**3N Process**

This 3N process involves in RI Spend Module

**Reinsurance:**

The process includes gross ceded reinsurance data (forecast and ultimate RI) transfer/import from source systems/files into FDM, automated processing/calculations performed on the data inputs within FDM and posting of RI adjustments from FDM to Agresso.

Finance relies on the data producers to ensure that the RI forecast and RI ultimate data is fit for purpose. The following data used in the ceded RI processes is received from non finance functions:  
- Forecast RI spend- This data is contained in the premium forecast template (PFT)  
- RI spend ultimate, overriding commission(ORC), reinstatements. This is contained in the RI Spend spread sheet obtained from Ceded RI team (Alan Fennell)  
- RI Spend Munich cede percentages (SharePoint)  
- Large Losses (SharePoint)

There are a number of data reconciliations which are manually performed by Finance in this process as shown below.  
- Reconciliation of Premium Forecast Templates (PFTs) to Wizi  
- Reconciliation of FDM to Wizi  
- Reconciliation of FDM to Agresso  
-Base allocation for 11100 matches external data source on LTD basis for Syndicate by Trifocus, YOA and Currency. Granularity must be by RI Policy with inception and expiry dates to be earned  
- Earning % for RI LOD policies match the inwards non-binder earning % calculation. Earning % for RI RAD policies match the inwards binder earning % calculation.  
- Calculation of non-Munich QS uses correct Cede % from PFT

A screenshot of a computer

Description automatically generated

**FDM Cube calculation for 3N Process**

// Sidecar Binder Adj

SCOPE ( [Process].[Process Code].[3N], [Time Series].[Time Series].[LTD]

, [Data Stage].[Data Stage].&[2],[Measures].[Cur Amount] , [Earnings].[Earnings Policy Type].[Earning Type].&[Binder].CHILDREN

);

([Account].[Account].&[11101]) = ( ([Account].[Account].&[11101], [Data Stage].[Data Stage].&[4],[Process].[Process Code].[UB]) +

([Account].[Account].&[11101], [Data Stage].[Data Stage].&[4],[Process].[Process Code].[1E]))

\* (([Account].[Account].&[1161003],[Measures].[UW Percentage],[Process].[Process Code].&[-1],[Entity].[Entity].&[-1],[Cur Transaction Currency].[Tran Curr].[All],[Client].[Client].&[-1],[Tri Focus].[Tri Focus].&[-1],[Policy Section].[Pk Policy Section].&[-1]

,[Trifocus Tree].[Trifocus Tree].[All],[YOA Group].[YOA Group].[All],[Account Tree].[Account Tree].[All],[Entity Tree].[Entity Tree].[All],[RI Policy].[RI Policy].&[-1]) - 1);

([Account].[Account].&[82101]) = ([Account].[Account].&[11101]) \* -1;

Format\_String( THIS ) = "#,##0.00 ;(#,##0.00)";

END SCOPE;

**Explanation:**

The calculation for account [Account].[Account].&[11101] involves two main parts:

Part 1: The sum of two values, each calculated with a specific combination of [Data Stage] and [Process] conditions.

Part 2: The first part adds the values of [Account].[Account].&[11101] corresponding to two different [Process] codes ([Process].[Process Code].[UB] and [Process].[Process Code].[1E]) and a specific [Data Stage] condition ([Data Stage].[Data Stage].&[4]).

The second part involves multiplying the value of account [Account].[Account].&[1161003] by [Measures].[UW Percentage] as -1(Unknown). The multiplication involves multiple dimensions such as [Process], [Entity], [Cur Transaction Currency], [Client], etc., with specific conditions specified for each dimension.

This calculation sets the value for account [Account].[Account].&[82101].

It's a straightforward multiplication by -1 of the value of account [Account].[Account].&[11101]. This indicates that the value for account [Account].[Account].&[82101] is the negative of the value for account [Account].[Account].&[11101].

**FDM Posting Query for 3N Process:**

SELECT [Measures].[Cur Amount] ON COLUMNS,

NONEMPTY ( {[Scenario].[Scenario].&[1] }

\* INTERSECT({[fdmParamEntityCode]},{[Entity].[Entity Code].[2623],[Entity].[Entity Code].[623]})

\* [Cur Transaction Currency].[Tran Curr].[Tran Curr].AllMembers

\* FILTER( { [fdmParamYOA] } \* { [fdmParamAccountingPeriod] }

,([Measures].[hBri YOAto YOA Group Count]

,[YOA Group].[YOA Group].&[1]

,[Data Stage].[Data Stage].&[2]) = 1

OR ([Measures].[hBri YOAto YOA Group Count]

,[YOA Group].[YOA Group].&[2],[Data Stage].[Data Stage].&[2]) = 1

OR ([Measures].[hBri YOAto YOA Group Count]

,[YOA Group].[YOA Group].&[3],[Data Stage].[Data Stage].&[2]) = 1 )

\* {[Policy Section].[Section Reference].[Section Reference].MEMBERS}

\* { [Earnings].[Inception Date].[Inception Date].MEMBERS}

\* { [Earnings].[Expiry Date].[Expiry Date].MEMBERS}

\* { [Tri Focus].[Trifocus Code].[Trifocus Code].ALLMEMBERS }

\* {[Process].[Process Code].[3N] }

\* {[Account].[Account Code].&[11101],[Account].[Account Code].&[82101]} ,[Measures].[Cur Amount] )

ON ROWS FROM [FinanceDataMart] WHERE ([Calculation Mode].[Calculation Mode].[Full],[Time Series].[Time Series].[LTD], [Data Stage].[Data Stage].&[3])

While posting the data for 3N process code the above posting query runs in FDM with respective parameters given by users.