Validating Medicare Beneficiary Identifier Submissions

Richard Pickett, RTI International®, Research Triangle Park, NC

ABSTRACT

Centers for Medicare and Medicaid Services (CMS) is removing Social Security Numbers (SSNs) from Medicare cards to fight medical identity theft for people with Medicare. A new Medicare Beneficiary Identifier (MBI) is replacing the SSN-based Health Insurance Claim Number (HICN). This presented a challenge to healthcare plans submitting data for the Medicare Health Outcomes Survey (HOS). Submissions included HICNs, Railroad Board Numbers (RRB) and MBIs mixed under one field which in turn presented a challenge to RTI in drawing additional information form the Enrollment Database using the identifier. This paper presents SAS® macro code to identify invalid MBIs and display them for review and communication with the healthcare plan for correction.

INTRODUCTION

Personally Identifiable Information (PII) requires special handling to insure compliance with the 1974 Privacy Act. This paper assumes familiarity with CMS data handling requirements. CMS defines the structure of MBIs at https://www.cms.gov/Medicare/New-Medicare-Card/Understanding-the-MBI.pdf The following macro cleans the MBI variable of punctuation, blanks, tabs, control characters, standardizes the character case, splits the MBI into 11 variables by position and checks the value for the position against the defined CMS MBI structure, flagging invalid MBIs. It then utilizes SAS ODS Excel to display invalid MBIs for review.

```
%MACRO ck_mbi(_MBI=, dsn=, outpath=) / DES = "Checks validity of Medicare Beneficiary Identifier " ;
   /* Parameters

           _MBI    - variable name of the MBI for the input data set

           dsn     - fully qualified SAS® input data set NAME

           outpath - location to send EXCEL® output
   */
   DATA ck_mbi ;
   /* https://www.cms.gov/Medicare/New-Medicare-Card/Understanding-the-MBI.pdf */
     LENGTH mbi1 mbi2 mbi3 mbi4 mbi5 mbi6 mbi7 mbi8 mbi9 mbi10 mbi11 $ 1  SSA_AREA $ 3
         pos1 pos2 pos3 pos4 pos5 pos6 pos7 pos8 pos9 pos10 pos11 badmbi num_SSA_AREA 3. ;
      SET &dsn ;


   /* C – Numeric 1 thru 9  */
   %LET c_set = '1','2','3','4','5','6','7','8','9' ;
   /* N – Numeric 0 thru 9 */
   %LET n_set = '0','1','2','3','4','5','6','7','8','9' ;
```

```
/* A – Alphabetic Character (A...Z); Excluding (S, L, O, I, B, Z) */
%LET a_set = 'A', 'C', 'D', 'E', 'F', 'G', 'H', 'J', 'K', 'M', 'N', 'P', 'Q', 'R', 'T', 'U', 'V', 'W', 'X', 'Y' ;
/* AN – Either A or N  */
%LET an_set =
'0','1','2','3','4','5','6','7','8','9','A','C','D','E','F','G','H','J','K','M','N','P','Q','R','T','U','V','W','X','Y';
/*
Position 1 - numeric values 1 thru 9
Position 2 - alphabetic values A thru Z (minus S, L, O, I, B, Z)
Position 3 - alpha-numeric values 0 thru 9 and A thru Z (minus S, L, O, I, B, Z)
Position 4 - numeric values 0 thru 9
Position 5 - alphabetic values A thru Z (minus S, L, O, I, B, Z)
Position 6 - alpha-numeric values 0 thru 9 and A thru Z (minus S, L, O, I, B, Z)
Position 7 - numeric values 0 thru 9
Position 8 - alphabetic values A thru Z (minus S, L, O, I, B, Z)
Position 9 - alphabetic values A thru Z (minus S, L, O, I, B, Z)
Position 10 - numeric values 0 thru 9
Position 11 - numeric values 0 thru 9
*/


/* Set up variables for each position of the MBI */
ARRAY mbid {*} mbi1 mbi2 mbi3 mbi4 mbi5 mbi6 mbi7 mbi8 mbi9 mbi10 mbi11 ;

IF ~MISSING(&_MBI.) THEN
DO ;
 /* Standardize case and remove punctuation, blanks, tabs, control characters */
 &_MBI. = UPCASE(STRIP(COMPRESS(&_MBI.,' psc'))) ;
  /* Assign values to each position from MBI */
  DO i = 1 TO DIM(mbid) ;
    mbid{i} = SUBSTR(&_MBI., i, 1) ;
  END ;
  /* Check each position against the CMS rule */
  IF mbi1  IN (&c_set)  THEN pos1 = 1;  ELSE pos1 = 0 ;
  IF mbi2  IN (&a_set)  THEN pos2 = 1;  ELSE pos2 = 0 ;
  IF mbi3  IN (&an_set) THEN pos3 = 1;  ELSE pos3 = 0 ;
  IF mbi4  IN (&n_set)  THEN pos4 = 1;  ELSE pos4 = 0 ;
  IF mbi5  IN (&a_set)  THEN pos5 = 1;  ELSE pos5 = 0 ;
  IF mbi6  IN (&an_set) THEN pos6 = 1;  ELSE pos6 = 0 ;
```

```
    IF mbi7  IN (&n_set)  THEN pos7 = 1;  ELSE pos7 = 0 ;

    IF mbi8  IN (&a_set)  THEN pos8 = 1;  ELSE pos8 = 0 ;

    IF mbi9  IN (&a_set)  THEN pos9 = 1;  ELSE pos9 = 0 ;

    IF mbi10 IN (&n_set)  THEN pos10 = 1; ELSE pos10 = 0 ;

    IF mbi11 IN (&n_set)  THEN pos11 = 1; ELSE pos11 = 0 ;


     /* Check MBI length and check all positions for invalid values. Set bad MBI flag */

     IF SUM(0,pos1, pos2, pos3, pos4, pos5, pos6, pos7, pos8, pos9, pos10, pos11) NE 11 OR (0 LT
LENGTHC(&_MBI.) LT 11) THEN badmbi = 1 ;

      ELSE badmbi = 0 ;

    END ;


   IF mbi1 IN (&n_set) AND mbi2 IN (&n_set) AND mbi3 IN (&n_set) THEN DO ;

     SSA_AREA = SUBSTR(&_MBI., 1,3) ;

     num_SSA_AREA = INPUT(SSA_AREA,3.) ;

   END ;


   LABEL

     mbi1   = "MBI position 1 numeric values 1 thru 9"

     mbi2   = "MBI position 2 alphabetic values A thru Z (minus S, L, O, I, B, Z)"

     mbi3   = "MBI position 3 alpha-numeric values 0 thru 9 and A thru Z (minus S, L, O, I, B, Z)"

     mbi4   = "MBI position 4 numeric values 0 thru 9"

     mbi5   = "MBI position 5 alphabetic values A thru Z (minus S, L, O, I, B, Z)"

     mbi6   = "MBI position 6 alpha-numeric values 0 thru 9 and A thru Z (minus S, L, O, I, B, Z)"

     mbi7   = "MBI position 7 numeric values 0 thru 9"

     mbi8   = "MBI position 8 alphabetic values A thru Z (minus S, L, O, I, B, Z)"

     mbi9   = "MBI position 9 alphabetic values A thru Z (minus S, L, O, I, B, Z)"

     mbi10  = "MBI position 10 numeric values 0 thru 9"

     mbi11  = "MBI position 11 numeric values 0 thru 9"

     pos1   = "MBI position 1 validation check"

     pos2   = "MBI position 2 validation check"

     pos3   = "MBI position 3 validation check"

     pos4   = "MBI position 4 validation check"

     pos5   = "MBI position 5 validation check"

     pos6   = "MBI position 6 validation check"

     pos7   = "MBI position 7 validation check"

     pos8   = "MBI position 8 validation check"
```

```sas
         pos9   = "MBI position 9 validation check"

         pos10  = "MBI position 10 validation check"

         pos11  = "MBI position 11 validation check"

         badmbi = "MBI invalid flag"

         SSA_AREA = "SSA Area Code"

         num_SSA_AREA = "Numeric SSA Area Code"

         ;

    RUN ;

   /* Send results to EXCEL® for review */

   %LET color=Blue BlueViolet CadetBlue CornFlowerBlue DodgerBlue LightBlue LightSkyBlue
LightSteelBlue SteelBlue AliceBlue PowderBlue SkyBlue DeepSkyBlue Azure MediumBlue SlateBlue
MediumSlateBlue DarkSlateBlue MidnightBlue DarkBlue Navy RoyalBlue DimGray lightGray Gray
DarkGray LightSlateGray SlateGray DarkSlateGray Gainsboro AntiqueWhite FloralWhite GhostWhite
NavajoWhite White WhiteSmoke Snow Ivory Oldlace Linen Seashell Black Brown RosyBrown
SaddleBrown SandyBrown Sienna Chocolate Moccasin Peru Beige Bisque BlanchedAlmond Burlywood
Tan Thistle Wheat Aqua Aquamarine MediumAquamarine Teal Green PaleGreen LightGreen DarkGreen
LawnGreen SpringGreen MediumSpringGreen SeaGreen LightSeaGreen MediumSeaGreen
DarkSeaGreen DarkOliveGreen ForestGreen GreenYellow LimeGreen Lime Chartreuse MintCream
Honeydew LightYellow Yellow YellowGreen Cornsilk LightGoldenrodYellow PaleGoldenrod Goldenrod
DarkGoldenrod Gold LemonChiffon Red DarkRed IndianRed MediumVioletRed OrangeRed
PaleVioletRed Crimson FireBrick Tomato MistyRose Purple MediumPurple Violet DarkViolet Indigo Plum
Lavender LavenderBlush Pink LightPink DeepPink HotPink Fuchsia Orange DarkOrange Turquoise
PaleTurquoise MediumTurquoise DarkTurquoise Khaki DarkKhaki Olive OliveDrab Coral LightCoral
PapayaWhip Peachpuff Salmon LightSalmon DarkSalmon Orchid MediumOrchid DarkOrchid Magenta
DarkMagenta Cyan LightCyan DarkCyan Maroon Silver ;


    /* ---------------------------------------------------------------- */
    ODS EXCLUDE NONE ;  /* Resume output to open destinations */
    ODS RESULTS OFF ;
    ODS LISTING CLOSE ; /* Close Listing                    */
    /* ---------------------------------------------------------------- */
    ODS EXCEL
      FILE="&outpath.invalid_mbi.xlsx" style=NORMAL
     OPTIONS (SHEET_INTERVAL      = 'PROC'
           Sheet_Name          = "MBI Invalid Position"
           Orientation        = 'Landscape'
           page_order_across    = 'Yes'
           suppress_bylines     = 'Yes'
           pages_fitwidth       = '1'
           FitToPage           = 'Yes'
           Center_Horizontal    = 'Yes'
           Center_Vertical      = 'Yes'
```

4

```
        embed_footnotes_once  = 'On'
        embedded_footnotes    = 'On'
        embed_titles_once     = 'On'
        embedded_titles       = 'On'
        TITLE_FOOTNOTE_WIDTH  = '127'
        absolute_row_height   = '24'
        gridlines             = 'Off'
        tab_color             = "%SCAN(&Color,1)"
        TITLE_FOOTNOTE_NOBREAK= 'yes'
        ) ;
    TITLE1 JUSTIFY=L "&dsn.: INVALID MBI posx = 0" ;
    TITLE2 JUSTIFY=L "Position 1 - numeric values 1 thru 9" ;
    TITLE3 JUSTIFY=L "Position 2,5,8,9 - alphabetic values A thru Z (minus S, L, O, I, B, Z)" ;
    TITLE4 JUSTIFY=L "Position 3,6 - alpha-numeric values 0 thru 9 and A thru Z (minus S, L, O, I, B,
Z)" ;
    TITLE5 JUSTIFY=L "Position 4,7,10,11 - numeric values 0 thru 9" ;
    PROC PRINT DATA = WORK.ck_mbi(WHERE=(badmbi=1)) ;
      VAR &_MBI. pos1 pos2 pos3 pos4 pos5 pos6 pos7 pos8 pos9 pos10 pos11 ;
    RUN ;
  QUIT ;
  /* ----------------------------------------------------------- */
  ODS EXCEL
   OPTIONS (SHEET_INTERVAL      = 'PROC'
        Sheet_Name          = "MBI Position/Validation Check"
        Orientation         = 'Landscape'
        page_order_across   = 'Yes'
        suppress_bylines    = 'Yes'
        pages_fitwidth      = '1'
        FitToPage           = 'Yes'
        Center_Horizontal   = 'Yes'
        Center_Vertical     = 'Yes'
        embed_footnotes_once  = 'On'
        embedded_footnotes    = 'On'
        embed_titles_once     = 'On'
        embedded_titles       = 'On'
        TITLE_FOOTNOTE_WIDTH  = '127'
        absolute_row_height   = '24'
```

```
          gridlines          = 'Off'
          tab_color          = "%SCAN(&Color,2)"
          TITLE_FOOTNOTE_NOBREAK= 'yes'
          ) ;
     TITLE1 JUSTIFY=L "&dsn.: INVALID MBI posx = 0" ;
     TITLE2 JUSTIFY=L "Position 1        - numeric values 1 thru 9" ;
     TITLE3 JUSTIFY=L "Position 2,5,8,9   - alphabetic values A thru Z (minus S, L, O, I, B, Z)" ;
     TITLE4 JUSTIFY=L "Position 3,6      - alpha-numeric values 0 thru 9 and A thru Z (minus S, L, O,
I, B, Z)" ;
     TITLE5 JUSTIFY=L "Position 4,7,10,11 - numeric values 0 thru 9" ;
      PROC PRINT DATA = WORK.ck_mbi(WHERE=(badmbi=1)) ;
       VAR &_MBI. mbi1 pos1 mbi2 pos2 mbi3 pos3 mbi4 pos4 mbi5 pos5 mbi6 pos6 mbi7 pos7 mbi8
pos8 mbi9 pos9 mbi10 pos10 mbi11 pos11 ;
      RUN ;
      TITLE1 ;
      TITLE2 ;
      TITLE3 ;
      TITLE4 ;
      TITLE5 ;
    QUIT ;
 /* -------------------------------------------------------------- */
      /* Add any other data views here */
`
 /* -------------------------------------------------------------- */
    ODS _ALL_ CLOSE ;
    QUIT ;
    RUN ;
    /* -------------------------------------------------------------- */
    ODS EXCEL CLOSE ; %*<-- Close the tagset and output reports ;
    ODS RESULTS ;    %*<-- Turn traditional SAS output back on ;
    ODS LISTING ;
    /* -------------------------------------------------------------- */
  %MEND ck_MBI. ;
```

CONCLUSION

Insuring accuracy of the Medicare Beneficiary Identifier is essential in linking beneficiary information to the correct beneficiary. This paper presents a versatile macro to assist in ensuring data integrity.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:
Richard Pickett
RTI International
Tel: 919-316-6798
Email: rpickett@rti.org

DISCLAIMER

The contents of this paper are purely the work of the author and do not necessarily represent the opinions, or recommendations of RTI International. SAS and all   other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are trademarks of their respective companies.