**TAS eBill SDD US2486**

System Design Document

IB\*2.0\*608



Department of Veterans Affairs

**April 2018**

Version 2.00

**User Story Number:** US-2486

**User Story Name:** RCB – Match COB Data to Payer Sequence

**Product Backlog ID:** n/a

**Rally ID:** US-2486

# Design/Resolution:

To resolve this request, the following items need to be done:

* The IBCEPTC3 routine needs to be modified to institute a new IB variable (IBRSBTST) that will be needed to identify that a claim is being resubmitted. The IBRSBTST variable will be referenced by the OUTPUT FORMATTER:

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCEPTC3 | | | | | | | | |
| **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 |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Current Logic** | | | | | | | | | |
| IBCEPTC3 ;ALB/ESG - EDI PREVIOUSLY TRANSMITTED CLAIMS ACTIONS ;12/19/05  ;;2.0;INTEGRATED BILLING;\*\*320,547\*\*;21-MAR-94;Build 119  ;;Per VA Directive 6402, this routine should not be modified.  ;  ; IB\*2.0\*547 Variable IBLOC is pre-defined (in IBCEPTC)   Q  ;   ; RESUB ; Resubmit selected claims  N DIR,X,Y,IBIFN,IB364,Z1,IBTYPPTC,DIRUT,DIROUT,DTOUT,DUOUT,IBFSKIP,IBABORT  D FULL^VALM1  I '$O(^TMP("IB\_PREV\_CLAIM\_SELECT",$J,0)) D  G RESUBQ  . N DIR,X,Y  . S DIR(0)="EA",DIR("A")="No claims have been selected - Press return to continue " D ^DIR K DIR  ;  ; Ask user if resubmit as production or as test  S DIR(0)="SA^P:Production;T:Test Only"  S DIR("A")="Resubmit Claims: "  S DIR("B")="Production"  S DIR("?",1)=" Select Production to resubmit the claims to the payer for payment."  S DIR("?")=" Select Test to resubmit the claims as Test claims only."  ; IB\*2.0\*547 Only allow locally printed claims to resubmit as Test  W ! I IBLOC'=1 D ^DIR K DIR  I $D(DIRUT) G RESUBQ  S IBTYPPTC="TEST"  I IBLOC'=1,Y="P" S IBTYPPTC="PRODUCTION"   ;  S DIR(0)="YA",DIR("B")="No"  S DIR("A",1)="You are about to resubmit "\_+^TMP("IB\_PREV\_CLAIM\_SELECT",$J)\_" claims as "\_IBTYPPTC\_" claims."  S DIR("A")="Are you sure you want to continue?: "  W ! D ^DIR K DIR  I Y'=1 G RESUBQ  ;  ; OK to proceed and resubmit  W !!,"Resubmission in process ... "  ;  ; loop thru selected claims and set into scratch globals  S IBFSKIP=0  KILL ^TMP("IBRCBOLD",$J)  S IBIFN=0 F  S IBIFN=$O(^TMP("IB\_PREV\_CLAIM\_SELECT",$J,IBIFN)) Q:'IBIFN  S Z1=+$G(^(IBIFN)),IB364=+$G(^(IBIFN,0)) I IB364 D  . ;  . I IBTYPPTC="TEST" D  .. S ^TMP("IBEDI\_TEST\_BATCH",$J,IB364)=""  .. S ^TMP("IBRESUBMIT",$J,IB364)=""  .. I Z1 D MARK(IBIFN,"",Z1,IB364,0,.VALMHDR)  .. Q  . ;  . I IBTYPPTC="PRODUCTION" D  .. I '$$TXOK(IBIFN) S IBFSKIP=IBFSKIP+1 Q    ; transmission not allowed  .. N Y S Y=$$ADDTBILL^IBCB1(IBIFN) ; new entry in file 364 - "X" status  .. I '$P(Y,U,3) Q                   ; quit if new entry didn't get added  .. S ^TMP("IBSELX",$J,+Y)=""  .. S ^TMP("IBRCBOLD",$J,IB364)=""   ; save list of old transmit bills  .. I Z1 D MARK(IBIFN,"",Z1,IB364,0,.VALMHDR)  .. Q  . ;  . Q  ;  ; set top level of scratch globals based on test or production  I IBTYPPTC="TEST" S ^TMP("IBRESUBMIT",$J)="^^0^1",^TMP("IBEDI\_TEST\_BATCH",$J)=1  E  KILL ^TMP("IBRESUBMIT",$J),^TMP("IBEDI\_TEST\_BATCH",$J),^TMP("IBONE",$J) S ^TMP("IBSELX",$J)=0  ;  ; resubmit call  D EN1^IBCE837B("","","",.IBABORT)  ;  ; if user aborted at the last minute, then get rid of the new entries  ; in file 364 that were added for production claim sending  I IBABORT D  . N IB,DIK,DA  . S IB=0 F  S IB=$O(^TMP("IBSELX",$J,IB)) Q:'IB  S DIK="^IBA(364,",DA=IB D ^DIK  . Q  ;  ; update EDI files for the old transmit bills  I 'IBABORT D  . N IB  . S IB=0 F  S IB=$O(^TMP("IBRCBOLD",$J,IB)) Q:'IB  D UPDEDI^IBCEM(IB,"R")  . Q  ;  ; cleanup  K ^TMP("IBEDI\_TEST\_BATCH",$J),^TMP("IBRESUBMIT",$J),^TMP("IBSELX",$J),^TMP("IBRCBOLD",$J)  I '$O(^TMP("IB\_PREV\_CLAIM\_SELECT",$J,0)) K ^TMP("IB\_PREV\_CLAIM\_SELECT",$J)  S DIR(0)="EA"  S DIR("A",1)="Selected claims have been resubmitted as "\_IBTYPPTC\_"."  I IBFSKIP D  . S DIR("A",2)="Please note: Some claims were not eligible to be resubmitted as live claims."  . S DIR("A",3)=" These claims are still indicated as being selected."  . Q  I IBABORT K DIR("A") S DIR("A",1)="No claims were resubmitted."  S DIR("A")="Press return to continue "  W ! D ^DIR K DIR  K VALMHDR  ; RESUBQ ;  S VALMBCK="R"  Q      ; TXOK(IBIFN) ; Function determines if claim is OK for live resubmission  NEW OK,IB364,IBD,IBSTAT  S OK=0  I '$P($G(^DGCR(399,+$G(IBIFN),"TX")),U,2) G TXOKX                  ; last electronic extract date  I '$F(".2.3.4.","."\_$P($G(^DGCR(399,IBIFN,0)),U,13)\_".") G TXOKX   ; claim status  S IB364=+$$LAST364^IBCEF4(+$G(IBIFN)) I 'IB364 G TXOKX             ; transmit bill exists  S IBD=$G(^IBA(364,IB364,0)) I IBD="" G TXOKX  S IBSTAT=$P(IBD,U,3) I IBSTAT="X" G TXOKX                          ; already awaiting extract  S OK=1 TXOKX ;  Q OK  ; | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCEPTC3 ;ALB/ESG - EDI PREVIOUSLY TRANSMITTED CLAIMS ACTIONS ;12/19/05  ;;2.0;INTEGRATED BILLING;\*\*320,547,608\*\*;21-MAR-94;Build 119  ;;Per VA Directive 6402, this routine should not be modified.  ;  ; IB\*2.0\*547 Variable IBLOC is pre-defined (in IBCEPTC)  ; IB\*2.0\*608 (vd) provided the ability to identify those claims that are resubmitted  ; and those that are skipped. (US2486)  Q  ;  RESUB ; Resubmit selected claims   N DIR,DIRCTR,DIRLN,DIROUT,DIRUT,DTOUT,DUOUT  N IB364,IBABORT,IBCLMNO,IBIFN,IBSKCTR,IBFSKIP,IBRSBTST,IBTYPPTC  N X,Y,Z1  ;/IB\*2\*608 - vd (US2486) - instituted the following variable to identify a claim as being resubmitted.  S IBRSBTST=0   D FULL^VALM1  I '$O(^TMP("IB\_PREV\_CLAIM\_SELECT",$J,0)) D  G RESUBQ  . N DIR,X,Y  . S DIR(0)="EA",DIR("A")="No claims have been selected - Press return to continue " D ^DIR K DIR  ;  ; Ask user if resubmit as production or as test  S DIR(0)="SA^P:Production;T:Test Only"  S DIR("A")="Resubmit Claims: "  S DIR("B")="Production"  S DIR("?",1)=" Select Production to resubmit the claims to the payer for payment."  S DIR("?")=" Select Test to resubmit the claims as Test claims only."  ; IB\*2.0\*547 Only allow locally printed claims to resubmit as Test  W ! I IBLOC'=1 D ^DIR K DIR  I $D(DIRUT) G RESUBQ  S IBTYPPTC="TEST"  I IBLOC'=1,Y="P" S IBTYPPTC="PRODUCTION"  ;/IB\*2\*608 (vd) - The following line indicates the claim is being resubmitted as a "TEST" Claim and should be handled  ; special concerning the COB, OFFSET, PRIOR PAYMENTS calculations by the Output Formatter. (US2486)  I IBTYPPTC="TEST" S IBRSBTST=1   ;  S DIR(0)="YA",DIR("B")="No"  S DIR("A",1)="You are about to resubmit "\_+^TMP("IB\_PREV\_CLAIM\_SELECT",$J)\_" claims as "\_IBTYPPTC\_" claims."  S DIR("A")="Are you sure you want to continue?: "  W ! D ^DIR K DIR  I Y'=1 G RESUBQ  ;  ; OK to proceed and resubmit  W !!,"Resubmission in process ... "  ;  ; loop thru selected claims and set into scratch globals  S IBFSKIP=0  ;KILL ^TMP("IBRCBOLD",$J)   K ^TMP("IBRCBOLD",$J)   K ^TMP("IBSKIPPED",$J) ;/IB\*2\*608 - vd  S IBIFN=0 F  S IBIFN=$O(^TMP("IB\_PREV\_CLAIM\_SELECT",$J,IBIFN)) Q:'IBIFN  S Z1=+$G(^(IBIFN)),IB364=+$G(^(IBIFN,0)) I IB364 D  . ;  . I IBTYPPTC="TEST" D   .. S ^TMP("IBEDI\_TEST\_BATCH",$J,IB364)=""  .. S ^TMP("IBRESUBMIT",$J,IB364)=""  .. I Z1 D MARK(IBIFN,"",Z1,IB364,0,.VALMHDR)  .. Q  . ;  . I IBTYPPTC="PRODUCTION" D   .. ;/IB\*2\*680 (vd) - modified the following line for US2486 as shown below.  .. ; I '$$TXOK(IBIFN) S IBFSKIP=IBFSKIP+1 Q ; transmission not allowed  .. I '$$TXOK(IBIFN) D  Q   ;transmission not allowed  ... S IBCLMNO=$$GET1^DIQ(399,IBIFN,.01)  ... S IBFSKIP=IBFSKIP+1  ... S ^TMP("IBSKIPPED",$J,IBCLMNO)=IBIFN  ; /IB\*2\*608 (vd) - Added to identify those claims that are Skipped   .. N Y S Y=$$ADDTBILL^IBCB1(IBIFN) ; new entry in file 364 - "X" status  .. I '$P(Y,U,3) Q                   ; quit if new entry didn't get added  .. S ^TMP("IBSELX",$J,+Y)=""  .. S ^TMP("IBRCBOLD",$J,IB364)=""   ; save list of old transmit bills  .. I Z1 D MARK(IBIFN,"",Z1,IB364,0,.VALMHDR)  .. Q  . ;  . Q  ;  ; set top level of scratch globals based on test or production  I IBTYPPTC="TEST" S ^TMP("IBRESUBMIT",$J)="^^0^1",^TMP("IBEDI\_TEST\_BATCH",$J)=1  E  KILL ^TMP("IBRESUBMIT",$J),^TMP("IBEDI\_TEST\_BATCH",$J),^TMP("IBONE",$J) S ^TMP("IBSELX",$J)=0  ;  ; resubmit call   D EN1^IBCE837B("","","",.IBABORT,$G(IBRSBTST)) ;/IB\*2\*608 (vd) - added the IBRSBTST parameter for US2486.   ;  ; if user aborted at the last minute, then get rid of the new entries  ; in file 364 that were added for production claim sending  I IBABORT D  . N IB,DIK,DA  . S IB=0 F  S IB=$O(^TMP("IBSELX",$J,IB)) Q:'IB  S DIK="^IBA(364,",DA=IB D ^DIK  . Q  ;  ; update EDI files for the old transmit bills  I 'IBABORT D  . N IB  . S IB=0 F  S IB=$O(^TMP("IBRCBOLD",$J,IB)) Q:'IB  D UPDEDI^IBCEM(IB,"R")  . Q  ;  ; cleanup  K ^TMP("IBEDI\_TEST\_BATCH",$J),^TMP("IBRESUBMIT",$J),^TMP("IBSELX",$J),^TMP("IBRCBOLD",$J)  I '$O(^TMP("IB\_PREV\_CLAIM\_SELECT",$J,0)) K ^TMP("IB\_PREV\_CLAIM\_SELECT",$J)  S DIR(0)="EA"  S DIR("A",1)="Selected claims have been resubmitted as "\_IBTYPPTC\_"."  I IBFSKIP D  . S DIR("A",2)="Please note: Some claims were not eligible to be resubmitted as live claims."  . S DIR("A",3)=" These claims are still indicated as being selected."  . S DIR("A",4)="The following are the claims that were skipped:"  . S (DIRLN,IBCLMNO)="",IBSKCTR=0,DIRCTR=4  . F  S IBCLMNO=$O(^TMP("IBSKIPPED",$J,IBCLMNO)) Q:IBCLMNO=""  D  . . S IBSKCTR=IBSKCTR+1 ; Increment # of claims on the display line.  . . I IBSKCTR>6 D     ; Want no more than 6 claim numbers displayed per display line.  . . . S DIRCTR=DIRCTR+1,DIR("A",DIRCTR)=DIRLN   ; increment the DIR("A",...) display line and set the line.  . . . S IBSKCTR=1,DIRLN=""   ; reset the line segment ctr and clear the display line.  . . ;  . . S DIRLN=DIRLN\_" "\_IBCLMNO   ; Append the claim # to the existing display line.  . I +IBSKCTR S DIRCTR=DIRCTR+1,DIR("A",DIRCTR)=DIRLN  . ;  . Q  K ^TMP("IBSKIPPED",$J) ;/IB\*2\*608 (vd) - delete the temporary list of skipped claims after reporting them.       ; TXOK(IBIFN) ; Function determines if claim is OK for live resubmission  NEW OK,IB364,IBD,IBSTAT  S OK=0   ;/IB\*2\*608 (vd) - added the following line for US2486.  I $D(^IBM(361.1,"ABS",+$G(IBIFN),$$COBN^IBCEF(IBIFN))) G TXOKX  ; Not okay for claim w/EOB for this payer sequence  ;   I '$P($G(^DGCR(399,+$G(IBIFN),"TX")),U,2) G TXOKX                  ; last electronic extract date  I '$F(".2.3.4.","."\_$P($G(^DGCR(399,IBIFN,0)),U,13)\_".") G TXOKX   ; claim status  S IB364=+$$LAST364^IBCEF4(+$G(IBIFN)) I 'IB364 G TXOKX             ; transmit bill exists  S IBD=$G(^IBA(364,IB364,0)) I IBD="" G TXOKX  S IBSTAT=$P(IBD,U,3) I IBSTAT="X" G TXOKX                          ; already awaiting extract  S OK=1 TXOKX ;  Q OK  ; | | | | | | | | | |

* The IBCE837B routine needs to be modified to reference the new IB variable (IBRESUBM):

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCE837B | | | | | | | | |
| **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 |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Current Logic** | | | | | | | | | |
| IBCE837B ;ALB/TMP - OUTPUT FOR 837 TRANSMISSION (cont) ;24-APR-01  ;;2.0;INTEGRATED BILLING;\*\*137,191,197,320\*\*;21-MAR-94  ; QTXMT(IBSITE8) ; Run now or queue EDI bill transmit (part of nightly job)  ; IBSITE8 = the '8' node of the site parameters file (350.9)  I $P(IBSITE8,U,6)="" D  . D EN1("",1)  E  D  . N Z,Z0  . F Z=1:1:4 S Z0=$P($P(IBSITE8,U,6),";",Z) Q:Z0=""  D  .. S IBQDT=$S($E(($$NOW^XLFDT()#1\*10000\1)\_"0000",1,4)>Z0:$$FMADD^XLFDT(DT,1),1:DT)\_"."\_Z0  ; If time to queue is after the current time, queue for the next day  .. D EN1("",1,+IBQDT)  Q  ; EN1(IBZTSK,IBNOASK,IBQDT,IBABORT) ;Txmit 837's for bills waiting transmission  ; Pass IBZTSK by ref to get task # if job is tasked  ; IBNOASK = flag: 0 = ask to queue, 1 = no ask to queue  ; IBQDT = if IBNOASK=1, this can be the date/time to queue the job  ; Pass IBABORT by reference to determine if user aborted process  N Z S Z=$O(^IBE(353,"B","IB 837 TRANSMISSION",0)),Z=$S($P($G(^IBE(353,+Z,2)),U,8):$P(^(2),U,8),1:Z)  I Z D  . I $D(^TMP("IBRESUBMIT",$J))!($G(IBNOASK)) D FORM^IBCEFG7(Z,$P($G(^IBE(350.9,1,8)),U),+$G(IBNOASK),$S($G(IBNOASK):$G(IBQDT),1:""),.IBZTSK,.IBABORT) Q  . D FORM^IBCEFG7(Z,$P($G(^IBE(350.9,1,8)),U),+$G(IBNOASK),$S($G(IBNOASK):$G(IBQDT),1:""),"",.IBABORT)  Q  ; CMN484 ;  W !,"YO, YO!!"  Q  ; | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCE837B ;ALB/TMP - OUTPUT FOR 837 TRANSMISSION (cont) ;24-APR-01  ;;2.0;INTEGRATED BILLING;\*\*137,191,197,320,608\*\*;21-MAR-94  ; QTXMT(IBSITE8) ; Run now or queue EDI bill transmit (part of nightly job)  ; IBSITE8 = the '8' node of the site parameters file (350.9)  I $P(IBSITE8,U,6)="" D  . D EN1("",1)  E  D  . N Z,Z0  . F Z=1:1:4 S Z0=$P($P(IBSITE8,U,6),";",Z) Q:Z0=""  D  .. S IBQDT=$S($E(($$NOW^XLFDT()#1\*10000\1)\_"0000",1,4)>Z0:$$FMADD^XLFDT(DT,1),1:DT)\_"."\_Z0  ; If time to queue is after the current time, queue for the next day  .. D EN1("",1,+IBQDT)  Q  ; EN1(IBZTSK,IBNOASK,IBQDT,IBABORT,IBRSBTST) ;Txmit 837's for bills waiting transmission  ; Pass IBZTSK by ref to get task # if job is tasked  ; IBNOASK = flag: 0 = ask to queue, 1 = no ask to queue  ; IBQDT = if IBNOASK=1, this can be the date/time to queue the job  ; Pass IBABORT by reference to determine if user aborted process   ;  ; IBRSBTST = 1 /IB\*2\*608 (vd) Instituted this variable for US2486 to indicate  ; a claim is being resubmitted and should be handled special  ; concerning the COB, OFFSET, PRIOR PAYMENTS calculation by the  ; OUTPUT FORMATTER. This is for "TEST" only.  ;   N Z S Z=$O(^IBE(353,"B","IB 837 TRANSMISSION",0)),Z=$S($P($G(^IBE(353,+Z,2)),U,8):$P(^(2),U,8),1:Z)  I Z D  . ;/IB\*2\*608 (vd) - US2486 added the parameter IBRESUBM to the following 2 lines.   . I $D(^TMP("IBRESUBMIT",$J))!($G(IBNOASK)) D FORM^IBCEFG7(Z,$P($G(^IBE(350.9,1,8)),U),+$G(IBNOASK),$S($G(IBNOASK):$G(IBQDT),1:""),.IBZTSK,.IBABORT,$G(IBRSBTST)) Q  . D FORM^IBCEFG7(Z,$P($G(^IBE(350.9,1,8)),U),+$G(IBNOASK),$S($G(IBNOASK):$G(IBQDT),1:""),"",.IBABORT,$G(IBRSBTST))  Q  ; CMN484 ;  W !,"YO, YO!!"  Q  ; | | | | | | | | | |

* The IBCEFG7 routine needs to be modified to reference the new IB variable (IBRESUBM):

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCEFG7 | | | | | | | | |
| **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 |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Current Logic** | | | | | | | | | |
| IBCEFG7 ;ALB/TMP - OUTPUT FORMATTER GENERIC FORM PROCESSING ;06-MAR-96  ;;2.0;INTEGRATED BILLING;\*\*52,84,96,51,137,191,320\*\*;21-MAR-94  ;  Q  ; FORM(IBFORM,IBQUE,IBNOASK,IBQDT,ZTSK,IBABORT) ;For ien IBFORM, extract data using  ; output generater  ; IBQUE = the output queue for transmitted forms or the printer queue  ; for printed output  ; IBNOASK = flag that says user interaction for queuing is not needed  ; 0 or null = ask, 1 = don't ask  ; IBQDT = the date/time to queue the job (optional)  ;  ; Sets ZTSK only if job is queued  ;  ; IBABORT = output parameter which says user aborted forms output.  ; Pass by reference. The $$QUEUE function returned false.  ;  N IBF2,IBTYP,POP,ZTIO,ZTRTN,ZTDESC,ZTSAVE,ZTREQ,ZTDTH,ZTREQ  S IBTYP=$P($G(^IBE(353,IBFORM,2)),U,2),IBQUE=$G(IBQUE),IBABORT=0  G:$S(IBTYP'="S":$G(^IBE(353,IBFORM,"EXT"))=""&($G(^IBE(353,+$P($G(^IBE(353,IBFORM,2)),U,5),"EXT"))=""),1:'$G(IBIFN)) FORMQ  I IBTYP="P",IBQUE="" D DEV(IBFORM) G:$G(POP) FORMQ  I IBTYP="T" D:$G(IBNOASK) Q:$G(IBNOASK) I '$$QUEUE(IBFORM) S:$O(^TMP("IBRESUBMIT",$J,0)) ^TMP("IBRESUBMIT",$J)="ABORT" S IBABORT=1 Q  . S ZTRTN="FORMOUT^IBCEFG7",ZTIO="",ZTDESC="OUTPUT FORMATTER - FORM: "\_$P($G(^IBE(353,IBFORM,0)),U),ZTSAVE("IB\*")="",ZTDTH=$S($G(IBQDT):IBQDT,1:$$NOW^XLFDT())  . S:$D(^TMP("IBRESUBMIT",$J)) ZTSAVE("^TMP(""IBRESUBMIT"",$J)")="",ZTSAVE("^TMP(""IBNOT"",$J)")="",ZTSAVE("^TMP(""IBRESUBMIT"",$J,")="",ZTSAVE("^TMP(""IBNOT"",$J,")=""  . I $D(^TMP("IBSELX",$J)) S ZTSAVE("^TMP(""IBSELX"",$J,")="",ZTSAVE("^TMP(""IBSELX"",$J)")=""  . S:'$G(DUZ) DUZ=.5  . D ^%ZTLOAD  I '$G(ZTSK) D FORMOUT FORMQ Q | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCEFG7 ;ALB/TMP - OUTPUT FORMATTER GENERIC FORM PROCESSING ;06-MAR-96  ;;2.0;INTEGRATED BILLING;\*\*52,84,96,51,137,191,320,608\*\*;21-MAR-94  ;  Q  ;  ;/IB\*2\*608 (vd) - US2486 added the parameter IBRSBTST to indicate a claim is a "TEST"  ; that is being RESUBMITTED and assists with the COB Output Formatter data.  FORM(IBFORM,IBQUE,IBNOASK,IBQDT,ZTSK,IBABORT,IBRSBTST) ;For ien IBFORM, extract data using  ; output generater  ; IBQUE = the output queue for transmitted forms or the printer queue  ; for printed output  ; IBNOASK = flag that says user interaction for queuing is not needed  ; 0 or null = ask, 1 = don't ask  ; IBQDT = the date/time to queue the job (optional)  ;  ; Sets ZTSK only if job is queued  ;  ; IBABORT = output parameter which says user aborted forms output.  ; Pass by reference. The $$QUEUE function returned false.  ;  ; IBRSBTST = 1, this indicates the claim is being resubmitted as a “TEST” claim and should be  ; used by the OUTPUT FORMATTER entries to determine what COB information is  ; going out. – IB\*2.0\*608  ;  N IBF2,IBTYP,POP,ZTIO,ZTRTN,ZTDESC,ZTSAVE,ZTREQ,ZTDTH,ZTREQ  S IBTYP=$P($G(^IBE(353,IBFORM,2)),U,2),IBQUE=$G(IBQUE),IBABORT=0  G:$S(IBTYP'="S":$G(^IBE(353,IBFORM,"EXT"))=""&($G(^IBE(353,+$P($G(^IBE(353,IBFORM,2)),U,5),"EXT"))=""),1:'$G(IBIFN)) FORMQ  I IBTYP="P",IBQUE="" D DEV(IBFORM) G:$G(POP) FORMQ  I IBTYP="T" D:$G(IBNOASK) Q:$G(IBNOASK) I '$$QUEUE(IBFORM) S:$O(^TMP("IBRESUBMIT",$J,0)) ^TMP("IBRESUBMIT",$J)="ABORT" S IBABORT=1 Q  . S ZTRTN="FORMOUT^IBCEFG7",ZTIO="",ZTDESC="OUTPUT FORMATTER - FORM: "\_$P($G(^IBE(353,IBFORM,0)),U),ZTSAVE("IB\*")="",ZTDTH=$S($G(IBQDT):IBQDT,1:$$NOW^XLFDT())  . S:$D(^TMP("IBRESUBMIT",$J)) ZTSAVE("^TMP(""IBRESUBMIT"",$J)")="",ZTSAVE("^TMP(""IBNOT"",$J)")="",ZTSAVE("^TMP(""IBRESUBMIT"",$J,")="",ZTSAVE("^TMP(""IBNOT"",$J,")=""  . I $D(^TMP("IBSELX",$J)) S ZTSAVE("^TMP(""IBSELX"",$J,")="",ZTSAVE("^TMP(""IBSELX"",$J)")=""  . S:'$G(DUZ) DUZ=.5  . D ^%ZTLOAD  I '$G(ZTSK) D FORMOUT FORMQ Q | | | | | | | | | |

* The IBCEU1 routine needs to be modified to reference the new IB variable (IBRSBTST):

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCEU1 | | | | | | | | |
| **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 |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Current Logic** | | | | | | | | | |
| IBCEU1 ;ALB/TMP - EDI UTILITIES FOR EOB PROCESSING ;10-FEB-99  ;;2.0;INTEGRATED BILLING;\*\*137,155,296,349,371,432,473,547\*\*;21-MAR-94;Build 119  ;;Per VA Directive 6402, this routine should not be modified.  ; CCOB1(IBIFN,NODE,SEQ) ; Extract Claim level COB data  ; for a bill IBIFN  ; NODE = the file 361.1 node(s) to be returned, separated by commas  ; SEQ = the specific insurance sequence you want returned. If not =  ; 1, 2, or 3, all are returned  ; Returns IBXDATA(COB,n,node) where COB = COB insurance sequence,  ; n is the entry number in file 361.1 and node is the node requested  ; = the requested node's data  ;  N IB,IBN,IBBILL,IBS,A,B,C,IBCURR,IBMRAF,Z  ;  K IBXDATA  ;  S:$G(NODE)="" NODE=1  S IB=$P($G(^DGCR(399,IBIFN,"M1")),U,5,7)  S IBCURR=$$COB^IBCEF(IBIFN)  ; ib\*2.0\*547 make sure you only set MRA flag if MRA on current sequence being checked  ;S IBMRAF=$$MCRONBIL^IBEFUNC(IBIFN)  S IBMRAF=$P($$MCRONBIL^IBEFUNC(IBIFN,$S(IBCURR="P":1,IBCURR="S":2,1:3)),U,2)  ;  S:"123"'[$G(SEQ) SEQ=""  ;  F B=1:1:3 S IBBILL=$P(IB,U,B) I IBBILL S C=0 F  S C=$O(^IBM(361.1,"B",IBBILL,C)) Q:'C  D  . I '$$EOBELIG(C,IBMRAF,IBCURR) Q      ; eob not eligible for secondary claim  . S IBS=$P($G(^IBM(361.1,C,0)),U,15) ; insurance sequence  . I $S('$G(SEQ):1,1:SEQ=IBS) D  .. F Z=1:1:$L(NODE,",") D  ... S A=$P(NODE,",",Z)  ... Q:A=""  ... S IBN=$G(^IBM(361.1,C,A))  ... ; Start IB\*2.0\*473 BI Added to null patient responsibility in OI1  ... ; if the data is contained at the line level to be sent in LCOB.  ... ; Perform the following for only OI1.19 using the dictionary 364.6 IEN.  ... S:+$G(IBX0)=2204&($$LPREXIST(C))&(A=1) $P(IBN,U,2)=""  ... ; End IB\*2.0\*473  ... I $TR(IBN,U)'="" S IBXDATA(IBS,C,A)=IBN  ;  Q  ; CCAS1(IBIFN,SEQ) ; Extract all MEDICARE COB claim level adjustment data  ; for a bill IBIFN (subfile 361.11 in file 361.1)  ; SEQ = the specific insurance sequence you want returned. If not =  ; 1, 2, or 3, all are returned  ; Returns IBXDATA(COB,n) where COB = COB insurance sequence,  ; n is the entry number in file 361.1 and  ; = the 0-node of the subfile entry (361.11)  ; and IBXDATA(COB,n,m) where m is a sequential # and  ; = this level's 0-node  N IB,IBA,IBS,IB0,IB00,IBBILL,B,C,D,E  ;  S IB=$P($G(^DGCR(399,IBIFN,"M1")),U,5,7)  S:"123"'[$G(SEQ) SEQ=""  ;  F B=1:1:3 S IBBILL=$P(IB,U,B) I IBBILL S C=0 F  S C=$O(^IBM(361.1,"B",IBBILL,C)) Q:'C  D  . I '$$EOBELIG(C) Q      ; eob not eligible for secondary claim  . S IBS=$P($G(^IBM(361.1,C,0)),U,15) ; insurance sequence  . I $S('$G(SEQ):1,1:SEQ=IBS) D  .. S (IBA,D)=0 F  S D=$O(^IBM(361.1,C,10,D)) Q:'D  S IB0=$G(^(D,0)) D  ... S IBXDATA(IBS,D)=IB0  ... S (IBA,E)=0  ... F  S E=$O(^IBM(361.1,C,10,D,1,E)) Q:'E  S IB00=$G(^(E,0)) D  .... S IBA=IBA+1  .... I $TR(IB00,U)'="" S IBXDATA(IBS,D,IBA)=IB00  ;  Q  ;  ; | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCEU1 ;ALB/TMP - EDI UTILITIES FOR EOB PROCESSING ;10-FEB-99  ;;2.0;INTEGRATED BILLING;\*\*137,155,296,349,371,432,473,547,608\*\*;21-MAR-94;Build 119  ;;Per VA Directive 6402, this routine should not be modified.  ; CCOB1(IBIFN,NODE,SEQ,IBRSBTST) ; Extract Claim level COB data  ; for a bill IBIFN  ; NODE = the file 361.1 node(s) to be returned, separated by commas  ; SEQ = the specific insurance sequence you want returned. If not =  ; 1, 2, or 3, all are returned  ; Returns IBXDATA(COB,n,node) where COB = COB insurance sequence,  ; n is the entry number in file 361.1 and node is the node requested  ; = the requested node’s data   ; IBRSBTST=1, this indicates the claim is being resubmitted as a "TEST"  ; claim and should be used be the OUTPUT FORMATTER entries  ; to determine what COB information is going out. - IB\*2\*608 (vd)   ;  N IB,IBN,IBBILL,IBS,A,B,C,IBCURR,IBMRAF,Z,CSEQ  ;  K IBXDATA  ;  S:$G(NODE)="" NODE=1  S IB=$P($G(^DGCR(399,IBIFN,"M1")),U,5,7)  S IBCURR=$$COB^IBCEF(IBIFN)  S CSEQ=$$COB^IBCEF(IBIFN)  ; ib\*2.0\*547 make sure you only set MRA flag if MRA on current sequence being checked  ;S IBMRAF=$$MCRONBIL^IBEFUNC(IBIFN)  S IBMRAF=$P($$MCRONBIL^IBEFUNC(IBIFN,$S(IBCURR="P":1,IBCURR="S":2,1:3)),U,2)  ;  S:"123"'[$G(SEQ) SEQ=""  ;  F B=1:1:3 S IBBILL=$P(IB,U,B) I IBBILL S C=0 F  S C=$O(^IBM(361.1,"B",IBBILL,C)) Q:'C  D  . I '$$EOBELIG(C,IBMRAF,IBCURR) Q      ; eob not eligible for secondary claim  . S IBS=$P($G(^IBM(361.1,C,0)),U,15) ; insurance sequence   . I +$G(IBRSBTST), ((CSEQ=IBS)!(CSEQ<IBS)) Q   ; IB\*2.0\*608/vd (US2486) added to prevent COB Data from being put on Resubmitted Claims for TEST.   . I $S('$G(SEQ):1,1:SEQ=IBS) D  .. F Z=1:1:$L(NODE,",") D  ... S A=$P(NODE,",",Z)  ... Q:A=""  ... S IBN=$G(^IBM(361.1,C,A))  ... ; Start IB\*2.0\*473 BI Added to null patient responsibility in OI1  ... ; if the data is contained at the line level to be sent in LCOB.  ... ; Pearform the following for only OI1.19 using the dictionary 364.6 IEN.  ... S:+$G(IBX0)=2204&($$LPREXIST(C))&(A=1) $P(IBN,U,2)=""  ... ; End IB\*2.0\*473  ... I $TR(IBN,U)'="" S IBXDATA(IBS,C,A)=IBN  ;  Q  ; CCAS1(IBIFN,SEQ,IBRSBTST) ; Extract all MEDICARE COB claim level adjustment data  ; for a bill IBIFN (subfile 361.11 in file 361.1)  ; SEQ = the specific insurance sequence you want returned. If not =  ; 1, 2, or 3, all are returned   ; IBRSBTST=1, this indicates the claim is being resubmitted as a "TEST"  ; claim and should be used be the OUTPUT FORMATTER entries  ; to determine what COB information is going out. - IB\*2\*608 (vd)   ; Returns IBXDATA(COB,n) where COB = COB insurance sequence,  ; n is the entry number in file 361.1 and  ; = the 0-node of the subfile entry (361.11)  ; and IBXDATA(COB,n,m) where m is a sequential # and  ; = this level's 0-node  N IB,IBA,IBS,IB0,IB00,IBBILL,B,C,D,E,CSEQ  ;  S IB=$P($G(^DGCR(399,IBIFN,"M1")),U,5,7)  S:"123"'[$G(SEQ) SEQ=""  S CSEQ=$$COB^IBCEF(IBIFN)  ;  F B=1:1:3 S IBBILL=$P(IB,U,B) I IBBILL S C=0 F  S C=$O(^IBM(361.1,"B",IBBILL,C)) Q:'C  D  . I '$$EOBELIG(C) Q      ; eob not eligible for secondary claim   . S IBS=$P($G(^IBM(361.1,C,0)),U,15) ; insurance sequence  . I +$G(IBRSBTST), ((CSEQ=IBS)!(CSEQ<IBS)) Q   ; IB\*2.0\*608/vd (US2486) added to prevent COB Data from being put on Resubmitted Claims for TEST.   . I $S('$G(SEQ):1,1:SEQ=IBS) D  .. S (IBA,D)=0 F  S D=$O(^IBM(361.1,C,10,D)) Q:'D  S IB0=$G(^(D,0)) D  ... S IBXDATA(IBS,D)=IB0  ... S (IBA,E)=0  ... F  S E=$O(^IBM(361.1,C,10,D,1,E)) Q:'E  S IB00=$G(^(E,0)) D  .... S IBA=IBA+1  .... I $TR(IB00,U)'="" S IBXDATA(IBS,D,IBA)=IB00  ;  Q   ;   ;  ;/IB\*2\*608 (vd) (US2486) - Added this module of code to be referenced by the Output Formatter. CKCOBTST(IBXIEN,IBXSAVE,Z0,Z,IBRSBTST) ; Check Primary, Secondary & Tertiary COBS for Claims Resubmitted as Test.  ; INPUT: IBXIEN - Current Claim number  ; IBXSAVE - Array containing current claim COB data.  ; Z0 - Will equal "INPT", "OUTPT" or "RX"  ; Z - Is the LINE  N A,CURSEQ,XX  I '+$G(IBRSBTST) M IBXSAVE("LCOB",Z)=IBXSAVE(Z0,Z) Q  ; Only concerned with Claims that are Resubmitted as Test.  S A="",CURSEQ=$$COBN^IBCEF(IBXIEN)  ; With the line below, ideally, we want to merge all of IBXSAVE(Z0,Z) into IBXSAVE("LCOB",Z),  ; but the COB node should be handled separately for the current sequence.  S IBXSAVE("LCOB",Z)=IBXSAVE(Z0,Z)   S XX="" F  S XX=$O(IBXSAVE(Z0,Z,XX)) Q:XX=""  I XX'="COB" M IBXSAVE("LCOB",Z,XX)=IBXSAVE(Z0,Z,XX)  ; Now handle the COB node for the current sequence.  F  S A=$O(IBXSAVE(Z0,Z,"COB",A)) Q:A=""  D   ; Only want to merge those COBS that are previous to the current  . I (CURSEQ=A)!(CURSEQ<A) Q   ; Only want to merge those COBS that are previous to the current sequence.  . M IBXSAVE("LCOB",Z,"COB",A)=IBXSAVE(Z0,Z,"COB",A)  Q  ; | | | | | | | | | |

* Modify the COB1 segment, piece 1.9 (OTHER INS DATA) so that it reflects patch IB\*2.0\*608. To accomplish this, the FORMAT CODE should be changed in the IB FORM FIELD CONTENT file (#364.5), record #225 (N-COB CLAIM LEVEL AMOUNTS). [NOTE: This record is also referenced by the OI1A segment, piece 6, below in this document.]

FROM:

PIECE 1.9 OTHER INS DATA 364.6[886] 364.7[808] 364.5[225]

\*\*\*\*\* CALCULATE ONLY, NO OUTPUT \*\*\*\*\*

>K IBXSAVE("CCOB") D CCOB1^IBCEU1(IBXIEN,"0,1,2,4")

>M IBXSAVE("CCOB")=IBXDATA K IBXDATA

TO:

PIECE 1.9 OTHER INS DATA 364.6[886] 364.7[808] 364.5[225]

\*\*\*\*\* CALCULATE ONLY, NO OUTPUT \*\*\*\*\*

>K IBXSAVE("CCOB") D CCOB1^IBCEU1(IBXIEN,"0,1,2,4",””,$G(IBRSBTST))

>M IBXSAVE("CCOB")=IBXDATA K IBXDATA

* Modify the MIA1 segment, piece 1.9 (OTHER INS COB MIA(1) DATA) so that it reflects patch IB\*2.0\*608. To accomplish this, the FORMAT CODE should be changed in the IB FORM FIELD CONTENT file (#364.5), record #226 (N-MEDICARE INPT CLAIM COB AMTS)

FROM:

PIECE 1.9 OTHER INS COB MIA(1) DATA 364.6[888] 364.7[819] 364.5[226]

\*\*\*\*\* CALCULATE ONLY, NO OUTPUT \*\*\*\*\*

>D CCOB1^IBCEU1(IBXIEN,"4,5")

>K IBXSAVE("MIA45") M IBXSAVE("MIA45")=IBXDATA K IBXDATA

TO:

PIECE 1.9 OTHER INS COB MIA(1) DATA 364.6[888] 364.7[819] 364.5[226]

\*\*\*\*\* CALCULATE ONLY, NO OUTPUT \*\*\*\*\*

>D CCOB1^IBCEU1(IBXIEN,"4,5",””,$G(IBRSBTST))

>K IBXSAVE("MIA45") M IBXSAVE("MIA45")=IBXDATA K IBXDATA

* Modify the MOA1 segment, piece 1.9 (OTHER INS COB MOA DATA) so that it reflects patch IB\*2.0\*608. To accomplish this, the FORMAT CODE should be changed in the IB FORM FIELD CONTENT file (#364.5), record #227 (N-MEDICARE OUTPT CLAIM COB AMT)

FROM:

PIECE 1.9 OTHER INS COB MOA DATA 364.6[887] 364.7[818] 364.5[227]

\*\*\*\*\* CALCULATE ONLY, NO OUTPUT \*\*\*\*\*

>D CCOB1^IBCEU1(IBXIEN,3)

>K IBXSAVE("MOA") M IBXSAVE("MOA")=IBXDATA K IBXDATA

TO:

PIECE 1.9 OTHER INS COB MOA DATA 364.6[887] 364.7[818] 364.5[227]

\*\*\*\*\* CALCULATE ONLY, NO OUTPUT \*\*\*\*\*

>D CCOB1^IBCEU1(IBXIEN,3,””,$G(IBRSBTST))

>K IBXSAVE("MOA") M IBXSAVE("MOA")=IBXDATA K IBXDATA

* Modify the CCAS segment, piece 1.9 (OTHER INS COB ADJ DATA) so that it reflects patch IB\*2.0\*608. To accomplish this, the FORMAT CODE should be changed in the IB FORM FIELD CONTENT file (#364.5), record #228 (N-COB CLAIM LEVEL ADJUSTMENTS)

FROM:

PIECE 1.9 OTHER INS COB ADJ DATA 364.6[890] 364.7[821] 364.5[228]

\*\*\*\*\* CALCULATE ONLY, NO OUTPUT \*\*\*\*\*

>D CCAS1^IBCEU1(IBXIEN)

>K IBXSAVE("CCAS") M IBXSAVE("CCAS")=IBXDATA K IBXDATA

TO:

PIECE 1.9 OTHER INS COB ADJ DATA 364.6[890] 364.7[821] 364.5[228]

\*\*\*\*\* CALCULATE ONLY, NO OUTPUT \*\*\*\*\*

>D CCAS1^IBCEU1(IBXIEN,””,$G(IBRSBTST))

>K IBXSAVE("CCAS") M IBXSAVE("CCAS")=IBXDATA K IBXDATA

* Modify the LCOB segment, piece 1.9 (EXTRACT LINE LEVEL ADJUSTMENTS) to determine what OUTPUT FORMATTER COB information is going out for “TEST” claims only. To accomplish this, the FORMAT CODE should be changed in the IB FORM FIELD CONTENT file (#364.7), record #985 (N-GET FROM PREVIOUS EXTRACT)

FROM:

PIECE 1.9 EXTRACT LINE LEVEL ADJUSTMENTS 364.6[983] 364.7[985] 364.5[5]

\*\*\*\*\* CALCULATE ONLY, NO OUTPUT \*\*\*\*\*

>Constant Value: ""

>N Z,Z0 K IBXDATA I $$COBN^IBCEF(IBXIEN)>1 F Z0="INPT","OUTPT","RX" S Z=0 F S Z=$O(IBXSAVE(Z0,Z)) K:'Z IBXSAVE(Z0) Q:'Z I $D(IBXSAVE(Z0,Z,"COB")) M IBXSAVE("LCOB",Z)=IBXSAVE(Z0,Z)

TO:

PIECE 1.9 EXTRACT LINE LEVEL ADJUSTMENTS 364.6[983] 364.7[985] 364.5[5]

\*\*\*\*\* CALCULATE ONLY, NO OUTPUT \*\*\*\*\*

>Constant Value: ""

>N Z,Z0 K IBXDATA I $$COBN^IBCEF(IBXIEN)>1 F Z0="INPT","OUTPT","RX" S Z=0 F S Z=$O(IBXSAVE(Z0,Z)) K:'Z IBXSAVE(Z0) Q:'Z D CKCOBTST^IBCEU1(IBXIEN,IBXSAVE,Z0,Z,+$G(IBRSBTST))

* The IBY608PR Pre-Install routine needs to be coded to include the Output Formatter modifications referenced in this design document:

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBY608PR | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** |  | | | | | | | | |
| **Related Options** | None | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  |  | | | | |  | | | |
| **Data Dictionary (DD) References** | IB DATA ELEMENT DEFINITION File [#364.7] | | | | | | | | |
| **Related Protocols** | None | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Current Logic** | | | | | | | | | |
| N/A | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBY608PR ;ALB/VD - Pre-Installation for IB patch 068 ; 4/06/17 4:33pm  ;;2.0;INTEGRATED BILLING;\*\*608\*\*;21-MAR-94;Build 119  ;;Per VA Directive 6402, this routine should not be modified.  ;  ; delete all output formatter (O.F.) data elements included in build  D DELOF  Q   ; INCLUDE(FILE,Y) ; function to determine if O.F. entry should be included in the build  ; FILE=5,6,7 indicating file 364.x  ; Y=ien to file  NEW OK,LN,TAG,DATA  S OK=0  F LN=2:1 S TAG="ENT"\_FILE\_"+"\_LN,DATA=$P($T(@TAG),";;",2) Q:DATA=""  I $F(DATA,U\_Y\_U) S OK=1 Q  Q OK  ;  ;Delete edited entries to insure clean install of new entries  ;Delete obsolete entries. DELOF   ; Delete included OF entries  NEW FILE,DIK,LN,TAG,TAGLN,DATA,PCE,DA,Y  F FILE=5,6,7 S DIK="^IBA(364."\_FILE\_"," D  . F TAG="ENT"\_FILE,"DEL"\_FILE D  .. F LN=2:1 S TAGLN=TAG\_"+"\_LN,DATA=$P($T(@TAGLN),";;",2) Q:DATA=""  D  ... F PCE=2:1 S DA=$P(DATA,U,PCE) Q:'DA  I $D(^IBA("364."\_FILE,DA,0)) D ^DIK  Q  ;  ; Example for ENT5, ENT6, ENT7, DEL5, DEL6, and DEL7:  ;;^^  ; Note: Must have beginning and ending up-carat  ;  ;-----------------------------------------------------------------------  ; 364.5 O.F. entries added:  ;  ; 225 N-COB CLAIM LEVEL AMOUNTS 'COB1-1.9' (US2486)  ; 226 N-MEDICARE INPT CLAIM COB AMTS 'MIA1-1.9' (US2486)  ; 227 N-MEDICARE OUTPT CLAIM COB AMT 'MOA1-1.9' (US2486)  ; 228 N-COB CLAIM LEVEL ADJUSTMENTS 'CCAS-1.9' (US2486)  ; ENT5 ; OF entries in file 364.5 to be included  ;  ;N^225^226^227^228^  ;  ;-----------------------------------------------------------------------  ; 364.6 entries modified:  ; ENT6 ; O.F. entries in file 364.6 to be included  ;  ;;  ;  ;-----------------------------------------------------------------------  ; 364.7 entries modified:  ;  ; 985 – LCOB-1.9 (N-GET FROM PREVIOUS EXTRACT) – US2486   ; ENT7 ; O.F. entries in file 364.7 to be included  ;  ;;^985^  ;  ;-----------------------------------------------------------------------  ; 364.5 entries deleted:  ; DEL5    ; remove O.F. entries in file 364.5 (not re-added)  ;  ;;  ;  ;-----------------------------------------------------------------------  ; 364.6 entries deleted:  ;  ; DEL6    ; remove O.F. entries in file 364.6 (not re-added)  ;  ;;^^  ;  ;-----------------------------------------------------------------------  ; 364.7 entries deleted:  ;   ; DEL7    ; remove O.F. entries in file 364.7 (not re-added)  ;  ;;^^   ; | | | | | | | | | |