Project S378: Highmark

Description of the Infogix Assure Controls to Convert a spreadsheet of Highmark Plans' Limits and Benefits into Translation Tables for the CSR valuation Project

Contents

Summary

This document will define and explain the Infogix Assure Controls and programming process that converts the Highmark client delivered spreadsheet of RX Plans Limits and Benefits for both Highmark and NEPA claims to four useable Translation Tables for discovering the Limits and Benefits related to the current RX claim being processed.

The input files

1. ***Infogix_FPL_2014_2015_Groups_07312015.xlsx*** – This file contains the Highmark RX Plans per line with all the Limits and Benefits distributed throughout the columns. All the rows and columns from this spreadsheet are loaded into the Assure control, *HM_RX_HM_Spreadsheet_Groups_Capture*, one plan per record.

2. ***NEPA RX Benefits 20160128_W_3STUBS_FORMAT_ALL_AS_TEXT_CELLS.xlsx*** - This file contains the NEPA RX Plans per line with all the Limits and Benefits distributed throughout the columns.

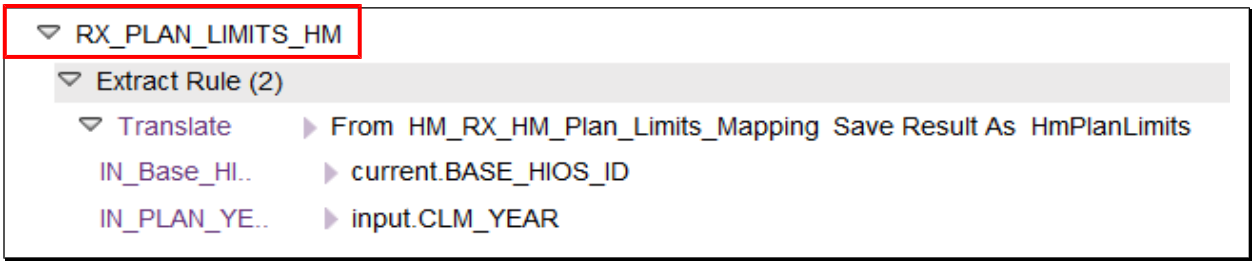
3. ***HM_AND_NEPA_RX_Benefits_Translation_Template.xlsx*** – This file contains two tabs; one for HM and the other for NEPA. This file takes each benefit column and makes it a row without a Plan ID specified since this will later be applied to each Plan. So if there are 100 columns of benefits this file would contain 100 rows with the Description of the column in each. If there are 20 plans then eventually each plan will have 100 rows.

The overall purpose of the conversion to table lookups

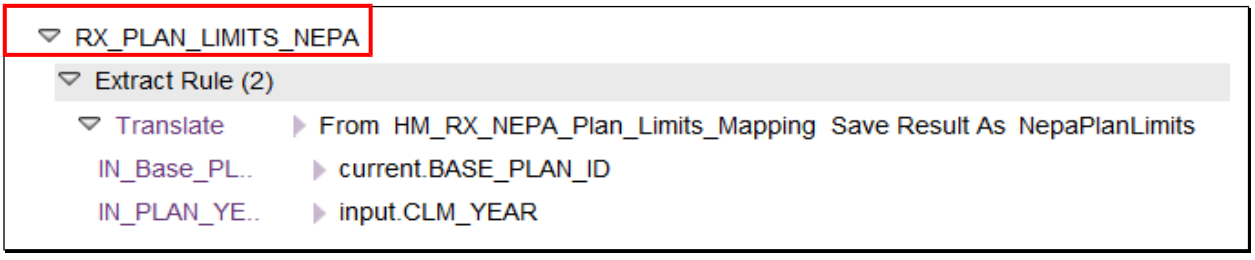
The overall purpose is to determine and calculate BASE Plan Limits and Benefits from a table rather than a spreadsheet.

1. ***Populate the Infogix Assure entity table with the Plan Limits*** to be used to look up and assign limits to the relevant RX Claims fields. The Translations demonstrated below for both HM and NEPA with their different input parameters deliver output values for Deductible, Out-Of-Pocket Limits, and any other specific limit settings.

   HM_RX_HM_Plan_Limits_Mapping table accessed during RX Claims processing

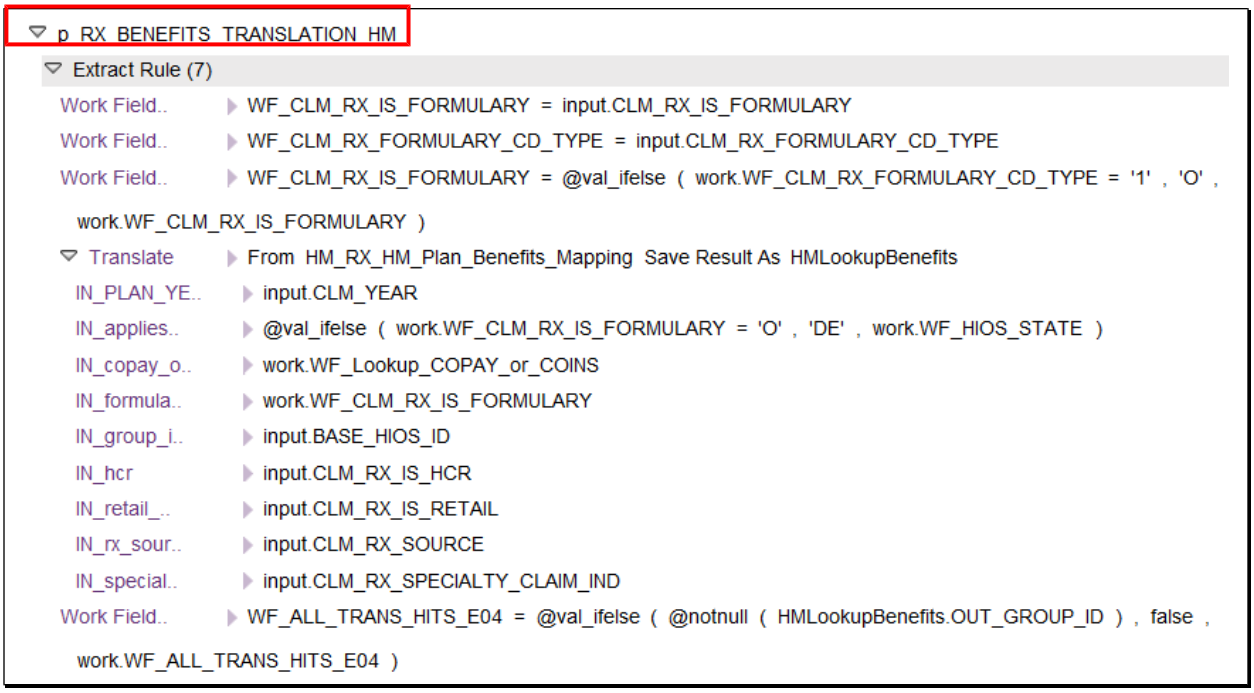


   HM_RX_NEPA_Plan_Limits_Mapping table accessed during RX Claims processing

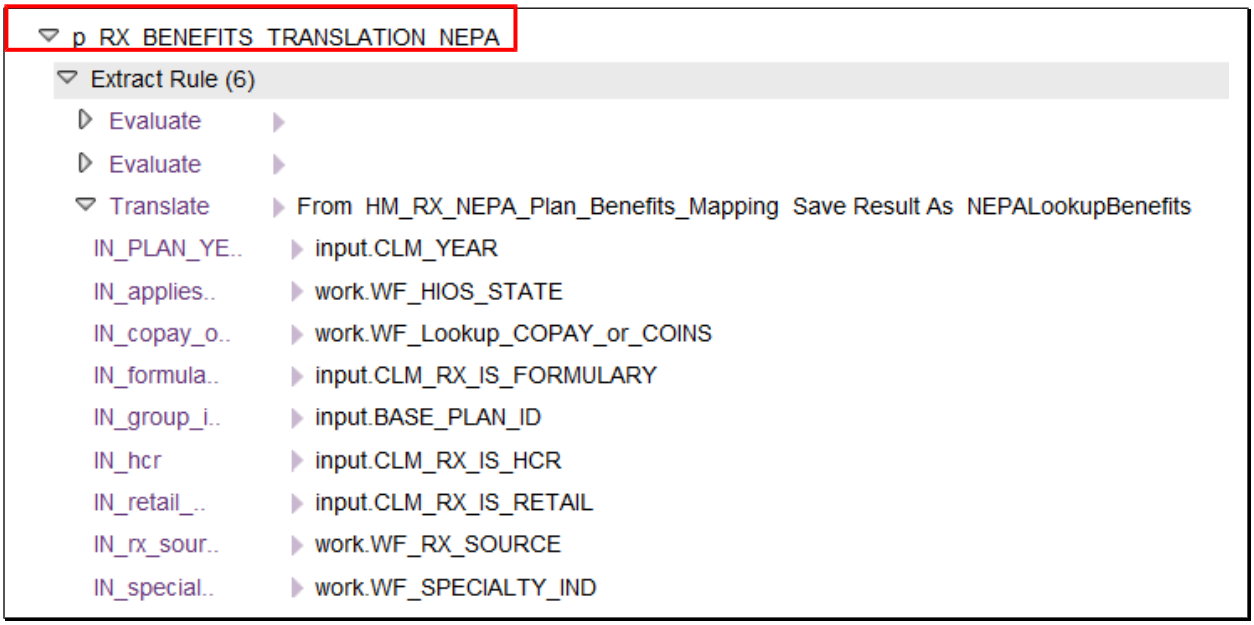


2. ***Populate the Infogix Assure entity table with the Plan Benefits*** values so a Mapping process can input values to locate which Benefit value to assign to the claim. This output value can be a Coinsurance Percent, Copay amount or an array of values based on Days' Supply.

HM_RX_HM_Plan_Benefits_Mapping table accessed during RX Claims processing



HM_RX_NEPA_Plan_Benefits_Mapping table accessed during RX Claims processing

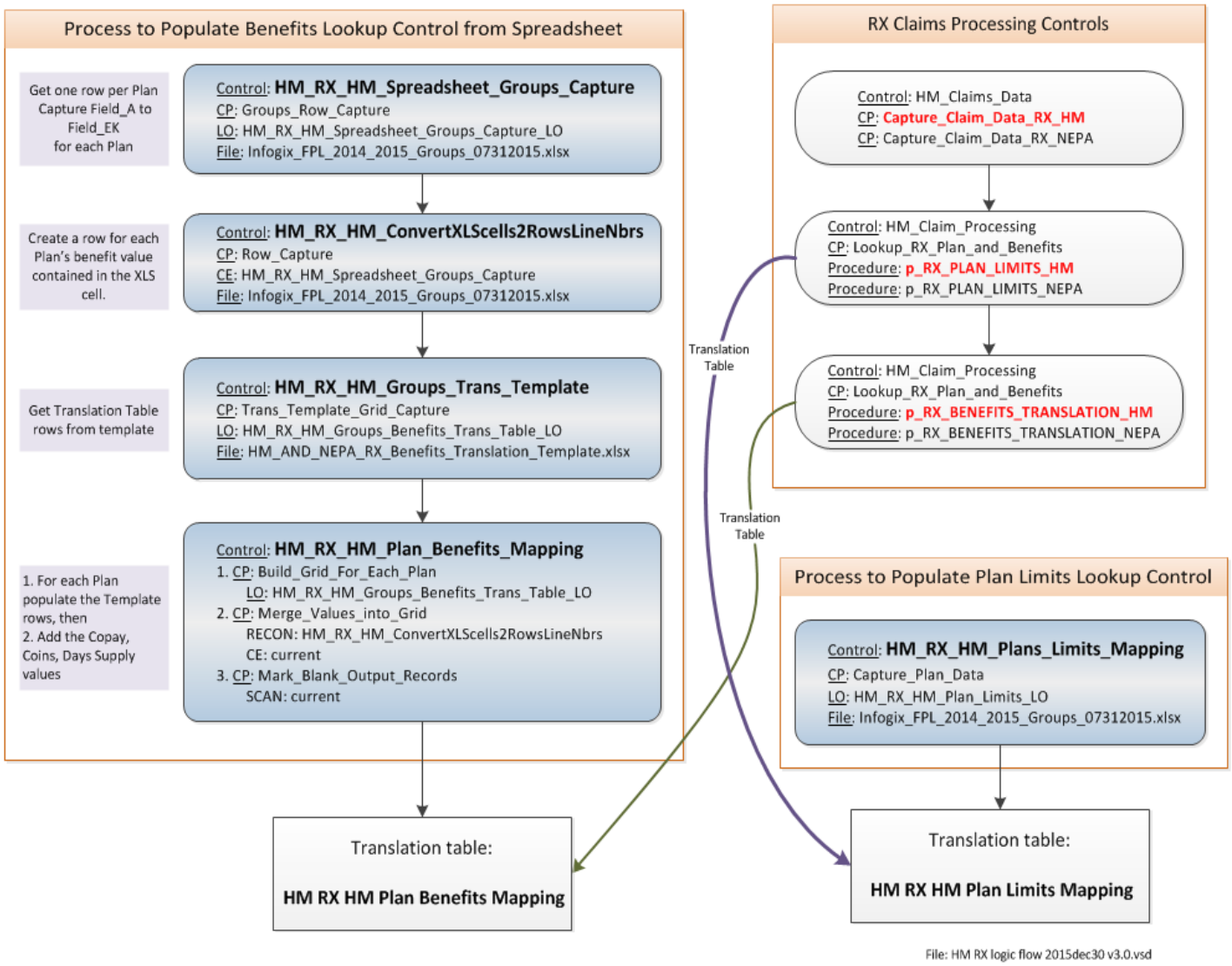


The different handling of the Benefits versus the Limits

The Limits conversion to a table is straight forward; one Plan line in the spreadsheet equals one record in the Assure table where the spreadsheet columns become field names. The benefits are handled different. The spreadsheet columns are converted into records using the PLAN_ID and spreadsheet column alphabetical locator as a synonym for the text description of the column. The reason for converting each Benefit column into a table record is because we need to locate which column value in the spreadsheet or row in the table from the Input values illustrated above in the RX_BENEFITS_TRANSLATION code to get the Copay, Coinsurance, or the Claim's Days' Supply dependent Copay value.

The Control Point Flow to Populate the Translation tables

Please note that there are parallel processes and controls for both HM and NEPA. The controls are appropriately prefixed with either *HM_RX_NEPA* or *HM_RX_HM*. Not all the NEPA entities are displayed below.



Steps for Populating the Translation Tables for HM and NEPA Plan Limits and Benefits

Note: All these steps are currently done automatically in Shell Scripts.

Clear all the Entities of any data.

Populate the Plans' Limits Lookup Control for Highmark and NEPA

1. Run the Control Point HM_RX_**HM**_Plans_Limits_Mapping -> Capture_Plan_Data
2. Run the Control Point HM_RX_**NEPA**_Plans_Limits_Mapping -> Capture_Plan_Data

Populate the Plans' Benefits Lookup Control for Highmark and NEPA

1. Run the Control Point HM_RX_**HM**_Spreadsheet_Groups_Capture -> Groups_Row_Capture
2. Run the Control Point HM_RX_**HM**_ConvertXLScells2RowsLineNbrs -> Row_Capture
3. Run the Control Point HM_RX_**HM**_Groups_Trans_Template -> Trans_Template_Grid_Capture
4. Run the Control Point HM_RX_**HM**_Plan_Benefits_Mapping -> Build_Grid_For_Each_Plan
5. Run the Control Point HM_RX_**HM**_Plan_Benefits_Mapping -> Merge_Values_into_Grid
6. Run the Control Point HM_RX_**HM**_Plan_Benefits_Mapping -> Mark_Blank_Output_Records
7. Run the Control Point HM_RX_**NEPA**_Spreadsheet_Groups_Capture -> Groups_Row_Capture
8. Run the Control Point HM_RX_**NEPA**_ConvertXLScells2RowsLineNbrs -> Row_Capture
9. Run the Control Point HM_RX_**NEPA**_Groups_Trans_Template -> Trans_Template_Grid_Capture
10. Run the Control Point HM_RX_**NEPA**_Plan_Benefits_Mapping -> Build_Grid_For_Each_Plan
11. Run the Control Point HM_RX_**NEPA**_Plan_Benefits_Mapping -> Merge_Values_into_Grid
12. Run the Control Point HM_RX_**NEPA**_Plan_Benefits_Mapping -> Mark_Blank_Output_Records

Description and Purpose of each of the four Controls

HM_RX_HM_Spreadsheet_Groups_Capture

The spreadsheet file*, Infogix_FPL_2014_2015_Groups_07312015.xlsx*, contains the Highmark RX Plans per line with all the Limits and Benefits distributed throughout the columns. All the rows and columns from this spreadsheet are loaded into this Assure control one plan per record.

A Run-Time field, ***RT_PlanYear***, is passed in to the capture since plans can exist in multiple years and may have different Limits and Benefits.

The Key fields for each row are:

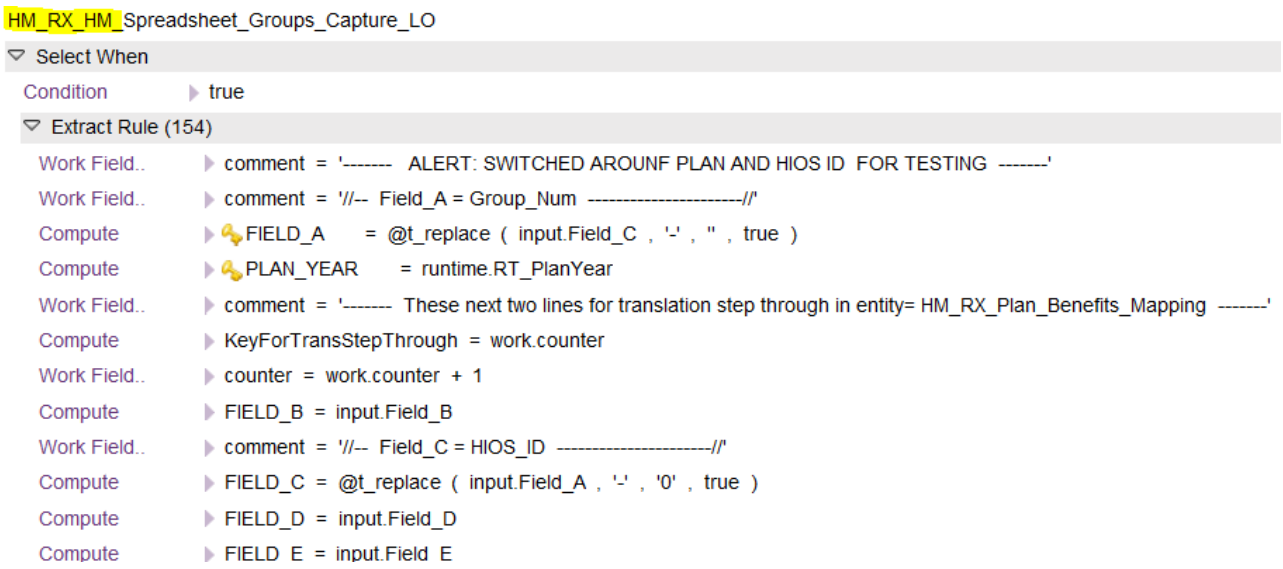
FIELD_A For Highmark this field contains the BASE_HIOS_ID

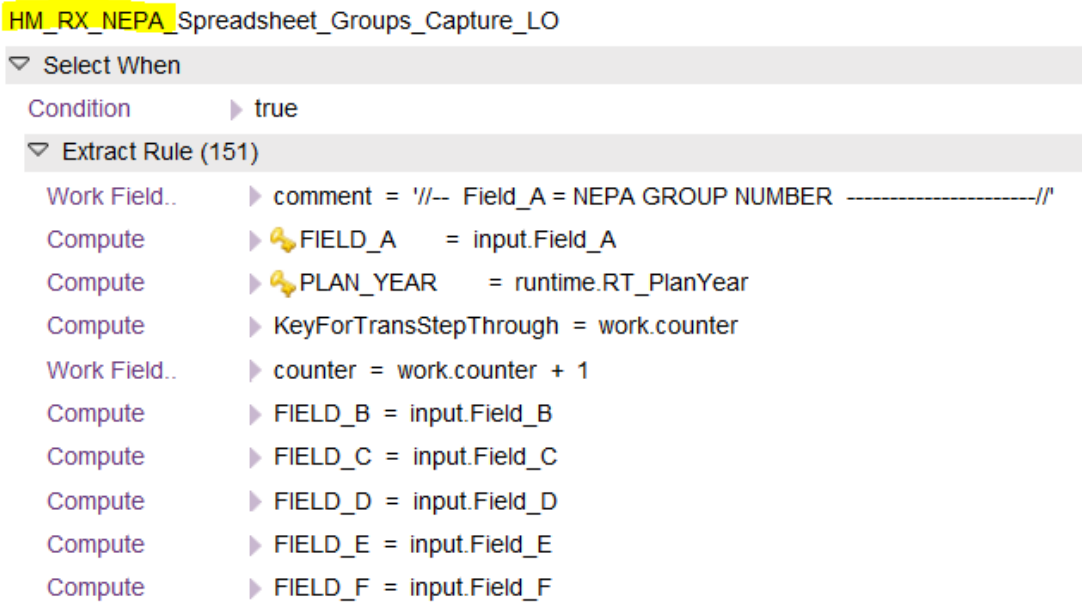
For NEPA this field contains the BASE_GROUP_ID which also means BASE_PLAN_ID

PLAN_YEAR This contains the value from the ***RT_PlanYear*** field. Filtering Plans by year will be accomplished by separating out the Plans into separate year file which are assigned in the Shell scripts.

The description of each plan is in FIELD_B. By setting the *Number of Header Rows* to zero in the layout we are able to use the alphabetical header in the spreadsheet as a field name and the column descriptor is assigned to FIELD_B. See images below. Also, notice several other issues:

1. Field A and C are switched around for Highmark since the BASE_HIOS_ID is used as the unique identifier of that plan rather than Group or Plan ID like with NEPA. The important point is that FIELD_A represents the unique identifier whatever it is.

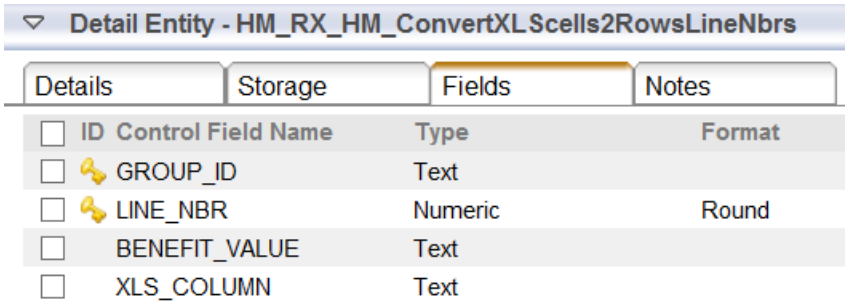




2. Field, KeyForTransStep, is no longer used. Its purpose has been replaced by using the map_get() and map_put() function calls to populate the standard template Benefits grid with the FIELD_A value (*Reference Figure A and B*). The code is stated in the Extract procedures, *p_BuildGroupKeyValueArray* and *p_IncrementThroughTransTbl*. This code is in the control point, *Build_Grid_For_Each_Plan*, of the control, *HM_RX_HM_Plan_Benefits_Mapping*. The code takes the currently 103 benefits rows in the template and creates 103 records for each PLAN or HIOS ID. All the IDs are stored in memory using the map_put() function.

HM_RX_HM_ConvertXLScells2RowsLineNbrs

This entity reads the RX Plans in the entity, *HM_RX_HM_Spreadsheet_Groups_Capture*, that are derived from the spreadsheet file*, Infogix_FPL_2014_2015_Groups_07312015.xlsx.* It then converts each field of the Plan record into a row as defined below. Note: Only Benefits values are populated. Limits for each RX Plan are stored in entity *HM_RX_HM_Plan_Limits_Mapping*.



HM_RX_HM_Groups_Trans_Template

This entity captured the spreadsheet file*, HM_AND_NEPA_RX_Benefits_Translation_Template.xlsx*, which contains the Template of all the Benefits columns converted into rows. The first column of the spreadsheet has a place holder value for the Group ID, also called FIELD_A (reference image below). This first field will be replaced with the value received from the map_get() function. This control data is utilized later in the control point *Build_Grid_For_Each_Plan* in the entity *HM_RX_HM_Plan_Benefits_Mapping*.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	GROUP_ID	XLS_COLUMN	SPREADSHEET_TEXT	APPLIES_TO_HIOS_STATE	RETAIL_OR_MAIL	HCR	SPECIALTY_CLAIM_IND	RX_SOURCE_DRUG_TYPE	MEMBER_YN	FORMULARY_YN	COPAY_OR_COINS	DAYS_SUPPLY	COPAY_AMT	COINS_PCT
2	Group Nbr	X	Retail Generic Coins Member Open	DE	Y	N	2	3	Y	O	COINS			
3	Group Nbr	Y	Retail Brand Coins Member Open	DE	Y	N	2	1	Y	O	COINS			
4	Group Nbr	Z	Retail Brand if Gen Avail Coins Member Open	DE	Y	N	2	5	Y	O	COINS			
5	Group Nbr	AA	Retail Brand Coins Member Formulary	DE	Y	N	2	1	Y	Y	COINS			
6	Group Nbr	AB	Retail Brand if Gen Avail Coins Member Formulary	DE	Y	N	2	5	Y	Y	COINS			
7	Group Nbr	AC	Retail Brand Coins Member Non-Formulary	DE	Y	N	2	1	Y	N	COINS			
8	Group Nbr	AD	Retail Brand if Gen Avail Coins Member Non-Formulary	DE	Y	N	2	5	Y	N	COINS			
9	Group Nbr	AE	Retail Generic Copay Open	DE	Y	N	2	3		O	COPAY			
10	Group Nbr	AF	Retail Brand Copay Open	DE	Y	N	2	1		O	COPAY			
11	Group Nbr	AG	Retail Brand if Gen Avail Copay Open	DE	Y	N	2	5		O	COPAY			
12	Group Nbr	AH	Retail Generic Copay Formulary	DE	Y	N	2	3		Y	COPAY			
13	Group Nbr	AI	Retail Brand Copay Formulary	DE	Y	N	2	1		Y	COPAY			
14	Group Nbr	AJ	Retail Brand if Gen Avail Copay Formulary	DE	Y	N	2	5		Y	COPAY			
15	Group Nbr	AK	Retail Generic Copay Non-Formulary	DE	Y	N	2	3		N	COPAY			
16	Group Nbr	AL	Retail Brand Copay Non-Formulary	DE	Y	N	2	1		N	COPAY			
17	Group Nbr	AM	Retail Brand if Gen Avail Copay Non-Formulary	DE	Y	N	2	5		N	COPAY			
18	Group Nbr	AN,AO	Retail Generic Copay Formulary Days Supply	DE	Y	N	2	3		Y	COPAY			
19	Group Nbr	AP,AQ,AR	Retail Generic Copay Formulary Days Supply	PA	Y	N	2	3		Y	COPAY			

Eventually, this template will be used so the GROUP_ID/column A will be populated with the Plan Group or HIOS ID and one of the pink fields to the right will hold the value for either Copay, Coinsurance or Copay based on Days' Supply from

the spreadsheet, *Infogix_FPL_2014_2015_Groups_07312015.xlsx,* that has been converted in the entity *HM_RX_HM_ConvertXLScells2RowsLineNbrs*.

Note: This file contins two tabs; one for HM RX benefits and the other for NEPA. The only difference is in the FORMULARY_YN field. The HM version can contain [Y,N,O] while NEPA only has [Y,N]. This is because HM can have Formulary, Non-Formulary, or Open while NEPA only has Formulary, Non-Formulary values.

HM_RX_HM_Plan_Benefits_Mapping

This entity puts the data together into the Translation table for RX Claims benefits lookups.

Control Points

Build_Grid_For_Each_Plan – This control point replicates the Benefits template for each Plan. For example, if there are 22 RX Plans and 103 Benefit columns in the original spreadsheet this control point will produce 2,266 rows in the table. But, these rows do not have the associated Benefits values until the next control point.

Merge_Values_Into_Grid – This Reconciliation control point places the Copay, Coinsurance, or Days' Supply value into the correct Plan and Benefits row. The two data Sources are this entity, called "*current*" and the entity *HM_RX_HM_ConvertXLScells2RowsLineNbrs*.

Mark_Blank_Output_Records – Any rows that don't contain a value in any of the three output fields will be marked with an "N". Those with values will be set to "Y". This eliminates the blank records during the translation process and therefore takes less iterations to find the relevant Benefits row.

APPENDIX

Figure A map_get() function

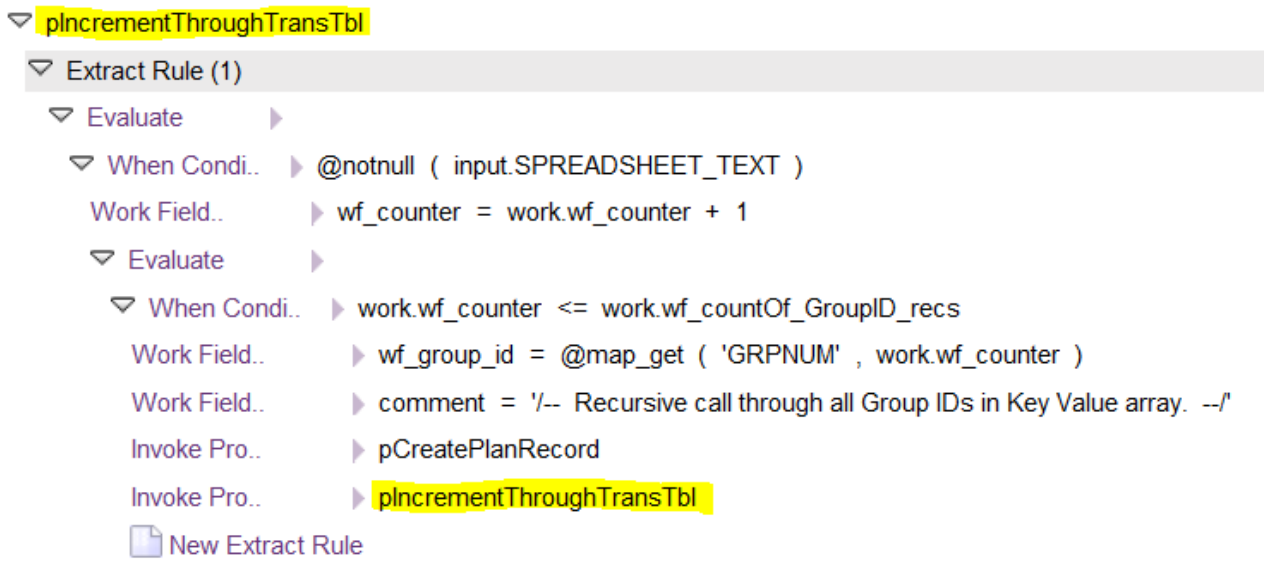


Figure B map_put() function