**MCCF EDI TAS US129**

System Design Document

IB\*2.0\*577



Department of Veterans Affairs

**March 2017**

Version 2.0

**User Story Number:** TSEB-02

**User Story Name:** Process Payer ID - Store HCCH Payer ID (TSEB-2)

**Product Backlog ID:**

**Rally ID:** US-129

# Design/Resolution:

# The system is to process a 277STAT message from Health Care Clearing House (HCCH) and evaluate/update Payer Id’s as follows:

# Update the Insurance file, field EDI ID NUMBER – PROF or EDI ID NUMBER – INST with the Payer IDs from the 277STAT transaction only when the value we have on file is blank. When updating these fields, also update new data elements to capture changes and date of update for report purposes (US976).

1. If the Payer ID from the 277STAT transaction is different than what we have stored and is not blank, then no update is to occur, but storage of the data into new data elements including the date for reporting purposes only. (US976) This is to occur only once per day.
2. Resolve intermittent issue of updating EDI ID numbers when we have no value stored.
3. In order to accomplish this, the following needs to be modified:
   1. File Modifications:
      1. File 36 (Insurance Company File)- add new fields:
         1. 277EDI ID Number – as a multiple
            1. 277Date EDI ID Number- as a subfile
            2. 277EDI Type (P)ROF or (I)NST- as subfile
            3. 277EDI Number on file– as a subfile
      2. Create Cross Reference on File 36 new fields for report usage (US976)
   2. Routine Modifications:
      1. Routine IBCEST- 277Stat Message processing
4. A Sample X12 and 277Stat file is included for testing changes.
5. Mail group to drop in the file in for testing.

| **Routine** | **Activities** | | | | |
| --- | --- | --- | --- | --- | --- |
| **File Name** | Insurance #36 | | | | |
| **Enhancement Category** | New | Modify | Delete | | No Change |
| **RTM** |  | | | | |
| **Related Options** | None | | | | |
| **Related Routines** | **Routines “Called By”** | | | **Routines “Called”** | |
|  |  | | |  | |
| **Data Dictionary (DD) References** | **File Modifications:**  FileMan  Option: 4- MODIFY FILE ATTRIBUTES  File: 36 (INSURANCE COMPANY)  Set up New fields:  **17.0 277EDI ID NUMBER**  MINIMUM LENGTH: 1  MAXIMUM LENGTH: 30  (OPTIONAL) PATTERN MATCH (IN 'X'):  WILL 277EDI ID NUMBER FIELD BE MULTIPLE? No// YES (Yes)  SUBSCRIPT: 17//  IS 277EDI NUMBER ENTRY MANDATORY (Y/N): NO// NO  SHOULD USER SEE AN "ADDING A NEW 277EDI NUMBER?" MESSAGE FOR NEW ENTRIES (Y/N): N NO  HAVING ENTERED OR EDITED ONE 277EDI NUMBER, SHOULD USER BE ASKED ANOTHER (Y/N): N NO  SUB-DICTIONARY NUMBER: 36.017//  ....  ....  'HELP'-PROMPT: Answer must be 1-30 characters in length.  Replace  XECUTABLE 'HELP':  DESCRIPTION:  THERE ARE NO LINES!  Edit? NO//  **17.01 277EDI ID NUMBER**  AS SUB-FIELD  DATA TYPE OF 277EDI ID NUMBER: 4 FREE TEXT  MINIMUM LENGTH: 1  MAXIMUM LENGTH: 30  (OPTIONAL) PATTERN MATCH (IN 'X'):  WILL 277EDI ID NUMBER FIELD BE MULTIPLE? No// N (No)  SUBSCRIPT: 0//  ^-PIECE POSITION: 1//  IS 277EDI ID NUMBER ENTRY MANDATORY (Y/N): NO// NO  ....  'HELP'-PROMPT: Answer must be 1-30 characters in length.  Replace  XECUTABLE 'HELP':  DESCRIPTION:  THERE ARE NO LINES!  Edit? NO// SUBSCRIPT:0  ^- PIECE POSITION: 1  **17.02 277DATE EDI ID NUMBER**  Select 277EDI NUMBER SUB-FIELD: .02  Are you adding a new FIELD (the 2ND)? No// YES (Yes)  LABEL: 277DATE EDI ID NUMBER  277EDI NUMBER SUB-FIELD NUMBER: .02//  DATA TYPE OF 277DATE EDI ID NUMBER: DATE/TIME  EARLIEST DATE (OPTIONAL):  CAN DATE BE IMPRECISE (Y/N): YES// NO  CAN TIME OF DAY BE ENTERED (Y/N): NO//  WILL 277DATEEDI ID NUMBER FIELD BE MULTIPLE? No// (No)  SUBSCRIPT: 0//  ^-PIECE POSITION: 2//  IS 277DATEEDI ID NUMBER ENTRY MANDATORY (Y/N): NO  // NO  ....  'HELP'-PROMPT:  XECUTABLE 'HELP':  DESCRIPTION:  THERE ARE NO LINES!  Edit? NO//  **17.03 277 EDI TYPE**  Select 277EDI NUMBER SUB-FIELD: .03  Are you adding a new FIELD (the 3RD)? No// YES (Yes)  LABEL: 277EDI TYPE  277EDI NUMBER SUB-FIELD NUMBER: .03//  DATA TYPE OF 277EDI TYPE: SET OF CODES  DATA TYPE OF 277EDI TYPE: SET OF CODES  INTERNALLY-STORED CODE: P WILL STAND FOR: PROFESSIONAL  INTERNALLY-STORED CODE: I WILL STAND FOR: INSTITUTIONAL  INTERNALLY-STORED CODE:  SHOULD '277EDI TYPE' ENTRIES BE SCREENED? No// (No)  WILL 277EDI TYPE FIELD BE MULTIPLE? No// (No)  SUBSCRIPT: 0//  ^-PIECE POSITION: 3//  IS 277EDI TYPE ENTRY MANDATORY (Y/N): NO// NO  ....  'HELP'-PROMPT:  XECUTABLE 'HELP':  DESCRIPTION:  THERE ARE NO LINES!  Edit? NO//  **17.04 277EDI ID NUMBER ON FILE**  Select 277EDI NUMBER SUB-FIELD: .04  Are you adding a new FIELD (the 4TH)? No// YES (Yes)  LABEL: 277EDI NUMBER ON FILE  277EDI NUMBER SUB-FIELD NUMBER: .04//  DATA TYPE OF 277EDI NUMBER ON FILE: FREE TEXT  MINIMUM LENGTH: 1  MAXIMUM LENGTH: 30  (OPTIONAL) PATTERN MATCH (IN 'X'):  WILL 277EDI NUMBER ON FILE FIELD BE MULTIPLE? No// (No)  SUBSCRIPT: 0//  ^-PIECE POSITION: 4//  IS 277EDI NUMBER ON FILE ENTRY MANDATORY (Y/N): NO  // NO  ....  'HELP'-PROMPT: Answer must be 1-30 characters in length.  Replace  XECUTABLE 'HELP':  DESCRIPTION:  THERE ARE NO LINES!  Edit? NO//    ***The Standard list of these new fields will look like this:***  CROSS REFERENCED BY: 277EDI ID NUMBER(B)  36.017,.01 277EDI ID NUMBER 0;1 FREE TEXT  INPUT TRANSFORM: K:$L(X)>30!($L(X)<1) X  MAXIMUM LENGTH: 30  LAST EDITED: MAR 06, 2017  HELP-PROMPT: Answer must be 1-30 characters in length.  DESCRIPTION: This is the ID number used to identify the  payer on a professional or an institutional  claim transmission. This is for reporting  purposes only.  CROSS-REFERENCE: 36.017^B  1)= S ^DIC(36,DA(1),17,"B",$E(X,1,30),DA)=""  2)= K ^DIC(36,DA(1),17,"B",$E(X,1,30),DA)  RECORD INDEXES: AEDIX (#1454) (WHOLE FILE #36)  36.017,.02 277DATE EDI ID NUMBER 0;2 DATE  INPUT TRANSFORM: S %DT="EX" D ^%DT S X=Y K:Y<1 X  LAST EDITED: MAY 08, 2017  HELP-PROMPT: Enter the date of the 277STAT transaction from  the clearing house.  DESCRIPTION: This is the date from the 277stat transmission  from the clearing house. This is for reporting  purposes only.  RECORD INDEXES: AEDIX (#1454) (WHOLE FILE #36)  36.017,.03 277EDI TYPE 0;3 SET  'P' FOR PROFESSIONAL;  'I' FOR INSTITUTIONAL;  LAST EDITED: MAY 08, 2017  HELP-PROMPT: Enter the type of claim "P" for professional or  "I" for institutional.  DESCRIPTION: This determines the type of claim - either "P"  for professional or "I" for institutional. For  reporting purposes only.  RECORD INDEXES: AEDIX (#1454) (WHOLE FILE #36)  36.017,.04 277EDI ID NUMBER ON FILE 0;4 FREE TEXT  INPUT TRANSFORM: K:$L(X)>30!($L(X)<1) X  MAXIMUM LENGTH: 30  LAST EDITED: MAR 06, 2017  HELP-PROMPT: Answer must be 1-30 characters in length.  DESCRIPTION: This is the ID number that was stored on file  to determine the payer on a claim prior to the  277stat transaction update. If there is no  number that means there will be an update.  This is for reporting purposes only.  RECORD INDEXES: AEDIX (#1454) (WHOLE FILE #36)  **Current file:**  CONDENSED DATA DICTIONARY---INSURANCE COMPANY FILE (#36)UCI: PLA,PLA VERSION:  2.0  STORED IN: ^DIC(36, FEB 6,2017 PAGE 1  --------------------------------------------------------------------------------  …  3.01 TRANSMIT ELECTRONICALLY (SX), [3;1]  3.02 EDI ID NUMBER - PROF (FXJ30a), [3;2]  3.03 BIN NUMBER (F), [3;3]  3.04 EDI ID NUMBER - INST (FXJ30a), [3;4]  …  16 ALTERNATE PROF PAYER ID TYPE (Multiple-36.016), [16;0]  .01 ALTERNATE PROF PAYER ID TYPE (M\*P355.98'), [0;1]  .02 ALTERNATE PROF PAYER ID (FX), [0;2]  New section of file:  17 277EDI ID NUMBER (Multiple-36.017), [17;0]  .01 277EDI ID NUMBER (FJ30), [0;1]  .02 277DATE EDI ID NUMBER (D), [0;2]  .03 277EDI TYPE (S), [0;3]  .04 277EDI ID NUMBER ON FILE (FJ30), [0;4] | | | | |
|  |  | | | | |

­­­­­­­­­­­­­­­­­­

| **Routine** | **Activities** | | | | |
| --- | --- | --- | --- | --- | --- |
| **File Name** | Insurance #36- Cross Reference | | | | |
| **Enhancement Category** | New | Modify | Delete | | No Change |
| **RTM** |  | | | | |
| **Related Options** | None | | | | |
| **Related Routines** | **Routines “Called By”** | | | **Routines “Called”** | |
|  |  | | |  | |
| **Data Dictionary (DD) References** | Create Cross Reference across entire Insurance file using Fileman   * OPTION 6: UTILITY FUNCTIONS   + UTILITY OPTION 2: CROSS-REFERENCE A FIELD OR FILE     - What type of cross-reference (Traditional or New)? Traditional// N NEW     - File: INSURANCE COMPANY     - Subfile: 17     - Index whole file: YES     - USED FOR: SORTING     - Index Name: “AEDIX”     - Mumps     Using New fields:  17.01 277EDI ID NUMBER  17.02 277DATE EDI ID NUMBER  17.03 277EDI TYPE  17.04 277EDI ID NUMBER ON FILE  Have the set M code be for record-level execution- which, when the entire record is set up make it equal to 17.04 277EDI ID NUMBER ON FILE  Do you want to cross reference the existing data now: NO  To have final result look like :  ^DIC(36,”AEDIX”,DATE,INSURANCE IEN,EDI ID NUMBER,TYPE “P” OR “I”)=EDI ID NUMBER ON FILE (If BLANK considered updated, otherwise attempted)  EXAMPLE SCREEN SHOTS:      Do you want to execute the old kill logic now? YES  Executing old kill logic ... DONE!  Do you want to cross reference existing data now? YES  Executing new set logic ... DONE!  Current Indexes on subfile #36.017:  1454 'AEDIX' whole file index (resides on file #36)    View the cross reference you just set up via :  Select OPTION: 8 DATA DICTIONARY UTILITIES  Select DATA DICTIONARY UTILITY OPTION: 1 LIST FILE ATTRIBUTES  START WITH What File: INSURANCE COMPANY// (1558 entries)  GO TO What File: INSURANCE COMPANY// (1558 entries)  Select SUB-FILE:  Select LISTING FORMAT: STANDARD// INDEXES ONLY  What type of cross-reference (Traditional or New)? Both// NEW  Which field: ALL//  DEVICE: ;;9999999 HOME (CRT) Right Margin: 80//  NEW-STYLE INDEX LIST -- FILE #36INDEX AND CROSS-REFERENCE LIST -- FILE #363/6/17  PAGE 1  -------------------------------------------------------------------------------  Subfile #36.017  New-Style Indexes:  AEDIX (#1454) RECORD MUMPS IR SORTING ONLY WHOLE FILE (#36)  Short Descr: 277STAT TRANSACTION PAYER STORAGE US129  Description:  This cross reference allows for reporting of 277STAT  Set Logic: S ^DIC(36,"AEDIX",X(1),DA(1),X(2),X(3))=X(4)  Set Cond: S X=(X(1)]""&(X(2)]"")&(X(3)]""))  Kill Logic: Q  X(1): 277DATE EDI ID NUMBER (36.017,.02) (Len 10) (forwards)  X(2): 277EDI ID NUMBER (36.017,.01) (Len 30) (forwards)  X(3): 277EDI TYPE (36.017,.03) (Len 1) (forwards)  X(4): 277EDI ID NUMBER ON FILE (36.017,.04) (Len 30)  (forwards) | | | | |
|  |  | | | | |

**Routine Changes**

| * **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCEST | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** |  | | | | | | | | |
| **Related Options** | None | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  |  | | | | |  | | | |
| **Data Dictionary (DD) References** |  | | | | | | | | |
| **Related Protocols** | None | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Routine Name** | IBCEST (changes are Highlighted) | | | | | | | | |
| Original:  IBCEST ;ALB/TMP - 837 EDI STATUS MESSAGE PROCESSING ;17-APR-96  ;;2.0;INTEGRATED BILLING;\*\*137,189,197,135,283,320,368,397,407\*\*;21-MAR-94;Build 29  ;;Per VHA Directive 2004-038, this routine should not be modified.  ; IA 4043 for call to AUDITX^PRCAUDT  Q  ; UPD361(IBTDA) ; Update IB BILL STATUS MESSAGES file  ; IBTDA = ien of return message in file 364.2  …   ;  I $S(IBMNUM="":1,1:'IBBILL&(IBBTCH="")) D DELMSG^IBCESRV2(IBTDA) G UPDQ  ;  ; Individual bill  I IBBILL D  G UPDQ  . N IBA1,IBMSG0,IBPID  . S IBPID="",IBA1=0  . F  S IBA1=$O(^IBA(364.2,IBTDA,2,IBA1)) Q:'IBA1  S IBMSG0=$P($G(^(IBA1,0)),"##RAW DATA: ",2) I +IBMSG0=277,$P(IBMSG0,U,5)="N" S IBPID=$P(IBMSG0,U,11) Q  . S IBSEQ=$P(IB00,U,8) S:IBSEQ="" IBSEQ="P"  . D STORE(IB0,IBBTCH,IBMNUM,IBTDA,IBBILL,IBSEQ,IBPID,1)  ;  ; Batch - update each bill separately  S IBBILL=""  F  S IBBILL=$O(^IBA(364,"ABABI",+IBBTCH,IBBILL)) Q:'IBBILL  D  . Q:$D(^TMP("IBCONF",$J,IBBILL)) ;Bill was rejected  . S IB=$O(^IBA(364,"ABABI",+IBBTCH,IBBILL,0)) Q:'IB  . S IBSEQ=$P($G(^IBA(364,IB,0)),U,8) S:IBSEQ="" IBSEQ="P"  . D STORE(IB0,IBBTCH,IBMNUM,IBTDA,IBBILL,IBSEQ,"",0)  ;  Q  ;  …  ; UPDINS(IBPID,IBINS,IBIFN) ; Update the insurance id or the bill printed at  ; the EDI contractor's print shop and mailed to the ins co.  ; IBPID = the id returned from the EDI contractor for the ins co  ; ("TYPE") = P if professional id or I if institutional id  ; IBINS = the ien of the insurance co it was sent to (file 36)  ; IBIFN = the ien of the claim (file 399)  ;  N IBID,IBIDFLD,IBPRT,IBLOOK,DA,DR,DIE,X,Y,Z  ;  Q:'$G(IBINS)!($G(IBPID)="")  ;  ; Strip spaces off the end of data  S IBLOOK=""  I $L(IBPID) F Z=$L(IBPID):-1:1 I $E(IBPID,Z)'=" " S IBLOOK=$E(IBPID,1,Z) Q  ;  S IBPRT=($E(IBLOOK,2,5)="PRNT")  I IBPRT D  ; Set printed via EDI field on bill  . S DA=IBIFN,DIE="^DGCR(399,",DR="26////1" D ^DIE  ;  S IBLOOK=$E($S('IBPRT:$P(IBLOOK,"PAYID=",2),1:""),1,5)  Q:IBLOOK=""!($E(IBLOOK,2,5)="PRNT")  S IBIDFLD="3.0"\_$S($G(IBPID("TYPE"))="I":4,1:2)  S IBID=$P($G(^DIC(36,+IBINS,3)),U,IBIDFLD\*100#100)  Q:IBID=IBLOOK  I IBID="" D  G UPDINSQ ; Update insurance co electronic id # if blank  . S DIE="^DIC(36,",DR=IBIDFLD\_"////"\_IBLOOK,DA=IBINS D ^DIE  I IBID'="",IBLOOK'="" D  ; Bulletin that the id on file and id returned  . ; are different  . N XMTO,XMDUZ,XMBODY,IBXM,XMSUBJ,XMZ  . S XMTO("I:G.IB EDI")=""  . S XMDUZ="",XMBODY="IBXM",XMSUBJ="PAYER ID RETURNED IS DIFFERENT THAN PAYER ID ON FILE"  . S IBXM(1)="BILL # : "\_$P($G(^DGCR(399,IBIFN,0)),U)  . S IBXM(2)="PAYER : "\_$P($G(^DIC(36,+IBINS,0)),U)  . S IBXM(3)="BILL TYPE : "\_$S($G(IBPID("TYPE"))="I":"INSTITUT",1:"PROFESS")\_"IONAL"  . S IBXM(4)="ID ON FILE : "\_IBID  . S IBXM(5)="ID RETURNED: "\_IBLOOK  . S IBXM(6)=" ",IBXM(7)=" Please determine which id number is correct and correct the id in the",IBXM(8)="insurance file for this payer, if needed"  . D SENDMSG^XMXAPI(XMDUZ,XMSUBJ,XMBODY,.XMTO,,.XMZ)  ; UPDINSQ Q …  Changes:  IBCEST ;ALB/TMP - 837 EDI STATUS MESSAGE PROCESSING ;17-APR-96  ;;2.0;INTEGRATED BILLING;\*\*137,189,197,135,283,320,368,397,407,577\*\*;21  -MAR-94;Build 29  ;;Per VA Directive 6402, this routine should not be modified.  ; IA 4043 for call to AUDITX^PRCAUDT  Q  ;  UPD361(IBTDA) ; Update IB BILL STATUS MESSAGES file  ; IBTDA = ien of return message in file 364.2  ;  N IB,IB0,IBSEQ,IB00,IBBILL,IBBTCH,IBMNUM,IBDATE,IBTYP  ;  I '$$LOCK^IBCEM(IBTDA) G UPDQ ;Lock message in file 364.2  ;  S IB0=$G(^IBA(364.2,IBTDA,0))  S IBMNUM=$P(IB0,U) ; Message number  S IB00=$G(^IBA(364,+$P(IB0,U,5),0)) ; Transmit bill entry  S IBBILL=+IB00 ; Actual bill ien in file 399  S IBBTCH=$P(IB0,U,4) ; Batch #  ;  ; Auto-audit bills based on status code on '10' record of status msg  ; flat file  I IBBILL,$P($T(PRCAUDT+1^PRCAUDT),"\*\*",2)[",173" D  . N Z,Z0,Z1,OK  . Q:+$$STA^PRCAFN(IBBILL)'=104  . S (Z,OK)=0  . F S Z=$O(^IBA(364.2,IBTDA,2,Z)) Q:'Z S Z0=$P($G(^(Z,0)),"##RAW DATA  : ",2) I +Z0=10 S Z0=$P(Z0,U,5) D Q:OK  .. ; Strip leading spaces  .. S Z0=$$TRIM^XLFSTR(Z0)  .. Q:Z0=""  .. I $$SCODE^IBCEST1(Z0),$P($G(^DGCR(399.3,+$P($G(^DGCR(399,IBBILL,0)),  U,7),0)),U,11) D AUDITX^PRCAUDT(IBBILL) S OK=1 ; IA 4043  ;  I $S(IBMNUM="":1,1:'IBBILL&(IBBTCH="")) D DELMSG^IBCESRV2(IBTDA) G UPDQ  ;  ; Individual bill ; KDM US129 IB\*2\*577 rework Individual vs. Batch to  Correct Storage of Payer ID  I IBBILL D UPDTBILL() G UPDQ  ;  ; Batch - update each bill separately  S IBBILL=""  F S IBBILL=$O(^IBA(364,"ABABI",+IBBTCH,IBBILL)) Q:'IBBILL D  . Q:$D(^TMP("IBCONF",$J,IBBILL)) ;Bill was rejected  . S IB=$O(^IBA(364,"ABABI",+IBBTCH,IBBILL,0))  . Q:'IB  . D UPDTBILL() ;KDM US129 IB\*2\*577 Correct Storage of PAYER ID  ;  Q  ;  UPDTBILL() ;KDM US129 IB\*2\*577 New section to Correct Storage of PAYER ID  N IBA1,IBMSG0,IBPID  S IBPID="",IBA1=0  ;  F S IBA1=$O(^IBA(364.2,IBTDA,2,IBA1)) Q:'IBA1 D Q:IBPID]""  . S IBMSG0=$P($G(^(IBA1,0)),"##RAW DATA: ",2)  . I +IBMSG0=277,$P(IBMSG0,U,5)="N" S IBPID=$P(IBMSG0,U,11)  ;  S IBSEQ=$P(IB00,U,8) S:IBSEQ="" IBSEQ="P"  D STORE(IB0,IBBTCH,IBMNUM,IBTDA,IBBILL,IBSEQ,IBPID,1)  Q  ;  STORE(IB0,IBBTCH,IBMNUM,IBTDA,IBBILL,IBSEQ,IBPID,IB1) ;  ;  ; IB0 = 0-node of message in file 364.2  ; IBBTCH = ien of batch in file 364.1  ; IBMNUM = actual message number  ; IBTDA = ien of message in file 364.2  ; IBBILL = ien of bill in 399  ; IBSEQ = P/S/T/ for COB sequence related to message  ; IBPID = the payer id returned from clearinghouse for the claim  ; IB1 = flag that says if the message was for a single bill or a batch.  ; Batch statuses have an additional standard text entry.  ; 1 = single bill 0 = batchtuff fields into existing entry  . ; (may be needed for reprocessing of aborted updates)  . S DIE="^IBM(361,",DA=IBDUP,DR=IBFLDS\_";1///@"  . D ^DIE  . I $D(Y) S IBY=-1 Q ;Update not successful  . S IBY=IBDUP  ;  K IBT  I 'IBDUP D ; Create new entry and stuff fields  . S DIC(0)="L",DIC="^IBM(361,",DLAYGO=361  . S DIC("DR")=IBFLDS  . D FILE^DICN  . K DO,DD,DLAYGO,DIC  . S IBY=+Y  . Q:IBY'>0  . ;  . ; IB\*2\*320 - Check for duplicate status message  . NEW IBNEW,IBOLD,PCE,Z,DIK,DA  . S IBNEW=""  . F PCE=3,4,5,7,8,11,15 S IBNEW=IBNEW\_$P($G(^IBM(361,IBY,0)),U,PCE)\_U  . S Z=0  . F S Z=$O(^IBM(361,"B",IBBILL,Z)) Q:'Z I Z'=IBY D Q:IBY'>0  .. S IBOLD=""  .. F PCE=3,4,5,7,8,11,15 S IBOLD=IBOLD\_$P($G(^IBM(361,Z,0)),U,PCE)\_U  .. I IBNEW'=IBOLD Q ; no duplicate so get the next one  .. S DIK="^IBM(361,",DA=IBY,IBY=-1 D ^DIK D DELMSG^IBCESRV2(IBTDA)  .. Q  . Q  ;  I IBY>0 D ;Move text over  . K IBT  . ;  . D BLDMSG(IB1,IBTDA,.IBT,.IBAUTO)  . ;  . ; IB\*2\*368 - ymg - 2Q,RE,RP messages will be filed as informational  . ; Z0 is the flag for 2Q code  . ; Z1 is the flag for RE code  . ; Z2 is the flag for RP code  . ; Z3 is the flag for autofiling the message  . I $P($G(^IBM(361,+IBY,0)),U,3)="R" D  .. S Z="",(Z0,Z1,Z2,Z3)=0 F S Z=$O(IBT(Z)) Q:Z=""!(Z3=1) D  ... S IBLN=$$UP^XLFSTR($G(IBT(Z)))  ... I (Z0!Z1!Z2)=0 D  .... S:IBLN?.E1"CODE:".P1"2Q".E Z0=1  .... S:IBLN?.E1"CODE:".P1"RE".E Z1=1  .... S:IBLN?.E1"CODE:".P1"RP".E Z2=1  ... I Z0=1 S:IBLN?.P1"CLAIM".P1"REJECTED".P1"BY".P1"CLEARINGHOUSE".E Z3  =1  ... I Z1=1 S:IBLN?.P1"ELECTRONIC".P1"CLAIM".P1"REJECTED".P1"BY".P1"EMDE  ON".E Z3=1  ... I Z2=1 S:IBLN?.P1"PAPER".P1"CLAIM".P1"REJECTED".P1"BY".P1"EMDEON".E  Z3=1  .. I Z3=1 S IBAUTO=1,DIE=361,DA=+IBY,DR=".03////I" D ^DIE  .. Q  . ;  . ; if info msg, ck for no review needed based on first line of text  . I $G(IBAUTO),$P($G(^IBM(361,+IBY,0)),U,3)="I" D  .. S DIE="^IBM(361,",DR=".09////2;.14////1;.1////F",DA=+IBY D ^DIE  .. I IB1,$P($G(^IBM(361,+IBY,0)),U,11) S Z="",Z0=0 F S Z=$O(IBT(Z)) Q:  Z=""!(Z0=1) D  ... S Z0=$$PRINTUPD^IBCEU0($$UP^XLFSTR($G(IBT(Z))),$P($G(^IBM(361,+IBY,  0)),U,11))  . ;  . D MSGLNSZ(.IBT) ; Convert Message Lines in IBT to be no longer than 7  0 chars  . D WP^DIE(361,+IBY\_",",1,"A","IBT") ; file message text  . ;  . ; Delete message after it successfully updates the database.  . D DELMSG^IBCESRV2(IBTDA)  . Q  ;  UPDQ L -^IBA(364.2,IBTDA,0)  Q  ;  BLDMSG(IB1,IBTDA,IBT,IBAUTO) ; Builds message text  ; IB1 = flag for batch message  ; IBTDA = ien of entry in file 364.2  ; IBT = array returned with message text  ; IBAUTO = if passed by reference, returns 1 if text indicates review  ; not needed  N IBDATA,IBCK,IBZ,IBZ0,IBZ1,Z  S (IBZ,IBZ0,IBDATA,IBAUTO,IBCK)=0  I 'IB1 S IBT(1)="Status message received for batch "\_$P($G(^IBA(364.1,I  BBTCH,0)),U)\_" dated "\_$$FMTE^XLFDT($P($G(^IBA(364.2,IBTDA,0)),U,10),2),IBZ0=1  ; Don't move the raw data over, just move the text of the message  F S IBZ=$O(^IBA(364.2,IBTDA,2,IBZ)) Q:'IBZ S IBZ1=$G(^(IBZ,0)) S IBDA  TA=($E(IBZ1,1,2)="##") Q:IBDATA S IBZ0=IBZ0+1,IBT(IBZ0)=IBZ1 I 'IBCK S Z=$$CKRE  VU^IBCEM4(IBZ1,,,.IBCK),IBAUTO=$S(IBCK:0,Z:1,1:IBAUTO)  Q  ;  UPDINS(IBPID,IBINS,IBIFN,IBTDA) ;KDM US129 IB\*2\*577  ; Update the insurance id or the bill printed at  ; the EDI contractor's print shop and mailed to the ins co.  ; IBPID = the id returned from the EDI contractor for the ins co  ; ("TYPE") = P if professional id or I if institutional id  ; IBINS = the ien of the insurance co it was sent to (file 36)  ; IBIFN = the ien of the claim (file 399)  ; IBTDA = ien of entry in file 364.2 ;KDM US129 IB\*2\*577  ;  N IBID,IBIDFLD,IBPRT,IBLOOK,DA,DR,DIE,X,Y,Z,UPD **;**KDM US129 IB\*2\*577  **;**  Q:'$G(IBINS)!($G(IBPID)="")  ;  ; Strip spaces off the end of data  S IBLOOK=""  I $L(IBPID) F Z=$L(IBPID):-1:1 I $E(IBPID,Z)'=" " S IBLOOK=$E(IBPID,1,Z  ) Q  ;  S IBPRT=($E(IBLOOK,2,5)="PRNT")  I IBPRT D ; Set printed via EDI field on bill  . S DA=IBIFN,DIE="^DGCR(399,",DR="26////1" D ^DIE  ;  ; KDM US129 IB\*2\*577 correct payer ID storage  ;S IBLOOK=$E($S('IBPRT:$P(IBLOOK,"PAYID=",2),1:""),1,5)  ;Q:IBLOOK=""!($E(IBLOOK,2,5)="PRNT")  I IBPRT Q  I IBLOOK'["PAYID=",IBLOOK'["COBID=" Q ;KDM US129 IB\*2\*577  S IBLOOK=$E($P(IBLOOK,"ID=",2),1,5)  Q:IBLOOK=""  ;  S IBIDFLD="3.0"\_$S($G(IBPID("TYPE"))="I":4,1:2)  S IBID=$P($G(^DIC(36,+IBINS,3)),U,IBIDFLD\*100#100)  Q:IBID=IBLOOK  S IBDATE=$P($P($G(^IBA(364.2,IBTDA,0)),U,10),".",1),IBTYP=$G(IBPID("TYP  E")) ;KDM US129 IB\*2\*577  I IBID="" D G UPDINSQ ; Update insurance co electronic id # if blank  . S DIE="^DIC(36,",DR=IBIDFLD\_"////"\_IBLOOK,DA=IBINS D ^DIE  . D UPDLOG(1,IBDATE,IBINS,IBLOOK,IBTYP,IBID) ;KDM US129 IB\*2\*577  I IBID'="",IBLOOK'="" D ; Bulletin that the id on file and id returned  . ; are different  . N XMTO,XMDUZ,XMBODY,IBXM,XMSUBJ,XMZ  . S XMTO("I:G.IB EDI")=""  . S XMDUZ="",XMBODY="IBXM",XMSUBJ="PAYER ID RETURNED IS DIFFERENT THAN  PAYER ID ON FILE"  . S IBXM(1)="BILL # : "\_$P($G(^DGCR(399,IBIFN,0)),U)  . S IBXM(2)="PAYER : "\_$P($G(^DIC(36,+IBINS,0)),U)  . S IBXM(3)="BILL TYPE : "\_$S($G(IBPID("TYPE"))="I":"INSTITUT",1:"PROF  ESS")\_"IONAL"  . S IBXM(4)="ID ON FILE : "\_IBID  . S IBXM(5)="ID RETURNED: "\_IBLOOK  . S IBXM(6)=" ",IBXM(7)=" Please determine which id number is correct  and correct the id in the",IBXM(8)="insurance file for this payer, if needed"  . D SENDMSG^XMXAPI(XMDUZ,XMSUBJ,XMBODY,.XMTO,,.XMZ)  . D UPDLOG(0,IBDATE,IBINS,IBLOOK,IBTYP,IBID) ;KDM US129, US976 IB\*2  \*577  ;  UPDINSQ Q  ;  UPDLOG(UPD,IBDATE,IBINS,IBLOOK,IBTYP,IBID) ;KDM US129, US976 IB\*2\*577 New sec  tion for New Payer Report  ; store flds for reporting purposes when updating or attempting to upda  te Payer information (US129)  ; ^DIC(36 -17.0 277EDI ID Number  ; 17.01 277EDI ID Number  ; 17.02 277Date EDI ID Number  ; 17.03 277EDI Type (P)ROF or (I)nst  ; 17.04 277EDI ID NUMBER ON FILE ;if blank it was an update ot  herwise it was an attempted update.  ;  Q:(($D(^DIC(36,"AEDIX",IBDATE,IBINS,IBLOOK,IBTYP)))&(UPD=0)) ;store  only one attempt a day  N ERROR,IBFDA,LEV  S LEV="+2,"\_IBINS\_","  S IBFDA(36.017,LEV,.01)=IBLOOK ;New Value from 277STAT  S IBFDA(36.017,LEV,.02)=IBDATE ;Date from 277STAT FM format  S IBFDA(36.017,LEV,.03)=IBTYP ;"P" or "I"  S IBFDA(36.017,LEV,.04)=$G(IBID) ;Value already on file- if blank it  was an update, otherwise attempted update  D UPDATE^DIE("","IBFDA","","ERROR")  Q  ;  MSGLNSZ(MSG) ; Change Input Message Lines to be no more than 70 characters long  each  ;  ; Input/Output: MSG - array of Input Message Lines; this is also the  Output Message  ; which is an array of Converted Message Lines (with lines no more than  70 chars each)  ;  N LN,XARY,XARYLN,CNT,OUTMSG,TMPMSG,LDNGSP,LDNGSPN  S LN="",CNT=0 F S LN=$O(MSG(LN)) Q:LN="" D ;  . ; Find any leading spaces in original message line,  . ; to be used if line got split below  . S TMPMSG=$$TRIM^XLFSTR(MSG(LN),"L"," ") ;Trim Leading Spaces  . S LDNGSP=$P(MSG(LN),TMPMSG,1) ;get leading spaces if any  . S LDNGSPN=$L(LDNGSP) S:LDNGSPN>30 LDNGSP=$E(LDNGSP,1,30) ;make sure t  here are no more than 30 leading spaces  . ; Converts a single line to multiple lines with a maximum width of 70  each  . ; If line is 70 chars or less, this call returns the exact line  . K XARY D FSTRNG^IBJU1(TMPMSG,70-LDNGSPN,.XARY)  . ; Scan lines and merge them into the final output array (OUTMSG)  . ; On lines 2 and higher, add Leading Spaces found above, if any.  . S XARYLN="" F S XARYLN=$O(XARY(XARYLN)) Q:XARYLN="" S CNT=CNT+1,OUT  MSG(CNT)=LDNGSP\_XARY(XARYLN)  ;  ; Move the final Message Lines (OUTMSG) into MSG array to be returned  K MSG M MSG=OUTMSG  Q  ;  ... | | | | | | | | | |

**Sample 277STAT flat file:**

402MCHRUEVDRN00254277STAT^P^20161124^1200^N^0000004^0000000^00000067498^00000000000^CIGNA ^623080000 ^N^EMDEON ^4020LQ ^ ~

402MCHRUEVDRN0025409^442-K702Q09 ^ ^ ^ ^ ~

402MCHRUEVDRN0025410^442-K702Q09 ^A^I^AE ^Accepted Claim sent out electronically ^ ~

402MCHRUEVDRN0000499^$

402MCHRUEVDRN00254277STAT^P^20161124^1200^N^0000004^0000000^00000067498^00000000000^AETNA US HEALTHCARE ^600540000 ^N^EMDEON ^4020LQ ^ ~

402MCHRUEVDRN0025409^442-K702Q0D ^ ^ ^ ^ ~

402MCHRUEVDRN0025410^442-K702Q0D ^A^I^AE ^Accepted Claim sent out electronically ^ ~

402MCHRUEVDRN0000499^$

402MCHRUEVDRN00254277STAT^P^20161124^1200^N^0000004^0000000^00000067498^00000000000^AETNA US HEALTHCARE ^600540000 ^N^EMDEON ^4020LQ ^ ~

402MCHRUEVDRN0025409^442-K702Q0C ^ ^ ^ ^ ~

402MCHRUEVDRN0025410^442-K702Q0C ^A^I^AE ^Accepted Claim sent out electronically ^ ~

402MCHRUEVDRN0000499^$

402MCHRUEVDRN00254277STAT^P^20161124^1200^N^0000004^0000000^00000067498^00000000000^BLUE CROSS CA (NE) ^TEST12000 ^N^EMDEON ^4020LQ ^ ~

402MCHRUEVDRN0025409^442-K702Q0B ^ ^ ^ ^ ~

402MCHRUEVDRN0025410^442-K702Q0B ^A^I^AE ^Accepted Claim sent out electronically ^ ~

402MCHRUEVDRN0025413^442-K702Q0B ^EP01220002AFRV9 ^ ^ ^ ^P^ ~

402MCHRUEVDRN0025415^442-K702Q0B ^IB ^PATIENT ^C ^ ^ ^ ^ ^R4545456 ^20161020^20161020 ~

402MCHRUEVDRN0000499^$

**Helpful hints for development and testing:**

Make sure the mail group “MCT” is set up with REMOTE MEMBER:

“S.IBCE MESSAGES [DNS](mailto:DNS)” in order to run and test the 277STAT file.

NAME: IBCE MESSAGES SERVER              MENU TEXT: Return Message Server

  TYPE: server                          CREATOR: SHURMAN,JILLIAN A

  PACKAGE: INTEGRATED BILLING

DESCRIPTION:   This option controls the reading and storing of return messages

generated as a result of the processing of Integrated Billing electronic

transmissions with the Austin translator.

  ROUTINE: SERVER^IBCESRV               SERVER ACTION: RUN IMMEDIATELY

  SERVER AUDIT: NO                      SUPRESS BULLETIN: YES, SUPRESS IT

  SERVER REPLY: NO REPLY (DEFAULT)

  UPPERCASE MENU TEXT: RETURN MESSAGE SERVER