Schematron Edits – New Method

Individual Edits will have the message displayed though a function call rather than hard-coded in the edit. The Error message is in the LOOKUPDBA.MICR\_EDITS\_LKP table which is generated into the micr\_edits\_lookup.xml file. The message function will be called in place of the hard-coded message like so:

<value-of select=*"mspfx:error-message('XXX')"*/> where ‘XXX’ is the error number.

Each Edit will also call a single function for its operational logic when it is not a simple edit.

<!-- Error 941 -->

<report test=*"mspfx:error-941($currentPropertyClass)"* role=*"error"*><value-of select=*"mspfx:error-message('941')"*/></report>

The functions are located in micr\_schematron\_validation.sch and are separated into three categories currently: Message Functions to generate messages returned by the edits, Edit Functions called be the edits, and Granular Functions that do the actual work that will be combined int the Edit Function calling it.

<!-- 941 Property Class 29-35 -->  
 <xsl:function name="mspfx:error-941" as="xs:boolean">  
 <xsl:param name="currentPropertyClassIN" as="xs:string"/>  
 <!-- Make sure Property Class is 29-35 and Offense is 12000, 12001, 22001, 22002, or 24002 so it is only for edit 941  
 Call function to evaluate Offense File Class/Property Class pairing-->  
 <xsl:sequence select="boolean(mspfx:ofc-prop-class-function-check('mspfx:error-941') and mspfx:all-ofc-prop-class($currentPropertyClassIN))"/>  
   
 </xsl:function>

Edit Functions take in parameters and return the logic output of the combination of the granular function logic outputs.

<!-- Check if Schematron function is in micr\_edits\_ofc\_prop\_class\_lookup -->  
 <xsl:function name="mspfx:ofc-prop-class-function-check" as="xs:boolean">  
 <xsl:param name="functionIn" as="xs:string"/>  
 <!-- Compare schematron funtion required by offense class -->   
 <xsl:sequence select="boolean(some $x in $completeOffenses satisfies tokenize(($micr\_edits\_ofc\_prop\_class\_lookup//ROW/COLUMN[@NAME = 'OFFENSE\_FILE\_CLASS' and text() = $x]/../COLUMN[@NAME = 'SCHEMATRON\_FUNCTIONS']/text()),',') = $functionIn)"/>  
   
 </xsl:function>

Granular Functions take in parameters and will operate on a lookup table. This one is used to check if the Edit Function calling is supposed to be operating on the offense file classes in the incident.

<!-- Check Offense File Class - Property Class Matirx -->  
 <xsl:function name="mspfx:all-ofc-prop-class" as="xs:boolean">  
 <xsl:param name="currentPropertyClassIN" as="xs:string"/>  
 <!-- Loop through the completed Offense File Class and using each offense code see if the Property Class is acceptable.  
 If no Offense File Class/Property Class pairing is acceptable returns false and is inverted to return to calling function  
 This also checks if a zero property record is acceptable the result is inverted because the edit triggers on a true result-->  
 <xsl:sequence select="not(boolean( some $x in $completeOffenses satisfies boolean(true()) = boolean(tokenize(($micr\_edits\_ofc\_prop\_class\_lookup//ROW/COLUMN[@NAME = 'OFFENSE\_FILE\_CLASS' and text() = $x]/../COLUMN[@NAME = 'PROPERTY\_CLASSES']/text()),',') = $currentPropertyClassIN)) or $zeroPropExitsts and boolean (some $i in $completeOffenses satisfies boolean(true()) = boolean(tokenize(($micr\_edits\_ofc\_prop\_class\_lookup//ROW/COLUMN[@NAME = 'OFFENSE\_FILE\_CLASS' and text() = $i]/../COLUMN[@NAME = 'PROPERTY\_CLASSES']/text()),',') = '00')))"/>  
   
 </xsl:function>

When combined with the previous Granular Function, the Edit Function will be able to validate for its edit. Not all edits will require a check like the previous Granular Function or multiple Granular Functions but Edits should always call and Edit Function that then calls a Granular Function so if changes need to be made it only needs to be in one place.

# Summary

* All Edits should get the message from a function
* Edits will provide parameters to a function that will perform the logic
* Edit Functions will combine the logic of Granular Functions to perform Edit logic
* Granular Functions will be table-driven when possible