**6.1.1.5.3 Temporary Tables**

<No Temporary Tables Exist>

**6.1.1.5.4 Scalars**

<No Scalars Exist>

**6.1.1.6 sub1\_cflow****6.1.1.6.1 Columns****6.1.1.6.1.1 bonus\_rate\_acc\_mthly**

```

if (t<= t_start )
  return 0.0;

return (1.+bonus_rate_acc_mthly(t-1))*(1.+bonus_rate_mthly(t))-1;

```



**6.1.1.6.1.2 bonus\_rate\_mthly**

```

if (t <= life->commence_period_w || t > life->maturity_period_w)
    return 0.0;

if(eq(life->ben_class, "phi") && eq(life->phi_type, "S"))
    return 0.0;

int proj_yr = xint(life->proj_year(t));
if(eq(life->projection_type_int, "Rollup"))
    proj_yr = xint(life->proj_year_rollup(t));

// calculate bonus rate
double rate = (1. + inv_rate_clm_mth_t[proj_yr])
    *(1.-mgt_fee_fixed_clm/1200.)-1;

if (rate > 0.)
    rate = rate * (1.-mgt_fee_var_clm/100.);

return rate - int_rate_res_mthly;

```

**6.1.1.6.1.3 err**

```

if (t <= life->commence_period_w || t >= life->maturity_period_w)
    return 0.0;

if (life->death_ben_w=="N" || life->err_sar_perc == 0.)
    return 0.0;

double factor = 1.;
if (life->err_spread_period>0.01)
    factor = min(1.,(life->pol_year(t) - 1 + (life->pol_month(t) + life->elapsed_months_extra) /
12.) / life->err_spread_period);

return max (0.0, life->err_sar_perc / 100. * factor * sum_at_risk_if(t) * (1-life-
>re_clm_rein_pc/100.));

```

**6.1.1.6.1.4 reserve**

```

if (t <= life->commence_period_w || t >= life->maturity_period_w)
    return 0.0;
if (life->zeroise_res=="Y")
    return max(0.0, reserve_basic(t) +reserve_basic_claims(t) +res_np_deficiency(t) );

return reserve_basic(t) +reserve_basic_claims(t) +res_np_deficiency(t);

```

**6.1.1.6.1.5 annuity\_factor**

```

if (t < life->commence_period_w || t >= life->prem_term - life->elapsed_months || inlist(life-
>paid_up,"Y,C"))
    return NO_AVG;

int age_now = xint(life->age_last(t+1));
int age_end = 0;

if (life->prem_lookup_temp=="Y") {
    age_end = xint(life->age_last((xint((life->pol_year(t+1)-1)/life-
>prem_lookup_freq_temp)+1)*life->prem_lookup_freq_temp*12-life->elapsed_months));
    if (age_end ==0)
        age_end = xint(life->age_last(life->maturity_period_w)+1.);
}

```

```

        age_end = xint(1. + age_end);
    }
else
    age_end = xint(life->age_at_issue
        + ceil(life->prem_term/12. * life->prem_freq)
        /life->prem_freq);

return (res_nx(age_now,0) - res_nx(age_end,0)) / res_dx(age_now,0);

```

#### 6.1.1.6.1.6 assurance\_factor

```

if (t < life->commence_period_w || t >= life->maturity_period_w)
    return NO_AVG;

int age_now = xint(life->age_last(t+1));
int age_end = 0;
if (life->prem_lookup_temp=="Y") {
    age_end = xint(life->age_last((xint((life->pol_year(t+1)-1)/life-
>prem_lookup_freq_temp)+1)*life->prem_lookup_freq_temp*12-life->elapsed_months));
    if (age_end ==0)
        age_end = xint(life->age_last(life->maturity_period_w)+1);

    age_end = xint(1. + age_end);
}
else
    age_end = xint(life->age_at_issue + life->benefit_term/12.);

return (res_mx(age_now,0) - res_mx(age_end,0)) / res_dx(age_now,0) * int_rate_res_hy;

```

#### 6.1.1.6.1.7 net\_prem\_deficiency\_b

```

if(t <= life->commence_period_w || t + life->elapsed_months > life->prem_term)
    return 0.0;

if(inlist(life->paid_up,"Y,C") || surv(t-1)==0.0 || annuity_factor(t-1)<0.0000001 || inlist(life-
>res_basis, "No_Reserve,Perc_Prem"))
    return 0.0;

double tat_shnatiut = 1.0;
if (life->mod_load_in_prem=="N")
    tat_shnatiut = (1 + life->tat_shnatiut_rate/100.);

if(life->prem_lookup_temp=="N") { //level premium
    if (t == life->commence_period_w + 1) {
        return max(net_premium_b(t) - life->prem_curr * tat_shnatiut * life->netprem_max /
100. * life->benefits_curr, 0);
    }
    return net_prem_deficiency_b(life->commence_period_w + 1);
}

//if life->prem_lookup = "Y"
double net_prem = 0.0;
if(t == life->commence_period_w + 1 || (xint(pol_sub_year(t))== 1 && xint(life->pol_month(t)) ==
1)){
    return max(net_premium_b(t) - premium_if_b(t) / surv(t-1) * surv(life->commence_period_w) *
life->netprem_max / 100. * life->benefits_curr, 0);
}
return net_prem_deficiency_b(t-1);

```

**6.1.1.6.1.8 net\_premium\_b**

```

if(t <= life->commence_period_w || t + life->elapsed_months > life->prem_term || inlist(life->paid_up,"Y,C"))
    return 0.0;

if (annuity_factor(t-1)<0.0000001) return 0.0;

if(life->prem_lookup_temp=="N") { //level premium
    if (t == life->commence_period_w + 1) {
        double net_prem = 0.0;
        net_prem = assurance_factor(t-1)/annuity_factor(t-1);
        return net_prem;
    }
    return net_premium_b(life->commence_period_w + 1);
}

//if life->prem_lookup = "Y"
double net_prem = 0.0;
if(t == life->commence_period_w + 1 || (xint(pol_sub_year(t))== 1 && xint(life->pol_month(t)) == 1)){
    net_prem = assurance_factor(t-1)/annuity_factor(t-1);
    return net_prem;
}
return net_premium_b(t-1);

```

**6.1.1.6.1.9 net\_premium\_e**

```

if (t < life->commence_period_w || t >= life->maturity_period_w || inlist(life->paid_up,"Y,C"))
    return 0.0;

return net_premium_b(t+1);

```

**6.1.1.6.1.10 res\_np\_deficiency**

```

if (t < life->commence_period_w || t >= life->maturity_period_w)
    return 0.0;

if (inlist(life->res_basis, "No_Reserve,Perc_Prem"))
    return 0.0;

//If res_basis is "Net_Prem"
double reserve_temp = 0;
double res_up=0.0;
int t_down=0;
double interpol = life->pol_month(t)/12.;
if(xint(life->pol_month(t))==12) {
    reserve_temp = net_prem_deficiency_b(t+1) * annuity_factor(t);
}
else {
    t_down = (xint(life->pol_year(t))-1)*12 - life->elapsed_months;
    res_up = 0.0;
    if(xint(pol_sub_year(t))== xint(life->prem_lookup_freq_temp))
        res_up = 0.0;
    else
        res_up = net_prem_deficiency_b(t_down+12+1) * annuity_factor(t_down+12);

    reserve_temp = (1. - interpol) * (net_prem_deficiency_b(t_down+1) * annuity_factor(t_down))
        + interpol * res_up;
}

```

```

}
reserve_temp = reserve_temp * surv(t);

return reserve_temp;

```

#### 6.1.1.6.1.11 reserve\_basic

```

if (t < life->commence_period_w || t >= life->maturity_period_w || life->paid_up == "C")
    return 0.0;

if (eq(life->res_basis, "No_Reserve"))
    return 0.0;

// for YRT premium return a percentage of annual premium in force
if (eq(life->res_basis, "Perc_Prem")) {
    int yr_ref = 0;
    yr_ref = round((t+life->elapsed_months)/12.,0);
    yr_ref = max(0,yr_ref);

    return life->res_perc_prem[yr_ref] /100. * surv(t)
        * premium_if_e(t) /(1.+life->tat_shnatiut_rate/100.);
}

//if res_basis is "Net_Prem"
double reserve_temp = 0.0;
double res_up=0.0;
int t_down=0;
double interpol = life->pol_month(t)/12.;
if(xint(life->pol_month(t))==12) {
    reserve_temp = vsa(t) - vnp(t);
}
else {
    t_down = (xint(life->pol_year(t))-1)*12 - life->elapsed_months;
    res_up = 0.0;
    if(xint(pol_sub_year(t))== xint(life->prem_lookup_freq_temp))
        res_up = 0.0;
    else
        res_up = vsa(t_down+12)-vnp(t_down+12);

    reserve_temp = (1. - interpol) * (vsa(t_down)-vnp(t_down))
        + interpol * res_up;
}
reserve_temp = reserve_temp * surv(t);

if (eq(life->ben_class,"ltc") && sum_insured_if_b(t)>0.) { // to add reserve i.r.o. paid-up benefit
    double reserve_temp_pup = 0.;
    if(xint(life->pol_month(t))==12)
        reserve_temp_pup = vsa(t);
    else {
        t_down = (xint(life->pol_year(t))-1)*12 - life->elapsed_months;
        res_up = 0.0;
        if(xint(pol_sub_year(t))== xint(life->prem_lookup_freq_temp))
            res_up = 0.0;
        else
            res_up = vsa(t_down+12);
        reserve_temp_pup = (1. - interpol) * vsa(t_down) + interpol * res_up;
    }
    reserve_temp_pup = reserve_temp_pup * surv(t)

```

```

        * sum_insured_if_b_2(t) / sum_insured_if_b(t); // adjust for
different paid-up inforce-si compared to premium paying
        reserve_temp = reserve_temp + reserve_temp_pup;
}

```

```
return reserve_temp;
```

#### 6.1.1.6.1.12 reserve\_basic\_claims

```

if ((!eq(life->ben_class,"phi")&&!eq(life->ben_class,"ltc")||!eq(life->paid_up,"C"))|| !eq(life-
>use_phi_claims_cf,"Y"))
    return NO_AVG;

```

```

if (t <= life->commence_period_w || t > life->maturity_period_w)
    return 0.0;

```

```
int t_start = 0;
```

```

if (eq(life->projection_type,"Rollup"))
    t_start = life->commence_period_w +1;

```

```

if(t==0 && life->paid_up == "C")
    return life->resinforce * life->benefits_curr;

```

```

if (t-t_start >0)
    return claims_inpay_res.sum_of_diagonal(t-t_start);

```

```
return 0;
```

#### 6.1.1.6.1.13 reserve\_risk\_premium

```

if (t < life->commence_period_w || t >= life->maturity_period_w)
    return 0.0;

```

```

double claims_pv = 0.0;
double decrem=0.0;
int i=0, j=0;
double acc_v = 1.0;

```

```

if (eq(life->res_basis, "Net_Prem")) {
    double qx_res_basis = 0.0;
    double qx_projection = 0.0;
    int age_now = xint(life->age_last(t+1));
    if (res_lx(age_now,0) > 0 ) {
        if (life->death_ben_w=="Y")
            qx_res_basis = (1.0 - res_lx(age_now+1,0) / res_lx(age_now,0));
        else {
            life->row_num = age_now;
            if (eq(life->use_uw_date,"Y"))
                decrem = life->decrem_rates_uw_res;
            else
                decrem = life->decrem_rates_res;
            qx_res_basis = decrem *
                life->decrem_mult_res/100. *
                (1.+max(life->health_occ_perc_min,life->health_occ_perc)/100.);
        }
    }
}

```

```

    if(life->death_ben_w=="Y")
        qx_projection = death_rate_dep(t);
    else
        qx_projection =decrem_rate_dep(t);

    if ( qx_projection > 0.0 ){

        if (eq(life->ben_class,"phi") && life->use_phi_claims_cf == "Y" && t - t_start >= 0)
            return claims_inpay_pv(t-t_start,0) / qx_projection * qx_res_basis / 12.0;

        return claims_total(t) / qx_projection * qx_res_basis / 12.0;
    }

    return 0.0;
}

// no reserve
return 0.0;

```

#### 6.1.1.6.1.14 vnp

```

if (t < life->commence_period_w || t >= life->maturity_period_w)
    return 0.0;

return annuity_factor(t) * net_premium_e(t);

```

#### 6.1.1.6.1.15 vsa

```

if (t < life->commence_period_w || t >= life->maturity_period_w)
    return 0.0;

return assurance_factor(t);

```

#### 6.1.1.6.1.16 surv

```

if (t < life->commence_period_w || t > life->maturity_period_w)
    return NO_AVG;

if (life->decrements_apply=="N")
    return 1.0;

if (inlist(life->ben_class,"phi,ltc") && eq(life->paid_up,"C") && life->use_phi_claims_cf == "Y"){
    if(t<=0)
        return 1.0;
    if (t-t_start>=0)
        return claims_inpay_rate.sum_of_diagonal(t-t_start);
}

if (t == 0) // At start
    return 1.0;

if (t > 0 && fabs(surv(t-1)) < .0000001)
    // No surv in previous period
    return NO_AVG;

if (t > 0)
    return surv(t-1) * surv_per(t);

// t < 0

```

```

if (life->gross_up_historic=="Y"){
  if(fabs(surv(t+1))<.0000001)
    return NO_AVG;
  else
    return surv(t+1) / surv_per(t+1);
}
return surv(t+1);

```

#### 6.1.1.6.1.17 **surv\_2**

```

if (t < life->commence_period_w || t > life->maturity_period_w)
  return NO_AVG;

if (t > 0 && eq(life->ben_class,"dd"))

  return surv_2(t-1)
    * (1. - death_rate(t))
    * (1. - decrem_rate(t))
    * (1. - lapse_rate(t)*life->secondary_lapse_mult/100.)
    + surv(t-1) * decrem_rate_dep(t)* life->secondary_prop_continue/100.;

if (t > 0 && eq(life->ben_class,"ltc")) {
  double check = 0;
  if (!eq(pup_ltc_key,"0"))
    check = pup_ltc_tbl;
  if (check > 0)
    return surv_2(t-1)
      * (1. - death_rate(t))
      * (1. - decrem_rate(t))
      + surv(t-1) * lapse_rate(t);
}

// for t <= 0, surv_2 remains 0 because not after a dd claim
return 0.;

```

#### 6.1.1.6.1.18 **surv\_2\_no\_dec**

```

if (t < life->commence_period_w || t > life->maturity_period_w)
  return NO_AVG;

if (t > 0 && eq(life->ben_class,"ltc")) {
  double check = 0;
  if (!eq(pup_ltc_key,"0"))
    check = pup_ltc_tbl;
  if (check > 0)
    return surv_2_no_dec(t-1)
      * (1. - death_rate(t))
      + surv_no_dec(t-1) * lapse_rate(t);
}

// for t <= 0, surv_2 remains 0 because not after a dd claim
return 0.;

```

#### 6.1.1.6.1.19 **surv\_no\_dec**

```

if (t < life->commence_period_w || t > life->maturity_period_w)
  return NO_AVG;

```

```

if (life->decrements_apply=="N")
  return 1.0;

if (inlist(life->ben_class,"phi,ltc") && eq(life->paid_up,"C") && life->use_phi_claims_cf == "Y"){
  if(t<=0)
    return 1.0;
  if (t-t_start>=0)
    return claims_inpay_rate.sum_of_diagonal(t-t_start);
}

if (t == 0) // At start
  return 1.0;

if (t > 0 && fabs(surv_no_dec(t-1)) < .0000001)
  // No surv in previous period
  return NO_AVG;

if (t > 0)
  return surv_no_dec(t-1) * surv_per_no_dec(t);

// t < 0

if (life->gross_up_historic=="Y"){
  if(fabs(surv_no_dec(t+1))<.0000001)
    return NO_AVG;
  else
    return surv_no_dec(t+1) / surv_per_no_dec(t+1);
}
return surv_no_dec(t+1);

```

#### 6.1.1.6.1.20 **surv\_no\_dth**

```

if (t < life->commence_period_w || t > life->maturity_period_w)
  return NO_AVG;

if (life->decrements_apply=="N")
  return 1.0;

if (inlist(life->ben_class,"phi,ltc") && eq(life->paid_up,"C") && life->use_phi_claims_cf == "Y"){
  if(t<=0)
    return 1.0;
  if (t-t_start>=0)
    return claims_inpay_rate.sum_of_diagonal(t-t_start);
}

if (t == 0) // At start
  return 1.0;

if (t > 0 && fabs(surv_no_dth(t-1)) < .0000001)
  // No surv in previous period
  return NO_AVG;

if (t > 0)
  return surv_no_dth(t-1) * surv_per_no_dth(t);

// t < 0

```



```

if (life->gross_up_historic=="Y"){
  if(fabs(surv_no_dth(t+1))<.0000001)
    return NO_AVG;
  else
    return surv_no_dth(t+1) / surv_per_no_dth(t+1);
}
return surv_no_dth(t+1);

```

#### 6.1.1.6.1.21 **surv\_per**

```

if (t <= life->commence_period_w || t > life->maturity_period_w)
  return NO_AVG;

if (life->decrements_apply=="N")
  return 1.0;

if (t <= 0 && life->gross_up_historic=="N")
  return 1.0;

if(life->death_ben_w=="Y")
  return (1. - death_rate(t)) * (1. - lapse_rate(t));
else {
  if (inlist(life->ben_class,"phi,ltc") && eq(life->paid_up,"C") && life->use_phi_claims_cf ==
"Y"){
    if(t<=0)
      return 1.0;
    if (surv(t-1)==0)
      return 0.;
    return surv(t)/surv(t-1);
  }
  else{
    if (eq(life->ben_class,"phi"))
      return (1. - death_rate(t)) * (1. - lapse_rate(t));
    }
    return (1. - death_rate(t)) * (1. - lapse_rate(t)) * (1 - decrem_rate(t));
  }
}

```

#### 6.1.1.6.1.22 **surv\_per\_no\_dec**

```

if (t <= life->commence_period_w || t > life->maturity_period_w)
  return NO_AVG;

if (life->decrements_apply=="N")
  return 1.0;

if (t <= 0 && life->gross_up_historic=="N")
  return 1.0;

if(life->death_ben_w=="Y")
  return (1. - death_rate(t)) * (1. - lapse_rate(t));
else {
  if (inlist(life->ben_class,"phi,ltc") && eq(life->paid_up,"C") && life->use_phi_claims_cf ==
"Y"){
    if(t<=0)
      return 1.0;
    if (surv_no_dec(t-1)==0)
      return 0.;
    return surv_no_dec(t)/surv_no_dec(t-1);
  }
}

```

```

        return (1. - death_rate(t)) * (1. - lapse_rate(t));
    }

```

#### 6.1.1.6.1.23 surv\_per\_no\_dth

```

if (t <= life->commence_period_w || t > life->maturity_period_w)
    return NO_AVG;

if (life->decrements_apply=="N")
    return 1.0;

if (t <= 0 && life->gross_up_historic=="N")
    return 1.0;

if(life->death_ben_w=="Y")
    return (1. - lapse_rate(t));
else {
    if (inlist(life->ben_class,"phi,ltc") && eq(life->paid_up,"C") && life->use_phi_claims_cf ==
"Y"){
        if(t<=0)
            return 1.0;
        if (surv_no_dth(t-1)==0)
            return 0.;
        return surv_no_dth(t)/surv_no_dth(t-1);
    }
    else{
        if (eq(life->ben_class,"phi"))
            return (1. - lapse_rate(t));
        }
        return (1. - lapse_rate(t)) * (1 - decrem_rate(t));
    }
}

```

#### 6.1.1.6.1.24 death\_rate

```

if (t <= life->commence_period_w || t > life->maturity_period_w)
    return NO_AVG;

if (t <= 0 && life->gross_up_historic=="N")
    return 0.0;

int pol_yr = max(xint(life->pol_year(t) + int(life->elapsed_months_extra/12.)),1);

// Assume all lives die at omega age (for death benefits)
if (life->age_last(t) >= life->omega_age_w && (life->death_ben_w=="Y"))
    return 1.0;

double rate = 0.0;

// set column to allow for selection in mortality table
life->col_dth = 0;
if (life->mort_sel_status=="Y")
    life->col_dth = min(pol_yr, life->select_periods);
else
    life->col_dth = life->select_periods;

double mult_factor = life->mort_mult/100.;
if (life->age_last(t) > life->mort_mult_end_age && !eq(life->ben_class,"ltc")) // adjust mortality
multiplier after age 75 to gradually reach 100%

```

```

        mult_factor = 1. + (mult_factor - 1.) * (life->omega_age_w - life->age_last(t))/(life-
->omega_age_w - life->mort_mult_end_age);

life->death_rate_row_key =life->age_last(t) - life->col_dth+1;
life->col_dth = life->col_dth; // reset lookup variable to avoid mutating lookup error
rate = life->death_rates_tbl;
rate = rate * mult_factor;

// Only apply the medical/occupational loading if there is a death benefit
if(life->death_ben_w=="Y")
    rate = rate * (1+max(life->health_occ_perc_min,life->health_occ_perc)/100.);

//***** add margin *****
if (life->margin_add=="Y")
    rate = rate * (1+life->margin_mort_pc/100);

//Margin for catastrophe
if (life->margin_add_cat == "Y" && life->death_ben_w == "Y" ){//Only apply to savings and death
risk

    double m_cat = 0;

    if (life->proj_year(t) == 1)
        m_cat = life->cat_risk;

    rate = rate + m_cat;
}

//***** convert to monthly *****

rate = max(0.0, min(1.0, rate));

return (1. - pow(1. - rate, 1./12.)); // convert to monthly

```

#### 6.1.1.6.1.25 death\_rate\_dep

```

if(life->death_ben_w=="Y")
    return death_rate(t);

//benefits for other causes of decrement
return death_rate(t) * (1 - 0.5 * decrem_rate(t));

```

#### 6.1.1.6.1.26 lapse\_rate

```

if (t <= life->commence_period_w || t > life->maturity_period_w)
    return NO_AVG;

if (t<=0 && life->gross_up_historic=="N")
    return NO_AVG;

if (eq(life->ben_class,"ltc") && life->paid_up=="Y")
    return NO_AVG;

if (eq(life->ben_class,"phi") && life->paid_up=="C")
    return NO_AVG;

double Mass_rate = 0;

```

```
if(life->lapse_force_month >0 && life->lapse_force_month == t)
    Mass_rate = life->lapse_force_rate;

if (life->submodel != "TERM")
    return 0;

double rate_basis1 = 0;
double rate_basis2 = 0;
double rate_rider = 0;
double rate = 0;

if(life->rider_ind == 1){ //this cover is rider

    if (life->savings_pol == "Y"){ //this cover is rider to savings policy

        life->lapse_type_col_key = "Surrender";
        life->lapse_expos_col_key = "premium";
        life->tarif_spec_row_key= xstring(life->tarif); //added to avoid mutating lookup term
error
        rate_basis1= life->lapse_rate_im/ 100.0;

        life->lapse_type_col_key = "PUP";
        life->lapse_expos_col_key = "premium";
        life->tarif_spec_row_key= xstring(life->tarif); //added to avoid mutating lookup term
error
        rate_basis2= life->lapse_rate_im/ 100.0;

    }
    else { // this cover is rider to non-savings policy
        life->lapse_type_col_key = "Lapse";
        life->lapse_expos_col_key = "premium";
        life->tarif_spec_row_key= xstring(life->tarif); //added to avoid mutating lookup term
error
        rate_basis1= life->lapse_rate_im/ 100.0;
    }

    // extra lapses for the rider
    life->lapse_type_col_key = "Lapse";
    life->lapse_expos_col_key = "premium";
    life->tarif_spec_row_key= xstring(life->tarif); //added to avoid mutating lookup term error
    rate_rider= life->lapse_rider_other/ 100.0;
}

if(life->rider_ind != 1){ //this cover is main cover of the policy
    life->lapse_type_col_key = "Lapse";
    life->lapse_expos_col_key = "premium";
    life->tarif_spec_row_key= xstring(life->tarif); //added to avoid mutating lookup term error
    rate_basis1= life->lapse_rate_im/ 100.0;
}

rate = ((rate_basis1 + rate_basis2)*life->lapse_factor_proj/100.0 + rate_rider*life->lapse_factor_proj_rider/100.0) * life->lapse_factor(t) ;

double margin = 0.;
if(life->margin_add=="Y")
    margin = life->margin_lapses;

rate = min( 0.999,rate * (1 + margin/100.));
```

```

rate = 1. - pow((1. - rate), 1./12.);
rate = min(rate * (1-Mass_rate) + Mass_rate,0.999);

return rate;

```

#### 6.1.1.6.1.27      **decrem\_rate**

```

if (t <= life->commence_period_w || t > life->maturity_period_w)
    return NO_AVG;

if (life->death_ben_w=="Y")
    return 0;

if (t <= 0 && life->gross_up_historic=="N")
    return 0.0;

double rate = 0.0;
life->row_num = life->age_last(t);
if (eq(life->use_uw_date, "Y"))
    rate = life->decrem_rates_uw;
else
    rate = life->decrem_rates;
rate = rate * life->decrem_mult / 100. *(1+max(life->health_occ_perc_min,life-
>health_occ_perc)/100.);
rate = rate * life->claims_multiplier[xint(life->pol_year(t) + round(life-
>elapsed_months_extra/12.,0))]/100.; //worsening of rate with years
rate = rate * claims_inflation(t);
rate = min(rate,0.999);

//***** add margin *****
if (life->margin_add=="Y"){
    double temp_1st_add = 0.;
    if(t>0-wp_phi && t<13-wp_phi)
        temp_1st_add = life->margin_1styr_clms_add;
    rate = rate * (1+life->margin_claims/100. + temp_1st_add /100. );
}

rate = (1. - pow(1. - rate, 1./12.)); // convert to monthly
return rate;

```

#### 6.1.1.6.1.28      **decrem\_rate\_dep**

```

if(life->death_ben_w=="Y")
    return 0;

//benefits for other causes of decrement
return decrem_rate(t) * (1 - 0.5 * death_rate(t));

```

#### 6.1.1.6.1.29      **pol\_fee**

```

if(t <= life->commence_period_w || (t + life->elapsed_months) > life->prem_term)
    return 0.0;

if (inlist(life->paid_up,"Y,C"))
    return 0.0;

if (fmod(xint(life->pol_month(t-1)), xint(12. / life->prem_freq))!=0)
    return 0.0; //not a premium due date

```

```

double tat_shnatiut = 1.0;
if (life->mod_load_in_prem=="N")
    tat_shnatiut = (1 + life->tat_shnatiut_rate/100.);

double temp = 1.0;
if (eq(life->ben_class,"dd") && (surv(t-1)>0.0))
    temp = (surv_2(t-1)+surv(t-1))/surv(t-1); // to calculate policy fee for secondary lives

return life->policy_fee_if
    * life->policies_b(t)
    / life->prem_freq
    * (1-life->pol_fee_disc_perc/100.)
    * temp
    * tat_shnatiut;

```

#### 6.1.1.6.1.30 prem\_gross\_no\_scen

```

if (t < life->commence_period_w || t > life->maturity_period_w || eq(life->paid_up,"C"))
    return 0.0;

if (fmod(xint(life->pol_month(t-1)), xint(12. / life->prem_freq))!=0)
    return 0.0; //not a premium due date

return premium(t) + pol_fee(t) - premium_disc_no_scen(t);

```

#### 6.1.1.6.1.31 premium

```

if (t < life->commence_period_w || t > life->maturity_period_w || eq(life->paid_up,"C"))
    return 0.0;

if (fmod(xint(life->pol_month(t-1)), xint(12. / life->prem_freq))!=0)
    return 0.0; //not a premium due date

if (eq(life->ben_class,"dd"))
    return (premium_if_b(t) + premium_if_b_2(t)) / life->prem_freq;

//else
return premium_if_b(t) / life->prem_freq;

```

#### 6.1.1.6.1.32 premium\_disc

```

if(eq(life->paid_up,"C"))
    return 0.;

double temp_total = 0.0;
double temp = 0.0;
double temp1 = 0.0;
double temp2 = 0.0;
double temp3 = 0.0;
double temp4 = 0.0;
double temp5 = 0.0;

//log_strm<<"prem_disc_perc term3: "<<life->prem_disc_perc<<endl;

if (t <= life->prem_disc_month)
    temp = life->prem_disc_perc/100.; // Premium discount

if (t <= life->prem_disc_month_2)

```

```

    temp = temp + life->prem_disc_perc_2/100.; // Premium discount

if (life->dump_vars == "Y")
    log_strm<<"Temp at time "<<t<<": "<<life->prem_disc_perc/100.<<"; "<<life->prem_disc_perc_2/100.<<endl;

if ( life-> prem_disc_step > 0){
    if (t <= life->prem_disc_step1_m)
        temp1 = life->prem_disc_step1_r/100.; // Premium discount
    if (t <= life->prem_disc_step2_m)
        temp2 = life->prem_disc_step2_r/100.; // Premium discount
    if (t <= life->prem_disc_step3_m)
        temp3 = life->prem_disc_step3_r/100.; // Premium discount
    if (t <= life->prem_disc_step4_m)
        temp4 = life->prem_disc_step4_r/100.; // Premium discount
    if (t <= life->prem_disc_step5_m)
        temp5 = life->prem_disc_step5_r/100.; // Premium discount
}

if (life->dump_vars == "Y")
    log_strm<<"Disc steps at time "<<t<<": "<<temp1<<"; "<<temp2<<"; "<<temp3<<"; "<<temp4<<"; "<<temp5<<endl;

temp_total = min(temp + temp1 + temp2 + temp3 + temp4 + temp5 , 1.);

if (life->dump_vars == "Y")
    log_strm<<"Total discount at time "<<t<<": "<<temp_total<<endl;

double margin = 0.0;

if (life->margin_add_discount == "Y")
    margin = prem_disc_scenario/100.;

temp_total = min(temp_total + margin, 1.);

double Shimur = 0.0;

if (life->prem_disc_shimur_flag == "Y")
    Shimur = Shimur + life->prem_disc_shimur_rate(t);

temp_total = min(temp_total + Shimur, 1.);

return temp_total * premium(t);

```

#### 6.1.1.6.1.33 premium\_disc\_no\_scen

```

if(eq(life->paid_up,"C"))
    return 0.;

double temp_total = 0.0;
double temp = 0.0;
double temp1 = 0.0;
double temp2 = 0.0;
double temp3 = 0.0;
double temp4 = 0.0;
double temp5 = 0.0;

```

```

if (t <= life->prem_disc_month)
    temp = life->prem_disc_perc/100.; // Premium discount

if (t <= life->prem_disc_month_2)
    temp = temp + life->prem_disc_perc_2/100.; // Premium discount

if (life->dump_vars == "Y")
    log_strm<<"Temp at time "<<t<<": "<<life->prem_disc_perc/100.<<"; "<<life->
    prem_disc_perc_2/100.<<endl;

if ( life-> prem_disc_step > 0){
    if (t <= life->prem_disc_step1_m)
        temp1 = life->prem_disc_step1_r/100.; // Premium discount
    if (t <= life->prem_disc_step2_m)
        temp2 = life->prem_disc_step2_r/100.; // Premium discount
    if (t <= life->prem_disc_step3_m)
        temp3 = life->prem_disc_step3_r/100.; // Premium discount
    if (t <= life->prem_disc_step4_m)
        temp4 = life->prem_disc_step4_r/100.; // Premium discount
    if (t <= life->prem_disc_step5_m)
        temp5 = life->prem_disc_step5_r/100.; // Premium discount
}

if (life->dump_vars == "Y")
    log_strm<<"Disc steps at time "<<t<<": "<<temp1<<"; "<<temp2<<"; "<<temp3<<"; "<<temp4<<";
    "<<temp5<<endl;

temp_total = min(temp + temp1 + temp2 + temp3 + temp4 + temp5 , 1.);

if (life->dump_vars == "Y")
    log_strm<<"Total discount at time "<<t<<": "<<temp_total<<endl;

return temp_total * premium(t);

```

#### 6.1.1.6.1.34 premium\_disc\_no\_shimur

```

if(eq(life->paid_up,"C"))
    return 0.;

double temp_total = 0.0;
double temp = 0.0;
double temp1 = 0.0;
double temp2 = 0.0;
double temp3 = 0.0;
double temp4 = 0.0;
double temp5 = 0.0;

if (t <= life->prem_disc_month)
    temp = life->prem_disc_perc/100.; // Premium discount

if (t <= life->prem_disc_month_2)
    temp = temp + life->prem_disc_perc_2/100.; // Premium discount

if ( life-> prem_disc_step > 0){
    if (t <= life->prem_disc_step1_m)

```



```

    temp1 = life->prem_disc_step1_r/100.; // Premium discount
    if (t <= life->prem_disc_step2_m)
        temp2 = life->prem_disc_step2_r/100.; // Premium discount
    if (t <= life->prem_disc_step3_m)
        temp3 = life->prem_disc_step3_r/100.; // Premium discount
    if (t <= life->prem_disc_step4_m)
        temp4 = life->prem_disc_step4_r/100.; // Premium discount
    if (t <= life->prem_disc_step5_m)
        temp5 = life->prem_disc_step5_r/100.; // Premium discount
}

if (life->dump_vars == "Y")
    log_strm<<"Disc steps at time "<<t<<": "<<temp1<< ", "<<temp2<< ", "<<temp3<< ", "<<temp4<< ",
"<<temp5<<endl;

temp_total = min(temp + temp1 + temp2 + temp3 + temp4 + temp5 , 1.);

if (life->dump_vars == "Y")
    log_strm<<"Total discount at time "<<t<<": "<<temp_total<<endl;

double margin = 0.0;

if (life->margin_add_discount == "Y")
    margin = prem_disc_scenario/100.;

temp_total = min(temp_total + margin, 1.);

return temp_total * premium(t);

```

#### 6.1.1.6.1.35 premium\_gross

```

if (t < life->commence_period_w || t > life->maturity_period_w)
    return 0.0;

if (eq(life->paid_up,"C"))
    return 0.0;

if (fmod(xint(life->pol_month(t-1)), xint(12. / life->prem_freq))!=0)
    return 0.0; //not a premium due date

return premium(t) + pol_fee(t) - premium_disc(t);

```

#### 6.1.1.6.1.36 claims\_inflation

```

if (t < 1 || t > life->maturity_period_w)
    return 1.0;

return min(life->claim_inflation_max/100.,claims_inflation(t-1) * (1.0 + claims_inflation_mthly));

```

#### 6.1.1.6.1.37 claims\_inpay

```

if (t <= life->commence_period_w || t > life->maturity_period_w)
    return 0.0;

if ((!eq(life->ben_class,"phi")&&!eq(life->ben_class,"ltc"))||!eq(life->paid_up,"C"))|| !eq(life-
>use_phi_claims_cf,"Y"))
    return NO_AVG;

if (surv(t-1) < 0.000001)

```

```

    return 0.0;

if(t-t_start >=0)
    return claims_inpayment.sum_of_diagonal(t-t_start)
        + premium_gross(t)/surv(t-1) * claims_inpay_rate.sum_of_diagonal(t-t_start);

return 0;

```

#### 6.1.1.6.1.38 claims\_inpay\_other

```

if (t <= life->commence_period_w || t > life->maturity_period_w)
    return 0.0;

if ((!eq(life->ben_class,"phi")&&!eq(life->ben_class,"ltc")||!eq(life->paid_up,"C"))|| !eq(life-
>use_phi_claims_cf,"Y"))
    return NO_AVG;

if (surv(t-1) < 0.000001)
    return 0.0;

return claims_inpay(t)-claims_inpay_q1(t)-claims_inpay_q2(t)-claims_inpay_q3(t)-claims_inpay_q4(t);

```

#### 6.1.1.6.1.39 claims\_inpay\_q1

```

if (t <= life->commence_period_w || t > life->maturity_period_w)
    return 0.0;

if ((!eq(life->ben_class,"phi")&&!eq(life->ben_class,"ltc")||!eq(life->paid_up,"C"))|| !eq(life-
>use_phi_claims_cf,"Y"))
    return NO_AVG;

if (surv(t-1) < 0.000001)
    return 0.0;

double temp1 = 0;
double temp2 = 0;

if (t-1>=0){
    temp1 = claims_inpayment(1,t-1);
    temp2 = claims_inpay_rate(1,t-1);
}

if (t-2>=0){
    temp1 = temp1 + claims_inpayment(2,t-2);
    temp2 = temp2 + claims_inpay_rate(2,t-2);
}

if (t-3>=0){
    temp1 = temp1 + claims_inpayment(3,t-3);
    temp2 = temp2 + claims_inpay_rate(3,t-3);
}

if(t-t_start >=0)
    return temp1
        + premium_gross(t)/surv(t-1) * temp2;

return 0;

```

**6.1.1.6.1.40 claims\_inpay\_q2**

```

if (t <= life->commence_period_w || t > life->maturity_period_w)
  return 0.0;

if ((!eq(life->ben_class,"phi")&&!eq(life->ben_class,"ltc")||!eq(life->paid_up,"C"))|| !eq(life-
>use_phi_claims_cf,"Y"))
  return NO_AVG;

if (surv(t-1) < 0.000001)
  return 0.0;

double temp1 = 0;
double temp2 = 0;

if (t-4>=0){
  temp1 = claims_inpayment(4,t-4);
  temp2 = claims_inpay_rate(4,t-4);
}

if (t-5>=0){
  temp1 = temp1 + claims_inpayment(5,t-5);
  temp2 = temp2 + claims_inpay_rate(5,t-5);
}

if (t-6>=0){
  temp1 = temp1 + claims_inpayment(6,t-6);
  temp2 = temp2 + claims_inpay_rate(6,t-6);
}

if(t-t_start >=0)
  return temp1
  + premium_gross(t)/surv(t-1) * temp2;

return 0;

```

**6.1.1.6.1.41 claims\_inpay\_q3**

```

if (t <= life->commence_period_w || t > life->maturity_period_w)
  return 0.0;

if ((!eq(life->ben_class,"phi")&&!eq(life->ben_class,"ltc")||!eq(life->paid_up,"C"))|| !eq(life-
>use_phi_claims_cf,"Y"))
  return NO_AVG;

if (surv(t-1) < 0.000001)
  return 0.0;

double temp1 = 0;
double temp2 = 0;

if (t-7>=0){
  temp1 = claims_inpayment(7,t-7);
  temp2 = claims_inpay_rate(7,t-7);
}

if (t-8>=0){
  temp1 = temp1 + claims_inpayment(8,t-8);

```

```

        temp2 = temp2 + claims_inpay_rate(8,t-8);
    }

    if (t-9>=0){
        temp1 = temp1 + claims_inpayment(9,t-9);
        temp2 = temp2 + claims_inpay_rate(9,t-9);
    }

    if(t-t_start >=0)
        return temp1
            + premium_gross(t)/surv(t-1) * temp2;

    return 0;

```

#### 6.1.1.6.1.42 claims\_inpay\_q4

```

if (t <= life->commence_period_w || t > life->maturity_period_w)
    return 0.0;

if ((!eq(life->ben_class,"phi")&&!eq(life->ben_class,"ltc")||!eq(life->paid_up,"C"))|| !eq(life-
>use_phi_claims_cf,"Y"))
    return NO_AVG;

if (surv(t-1) < 0.000001)
    return 0.0;

double temp1 = 0;
double temp2 = 0;

if (t-10>=0){
    temp1 = claims_inpayment(10,t-10);
    temp2 = claims_inpay_rate(10,t-10);
}

if (t-11>=0){
    temp1 = temp1 + claims_inpayment(11,t-11);
    temp2 = temp2 + claims_inpay_rate(11,t-11);
}

if (t-12>=0){
    temp1 = temp1 + claims_inpayment(12,t-12);
    temp2 = temp2 + claims_inpay_rate(12,t-12);
}

if(t-t_start >=0)
    return temp1
        + premium_gross(t)/surv(t-1) * temp2;

return 0;

```

#### 6.1.1.6.1.43 claims\_rate\_per

```

if (t<= life->commence_period_w || t > life->maturity_period_w)
    return 0.0;

if((eq(life->ben_class, "phi") || (eq(life->ben_class, "ltc") && eq(life->paid_up, "C")))) && life-
>use_phi_claims_cf == "Y" && t-t_start>=0)
    return claims_inpay_rate.sum_of_diagonal(t-t_start);

```

```

if(life->death_ben_w == "Y")
    return death_rate_dep(t) * surv(t-1);

return decem_rate_dep(t) * surv(t-1); //Other

```

#### 6.1.1.6.1.44 claims\_rate\_per\_other

```

if (t <= life->commence_period_w || t > life->maturity_period_w)
    return 0.0;

if ((!eq(life->ben_class,"phi")&&!eq(life->ben_class,"ltc"))||!eq(life->paid_up,"C"))|| !eq(life-
>use_phi_claims_cf,"Y"))
    return NO_AVG;

if (surv(t-1) < 0.000001)
    return 0.0;

return claims_rate_per(t)-claims_rate_per_q1(t)-claims_rate_per_q2(t)-claims_rate_per_q3(t)-
claims_rate_per_q4(t);

```

#### 6.1.1.6.1.45 claims\_rate\_per\_q1

```

if (t <= life->commence_period_w || t > life->maturity_period_w)
    return 0.0;

if ((!eq(life->ben_class,"phi")&&!eq(life->ben_class,"ltc"))||!eq(life->paid_up,"C"))|| !eq(life-
>use_phi_claims_cf,"Y"))
    return NO_AVG;

if (surv(t-1) < 0.000001)
    return 0.0;

double temp1 = 0;

if (t-1>=0){
    temp1 = claims_inpay_rate(1,t-1);
}

if (t-2>=0){
    temp1 = temp1 + claims_inpay_rate(2,t-2);
}

if (t-3>=0){
    temp1 = temp1 + claims_inpay_rate(3,t-3);
}

if(t-t_start >=0)
    return temp1;

return 0;

```

**6.1.1.6.1.46 claims\_rate\_per\_q2**

```
if (t <= life->commence_period_w || t > life->maturity_period_w)
    return 0.0;

if ((!eq(life->ben_class,"phi")&&!eq(life->ben_class,"ltc")||!eq(life->paid_up,"C"))|| !eq(life-
>use_phi_claims_cf,"Y"))
    return NO_AVG;

if (surv(t-1) < 0.000001)
    return 0.0;

double temp1 = 0;

if (t-4>=0){
    temp1 = claims_inpay_rate(4,t-4);
}

if (t-5>=0){
    temp1 = temp1 + claims_inpay_rate(5,t-5);
}

if (t-6>=0){
    temp1 = temp1 + claims_inpay_rate(6,t-6);
}

if(t-t_start >=0)
    return temp1;

return 0;
```

**6.1.1.6.1.47 claims\_rate\_per\_q3**

```
if (t <= life->commence_period_w || t > life->maturity_period_w)
    return 0.0;

if ((!eq(life->ben_class,"phi")&&!eq(life->ben_class,"ltc")||!eq(life->paid_up,"C"))|| !eq(life-
>use_phi_claims_cf,"Y"))
    return NO_AVG;

if (surv(t-1) < 0.000001)
    return 0.0;

double temp1 = 0;

if (t-7>=0){
    temp1 = claims_inpay_rate(7,t-7);
}

if (t-8>=0){
    temp1 = temp1 + claims_inpay_rate(8,t-8);
}
```

```

}

if (t-9>=0){

    temp1 = temp1 + claims_inpay_rate(9,t-9);
}

if(t-t_start >=0)
    return temp1;

return 0;

```

#### 6.1.1.6.1.48 claims\_rate\_per\_q4

```

if (t <= life->commence_period_w || t > life->maturity_period_w)
    return 0.0;

if ((!eq(life->ben_class,"phi")&&(!eq(life->ben_class,"ltc"))||!eq(life->paid_up,"C"))|| !eq(life-
>use_phi_claims_cf,"Y"))
    return NO_AVG;

if (surv(t-1) < 0.000001)
    return 0.0;

double temp1 = 0;

if (t-10>=0){

    temp1 = claims_inpay_rate(10,t-10);
}

if (t-11>=0){

    temp1 = temp1 + claims_inpay_rate(11,t-11);
}

if (t-12>=0){

    temp1 = temp1 + claims_inpay_rate(12,t-12);
}

if(t-t_start >=0)
    return temp1;

return 0;

```

#### 6.1.1.6.1.49 claims\_total

```

if (t <= life->commence_period_w || t > life->maturity_period_w)
    return 0.0;

if ((eq(life->ben_class,"phi") || (eq(life->ben_class,"ltc") && eq(life->paid_up,"C"))) && life-
>use_phi_claims_cf == "Y")
    return claims_inpay(t);

if(life->death_ben_w=="Y")
    return death_rate(t) * sum_insured_if_b(t);

```

```

double margin = 0.0;
if (life->prod_code == "phi-mitriya" && life->margin_add == "Y")
    margin = life->margin_claim_cost_mitriya;

if (eq(life->ben_class,"dd")||eq(life->ben_class,"ltc"))
    return decrem_rate_dep(t) * sum_insured_if_b(t)
        + decrem_rate_dep(t) * sum_insured_if_b_2(t);
else
    return decrem_rate_dep(t) * sum_insured_if_b(t) * (1 + margin/100);

```

#### 6.1.1.6.1.50 premium\_if\_b

```

if(t <= life->commence_period_w || (t + life->elapsed_months) > life->prem_term)
    return 0.0;

if(inlist(life->paid_up,"Y,C"))
    return 0.0;

double prem_temp = life->prem_curr;

double tat_shnatiut = 1.0;
if (life->mod_load_in_prem=="N")
    tat_shnatiut = (1 + life->tat_shnatiut_rate/100.);

if (life->prem_curr_changed == "N")
    prem_temp = prem_temp - life->policy_fee_if* life->policies_curr/max(1.0,life->benefits_curr);

// Calc increase for this month
double si_inc_pct = 0.0;
if (xint(life->pol_month(t)) == 1 && (t>0))
    si_inc_pct = life->sum_ins_inc(t) / 100.;

if(life->prem_lookup_temp=="N") { //level premium
    if (t == 1)
        return prem_temp * life->benefits_curr * (1+si_inc_pct) * tat_shnatiut;
    if (t > 1)
        return premium_if_b(t-1) * (1+si_inc_pct)* surv_per(t-1);

    // t<1
    if (surv_per(t) == 0)
        return 0;
    else
        return premium_if_b(t+1) / surv_per(t);
}

double prate = 0.0;
double factor = 0.0;

if (t == 0){
    if (eq(life->prod_code,"mrtg-y") || life->prem_curr_changed == "Y")
        return prem_temp * life->benefits_curr * surv(t-1); /* check why no tat_shnatiut
here? */
    else
        return prem_temp * life->benefits_curr * surv(t-1) * tat_shnatiut;
}

```



```

if (xint(pol_sub_year(t)) == 1 && xint(life->pol_month(t)) == 1){
  life->tarif_spec_row_key= xstring(life->tarif);
  factor = atoi(life->prem_factor);

  life->charge_rate_tt_col=xstring(life->age_last(t));
  prate = life->prem_if_rates;

  // apply prem factor to adjust phi rates for murchav, PHI only at the moment
  if (life->use_tarif_spec_prem == "Y")
    prate=prate * factor/100.;

  prate = prate*(1+ life->health_occ_perc/100.);
  if (inlist(life->ben_class,"ltc,phi"))
    prate = prate * life->sum_ins_curr * life->benefits_b_prm(t)/life->prem_rate_scale_w
      * life->sum_ins_inc_acc(t);
  else
    prate = prate * sum_insured_if_b(t)/life->prem_rate_scale_w;

  tat_shnatiut = (1 + life->tat_shnatiut_rate/100.);

  return prate * tat_shnatiut;
}

if (t<0){
  if (life->gross_up_historic=="N" || (surv_per(t+1)<0.000001))
    return premium_if_b(t+1);
  else
    return premium_if_b(t+1) / surv_per(t+1);
}
// t>0
return premium_if_b(t-1) * surv_per(t-1);

```

#### 6.1.1.6.1.51 premium\_if\_b\_2

```

if(t <= life->commence_period_w || (t + life->elapsed_months) > life->prem_term)
  return 0.0;

if (eq(life->ben_class,"dd") && (surv(t-1) >0.0))
  return premium_if_b(t) * surv_2(t-1)/surv(t-1);

return 0.;

```

#### 6.1.1.6.1.52 premium\_if\_e

```

if (t < life->commence_period_w || (t + life->elapsed_months) >= life->prem_term)
  return 0.0;

if (inlist(life->paid_up,"Y,C"))
  return 0.0;

if (t<0 && t == life->commence_period_w )
  return 0.0;

return premium_if_b(t+1);

```

#### 6.1.1.6.1.53 sum\_at\_risk\_if

```

// Sum at risk at end of period t for period t to t+1

```

```
if (t < life->commence_period_w || t + life->elapsed_months >= life->benefit_term)
    return 0.0;
```

```
return max(sum_insured_if_b(t+1) - reserve_basic(t), 0.);
```

#### 6.1.1.6.1.54 sum\_insured

```
if (t < life->commence_period_w || t > life->maturity_period_w)
    return 0.0;
```

```
double si_inc_pct = 0.0;
```

```
if(!eq(life->paid_up, "C") && (xint(life->pol_month(t)) == 1) && (t > 0))
    si_inc_pct = life->sum_ins_inc(t) / 100.;
```

```
if(inlist(life->ben_class, "fib, phi, ltc")){//use claims cost
```

```
    double mult = 1. ;
```

```
        if(eq(life->ben_class, "FIB")) // calculate claims cost for FIB
```

```
            mult = (1 - pow(1+int_rate_res_mthly, t-life->maturity_period_w-1)) /
```

```
int_rate_res_mthly;
```

```
        else {
```

```
            if ((eq(life->ben_class, "phi") || (eq(life->ben_class, "ltc") && eq(life->paid_up, "C")) && life->use_phi_claims_cf == "Y"))
                mult=1.;
```

```
            else {
```

```
                if (t > life->commence_period_w) // claims cost from table
                    mult = claims_cost_factors(xint(life->age_last(t)), sexcode);
```

```
            }
        }
    }
    else // (t= commence_period)
        mult = claims_cost_factors(xint(life->age_last(t+1)), sexcode); //age last not
```

```
defined at comm. period
    }
} // end else
return life->sum_ins_curr * life->benefits_curr * mult * life->sum_ins_inc_acc(t);
}
```

```
if (t == 0)
```

```
    return life->sum_ins_curr * life->benefits_curr;
```

```
if (t > 0){
```

```
    if (eq(life->ben_class, "mortg"))
```

```
        return sum_insured(t-1)*(1 + life->mortg_int_mth_w) - life->mortg_pmt_fix_w * life->benefits_curr;
```

```
    else
```

```
        return sum_insured(t-1) * (1+si_inc_pct);
```

```
}
```

```
// t<0
```

```
if (eq(life->ben_class, "mortg"))
```

```
    return (sum_insured(t+1)+ life->mortg_pmt_fix_w * life->benefits_curr)/(1 + life->mortg_int_mth_w) ;
```

```
else
```

```
    return sum_insured(t+1) / (1+si_inc_pct);
```

#### 6.1.1.6.1.55 sum\_insured\_if\_b

```
if (t <= life->commence_period_w || t > life->maturity_period_w)
    return 0.0;
```

```

if ((life->paid_up=="Y") && eq(life->ben_class,"ltc")) // LTC paid-up cover in force uses secondary
sum-life->insured
    return 0.0;

```

```

return sum_insured(t) * surv(t-1);

```

#### 6.1.1.6.1.56 sum\_insured\_if\_b\_2

```

if (t <= life->commence_period_w || t > life->maturity_period_w)
    return 0.0;

```

```

if (eq(life->ben_class,"dd"))
    return sum_insured(t-1) * surv_2(t-1);

```

```

// for LTC (siluk)

```

```

if (eq(life->ben_class,"ltc")) {
    if (t <= 1) {
        if (life->paid_up=="Y") // LTC paid-up cover in force
            return sum_insured(t);
        else
            return 0.;
    }
}

```

```

// get current PUV factor

```

```

double temp = 0;
if (!eq(pup_ltc_key,"0"))
    temp = pup_ltc_tbl;

```

```

double siluk_factor = 0.0;

```

```

if (temp > 0) { // only if PUV already exists according to policy term, then interpolate to
get factor for policy month

```

```

    double temp_next = pup_ltc_tbl_next;
    siluk_factor = (12. - life->pol_month(t-1))/12. * temp
                  + life->pol_month(t-1)/12. * temp_next;
}

```

```

return ( sum_insured_if_b_2(t-1)
        * (1. - death_rate(t-1))
        * (1. - decrem_rate(t-1))
        + sum_insured_if_b(t-1) // addition for new silukim
        * siluk_factor/1000.
        * lapse_rate(t-1) )
        * sum_insured(t)/sum_insured(t-1);
}

```

```

return 0;

```

#### 6.1.1.6.1.57 sum\_insured\_if\_b\_2\_no\_dec

```

if (t <= life->commence_period_w || t > life->maturity_period_w)
    return 0.0;

```

```

if (eq(life->ben_class,"dd"))
    return sum_insured(t-1) * surv_2_no_dec(t-1);

```

```

// for LTC (siluk)

```

```

if (eq(life->ben_class,"ltc")) {
    if (t <= 1) {

```

```

        if (life->paid_up=="Y")          // LTC paid-up cover in force
            return sum_insured(t);
        else
            return 0.;
    }

    // get current PUV factor
    double temp = 0;
    if (!eq(pup_ltc_key,"0"))
        temp = pup_ltc_tbl;

    double siluk_factor = 0.0;
    if (temp > 0) { // only if PUV already exists according to policy term, then interpolate to
get factor for policy month

        double temp_next = pup_ltc_tbl_next;
        siluk_factor = (12. - life->pol_month(t-1))/12. * temp
                        + life->pol_month(t-1)/12. * temp_next;
    }

    return ( sum_insured_if_b_2_no_dec(t-1)
            * (1. - death_rate(t-1))
            + sum_insured_if_b_no_dec(t-1) // addition for new silukim
            * siluk_factor/1000.
            * lapse_rate(t-1) )
            * sum_insured(t)/sum_insured(t-1);
}

return 0;

```

#### 6.1.1.6.1.58 sum\_insured\_if\_b\_no\_dec

```

if (t <= life->commence_period_w || t > life->maturity_period_w)
    return 0.0;

if ((life->paid_up=="Y") && eq(life->ben_class,"ltc")) // LTC paid-up cover in force uses secondary
sum-life->insured
    return 0.0;

return sum_insured(t) * surv_no_dec(t-1);

```

#### 6.1.1.6.1.59 sum\_insured\_if\_b\_no\_dth

```

if (t <= life->commence_period_w || t > life->maturity_period_w)
    return 0.0;

if ((life->paid_up=="Y") && eq(life->ben_class,"ltc")) // LTC paid-up cover in force uses secondary
sum-life->insured
    return 0.0;

return sum_insured(t) * surv_no_dth(t-1);

```

#### 6.1.1.6.1.60 pup\_ltc\_col

```

return xint((life->elapsed_months+t-2)/12.);

```

#### 6.1.1.6.1.61 pup\_ltc\_col\_next

```

return xint((life->elapsed_months+t-1)/12.)+1;

```

**6.1.1.6.1.62 claims\_re**

```

if (t <= life->commence_period_w || t > life->maturity_period_w || life->reinsurance=="N" ||
eq(life->re_type,"NONE"))
    return 0.0;

return claims_total(t) *(life->re_ratio_w );

```

**6.1.1.6.1.63 comm\_re**

```

if (t <= life->commence_period_w || t > life->maturity_period_w || eq(life->re_type,"NONE"))
    return 0.0;

double res = 0.0; //regular commission

int yr;

if (atof(life->comm_by_cal)==1)
    yr=xint(life->cal_duration(t)+1);
else
    yr=xint(life->pol_year_ext(t));

res = life->comm_ren_re[yr] / 100. * premium_re(t) ;

return res;

```

**6.1.1.6.1.64 comm\_re\_prof**

```

if (t <= life->commence_period_w || t > life->maturity_period_w ||eq(life->paid_up,"C"))
    return 0.0;

return max(0,life->comm_prof_re / 100. * profit_re(t)) ;

```

**6.1.1.6.1.65 exp\_re\_nom**

```

if (t <= life->commence_period_w || t > life->maturity_period_w || life->reinsurance=="N" ||
eq(life->re_type,"NONE"))
    return 0.0;

return life->expense_re_nom_temp / 100. * premium_re(t);

```

**6.1.1.6.1.66 interest\_re**

```

// Investment income in the period

if (t <= life->commence_period_w || t > life->maturity_period_w)
    return 0.0;

if (life->interest_re_calculate=="N")
    return 0.0;

int proj_yr = xint(life->proj_year(t));
if(eq(life->projection_type_int, "Rollup"))
    proj_yr = xint(life->proj_year_rollup(t));
proj_yr = max(proj_yr, 0);

if (eq(life->paid_up,"C")) {
    if (life->elapsed_months <= 24 || (life->elapsed_months > 24 && !eq(life->par_nonpar,"P")))
        return int_rate_res_mthly * (reserve_re(t-1));
}

```

```

        else //Participating
            return (life->inv_rate_mth_t[proj_yr]) * (reserve_re(t-1)); }

else
    return int_rate_res_mthly * (reserve_re(t-1));

```

#### 6.1.1.6.1.67 premium\_if\_b\_re

```

if (t <= life->commence_period_w || t > life->maturity_period_w)
    return 0.0;

if(life->reinsurance=="N" || life->paid_up=="C")
    return 0.0;

if(eq(life->re_type,"simple")){
    return life->prem_freq * claims_re(t) * (1.0 + life->re_cost_perc/100.0);
}

if (xint((life->pol_month(t)) != 1) && (t>12))
    return premium_if_b_re(t-1)*surv_per(t-1);

if (eq(life->re_type,"NONE") || (life->re_ratio_w==0.0))
    return 0.0;

double prem_re = 0.0;
double prate = 0.0;

//Premium lookup definitions

if(eq(life->re_type,"YRT")){
    prate = life->prem_rates_re * (1+ max(life->health_occ_perc_min,life->health_occ_perc)/100.) +
    life->prem_per_unit_si_re;
    if (inlist(life->ben_class,"ltc,phi")) // *** not good with monthly SI, life->prem_freq,
    etc. ?
        prem_re = prate * life->sum_ins_curr * life->benefits_b_prm(t)/life-
        >prem_rate_scale_w
        *( life->re_ratio_w) * life->sum_ins_inc_acc(t);
    else
        prem_re = prate/life->prem_rate_scale_w * sum_insured_if_b(t)*( life->re_ratio_w);
}

if(eq(life->re_type,"OT"))
    prem_re = premium_if_b(t) * (life->re_ratio_w) ;

return prem_re;

```

#### 6.1.1.6.1.68 premium\_re

```

if (t < life->commence_period_w || t > life->maturity_period_w)
    return 0.0;

if (fmod(xint(life->pol_month(t-1)), xint(12. / life->prem_freq))!=0)
    return 0.0; //not a premium due date

```

```
return premium_if_b_re(t) * atof(life->prem_re_mult) / life->prem_freq;
```

#### 6.1.1.6.1.69 profit\_re

```
if (t <= life->commence_period_w || t > life->maturity_period_w)
    return 0.0;
```

```
if (t==0)
    return 0.0;
```

```
return premium_re(t) - claims_re(t) - exp_re_nom(t)
    - comm_re(t)
    + interest_re(t)
    - reserve_re_increase(t);
```

#### 6.1.1.6.1.70 reserve\_re

```
if (t <= life->commence_period_w || t >= life->maturity_period_w || eq(life->re_type,"NONE") ||
(life->re_ratio_w==0.0) || life->reinsurance=="N" || eq(life->re_type,"simple"))
    return 0.0;
```

```
double res_re1 = 0.0; //quota share reinsurance part
double res_re2 = 0.0; // claims reserves part
```

```
//Quota share reinsurance - res_re1
```

```
if(eq(life->re_type,"OT"))
    res_re1 = (reserve_basic(t)+ err(t)) * (life->re_ratio_w);
```

```
if (inlist(life->ben_class,"phi,ltc") && eq(life->use_phi_claims_cf,"Y"))
    res_re2 = reserve_basic_claims(t)*(life->re_ratio_w);
```

```
return res_re1 + res_re2 ;
```

#### 6.1.1.6.1.71 reserve\_re\_increase

```
if (t <= life->commence_period_w || t > life->maturity_period_w)
    return 0.0;
```

```
if (life->reserve_re_increase_calculate=="N")
    return 0.0;
```

```
return reserve_re(t) - reserve_re(t-1);
```

#### 6.1.1.6.1.72 startup

```
if (life->submodel != "TERM")
    return 0;
```

```
//***** set variables *****/
```

```
set_other_variables();
```

```
// Calculate fixed payback for mortgage
```

```
if (eq(life->ben_class,"mortg")){
    life->mortg_int_mth_w = monthly_rate(life->mortg_int);
    double v = 1/(1 + life->mortg_int_mth_w);
    double a_n = (1 - pow(v,life->benefit_term))/life->mortg_int_mth_w;
```

```

    if (a_n != 0)
        life->mortg_pmt_fix_w = life->sum_ins_curr / a_n ;//monthly level payback

    // calculate current sum-life->insured
    if (life->elapsed_months > 0) {
        a_n = (1 - pow(v,life->maturity_period_w))/life->mortg_int_mth_w;
        life->sum_ins_curr = a_n * life->mortg_pmt_fix_w;
    }
}

if (eq(life->done_startup_w,"false")){
    validate_data();
    life->done_startup_w = "true";
}

if ((eq(life->ben_class,"phi") || (eq(life->ben_class,"ltc") && eq(life->paid_up,"C")))) && life->use_phi_claims_cf == "Y"){

    claims_inpay_rate.resize(temp_tbl_size, temp_tbl_size);
    claims_inpayment.resize(temp_tbl_size, temp_tbl_size);
    claims_inpay_pv.resize(temp_tbl_size, temp_tbl_size);
    claims_inpay_res.resize(temp_tbl_size, temp_tbl_size);
    claims_inpay_rate_pv.resize(temp_tbl_size, temp_tbl_size);
    claims_inpay_res_factor.resize(temp_tbl_size, temp_tbl_size);
}
return 0.0;

```

#### 6.1.1.6.1.73 pol\_sub\_year

```

if (t < life->commence_period_w || t > life->maturity_period_w)
    return NO_AVG;

if(life->prem_lookup_temp=="N" || life->prem_lookup_freq_temp == 0) //level premium
    return life->pol_year(t);

//YRT or stepped premium
if (xint(fmod(xint(life->pol_year(t)),life->prem_lookup_freq_temp)) == 0)
    return life->prem_lookup_freq_temp;
else
    return fmod(xint(life->pol_year(t)),life->prem_lookup_freq_temp);

```

#### 6.1.1.6.1.74 proj\_month

```

if (t < life->commence_period_w || t > life->maturity_period_w)
    return NO_AVG;

if (t == 0)
    return NO_AVG;

int result = xint(fmod(t, 12));

if (result == 0) {
    if (t > 0)
        result = 12;
    else
        result = -12;
}
return result;

```



### 6.1.1.6.2 External Functions

#### 6.1.1.6.2.1 monthly\_rate

```
double monthly_rate(double annual_rate) {
    return (pow(1. + annual_rate / 100. , 1. / 12.) - 1.);
}
```

#### 6.1.1.6.2.2 set\_other\_variables

```
void set_other_variables (void) {

    int mth=0, year=0, i=0;
    double total=0.0,mult=0.0;

    // set Capital requirement as a percentage of DAC-Books
    if (life->dac_cap_apply=="N")
        life->dac_cap_perc_w = 0.0;
    else {
        if (life->prod_yr_w < 1999)
            life->dac_cap_perc_w = 0.0;
        if (life->prod_yr_w >= 1999)
            life->dac_cap_perc_w = 30.0;
        if (life->prod_yr_w >= 2004)
            life->dac_cap_perc_w = 100.0;
    }

    if(eq(life->ben_class,"ltc"))
        pup_ltc_key = life->prod_code + "_" + life->sex;

    // calculate maximum age in claims rates table
    life->omega_age_w = life->omega_age_cmi; // max age in rows of CMI table

    if(life->death_ben_w=="N")
        life->omega_age_w = life->omega_age_dec; //max age in rows of decrement tables

    {
        life->prem_term = min(life->prem_term_input*12,life->benefit_term);
        if (eq(life->ben_class,"mortg") && life->prem_lookup_temp=="N") // shoham kavua
            life->prem_term = life->benefit_term - 36;
        if (eq(life->ben_class,"fib")) // bareket
            life->prem_term = life->benefit_term - 36;
    }

    if (life->maturity_period_w >= 12*xint(t_high/12.)) {
        log_strm << "maturity_period_w: " << life->maturity_period_w << endl;
        log_strm << "valuation_high_w: " << 12*xint(t_high/12.) << endl;
        log_strm << "policy: " << life->pol_number << endl;
        throw("Benefit Term exceeds projection period, rerun with larger t_high\n ");
    }

    // for a rollup run, allow for extra elapsed months
    if (eq(life->projection_type,"Rollup") && (life->elapsed_months+life->elapsed_months_extra<=12)) {
        life->commence_period_w = -(life->elapsed_months+life->elapsed_months_extra);
        life->benefit_term = life->maturity_period_w - life->commence_period_w ;
    }
    life->mat_period_min = life->maturity_period_w;
```

```
// ***** Set Interest rates *****

// For net bonus calc. of phi claims in payment
if (life->par_nonpar=="P" && inlist(life->ben_class,"phi,ltc") && eq(life-
>use_phi_claims_cf,"Y")){
  for (i=0; i<=119.;i++){
    inv_rate_clm_mth_t[i] = life->inv_rate_mth_t[i];
  }
  mgt_fee_var_clm =life->var_mgt_fee;
  mgt_fee_fixed_clm =life->fixed_mgt_fee_term;
}
// For phi level and ltc (pre2004) remove interest income would should go to bonus
if (life->par_nonpar=="P" && (eq(life->prod_code,"phi-1") || eq(life->ben_class,"ltc"))&& !eq(life-
>paid_up,"C")) {
  life->mgt_fee_variable =life->var_mgt_fee;
  life->mgt_fee_fixed =life->fixed_mgt_fee_term;

  for (i=0; i<=119.;i++){
    if (eq(life->ben_class,"ltc")) // for LTC reduce bonus by 50% as bonus only paid on
ltc claim
      life->inv_rate_mth_t[i] = life->inv_rate_mth_t[i] -
        ((life->inv_rate_mth_t[i] - life->mgt_fee_fixed/1200)
        * (1-life->mgt_fee_variable/100) - int_rate_res_mthly ) * 0.5;
    else
      life->inv_rate_mth_t[i] = life->inv_rate_mth_t[i] -
        ((life->inv_rate_mth_t[i] - life->mgt_fee_fixed/1200)
        * (1-life->mgt_fee_variable/100) - int_rate_res_mthly );
  } //end for loop
} // end if

life->v_month_w = 1. / (1. + monthly_rate(life->ev_disc_rate));

// if discount type = Single, then replace the discount rate vector with the input value
if (eq(life->ev_discount_rate_type,"Single")) {
  for (i=0; i<=119.;i++){
    life->v_month_t[i] = life->v_month_w;
  }
}

// ***** set commission variables *****
// reduction for short premium terms (2n)
mult = 1.;
if(life->comm_min_prem_term > 0 || eq(life->ben_class,"ltc")){
  if (eq(life->ben_class,"ltc")) { // special reducing formula for LTC with old entry age
    if (life->age_at_issue>=65)
      mult = (110. - life->age_at_issue)/50.;
    else
      mult = 1.;
  }
  else
    mult = life->prem_term/life->comm_min_prem_term;
  mult = min(1.,mult);

  for(i = 0; i<116; i++){
    life->comm_regular_pc[i] = life->comm_regular_pc[i]* mult;
  }
}
```

```

}
// set total % of init reg comm
double comm_tot = 0.;
for (i=0; i<115; i++)
    comm_tot = comm_tot + life->comm_regular_pc[i];
life->comm_reg_tot_w = comm_tot;

// Set DAC amortisation period
if(eq(life->dac_amort_type,"Lifetime"))
    life->dac_amort_per = life->prem_term;
if(life->dac_amort_per > life->prem_term)
    life->dac_amort_per = life->prem_term;
if(life->dac_amort_per_tax > life->prem_term)
    life->dac_amort_per_tax = life->prem_term;

// close function
}

```

### 6.1.1.6.2.3 validate\_data

```

void validate_data(void) {

if (life->prem_term > life->benefit_term)
    life->error_msg = "prem_term_>_ben_term";

if (pv_period != 12)
    throw NonFatalError("Template set up for monthly projections. Change the discount period in
the projection task.");

if (eq(life->projection_type,"Valn") && life->elapsed_months < 0)
    life->error_msg = "elapsed_months_<_0";

if (life->elapsed_months > life->benefit_term)
    life->error_msg = "elapsed_months_>_ben_term";

if (12 % life->prem_freq != 0.0) // premium frequency must be a multiple of 12 and must not be zero
whilst premiums are being paid.
    throw NonFatalError("Premium frequency must be 1, 2, 3, 4, 6, or 12.");

if ((life->prem_freq == 0 || eq(life->paid_up,"Y")) && (life->promil >0.3) && !eq(life-
>ben_class,"ltc"))
    life->error_msg = "paid_up";

if (life->sum_ins_curr <= 0.0001)
    life->error_msg = "Sum_Insured_<=_0";

if (eq(life->res_basis,"Net_Prem") && eq(life->prem_lookup_temp,"Y") && life-
>prem_lookup_freq_temp==1)
    throw NonFatalError("Cannot have a Net Premium Reserve for YRT product (Product Code:
"+life->prod_code+"");

// close function
}

```

### 6.1.1.6.3 Temporary Tables

#### 6.1.1.6.3.1 claims\_inpay\_res

```

// Present value of future claims payment by projection months(row) and duration (column)

```

```

if ((!eq(life->ben_class,"phi")&&!eq(life->ben_class,"ltc")||!eq(life->paid_up,"C"))|| !eq(life-
>use_phi_claims_cf,"Y"))
    return 0;

if (r == 0)
    return 0.0;

if (eq(life->projection_type,"Rollup")&& (r < wp_phi))
    return 0.0;

// reserve factor for השלמת is fixed to 2
if(eq(life->prod_code ,"phi-k-fra"))
    return (claims_inpayment(r,c) + premium_gross(r)*claims_inpay_rate(r,c)) * 2 ;

if (r+c == temp_tbl_size-1)
    return 0.0;

if (c<wp_phi && !eq(life->paid_up,"C")) {
    double rate = 2.5;
    if (life->year_start <= 1992)
        rate = 4;

    double disc_factor = 1. / (1. + monthly_rate(rate));

    if (life->phi_res_discount_rate_type == "RF")
        disc_factor = life->v_month_t[xint((r+c +t_start-1)/12) +1];

    return claims_inpay_res(r,c+1)*disc_factor; }

return (claims_inpayment(r,c) + premium_gross(r)*claims_inpay_rate(r,c)) *
claims_inpay_res_factor(r,c);

```

#### 6.1.1.6.3.2 res\_cx

```

// Commutation Function  $C_x = v^{(x+1)} * (l_x - l_{(x+1)})$ 
// based on average sum life->insured during the next year (Cx bar)
// r = current age in years

double d=0.0;
double decrem =0.0;
if (life->death_ben_w=="Y")
    d = res_lx(r,0) - res_lx(r+1,0); //deaths
else {
    life->row_num = r;
    if (eq(life->use_uw_date,"Y"))
        decrem = life->decrem_rates_uw_res;
    else
        decrem = life->decrem_rates_res;

    d = res_lx(r,0) * decrem * life->decrem_mult_res/100. * (1.+max(life-
>health_occ_perc_min,life->health_occ_perc)/100.);
}

int tee = xint((r-life->age_at_issue)*12.) - life->elapsed_months + 1;

```

```

if (d>0.0){
  if (tee <= t_high && r<life->omega_age_w && tee>=life->commence_period_w && tee+6<=life-
>maturity_period_w){
    if (eq(life->ben_class,"phi") && eq(life->use_phi_claims_cf,"Y"))
      d = d *
(sum_insured(tee+5)*claims_cost_factors(r,sexcode)+sum_insured(tee+6)*claims_cost_factors(r,sexcode
))/2.* life->claims_cost_multiplier;
    else
      d = d * (sum_insured(tee+5)+sum_insured(tee+6))/2.; // sum insured in middle
of next year
  }
  else
    d = 0.0;
}

return d * res_vx(r+1,0);

```

#### 6.1.1.6.3.3 res\_dx

```

// Commutation Function Dx Yearly Dx = lx * v^x
// r = current age in years

if (r > life->omega_age_w) //omega_age from underlying table
  return 0.0;

return res_lx(r, 0) * res_vx(r,0);

```

#### 6.1.1.6.3.4 res\_lx

```

// Commutation Function lx

if (r <= 0)
  return 100.0;

double decrem=0.0;

if (r > life->omega_age_w) { // omega age allows for table adjustment
  if (life->death_ben_w=="Y") {
    return 0.0; // all die
  }
  else {
    life->row_num = life->omega_age_w;

    if (eq(life->use_uw_date,"Y"))
      decrem = life->decrem_rates_uw_res;
    else
      decrem = life->decrem_rates_res;

    return res_lx(r-1, 0) // decrement rates do not have to occur (like death)
      * (1. - decrem * life->decrem_mult_res/100.)
      * (1+max(life->health_occ_perc_min,life->health_occ_perc)/100.);
  }
}

double q=0.0;
double q_i=0.0;
life->row_num = r-1;
q =life->death_rates_res_tbl;

```

```

q = q + life->mort_addn_res/1000.;

if ((life->death_ben_w=="Y")) {
    q_i = 0.0;
}
else {
    life->row_num = r - 1;

    if (eq(life->use_uw_date,"Y"))
        decrem = life->decrem_rates_uw_res;
    else
        decrem = life->decrem_rates_res;

    q_i = decrem * life->decrem_mult_res/100.
    * (1+max(life->health_occ_perc_min,life->health_occ_perc)/100.);
}

return res_lx(r-1, 0) * (1. - q)*(1. - q_i);

```

#### 6.1.1.6.3.5 res\_mx

```

if (r>life->omega_age_w)
    return 0.0;

if (r==life->omega_age_w)
    return res_cx(r,0);

return res_cx(r,0) + res_mx(r+1,0);

```

#### 6.1.1.6.3.6 res\_nx

```

//Nx

if (r >= life->omega_age_w)
    return res_dx(r, 0);

return res_nx(r+1, 0) + res_dx(r, 0);

```

#### 6.1.1.6.3.7 res\_vx

```

// Commutation Function vx = v^(x)
// r = current age in years
//TYPE ALL

if (r<=0)
    return 1.0;
return res_vx(r-1,0)/(1.+life->int_rate_res/100.);

```

#### 6.1.1.6.3.8 claims\_cost\_factors

```

// claims cost factors by age (rows) and life->sex (columns)
// age is 0 to 100
// Sex, 0=male, 1=females

xstring sex2 = "M";
if (c==1)
    sex2="F";

```

```

double rate=0.0;

if (eq(life->ben_class,"phi") || eq(life->ben_class,"ltc")) {    // new claims cost format
    life->key_temp = life->claims_cost_key;
    life->series_col_key = xstring(r);

    rate = life->claims_cost_factors_tbl;

    if(rate!= 9999999. && rate!=10000000.)
        return rate * life->claims_cost_multiplier;

    if(rate== 9999999.)
        return 0.0;

    if(rate==10000000.)
        throw NonFatalError("Key " + life->claims_cost_key + " not found in claims cost
table.");
}

else { //previous claims cost format
    life->key_temp = xstring(r);
    life->series_col_key = sex2;

    rate = life->claims_cost_factors_tbl;

    if(rate!= 9999999. && rate!=10000000.)
        return rate;

    if(rate==10000000.)
        return 0.0;

    if(rate== 9999999.)
        throw NonFatalError("Error looking up .... "+ xstring(r) + ", "+sex2+ ", in
claims_cost_factors_tbl");
}

return 0.0;

```

#### 6.1.1.6.3.9 claims\_inpay\_pv

```

// Present value of future claims payment by projection months(row) and duration (column)
// Rows are projection months
// Columns are duration (in months) of claim.

if (t <= life->commence_period_w || t > life->maturity_period_w)
    return 0.0;

if ((!eq(life->ben_class,"phi")&&(!eq(life->ben_class,"ltc"))||!eq(life->paid_up,"C"))|| !eq(life-
>use_phi_claims_cf,"Y"))
    return 0;

if (r <= 0)
    return 0.0;

if(r+c == temp_tbl_size-1)
    return (claims_inpayment(r,c)) *life->v_month_t[xint((r+c +t_start-1)/12) +1];

```

```
return (claims_inpay_pv(r,c+1)+claims_inpayment(r,c)) *life->v_month_t[xint((r+c+t_start-1)/12)
+1];
```

#### 6.1.1.6.3.10 claims\_inpay\_rate

```
// claims in payment rates by projection months(row) and duration (column)
```

```
if ((!eq(life->ben_class,"phi")&&(!eq(life->ben_class,"ltc"))||!eq(life->paid_up,"C"))|| !eq(life-
>use_phi_claims_cf,"Y"))
    return 0;
```

```
if (r==0 )
    return 0.0;
```

```
if((r < 1) && !eq(life->prod_code, "phi-k-fra") && !eq(life->paid_up, "C"))
    return 0.0;
```

```
if((c >= 3+wp_phi) && eq(life->prod_code , "phi-k-fra"))
    return 0.0;
```

```
if (c<wp_phi && !eq(life->paid_up, "C")) return 0.0;
```

```
int durn = c-wp_phi;
int age = life->age_last(max(r+t_start,life->commence_period_w+1));
```

```
if (life->paid_up == "C"){
    if (c==0){
        if((r)==1)
            return 1.;
        return 0.;
    }
    durn = c + life->elapsed_months-1;
    age =life->age_at_issue;
}
```

```
if (c==wp_phi && !eq(life->paid_up, "C")){
```

```
    double clm_rate = 0.0;
    clm_rate = claims_inpay_rate.sum_of_diagonal(r+wp_phi-1);

    return max(decrem_rate_dep(max(r + t_start,life->commence_period_w+1))
        * life->claims_cost_multiplier
        *(surv(r+t_start+wp_phi -1)- clm_rate),0);
}
```

```
life->recovery_rates_row = min(max(age,18),67);
life->recovery_rates_col = durn;
double termination_rate = life->recovery_rates_tbl;
```

```
//***** add margin recovery rates*****
```

```
if (life->margin_add=="Y")
    termination_rate = termination_rate * (1+life->margin_recover/100);
```

```
return claims_inpay_rate(r,c-1) *(1.- termination_rate);
```



**6.1.1.6.3.11 claims\_inpay\_rate\_pv**

```

if (t <= life->commence_period_w || t > life->maturity_period_w)
    return 0.0;

if ((!eq(life->ben_class,"phi")&&!eq(life->ben_class,"ltc")||!eq(life->paid_up,"C"))|| !eq(life-
>use_phi_claims_cf,"Y"))
    return 0;

if (r <= 0)
    return 0.0;

double rate = 2.5;
    if (life->year_start <= 1992)
        rate = 4;

double disc_factor = 1. / (1. + monthly_rate(rate));

if (life->phi_res_discount_rate_type == "RF")
    disc_factor = life->v_month_t[xint((r+c +t_start-1)/12) +1];

if(r+c >= temp_tbl_size-1)
    return (claims_inpay_rate(r,c)) * disc_factor;

return (claims_inpay_rate_pv(r,c+1)+claims_inpay_rate(r,c)) * disc_factor;

```

**6.1.1.6.3.12 claims\_inpay\_res\_factor**

```

if (t <= life->commence_period_w || t > life->maturity_period_w)
    return 0.0;

if ((!eq(life->ben_class,"phi")&&!eq(life->ben_class,"ltc")||!eq(life->paid_up,"C"))|| !eq(life-
>use_phi_claims_cf,"Y"))
    return 0;

if (r <= 0)
    return 0.0;

if(r+c == temp_tbl_size-1)
    return 0.0;

if (c<wp_phi && !eq(life->paid_up,"C")) {
    double rate = 2.5;
    if (life->year_start <= 1992)
        rate = 4;

    double disc_factor = 1. / (1. + monthly_rate(rate));

    if (life->phi_res_discount_rate_type == "RF")
        disc_factor = life->v_month_t[xint((r+c +t_start-1)/12) +1];
    return claims_inpay_res_factor(r,c+1)*disc_factor;
}

double claims_rate = claims_inpay_rate(r,c) ;

if (claims_rate == 0) return 0.0;

```

```
return claims_inpay_rate_pv(r,c)/claims_rate;
```

#### 6.1.1.6.3.13 claims\_inpayment

```
// claims in payment by projection months(row) and duration (column)
```

```
// Rows are projection months
```

```
// Columns are duration (in months) of claim.
```

```
if ((!eq(life->ben_class,"phi")&&!eq(life->ben_class,"ltc")||!eq(life->paid_up,"C"))|| !eq(life->use_phi_claims_cf,"Y"))
    return 0;
```

```
double payment=0.0;
```

```
int bonus_WP = 24;
```

```
int curr_dur = life->elapsed_months;
```

```
if( !eq(life->paid_up , "C")){
```

```
payment = claims_inpay_rate(r,c) *sum_insured(r+t_start+wp_phi);
```

```
//Bonus payments from 25th month for participation
```

```
if(life->par_nonpar=="P" && c >= bonus_WP+wp_phi)
```

```
    payment = payment * (1 + bonus_rate_acc_mthly(r+t_start+c))/(1 +
bonus_rate_acc_mthly(r+wp_phi+t_start+bonus_WP-1));
```

```
return payment;
```

```
}
```

```
//If already Claims in payment
```

```
if (r > 1)
```

```
    return 0.;
```

```
payment = claims_inpay_rate(r,c) * sum_insured(r+t_start);
```

```
if(life->par_nonpar=="P"){
```

```
    if( c==0 || c + curr_dur < bonus_WP)
```

```
        return payment;
```

```
    if(curr_dur >= bonus_WP)
```

```
        return payment * pizui_prop_pup_stat_c * (1 + bonus_rate_acc_mthly(r+t_start+c-1)) +
payment * (1-pizui_prop_pup_stat_c);
```

```
        return payment * pizui_prop_pup_stat_c * (1 + bonus_rate_acc_mthly(r+t_start+c))/(1 +
bonus_rate_acc_mthly(r+t_start+bonus_WP-curr_dur-1)) + payment * (1-pizui_prop_pup_stat_c);
```

```
}
```

```
return payment;
```

#### 6.1.1.6.4 Scalars

##### 6.1.1.6.4.1 claims\_inflation\_mthly

```
return monthly_rate(life->claim_inflation_perc);
```

##### 6.1.1.6.4.2 interest\_rein\_mthly

```
return monthly_rate(life->interest_rein);
```

##### 6.1.1.6.4.3 sexcode

```
if (life->sex=="F")
```

```
    return 1;
```

```
return 0;
```

#### 6.1.1.6.4.4 t\_start

```
if ((eq(life->ben_class,"phi") || (eq(life->ben_class,"ltc") && eq(life->paid_up,"C")))) && life-
>use_phi_claims_cf == "Y"){

    if (eq(life->projection_type,"Rollup")) // Assume that always rollup for 1 year
        return max(-11,life->commence_period_w +1);

    return 0;
}
return 0;
```

#### 6.1.1.6.4.5 temp\_tbl\_size

```
if ((eq(life->ben_class,"phi") || (eq(life->ben_class,"ltc") && eq(life->paid_up,"C")))) && life-
>use_phi_claims_cf == "Y") {
    if (eq(life->projection_type,"Rollup")) // Assume that always rollup for 1 year
        return min(life->benefit_term,life->benefit_term +life->commence_period_w + 11) ;

    return max(life->maturity_period_w+1,1);
}
return 0;
```

#### 6.1.1.6.4.6 wp\_phi

```
int temp = 0;
if (eq(life->ben_class,"phi")&& (life->use_phi_claims_cf=="Y")){
    life->tarif_spec_row_key= xstring(life->tarif);
    temp= atoi(life->waiting_period_modeled);

    if (temp!= 1 && temp!= 3 && temp != 6)
        temp = 3 ;
}
return temp;
```

#### 6.1.1.6.4.7 int\_rate\_res\_hy

```
return pow(1. + life->int_rate_res/100., 0.5);
```

#### 6.1.1.6.4.8 int\_rate\_res\_mthly

```
return pow(1. + life->int_rate_res/100., 1/12.)-1.0;
```

### 6.1.2 Externs

<No Externs Exist>

## 6.2 Lookup Settings

### 6.2.1 Input Manager: Input Manager

#### 6.2.1.1 Input Page: Annuity

##### 6.2.1.1.1 Assumption Set: Base

##### 6.2.1.1.1.1 res\_prop\_data

Description: Tzeva Kesef

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: res_prop_key	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: pol_number	Absolute	Character	Not Applicable	Value Of: - 9999	Not Applicable	
N/A	Code Variable: life: fund	Absolute	Character	Not Applicable	Value Of: - 9999	Not Applicable	
N/A	Code Variable: life: ben_class	Absolute	Character	Not Applicable	Value Of: - 9999	Not Applicable	

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: policy_type	Absolute	Character	Not Applicable	Value Of: - 9999	Not Applicable	
N/A	Input Variable: life_Data: life: paid_up	Absolute	Character	Not Applicable	Value Of: - 9999	Not Applicable	

#### 6.2.1.1.1.2 piz\_antiselection\_adj

Description: AnnuityTU

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: AS prop piz	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: 60	Absolute	Numeric	First	Default	Default	
N/A	Code Variable: life: sex	Absolute	Character	Not Applicable	Error	Not Applicable	
N/A	Code Scalar: life: fund_type	Absolute	Character	Not Applicable	Retry With: P	Not Applicable	
N/A	Code Variable: life: ben_class	Absolute	Character	Not Applicable	Retry With: adif	Not Applicable	end: klasi gimla: klasi wol: klasi

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
							ytron: klasi adif: adif profil: profil
N/A	Constant: 1	Absolute	Numeric	Error	Error	Error	
N/A	Code Variable: life: pol_type_annuity_tu	Absolute	Character	Not Applicable	Retry With: managers	Not Applicable	

#### 6.2.1.1.1.3 prat\_antiselection\_adj

Description: AnnuityTU

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: AS prop prat	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: 60	Absolute	Numeric	First	Default	Default	
N/A	Code Variable: life: sex	Absolute	Character	Not Applicable	Error	Not Applicable	

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Scalar: life: fund_type	Absolute	Character	Not Applicable	Retry With: P	Not Applicable	
N/A	Code Variable: life: ben_class	Absolute	Character	Not Applicable	Retry With: adif	Not Applicable	end: klasi gimla: klasi wol: klasi ytron: klasi adif: adif profil: profil
N/A	Constant: 1	Absolute	Numeric	Error	Error	Error	
N/A	Code Variable: life: pol_type_annuity_tu	Absolute	Character	Not Applicable	Retry With: managers	Not Applicable	

#### 6.2.1.1.1.4 old\_antiselection\_adj

Description: AnnuityTU

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: AS prop old	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
----------	-------------	---------------------	------	--------------	---------	------------	-------------------

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: 60	Absolute	Numeric	First	Default	Default	
N/A	Code Variable: life: sex	Absolute	Character	Not Applicable	Error	Not Applicable	
N/A	Code Scalar: life: fund_type	Absolute	Character	Not Applicable	Retry With: P	Not Applicable	
N/A	Code Variable: life: ben_class	Absolute	Character	Not Applicable	Retry With: adif	Not Applicable	end: klasi gimla: klasi wol: klasi ytron: klasi adif: adif profil: profil
N/A	Constant: 1	Absolute	Numeric	Error	Error	Error	
N/A	Code Variable: life: pol_type_annuity_tu	Absolute	Character	Not Applicable	Retry With: managers	Not Applicable	

#### 6.2.1.1.1.5 gimla\_table

Description: gimla

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: gimla_col_key	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No



Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: gimla_row_key	Absolute	Numeric	First	Previous	Last	

**6.2.1.1.1.6      takeupt\_age**

Description: Annuity

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: Assumption	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: <*code_variable*>	Absolute	Character	Not Applicable	Error	Not Applicable	
N/A	Input Variable: life_Data: life: sex	Absolute	Character	Not Applicable	Retry With: M	Not Applicable	

**6.2.1.1.1.7      Types of Annuity Prop**

Description: AnnuityTU

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: <*code_variable*>	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: takeup_age	Absolute	Numeric	First	Default	Default	
N/A	Code Variable: life: sex	Absolute	Character	Not Applicable	Error	Not Applicable	
N/A	Code Scalar: life: fund_type	Absolute	Character	Not Applicable	Retry With: P	Not Applicable	
N/A	Code Variable: life: ben_class	Absolute	Character	Not Applicable	Retry With: adif	Not Applicable	end: klasi gimla: klasi wol: klasi ytron: klasi adif: adif profil: profil
N/A	Code Variable: annuity: ann_series_prop	Absolute	Numeric	Retry With: 1	Error	Error	
N/A	Code Variable: life: pol_type_annuity_tu	Absolute	Character	Not Applicable	Retry With: managers	Not Applicable	

**6.2.1.1.1.8 Retirement rate**

Description: AnnuityTU

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: <*code_variable*>	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Column: life: retirement_age_looku p	Absolute	Numeric	Default	Default	Default	
N/A	Code Variable: life: sex	Absolute	Character	Not Applicable	Error	Not Applicable	
N/A	Code Scalar: life: fund_type	Absolute	Character	Not Applicable	Retry With: P	Not Applicable	
N/A	Code Variable: life: ben_class	Absolute	Character	Not Applicable	Retry With: adif	Not Applicable	end: klasi gimla: klasi wol: klasi ytron: klasi adif: adif profil: profil
N/A	Code Variable: life: ann_series	Absolute	Numeric	Retry With: 1	Error	Error	
N/A	Code Variable: life: pol_type_annuity_tu	Absolute	Character	Not Applicable	Retry With: managers	Not Applicable	

**6.2.1.1.1.9 annuity\_value\_res\_tbl**

Description: T\_Factors

## Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: Base	Relative	Character	Not Applicable	Error	Not Applicable	

## Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Scalar: life: yob	Absolute	Numeric	First	Previous	Last	
N/A	Code Variable: annuity: takeup_age	Absolute	Numeric	Error	Error	Error	
N/A	Code Variable: annuity: fund_t_factor	Absolute	Numeric	Error	Error	Error	
N/A	Code Variable: life: sex	Absolute	Character	Not Applicable	Error	Not Applicable	

**6.2.1.1.1.10 annuity\_detail\_gtee\_tbl**

Description: AnnuityDetails

## Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: ANN_FAC_GTEE	Absolute	Character	Not Applicable	Error	Not Applicable	

## Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: annuity: annuity_code	Absolute	Character	Not Applicable	Error	Not Applicable	

**6.2.1.1.11 annuity\_details\_tbl**

Description: AnnuityDetails

## Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: <*code_variable*>	Absolute	Character	Not Applicable	Error	Not Applicable	

## Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: annuity: annuity_code	Absolute	Character	Not Applicable	Error	Not Applicable	

**6.2.1.1.12 Retirement rate ann**

Description: AnnuityTU

## Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: <*code_variable*>	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: annuity: takeup_age	Absolute	Numeric	Error	Error	Error	
N/A	Code Variable: annuity: sex1	Absolute	Character	Not Applicable	Error	Not Applicable	
N/A	Code Scalar: annuity: fund_type	Absolute	Character	Not Applicable	Retry With: P	Not Applicable	
N/A	Code Variable: life: ben_class	Absolute	Character	Not Applicable	Retry With: adif	Not Applicable	end: klasi gimla: klasi wol: klasi ytron: klasi adif: adif profil: profil
N/A	Code Variable: life: ann_series	Absolute	Numeric	Retry With: 1	Error	Error	
N/A	Code Variable: life: pol_type_annuity_tu	Absolute	Character	Not Applicable	Retry With: managers	Not Applicable	

**6.2.1.1.1.13 annuity\_details\_temp\_tbl**

Description: AnnuityDetails

## Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: ANN_FAC_NO_GTE E	Absolute	Character	Not Applicable	Error	Not Applicable	

## Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: annuity: temp_annuity_code	Absolute	Character	Not Applicable	Value Of: - 99999	Not Applicable	

**6.2.1.2 Input Page: Charges****6.2.1.2.1 Assumption Set: Base****6.2.1.2.1.1 mgtfee\_tbl**

Description: format\_mgtfee

## Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: <*code_variable*>	Absolute	Character	Not Applicable	Error	Not Applicable	

## Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: mgtfee_format	Absolute	Numeric	Value Of: - 99999	Value Of: - 99999	Value Of: - 99999	

**6.2.1.2.1.2 mgt\_deficit\_perc**

Description: Economic

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: Current	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: int_rate_cumm	Absolute	Character	Not Applicable	Value Of: 0	Not Applicable	
N/A	Code Variable: life: company	Absolute	Character	Not Applicable	Retry With: clal	Not Applicable	



## 6.2.1.3 Input Page: Claims

### 6.2.1.3.1 Assumption Set: Base

#### 6.2.1.3.1.1 claim\_cost\_factors\_tbl

Description: profil\_rider\_claims\_annuity\_fac

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: row	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: age_adj	Absolute	Numeric	Error	Error	Error	

#### 6.2.1.3.1.2 claim\_rates\_tbl

Description: profil\_decrement\_rates

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: col	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: age_adj	Absolute	Numeric	Error	Error	Error	

#### 6.2.1.3.1.3 claims\_cost\_factors\_tbl

Description: claim\_cost\_phi12\_ltc07

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: series_col_key	Absolute	Character	Not Applicable	Value Of: 9999999	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: key_temp	Absolute	Character	Not Applicable	Value Of: 10000000	Not Applicable	

**6.2.1.3.1.4 clms\_mult\_infl**

Description: clms\_mult

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: 0	Absolute	Numeric	Error	Error	Error	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: clms_mult_set_temp	Absolute	Character	Not Applicable	Error	Not Applicable	

**6.2.1.3.1.5 clms\_mult\_i**

Description: clms\_mult

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: clms_mult_i_col	Absolute	Numeric	Error	Error	Error	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: clms_mult_set_temp	Absolute	Character	Not Applicable	Error	Not Applicable	

**6.2.1.3.1.6 clms\_mult\_tt**

Description: clms\_mult

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	System Variable: r	Absolute	Numeric	Error	Value Of: - 99999	Error	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: prod_assumpt_rider_clms_tbl	Absolute	Character	Not Applicable	Error	Not Applicable	

**6.2.1.3.1.7 recovery\_rates\_tbl**

Description: phi\_recover

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: recovery_rates_col	Relative	Numeric	First	Error	Last	
N/A	Code Variable: life: pol_type_recovery_rates	Absolute	Character	Not Applicable	Retry With: private	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: recovery_rates_row	Relative	Numeric	First	Error	Last	

#### 6.2.1.3.1.8 Various\_Parameters

Description: Parameters

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: Value	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
----------	-------------	---------------------	------	--------------	---------	------------	-------------------

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: <*code_variable*>	Absolute	Character	Not Applicable	Error	Not Applicable	

## 6.2.1.4 Input Page: Commission

### 6.2.1.4.1 Assumption Set: Base

#### 6.2.1.4.1.1 comm\_extra\_tbl

Description: comm\_extra

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: comm_ext_col_key	Absolute	Character	Not Applicable	Value Of: 9999	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: temp_comm_set	Absolute	Character	Not Applicable	Value Of: 10000	Not Applicable	
N/A	Code Variable: life: pol_type_comm_heke f	Absolute	Character	Not Applicable	Retry With: default	Not Applicable	
N/A	Code Variable: life: channel	Absolute	Character	Not Applicable	Retry With: default	Not Applicable	

**6.2.1.4.1.2 comm\_extra\_agent\_tbl**

Description: comm\_extra\_agent

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: comm_ext_col_key	Absolute	Character	Not Applicable	Value Of: 10000	Not Applicable	
N/A	Code Variable: life: comm_set_temp	Absolute	Character	Not Applicable	Retry With: default	Not Applicable	
N/A	Code Variable: life: pol_type_comm_heke f	Absolute	Character	Not Applicable	Retry With: default	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: channel	Absolute	Character	Not Applicable	Retry With: default	Not Applicable	
N/A	Code Variable: life: temp_agency_no	Absolute	Character	Not Applicable	Retry With: default	Not Applicable	

**6.2.1.4.1.3 comm\_claw\_prpn\_tbl**

Description: commclaw

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: comm_ext_col_key	Absolute	Character	Not Applicable	Value Of: 9999	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: comm_claw_row_key	Absolute	Character	Not Applicable	Value Of: 10000	Not Applicable	

#### 6.2.1.4.1.4 comm\_ren\_perc\_prem\_mrtg

Description:

Parameters

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: Value	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: comm_ren_perc_prem_mrtg	Absolute	Character	Not Applicable	Error	Not Applicable	



## 6.2.1.5 Input Page: DAC

### 6.2.1.5.1 Assumption Set: Base

#### 6.2.1.5.1.1 dac\_amort\_type

Description:

Parameters

Column Lookup Details:

Key Names:

No

Final Column Uses Position:

No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: Value	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names:

No

Final Row Uses Position:

No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: dac_amort_type	Absolute	Character	Not Applicable	Error	Not Applicable	

#### 6.2.1.5.1.2 dac\_cap\_apply

Description:

Parameters

Column Lookup Details:

Key Names:

No

Final Column Uses Position:

No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: Value	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: dac_cap_apply	Absolute	Character	Not Applicable	Error	Not Applicable	

#### 6.2.1.5.1.3 dac\_book\_adj\_factor

Description: Economic

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: Current	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: dac_book_adj_factor	Absolute	Character	Not Applicable	Error	Not Applicable	

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Input Variable: life_Data: life: company	Absolute	Character	Not Applicable	Retry With: clal	Not Applicable	

#### 6.2.1.5.1.4 dac\_tax\_adj\_factor

Description: Economic

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: Current	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: dac_tax_adj_factor	Absolute	Character	Not Applicable	Error	Not Applicable	
N/A	Input Variable: life_Data: life: company	Absolute	Character	Not Applicable	Retry With: clal	Not Applicable	

## 6.2.1.6 Input Page: Economic

### 6.2.1.6.1 Assumption Set: Solv\_Base

#### 6.2.1.6.1.1 fund\_rates\_tbl

Description: fundrate

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: <*code_variable*>	Absolute	Character	Not Applicable	Error	Not Applicable	
N/A	Code Variable: life: sex	Absolute	Character	Not Applicable	Retry With: M	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Input Variable: life_Data: life: fund	Absolute	Character	Not Applicable	Error	Not Applicable	

#### 6.2.1.6.1.2 fund\_rates\_code\_tbl

Description: fundrate

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: temp_col_fund	Absolute	Character	Not Applicable	Error	Not Applicable	
N/A	Code Variable: life: sex	Absolute	Character	Not Applicable	Retry With: M	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Input Variable: life_Data: life: fund	Absolute	Character	Not Applicable	Error	Not Applicable	

**6.2.1.6.1.3 inv\_rates**

Description: Economic

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: Current	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
----------	-------------	---------------------	------	--------------	---------	------------	-------------------

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: <*code_variable*>	Absolute	Character	Not Applicable	Error	Not Applicable	
N/A	Input Variable: life_Data: life: company	Absolute	Character	Not Applicable	Retry With: clal	Not Applicable	

#### 6.2.1.6.1.4 Yield\_pre\_ret

Description: RFR\_Solv

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: <*code_variable*>	Absolute	Character	Not Applicable	Error	Not Applicable	
N/A	Code Variable: life: prod_group_yessodi_ portfolio	Absolute	Character	Not Applicable	Retry With: default	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: risk_free_row_key	Absolute	Numeric	First	Previous	Last	

#### 6.2.1.6.1.5 Yield\_post\_ret

Description: RFR\_Solv

## Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: <*code_variable*>	Absolute	Character	Not Applicable	Error	Not Applicable	
N/A	Code Variable: life: prod_group_yessodi_portfolio	Absolute	Character	Not Applicable	Retry With: default	Not Applicable	

## Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: risk_free_row_key	Absolute	Numeric	First	Previous	Last	

**6.2.1.6.1.6 Discounting\_pre**

Description: RFR\_Solv

## Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: inv_rate_m	Absolute	Character	Not Applicable	Error	Not Applicable	
N/A	Code Variable: life: prod_group_yessodi_portfolio	Absolute	Character	Not Applicable	Retry With: default	Not Applicable	

## Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: risk_free_row_key	Absolute	Numeric	First	Previous	Last	

**6.2.1.6.1.7 Yield\_pre\_ret\_ifrs**

Description: RFR\_IFRS

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: inv_rate_m	Absolute	Character	Not Applicable	Error	Not Applicable	
N/A	Constant: SAV_NPAR	Absolute	Character	Not Applicable	Retry With: default	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: risk_free_row_key	Absolute	Numeric	First	Previous	Last	

**6.2.1.6.1.8 Yield\_post\_ret\_ifrs**

Description: RFR\_IFRS

Column Lookup Details:



Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: ann_inv_rate_m	Absolute	Character	Not Applicable	Error	Not Applicable	
N/A	Constant: SAV_NPAR	Absolute	Character	Not Applicable	Retry With: default	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: risk_free_row_key	Absolute	Numeric	First	Previous	Last	

**6.2.1.6.1.9 Discounting\_pos**

Description: RFR\_Solv

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: ann_inv_rate_m	Absolute	Character	Not Applicable	Error	Not Applicable	
N/A	Code Variable: life: prod_group_yessodi_ portfolio	Absolute	Character	Not Applicable	Retry With: default	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: risk_free_row_key	Absolute	Numeric	First	Previous	Last	

**6.2.1.6.1.10 Discounting\_NoVA**

Description: RFR\_Solv

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: <*code_variable*>	Absolute	Character	Not Applicable	Error	Not Applicable	
N/A	Code Variable: life: prod_group_yessodi_portfolio	Absolute	Character	Not Applicable	Retry With: default	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: risk_free_row_key	Absolute	Numeric	First	Previous	Last	

**6.2.1.6.1.11 tax\_rate**

Description: tax\_rates

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: <*code_variable*>	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Column: life: cal_year	Absolute	Numeric	First	Previous	Last	

**6.2.1.6.1.12 CU\_Discounted**

Description: Parameters

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: Value	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: CU_Discounted	Absolute	Character	Not Applicable	Error	Not Applicable	

**6.2.1.6.1.13 fund\_group**

Description: fundrate

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: <*code_variable*>	Absolute	Character	Not Applicable	Error	Not Applicable	
N/A	Code Variable: life: sex	Absolute	Character	Not Applicable	Retry With: M	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: fund	Absolute	Character	Not Applicable	Error	Not Applicable	

**6.2.1.6.1.14 fundgroup\_manual**

Description: fundrate

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: <*code_variable*>	Absolute	Character	Not Applicable	Error	Not Applicable	

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: sex	Absolute	Character	Not Applicable	Retry With: M	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: fund	Absolute	Character	Not Applicable	Error	Not Applicable	

**6.2.1.6.1.15 free\_inv\_ratio\_tbl**

Description: FreeInvRatio

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Input Variable: life_Data: life: fund	Absolute	Character	Not Applicable	Retry With: 0	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: free_inv_row_key	Absolute	Numeric	Error	Previous	Last	

**6.2.1.6.1.16 fund\_rates\_tbl\_yesodi**

Description: fundrate

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: par_npar	Absolute	Character	Not Applicable	Error	Not Applicable	
N/A	Code Variable: life: sex	Absolute	Character	Not Applicable	Retry With: M	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Input Variable: life_Data: life: fund_yesodi	Absolute	Character	Not Applicable	Value Of: 0	Not Applicable	

**6.2.1.6.1.17 Economic\_Char**

Description: Economic

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: Current	Absolute	Character	Not Applicable	Error	Not Applicable	

## Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: <*code_variable*>	Absolute	Character	Not Applicable	Error	Not Applicable	
N/A	Input Variable: life_Data: life: company	Absolute	Character	Not Applicable	Retry With: clal	Not Applicable	

**6.2.1.6.1.18 Economic\_Num**

Description: Economic

## Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: Current	Absolute	Character	Not Applicable	Error	Not Applicable	

## Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: <*code_variable*>	Absolute	Character	Not Applicable	Error	Not Applicable	
N/A	Input Variable: life_Data: life: company	Absolute	Character	Not Applicable	Retry With: clal	Not Applicable	

**6.2.1.6.1.19 Cols\_of\_Money\_Prop**

Description: AnnuityTU

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: <*code_variable*>	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: annuity: takeup_age	Absolute	Numeric	Error	Error	Error	
N/A	Code Variable: annuity: sex1	Absolute	Character	Not Applicable	Error	Not Applicable	
N/A	Code Scalar: annuity: fund_type	Absolute	Character	Not Applicable	Retry With: P	Not Applicable	
N/A	Code Variable: life: ben_class	Absolute	Character	Not Applicable	Retry With: adif	Not Applicable	end: klasi gimla: klasi wol: klasi ytron: klasi adif: adif profil: profil
N/A	Code Variable: annuity: ann_series	Absolute	Numeric	Retry With: 1	Error	Error	
N/A	Code Variable: life: pol_type_annuity_tu	Absolute	Character	Not Applicable	Retry With: managers	Not Applicable	



**6.2.1.6.1.20 temp\_fund\_rates\_tbl ann**

Description: fundrate

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: ann_series	Absolute	Character	Not Applicable	Error	Not Applicable	
N/A	Code Variable: life: sex	Absolute	Character	Not Applicable	Retry With: M	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Scalar: annuity: temp_fund_scalar	Absolute	Character	Not Applicable	Error	Not Applicable	

**6.2.1.6.1.21 fund\_t\_factor**

Description: fundrate

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: <*code_variable*>	Absolute	Character	Not Applicable	Error	Not Applicable	

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: sex	Absolute	Character	Not Applicable	Retry With: M	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Scalar: annuity: temp_fund_scalar	Absolute	Character	Not Applicable	Error	Not Applicable	

#### 6.2.1.6.1.22 fund\_rates\_tbl temp

Description: fundrate

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: ann_series	Absolute	Character	Not Applicable	Error	Not Applicable	
N/A	Code Variable: life: sex	Absolute	Character	Not Applicable	Retry With: M	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
----------	-------------	---------------------	------	--------------	---------	------------	-------------------

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: fund_name_temp	Absolute	Character	Not Applicable	Retry With: 1	Not Applicable	

## 6.2.1.7 Input Page: Expenses

### 6.2.1.7.1 Assumption Set: Base

#### 6.2.1.7.1.1 exp\_dac\_perc

Description:

Parameters

Column Lookup Details:

Key Names:

No

Final Column Uses Position:

No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: Value	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names:

No

Final Row Uses Position:

No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: exp_dac_perc	Absolute	Character	Not Applicable	Error	Not Applicable	

#### 6.2.1.7.1.2 exp\_mult\_tbl

Description:

exp\_mult

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: exp_col_key	Absolute	Character	Not Applicable	Value Of: 99999	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: exp_row_key	Absolute	Character	Not Applicable	Value Of: 100000	Not Applicable	

**6.2.1.7.1.3 expense\_tbl**

Description: expense

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: <*code_variable*>	Absolute	Character	Not Applicable	Value Of: 100000	Not Applicable	
N/A	Code Variable: life: par_nonpar_yesodi	Absolute	Character	Not Applicable	Retry With: P	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: exp_row_lookup	Absolute	Character	Not Applicable	Value Of: 99999	Not Applicable	
N/A	Input Variable: life_Data: life: channel	Absolute	Character	Not Applicable	Retry With: default	Not Applicable	

## 6.2.1.8 Input Page: Lapses

### 6.2.1.8.1 Assumption Set: Solv\_Base

#### 6.2.1.8.1.1 various parameters

Description:

Parameters

Column Lookup Details:

Key Names:

No

Final Column Uses Position:

No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: Value	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names:

No

Final Row Uses Position:

No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: <*code_variable*>	Absolute	Character	Not Applicable	Error	Not Applicable	

#### 6.2.1.8.1.2 sur\_val\_method

Description:

Parameters

## Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: Value	Absolute	Character	Not Applicable	Error	Not Applicable	

## Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: <*code_variable*>	Absolute	Character	Not Applicable	Error	Not Applicable	

**6.2.1.8.1.3 lapse\_factor\_y1**

Description: lapse\_factor

## Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: lapse_factor_y1_row	Absolute	Character	Not Applicable	Value Of: - 99999	Not Applicable	

## Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: agency_no_lookup	Absolute	Character	Not Applicable	Value Of: 99999	Not Applicable	
N/A	Input Variable: life_Data: life: ben_class	Absolute	Character	Not Applicable	Retry With: default	Not Applicable	

#### 6.2.1.8.1.4 lapse\_clawback\_factor

Description: lapse\_factor

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: claw_fact_set	Absolute	Character	Not Applicable	Value Of: - 99999	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: agency_no_lookup	Absolute	Character	Not Applicable	Value Of: 99999	Not Applicable	
N/A	Input Variable: life_Data: life: ben_class	Absolute	Character	Not Applicable	Retry With: default	Not Applicable	

#### 6.2.1.8.1.5 lapse\_factor\_yplus

Description: lapse\_factor

## Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: lapse_factor_y_col	Absolute	Character	Not Applicable	Value Of: - 99999	Not Applicable	

## Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: agency_no_lookup	Absolute	Character	Not Applicable	Value Of: 99999	Not Applicable	
N/A	Input Variable: life_Data: life: ben_class	Absolute	Character	Not Applicable	Retry With: default	Not Applicable	

**6.2.1.8.1.6 lapse\_rate\_im**

Description: lapse

## Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Column: life: pol_year_ext	Absolute	Numeric	Error	Error	Last	

## Row Lookup Details:

Key Names: No

Final Row Uses Position: No



Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: lapse_set	Absolute	Character	Not Applicable	Error	Not Applicable	
N/A	Code Variable: life: pol_type_lapse	Absolute	Character	Not Applicable	Retry With: default	Not Applicable	
N/A	Constant: 0	Absolute	Numeric	Error	Retry With: 0	Error	
N/A	Constant: n/a	Absolute	Character	Not Applicable	Retry With: n/a	Not Applicable	
N/A	Code Variable: life: lapse_tarif_set	Absolute	Character	Not Applicable	Retry With: 0	Not Applicable	
N/A	Code Variable: life: pup_ind	Absolute	Numeric	Error	Retry With: 0	Error	
N/A	Code Column: life: age_last	Absolute	Numeric	First	Previous	Last	
N/A	Code Variable: life: fund_group	Absolute	Character	Not Applicable	Retry With: P	Not Applicable	
N/A	Input Variable: life_Data: life: channel	Absolute	Character	Not Applicable	Retry With: default	Not Applicable	
N/A	Code Variable: life: foreign_id	Absolute	Numeric	Retry With: 0	Retry With: 0	Retry With: 0	
N/A	Code Variable: life: lapse_type_col_key	Absolute	Character	Not Applicable	Error	Not Applicable	
N/A	Code Variable: life: lapse_expos_col_key	Absolute	Character	Not Applicable	Error	Not Applicable	

#### 6.2.1.8.1.7 lapse\_rider\_profil\_dth

Description: lapse

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: 0	Absolute	Numeric	Error	Error	Last	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: profil	Absolute	Character	Not Applicable	Error	Not Applicable	
N/A	Code Variable: life: pol_type_lapse_rider	Absolute	Character	Not Applicable	Retry With: default	Not Applicable	
N/A	Constant: 1	Absolute	Numeric	Error	Error	Error	
N/A	Code Variable: life: prod_assumpt_rider_lapse_tbl	Absolute	Character	Not Applicable	Retry With: n/a	Not Applicable	
N/A	Code Variable: life: lapse_tarif_set	Absolute	Character	Not Applicable	Retry With: 0	Not Applicable	
N/A	Constant: 0	Absolute	Numeric	Error	Retry With: 0	Error	
N/A	Constant: 0	Absolute	Numeric	First	Previous	Last	
N/A	Code Variable: life: fund_group	Absolute	Character	Not Applicable	Retry With: P	Not Applicable	
N/A	Input Variable: life_Data: life: channel	Absolute	Character	Not Applicable	Retry With: default	Not Applicable	
N/A	Code Variable: life: foreign_id	Absolute	Numeric	Retry With: 0	Retry With: 0	Retry With: 0	

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: lapse_type_col_key	Absolute	Character	Not Applicable	Error	Not Applicable	
N/A	Code Variable: life: lapse_expos_col_key	Absolute	Character	Not Applicable	Error	Not Applicable	

#### 6.2.1.8.1.8 lapse\_rider\_other

Description: lapse

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Column: life: pol_year_ext	Absolute	Numeric	Error	Error	Last	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: lapse_set	Absolute	Character	Not Applicable	Error	Not Applicable	
N/A	Code Variable: life: pol_type_lapse_rider	Absolute	Character	Not Applicable	Retry With: default	Not Applicable	
N/A	Constant: 1	Absolute	Numeric	Error	Retry With: 1	Error	
N/A	Code Variable: life: lapse_set_riders	Absolute	Character	Not Applicable	Retry With: default	Not Applicable	

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: lapse_tarif_set	Absolute	Character	Not Applicable	Retry With: 0	Not Applicable	
N/A	Constant: 0	Absolute	Numeric	Error	Retry With: 0	Error	
N/A	Code Column: life: age_last	Absolute	Numeric	First	Previous	Last	
N/A	Code Variable: life: fund_group	Absolute	Character	Not Applicable	Retry With: P	Not Applicable	
N/A	Input Variable: life_Data: life: channel	Absolute	Character	Not Applicable	Retry With: default	Not Applicable	
N/A	Code Variable: life: foreign_id	Absolute	Numeric	Retry With: 0	Retry With: 0	Retry With: 0	
N/A	Code Variable: life: lapse_type_col_key	Absolute	Character	Not Applicable	Error	Not Applicable	
N/A	Code Variable: life: lapse_expos_col_key	Absolute	Character	Not Applicable	Error	Not Applicable	

#### 6.2.1.8.1.9 lapse\_rate\_pup\_im

Description: lapse

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Column: life: pol_year_ext	Absolute	Numeric	Error	Error	Last	

Row Lookup Details:

Key Names: No

Final Row Uses Position:

No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: lapse_set	Absolute	Character	Not Applicable	Error	Not Applicable	
N/A	Code Variable: life: pol_type_lapse	Absolute	Character	Not Applicable	Retry With: default	Not Applicable	
N/A	Constant: 0	Absolute	Numeric	Error	Retry With: 0	Error	
N/A	Constant: n/a	Absolute	Character	Not Applicable	Retry With: n/a	Not Applicable	
N/A	Code Variable: life: lapse_tarif_set	Absolute	Character	Not Applicable	Retry With: 0	Not Applicable	
N/A	Constant: 1	Absolute	Numeric	Error	Retry With: 1	Error	
N/A	Code Column: life: age_last	Absolute	Numeric	First	Previous	Last	
N/A	Code Variable: life: fund_group	Absolute	Character	Not Applicable	Retry With: P	Not Applicable	
N/A	Input Variable: life_Data: life: channel	Absolute	Character	Not Applicable	Retry With: default	Not Applicable	
N/A	Code Variable: life: foreign_id	Absolute	Numeric	Retry With: 0	Retry With: 0	Retry With: 0	
N/A	Code Variable: life: lapse_type_col_key	Absolute	Character	Not Applicable	Error	Not Applicable	
N/A	Code Variable: life: lapse_expos_col_key	Absolute	Character	Not Applicable	Error	Not Applicable	

**6.2.1.8.1.10 puv\_09\_tbl**

Description:

puv\_composite

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: puv_col_key	Absolute	Character	Not Applicable	Value Of: 0	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: puv_row_key	Absolute	Character	Not Applicable	Value Of: 0	Not Applicable	

**6.2.1.8.1.11 masslaps\_tbl**

Description: mass\_lapse\_tab

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: col_char	Absolute	Character	Not Applicable	Value Of: - 999999	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
----------	-------------	---------------------	------	--------------	---------	------------	-------------------

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: flag_code	Absolute	Numeric	Error	Value Of: - 999999	Error	

#### 6.2.1.8.1.12 **surr\_chg\_tbl**

Description: surr\_chg

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: col_char	Absolute	Character	Not Applicable	Value Of: - 999999	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	System Variable: r	Absolute	Numeric	Error	Value Of: -9999	Last	

**6.2.1.8.1.13**      **pup\_ltc\_tbl**

Description: pup

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
----------	-------------	---------------------	------	--------------	---------	------------	-------------------

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Column: pup_ltc_col	Relative	Numeric	First	Previous	Last	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Input Variable: life_Data: age_exact_issue	Relative	Numeric	First	Previous	Last	

#### 6.2.1.8.1.14 pup\_ltc\_tbl\_next

Description: pup

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Column: pup_ltc_col_next	Relative	Numeric	First	Previous	Last	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Input Variable: life_Data: age_exact_issue	Relative	Numeric	First	Previous	Last	



**6.2.1.8.1.15 lapse\_factor\_proj**

Description: lapse\_factor\_proj

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Column: life: cal_year	Absolute	Numeric	First	Previous	Last	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: lapse_set	Absolute	Character	Not Applicable	Value Of: 100	Not Applicable	
N/A	Code Variable: life: pol_type_lapse	Absolute	Character	Not Applicable	Retry With: default	Not Applicable	
N/A	Constant: 0	Absolute	Numeric	Error	Retry With: 0	Error	

**6.2.1.8.1.16 lapse\_factor\_proj\_rider**

Description: lapse\_factor\_proj

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
----------	-------------	---------------------	------	--------------	---------	------------	-------------------

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Column: life: cal_year	Absolute	Numeric	First	Previous	Last	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: lapse_set	Absolute	Character	Not Applicable	Value Of: 100	Not Applicable	
N/A	Code Variable: life: pol_type_lapse	Absolute	Character	Not Applicable	Retry With: default	Not Applicable	
N/A	Constant: 1	Absolute	Numeric	Error	Retry With: 0	Error	

**6.2.1.8.1.17 lapse\_factor\_profil\_rider**

Description: lapse\_factor\_proj

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: valn_year	Absolute	Numeric	Retry With: 0	Previous	Last	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: lapse_set	Absolute	Character	Not Applicable	Value Of: 100	Not Applicable	
N/A	Code Variable: life: pol_type_lapse	Absolute	Character	Not Applicable	Retry With: default	Not Applicable	
N/A	Constant: 1	Absolute	Numeric	Error	Retry With: 0	Error	

## 6.2.1.9 Input Page: Margins

### 6.2.1.9.1 Assumption Set: Solv\_Base

#### 6.2.1.9.1.1 asset\_shock

Description: Asset\_Shocks

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: Base	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: ben_class	Absolute	Character	Not Applicable	Value Of: 0	Not Applicable	

**6.2.1.9.1.2 prem\_disc\_scenario**

Description: Discount\_Scenarios

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: prem_disc	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Column: life: proj_year	Absolute	Numeric	Retry With: 0	Previous	Last	

**6.2.1.9.1.3 mgt\_fee\_disc**

Description: Discount\_Scenarios

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: margin_disc_col_key	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Column: life: proj_year	Absolute	Numeric	Retry With: 0	Previous	Last	

**6.2.1.9.1.4 Margins\_Char**

Description: margins

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: Base	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: <*code_variable*>	Absolute	Character	Not Applicable	Error	Not Applicable	
N/A	System Variable: group	Absolute	Character	Not Applicable	Retry With: default	Not Applicable	

**6.2.1.9.1.5 Margins\_Number**

Description: margins

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: Base	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: <*code_variable*>	Absolute	Character	Not Applicable	Error	Not Applicable	
N/A	System Variable: group	Absolute	Character	Not Applicable	Retry With: default	Not Applicable	

## 6.2.1.10 Input Page: Mortality

### 6.2.1.10.1 Assumption Set: Base

#### 6.2.1.10.1.1 death\_rates\_tbl

Description: death\_rates\_comp

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: col_dth	Relative	Numeric	First	Error	Last	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: death_rate_row_key	Relative	Numeric	First	Error	Last	

**6.2.1.10.1.2 sv\_09\_tbl**

Description: sv\_composite

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: sv_col_key	Relative	Character	Not Applicable	Value Of: 0	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: sv_row_key	Relative	Character	Not Applicable	Value Of: 0	Not Applicable	

**6.2.1.10.1.3 sv\_09\_tbl\_check**

Description: sv\_composite

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: sv_col_key	Relative	Character	Not Applicable	Value Of: 0	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: sv_row_key	Relative	Character	Not Applicable	Value Of: 99999	Not Applicable	

#### 6.2.1.10.1.4 decrement rates

Description: decrem\_rates\_com

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: decrem_rate_key	Relative	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: row_num	Relative	Numeric	Error	Error	Last	



**6.2.1.10.1.5 decrement rates check**

Description: decrement\_rates\_com

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: col_char	Relative	Character	Not Applicable	Value Of: -999999	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: 30	Relative	Numeric	Error	Error	Error	

**6.2.1.10.1.6 decrement rates by UW date**

Description: decrement\_rates\_uw\_com

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: decrement_rate_key	Relative	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Scalar: origdate	Relative	Numeric	Error	Previous	Last	
N/A	Code Variable: row_num	Relative	Numeric	Error	Error	Last	

**6.2.1.10.1.7      decrem\_mult\_tbl**

Description: decrmult

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: decrem_mult_col_key	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: decrem_mult_row_key	Absolute	Character	Not Applicable	Error	Not Applicable	

**6.2.1.10.1.8      mort\_mult\_end\_age**

Description: Parameters

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: Value	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: mort_mult_end_age	Absolute	Character	Not Applicable	Error	Not Applicable	

#### 6.2.1.10.1.9 antisel\_margin

Description: Parameters

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: Value	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: antisel_margin	Absolute	Character	Not Applicable	Error	Not Applicable	

**6.2.1.10.1.10 mort\_mult\_tbl**

Description: mortmult

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: mort_mult_col_key	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: mort_mult_set	Absolute	Character	Not Applicable	Error	Not Applicable	

**6.2.1.10.1.11 survive\_tbl**

Description: Survival\_Rates

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Column: life: retirement_age_lookup	Absolute	Numeric	First	Previous	Last	
N/A	Code Variable: life: sex	Absolute	Character	Not Applicable	Error	Not Applicable	

## Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Column: life: age_last	Absolute	Numeric	Error	Error	Error	

**6.2.1.10.1.12 select\_periods**

Description: Parameters

## Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: Value	Absolute	Character	Not Applicable	Error	Not Applicable	

## Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: <*code_variable*>	Absolute	Character	Not Applicable	Error	Not Applicable	

**6.2.1.10.1.13 omega\_age**

Description: Parameters

## Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: Value	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: <*code_variable*>	Absolute	Character	Not Applicable	Error	Not Applicable	

**6.2.1.10.1.14 death\_rates\_ann\_m\_1**

Description: ann\_mort\_08\_M\_BE

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Column: life: mort_year	Absolute	Numeric	First	Previous	Last	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Column: annuity:	Absolute	Numeric	First	Previous	Last	

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
	death_rates_row_1						

**6.2.1.10.1.15 sel\_ret\_qx\_im\_dth\_1**

Description: Sel\_Ret\_Qx

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: 1	Absolute	Numeric	Error	Error	Error	
N/A	Code Variable: annuity: sel_death_rate_col	Absolute	Numeric	First	Previous	Last	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: annuity: takeup_age	Absolute	Numeric	First	Previous	Last	
N/A	Code Variable: annuity: sex1	Absolute	Character	Not Applicable	Error	Not Applicable	

**6.2.1.10.1.16 sel\_ret\_qx\_im\_dth\_2**

Description: Sel\_Ret\_Qx

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: 2	Absolute	Numeric	Error	Error	Error	
N/A	Code Variable: annuity: sel_death_rate_col	Absolute	Numeric	First	Previous	Last	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: annuity: takeup_age	Absolute	Numeric	First	Previous	Last	
N/A	Code Variable: annuity: sex2	Absolute	Character	Not Applicable	Error	Not Applicable	

**6.2.1.10.1.17 death\_rates\_ann\_f\_1**

Description: ann\_mort\_08\_F\_BE

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Column: life: mort_year	Absolute	Numeric	First	Previous	Last	

Row Lookup Details:

Key Names: No



Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Column: annuity: death_rates_row_1	Absolute	Numeric	First	Previous	Last	

**6.2.1.10.1.18 death\_rates\_ann\_m\_2**

Description: ann\_mort\_08\_M\_BE

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Column: life: mort_year	Absolute	Numeric	First	Previous	Last	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Column: annuity: death_rates_row_2	Absolute	Numeric	First	Previous	Last	

**6.2.1.10.1.19 death\_rates\_ann\_f\_2**

Description: ann\_mort\_08\_F\_BE

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Column: life: mort_year	Absolute	Numeric	First	Previous	Last	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Column: annuity: death_rates_row_2	Absolute	Numeric	First	Previous	Last	

#### 6.2.1.10.1.20 death\_rates\_ann\_m\_b3\_2

Description: ann\_mort\_b3\_08\_M\_BE

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Column: life: mort_year	Absolute	Numeric	First	Previous	Last	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Column: annuity: death_rates_row_2	Absolute	Numeric	First	Previous	Last	

**6.2.1.10.1.21 death\_rates\_ann\_f\_b3\_2**

Description: ann\_mort\_b3\_08\_F\_BE

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Column: life: mort_year	Absolute	Numeric	First	Previous	Last	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Column: annuity: death_rates_row_2	Absolute	Numeric	First	Previous	Last	

**6.2.1.10.1.22 death\_rates\_ann\_m\_res\_1**

Description: ann\_mort\_08\_M\_res

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Column: life: mort_year	Absolute	Numeric	First	Previous	Last	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Column: annuity: death_rates_row_1	Absolute	Numeric	First	Previous	Last	

**6.2.1.10.1.23 death\_rates\_ann\_m\_res\_2**

Description: ann\_mort\_08\_M\_res

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Column: life: mort_year	Absolute	Numeric	First	Previous	Last	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Column: annuity: death_rates_row_2	Absolute	Numeric	First	Previous	Last	

**6.2.1.10.1.24 death\_rates\_ann\_m\_res\_tt**

Description: ann\_mort\_08\_M\_res

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: mort_year_tt	Absolute	Numeric	First	Previous	Last	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: annuity: dth_rts_m_row_key_t t	Absolute	Numeric	First	Previous	Last	

**6.2.1.10.1.25 death\_rates\_ann\_f\_res\_1**

Description: ann\_mort\_08\_F\_res

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Column: life: mort_year	Absolute	Numeric	First	Previous	Last	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Column: annuity:	Absolute	Numeric	First	Previous	Last	

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
	death_rates_row_1						

**6.2.1.10.1.26 death\_rates\_ann\_f\_res\_2**

Description: ann\_mort\_08\_F\_res

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Column: life: mort_year	Absolute	Numeric	First	Previous	Last	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Column: annuity: death_rates_row_2	Absolute	Numeric	First	Previous	Last	

**6.2.1.10.1.27 death\_rates\_ann\_f\_res\_tt**

Description: ann\_mort\_08\_F\_res

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: mort_year_tt	Absolute	Numeric	First	Previous	Last	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: annuity: dth_rts_m_row_key_t t	Absolute	Numeric	First	Previous	Last	

**6.2.1.10.1.28 death\_rates\_ann\_m\_res\_b3\_2**

Description: ann\_mort\_b3\_08\_M\_res

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Column: life: mort_year	Absolute	Numeric	First	Previous	Last	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Column: annuity:	Absolute	Numeric	First	Previous	Last	

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
	death_rates_row_2						

**6.2.1.10.1.29 death\_rates\_ann\_m\_res\_b3\_tt**

Description: ann\_mort\_b3\_08\_M\_res

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: mort_year_tt	Absolute	Numeric	First	Previous	Last	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: annuity: dth_rts_m_row_key_t t	Absolute	Numeric	First	Previous	Last	

**6.2.1.10.1.30 death\_rates\_ann\_f\_res\_b3\_2**

Description: ann\_mort\_b3\_08\_F\_res

Column Lookup Details:

Key Names: No

Final Column Uses Position: No



Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Column: life: mort_year	Absolute	Numeric	First	Previous	Last	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Column: annuity: death_rates_row_2	Absolute	Numeric	First	Previous	Last	

**6.2.1.10.1.31 death\_rates\_ann\_f\_res\_b3\_tt**

Description: ann\_mort\_b3\_08\_F\_res

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: mort_year_tt	Absolute	Numeric	First	Previous	Last	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: annuity: dth_rts_m_row_key_t	Absolute	Numeric	First	Previous	Last	

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
	t						

6.2.1.10.1.32 death\_rates\_res\_tbl

Description: death\_rates\_res\_comp

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: sex	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: row_num	Absolute	Numeric	Error	Error	Last	

6.2.1.11 Input Page: Premium

6.2.1.11.1 Assumption Set: Base

6.2.1.11.1.1 Various\_Parameters

Description: Parameters

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: Value	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: <*code_variable*>	Absolute	Character	Not Applicable	Error	Not Applicable	

#### 6.2.1.11.1.2 prem\_rates\_series\_end\_im

Description: prem\_rates

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: Series_End	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
----------	-------------	---------------------	------	--------------	---------	------------	-------------------

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: row_char	Absolute	Character	Not Applicable	Error	Not Applicable	

#### 6.2.1.11.1.3 prem\_rates\_series

Description: prem\_rates

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: Series_End	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: prem_code_test_tem p	Absolute	Character	Not Applicable	Value Of: - 99999	Not Applicable	

#### 6.2.1.11.1.4 prem\_key\_temp\_rates

Description: prem\_rates

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: Series_End	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: prem_key_temp	Absolute	Character	Not Applicable	Error	Not Applicable	

#### 6.2.1.11.1.5 prem\_rates\_charge\_tt

Description: prem\_rates

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: charge_rate_tt_col	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: charge_rate_tt_row	Absolute	Character	Not Applicable	Error	Not Applicable	

**6.2.1.11.1.6 prem\_if\_rates**

Description: prem\_rates

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: charge_rate_tt_col	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: prem_code	Absolute	Character	Not Applicable	Value Of: 0	Not Applicable	

**6.2.1.11.1.7 prem\_rates\_others**

Description: prem\_rates

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: <*code_variable*>	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: prem_code	Absolute	Character	Not Applicable	Error	Not Applicable	

**6.2.1.11.1.8 prem\_rates\_si**

Description: prem\_rates

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: prem_rates_si_col	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: prem_rates_si_row	Absolute	Character	Not Applicable	Error	Not Applicable	

**6.2.1.11.1.9 sal\_tbl**

Description: sal\_inc

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Column: life: age_last	Absolute	Numeric	First	Previous	Last	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: pol_type_sal_inc	Absolute	Character	Not Applicable	Retry With: default	Not Applicable	
N/A	Code Variable: life: sal_inc_set	Absolute	Character	Not Applicable	Retry With: default	Not Applicable	
N/A	Code Variable: life: term_in_profil	Absolute	Character	Not Applicable	Retry With: N	Not Applicable	

#### 6.2.1.11.1.10 sal\_rider\_tbl

Description: sal\_inc

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: 0	Absolute	Numeric	First	Previous	Last	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No



Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: pol_type_sal_inc	Absolute	Character	Not Applicable	Retry With: default	Not Applicable	
N/A	Code Variable: life: sal_inc_set_rider	Absolute	Character	Not Applicable	Retry With: default	Not Applicable	
N/A	Constant: Y	Absolute	Character	Not Applicable	Retry With: N	Not Applicable	

#### 6.2.1.11.1.11 prem\_rates\_extra\_prm

Description: prem\_rates\_extra

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: prem_rate_col	Absolute	Character	Not Applicable	Value Of: 99999	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: prem_rate_row	Relative	Numeric	First	Value Of: - 99999	Last	

#### 6.2.1.11.1.12 prem\_rates\_risk\_1

Description: prem\_rates\_risk\_comp

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: MS	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: -1	Absolute	Numeric	Error	Error	Error	

**6.2.1.11.1.13 prem\_rates\_risk\_2**

Description: prem\_rates\_risk\_comp

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: FS	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: -1	Absolute	Numeric	Error	Error	Error	

#### 6.2.1.11.1.14 prem\_rates\_risk\_rider

Description: prem\_rates\_risk\_rider

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: col_char	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: prem_rates_row	Absolute	Numeric	Error	Error	Error	

#### 6.2.1.11.1.15 prem\_code\_map\_tbl

Description: prem\_code\_map

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: col	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: row	Absolute	Character	Not Applicable	Error	Not Applicable	

#### 6.2.1.11.1.16 prate\_level\_tbl

Description: prem\_rates\_level\_comp

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: col_char	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: row_num	Absolute	Numeric	Error	Error	Error	

**6.2.1.11.1.17 zillmer\_pr\_tbl**

Description: zillmer\_prm

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Column: life: pol_year	Absolute	Numeric	First	Error	Last	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: row_char	Absolute	Character	Not Applicable	Error	Not Applicable	

**6.2.1.11.1.18 prem\_disc\_shimur**

Description: Shimur\_disc

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: Base	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: ben_class	Absolute	Character	Not Applicable	Retry With: default	Not Applicable	
N/A	Code Column: life: pol_year_ext	Absolute	Numeric	Value Of: 0	Previous	Last	

## 6.2.1.12 Input Page: Product Details

### 6.2.1.12.1 Assumption Set: Base

#### 6.2.1.12.1.1 prod\_assumpt\_tbl

Description: prod\_ass

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: <*code_variable*>	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Input Variable: life_Data: life: prod_code	Absolute	Character	Not Applicable	Value Of: N/A	Not Applicable	

**6.2.1.12.1.2 savings\_pol\_prod\_code**

Description: prod\_ass

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: savings_pol	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: prod_code	Absolute	Character	Not Applicable	Value Of: N/A	Not Applicable	

**6.2.1.12.1.3 prod\_assumpt\_base\_tbl**

Description: prod\_ass

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: <*code_variable*>	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Input Variable: life_Data: life: prod_code_base	Absolute	Character	Not Applicable	Value Of: N/A	Not Applicable	

**6.2.1.12.1.4 prod\_assumpt\_key\_tbl**

Description: prod\_ass

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: key_temp	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Input Variable: life_Data: life: prod_code	Absolute	Character	Not Applicable	Value Of: N/A	Not Applicable	

**6.2.1.12.1.5 prod\_assumpt\_rider\_exp\_tbl**

Description: prod\_ass

Column Lookup Details:

Key Names: No

Final Column Uses Position: No



Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: exp_set_cvr	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: prod_code_rider	Absolute	Character	Not Applicable	Value Of: N/A	Not Applicable	

#### 6.2.1.12.1.6 sal\_inc\_set\_rider

Description: prod\_ass

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: sal_inc_set	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: prod_code_rider	Absolute	Character	Not Applicable	Value Of: N/A	Not Applicable	

**6.2.1.12.1.7 prod\_assumpt\_rider\_lapse\_tbl**

Description: prod\_ass

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: lapse_set_riders	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: prod_code_rider	Absolute	Character	Not Applicable	Value Of: N/A	Not Applicable	

**6.2.1.12.1.8 prod\_assumpt\_rider\_clms\_tbl**

Description: prod\_ass

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: clms_mult_set	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: prod_code_rider	Absolute	Character	Not Applicable	Value Of: N/A	Not Applicable	

**6.2.1.12.1.9 prod spec term**

Description: prod\_spec\_term

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: <*code_variable*>	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Input Variable: life_Data: life: prod_code	Absolute	Character	Not Applicable	Value Of: N/A	Not Applicable	

**6.2.1.12.1.10 prod spec trad**

Description: prod\_spec\_trad

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: <*code_variable*>	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Input Variable: life_Data: life: prod_code	Absolute	Character	Not Applicable	Value Of: N/A	Not Applicable	

**6.2.1.12.1.11 prem\_lookup\_trad**

Description: prod\_spec\_trad

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: prem_lookup	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Input Variable: life_Data: life: prod_code	Absolute	Character	Not Applicable	Value Of: N/A	Not Applicable	

**6.2.1.12.1.12 prem\_lookup\_freq\_trad**

Description: prod\_spec\_trad

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: prem_lookup_freq	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Input Variable: life_Data: life: prod_code	Absolute	Character	Not Applicable	Value Of: N/A	Not Applicable	

**6.2.1.12.1.13 pup\_sv\_charge\_rebate**

Description: prod\_spec\_unit

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: pup_sv_charge_rebate	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Input Variable: life_Data: life: prod_code	Absolute	Character	Not Applicable	Value Of: N/A	Not Applicable	

**6.2.1.12.1.14 prod\_specs\_max\_perc**

Description: prod\_spec\_unit

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Scalar: life: prod_specs_max_per c	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Input Variable: life_Data: life: prod_code	Absolute	Character	Not Applicable	Value Of: N/A	Not Applicable	

**6.2.1.12.1.15 prod\_spec\_risk\_code**

Description: prod\_spec\_unit

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: prem_lookup_freq	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: risk_code	Absolute	Character	Not Applicable	Value Of: N/A	Not Applicable	

**6.2.1.12.1.16 prod\_specs\_rider**

Description: prod\_spec\_unit

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: prod_specs_rider_col	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: prod_code_rider	Absolute	Character	Not Applicable	Value Of: N/A	Not Applicable	

**6.2.1.12.1.17 rider\_tarif\_tbl**

Description: profil\_rider\_tarif\_map

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: procd	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: rider_tarif_row_key	Absolute	Character	Not Applicable	Value Of: N/A	Not Applicable	

**6.2.1.12.1.18 tarif\_spec**

Description: tarif\_spec

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: <*code_variable*>	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No



Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: tarif_spec_row_key	Absolute	Character	Not Applicable	Value Of: N/A	Not Applicable	

**6.2.1.12.1.19 tarif\_spec\_lookup\_freq**

Description: tarif\_spec

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: Prem_Lookup_Freq	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: tarif_spec_row_key	Absolute	Character	Not Applicable	Value Of: N/A	Not Applicable	

**6.2.1.12.1.20 alloc\_rate\_stri**

Description: alloc

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: stri	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: alloc_rate_row	Absolute	Character	Not Applicable	Error	Not Applicable	

#### 6.2.1.12.1.21 suminisba\_tbl

Description: suminisba

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: suminsbas_col	Relative	Character	Not Applicable	Value Of: 99999	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: suminsbas_row	Relative	Numeric	Value Of: - 99999	Value Of: - 99999	Value Of: - 99999	

**6.2.1.12.1.22 claims\_factor\_occ**

Description: tarif\_spec\_occ

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: occ_key	Absolute	Character	Not Applicable	Retry With: 3	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: tarif	Absolute	Numeric	Value Of: 100	Value Of: 100	Value Of: 100	

**6.2.1.12.1.23 bonus\_tbl**

Description: bonus5

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: surr_charge_set_tem p	Absolute	Character	Not Applicable	Retry With: managers_z ero	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: bonus_tbl_row	Absolute	Numeric	Error	Error	Error	

**6.2.1.12.1.24    gorem\_mult**

Description: prod\_ass

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: <*code_variable*>	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Input Variable: life_Data: life: prod_code	Absolute	Character	Not Applicable	Error	Not Applicable	

**6.2.1.13        Input Page: Reinsurance****6.2.1.13.1      Assumption Set: Base****6.2.1.13.1.1    prem\_rates\_re**

Description: premium\_rates\_rein\_life

## Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Column: life: age_last	Absolute	Numeric	Error	Error	Last	

## Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: prem_re_row_key	Absolute	Character	Not Applicable	Value Of: 0	Not Applicable	
N/A	Code Variable: life: prem_re_sex	Absolute	Character	Not Applicable	Retry With: default	Not Applicable	
N/A	Code Variable: life: prem_re_bw	Absolute	Character	Not Applicable	Retry With: default	Not Applicable	
N/A	Code Variable: life: prem_re_wp	Absolute	Character	Not Applicable	Retry With: default	Not Applicable	
N/A	Code Variable: life: prem_re_occ	Absolute	Character	Not Applicable	Retry With: default	Not Applicable	
N/A	Code Variable: life: prem_re_endage	Absolute	Character	Not Applicable	Retry With: default	Not Applicable	
N/A	Code Variable: life: policy_type	Absolute	Character	Not Applicable	Retry With: default	Not Applicable	
N/A	Code Scalar: origdate	Relative	Numeric	First	Previous	Last	

## 6.2.1.13.1.2 reinsur\_w

Description: life\_treaty\_details

## Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: <*code_variable*>	Absolute	Character	Not Applicable	Error	Not Applicable	

## Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: rein_set	Absolute	Character	Not Applicable	Retry With: 0_0	Not Applicable	

**6.2.1.13.1.3      rein\_series\_end\_key\_temp**

Description: life\_treaty\_details

## Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: Series_End	Absolute	Character	Not Applicable	Error	Not Applicable	

## Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: rein_key_temp	Absolute	Character	Not Applicable	Retry With: 0_0	Not Applicable	

**6.2.1.13.1.4 reinsur\_kod\_tavla**

Description: life\_treaty\_details

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Scalar: life: reinsur_kodtavla	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: rein_set	Absolute	Character	Not Applicable	Retry With: 0_0	Not Applicable	

**6.2.1.13.1.5 reinsur\_comm**

Description: life\_treaty\_details

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: reinsur_comm_key	Absolute	Character	Not Applicable	Value Of: 99999	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: rein_set	Absolute	Character	Not Applicable	Value Of: - 99999	Not Applicable	

#### 6.2.1.13.1.6 reinsur\_simple\_perc

Description: LifeReins

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: ReinsPerc	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Scalar: life: reins_simple_row	Absolute	Character	Not Applicable	Value Of: - 99999	Not Applicable	



**6.2.1.13.1.7 reinsur\_simple\_cost**

Description: LifeReins

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: ReinsCost	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Scalar: life: reins_simple_row	Absolute	Character	Not Applicable	Error	Not Applicable	

**6.2.1.13.1.8 reinsur\_simple\_rider\_cost**

Description: LifeReins

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: ReinsCost	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Scalar: life: reins_simple_rider_row	Absolute	Character	Not Applicable	Value Of: -99999	Not Applicable	

## 6.2.1.14 Input Page: Reserve

### 6.2.1.14.1 Assumption Set: Base

#### 6.2.1.14.1.1 err\_sar\_perc

Description: Parameters

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: Value	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: err_sar_perc	Absolute	Character	Not Applicable	Error	Not Applicable	

#### 6.2.1.14.1.2 err\_spread\_period

Description: Parameters

## Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: Value	Absolute	Character	Not Applicable	Error	Not Applicable	

## Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: err_spread_period	Absolute	Character	Not Applicable	Error	Not Applicable	

**6.2.1.14.1.3 zeroise\_res**

Description: Parameters

## Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: Value	Absolute	Character	Not Applicable	Error	Not Applicable	

## Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: zeroise_res	Absolute	Character	Not Applicable	Error	Not Applicable	

#### 6.2.1.14.1.4 reserve\_factors\_tbl

Description: Reserve\_Factors

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: prod_code	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: res_fac_row_key	Absolute	Numeric	Error	Error	Error	

#### 6.2.1.14.1.5 zillmer\_adj\_factor

Description: Economic

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: Current	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: zillmer_adj_factor	Absolute	Character	Not Applicable	Error	Not Applicable	
N/A	Input Variable: life_Data: life: company	Absolute	Character	Not Applicable	Retry With: clal	Not Applicable	

6.2.1.14.1.6      AnnuitySets

Description: Annuity

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: Assumption	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
----------	-------------	---------------------	------	--------------	---------	------------	-------------------

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: <*code_variable*>	Absolute	Character	Not Applicable	Error	Not Applicable	
N/A	Input Variable: life_Data: life: sex	Absolute	Character	Not Applicable	Retry With: M	Not Applicable	

#### 6.2.1.14.1.7 Res\_Adj\_Factor

Description: Reserve\_Manual

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: Adj_Factor	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: fundgroup_manual	Absolute	Character	Not Applicable	Retry With: default	Not Applicable	
N/A	Code Variable: life: company	Absolute	Character	Not Applicable	Value Of: 0	Not Applicable	
N/A	Code Variable: life: submodel	Absolute	Character	Not Applicable	Value Of: 0	Not Applicable	
N/A	Code Variable: life: paid_up_input	Absolute	Character	Not Applicable	Value Of: 0	Not Applicable	

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: ben_class	Absolute	Character	Not Applicable	Retry With: default	Not Applicable	

**6.2.1.14.1.8 Comm\_reserves\_AddVAT**

Description:

Parameters

Column Lookup Details:

Key Names:

No

Final Column Uses Position:

No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: Value	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names:

No

Final Row Uses Position:

No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: Comm_reserves_Add VAT	Absolute	Character	Not Applicable	Error	Not Applicable	

**6.2.1.15 Input Page: Setup****6.2.1.15.1 Assumption Set: Solv\_Base****6.2.1.15.1.1 RunControl\_Char**

Description:

Run\_Control

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: Base	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: <*code_variable*>	Absolute	Character	Not Applicable	Error	Not Applicable	

#### 6.2.1.15.1.2 RunControl\_Num

Description: Run\_Control

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: Base	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
----------	-------------	---------------------	------	--------------	---------	------------	-------------------



Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: <*code_variable*>	Absolute	Character	Not Applicable	Error	Not Applicable	

#### 6.2.1.15.1.3 Param\_Switch

Description:

Parameters

Column Lookup Details:

Key Names:

No

Final Column Uses Position:

No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: Value	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names:

No

Final Row Uses Position:

No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: <*code_variable*>	Absolute	Character	Not Applicable	Error	Not Applicable	

#### 6.2.1.15.1.4 RA\_factor

Description:

RA\_Factor

Column Lookup Details:

Key Names:

No

Final Column Uses Position:

No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: <*code_variable*>	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: groups_sol	Absolute	Character	Not Applicable	Error	Not Applicable	

**6.2.1.15.1.5 Serv\_Units\_Dur**

Description: Serv\_Units\_Dur

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Variable: life: sex	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Code Column: life: age_last	Absolute	Numeric	First	Previous	Last	

**6.2.1.15.1.6 dump\_vars**

Description: Run\_Control

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: Base	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: <*code_variable*>	Absolute	Character	Not Applicable	Error	Not Applicable	

**6.2.1.15.1.7 dump\_vars (2)**

Description: Run\_Control

Column Lookup Details:

Key Names: No

Final Column Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: Base	Absolute	Character	Not Applicable	Error	Not Applicable	

Row Lookup Details:

Key Names: No

Final Row Uses Position: No

Key Name	Lookup term	Lookup Model Object	Type	Before First	Missing	After Last	Transformed Value
N/A	Constant: dump_vars	Absolute	Character	Not Applicable	Error	Not Applicable	