**TAS eBill SDD US11**

System Design Document

IB\*2.0\*577



Department of Veterans Affairs

**May 2017**

Version 1.02

**User Story Number:** USEB-15

**User Story Name:** Unit or Basis for Measurements Codes for Drugs (Backlog #142)

**Rally ID:** US-11

**Resolution:**

To satisfy the enhancement request described by the aforementioned User Story, the following needs to occur:

1. CREATE the new UNITS/BASIS OF MEASUREMENT field [#399.0304, 52]. This field is required if there is an NDC number.
2. MODIFY a line of code in the routine ^IBCU7 to prompt for the NDC field [#399.0304, 53], then prompt for the new UNITS/BASIS OF MEASUREMENT field [#399.0304, 52] and then prompt for the UNITS field [#399.0304, 54].
3. ADD two new triggers to the NDC field [#399.0304, 53] that require the UNITS/BASIS OF MEASUREMENT field [#399.0304, 52] and the UNITS field [#399.0304, 54] to be populated if there is an NDC number present.
4. ADD a new validation code to check claim lines and confirm that if the NDC field is populated, then the UNITS and UNITS/BASIS OF MEASUREMENT fields are populated as well.
5. MODIFY [#364.7, 941] to properly include the UNITS/BASIS OF MEASUREMENT in the “PRF” (Professional) Segment of the 837, piece 25.
6. MODIFY [#364.7, 1950] to properly include the UNITS/BASIS OF MEASUREMENT in the “INS” (Institutional) Segment of the 837, piece 17.
7. ADD a new error code to the Error Code file [#350.8] for the line level check on NDC number and the UNITS/BASIS OF MEASUREMENT and the UNITS fields.
8. ADD the #364.7 file to the build for all of the updated OUTPUT FORMATTER entries.
9. MODIFY the ^IBCEF11 and ^IBCF23A routines to update the 837 and printed CMS 1500 for the new UNITS/BASIS OF MEASUREMENT field.
10. MODIFY the ^IBCEF22 routines to update the printed UB-04 for the new UNITS/BASIS OF MEASUREMENT field.
11. MODIFY the acceptable format of the UNITS field (#399.0304, 54) so that it’s format can be “99999999999.999”.
12. The FORMAT CODE of INS-16 and PRF-23 for the 837 Transmission (in the IB FORM FIELD CONTENT file [#364.7], entries 939 and 1949) needs to be changed remove the decimal point “.” from the output formatter.
13. CREATE the New ^IBY577PR Pre-Install Routine

**Data Design:**

1. CREATE the new UNITS/BASIS OF MEASUREMENT field [#399.0304, 52]. This field is required if there is an NDC number.

399.0304,52 UNITS/BASIS OF MEASUREMENT 2;1 SET

Units/Basis of Measurement

'F2' FOR International Unit;

'GR' FOR Gram;

'ME' FOR Milligram;

'ML' FOR Milliliter;

'UN' FOR Unit;

LAST EDITED: MAY 3, 2017

HELP-PROMPT: Enter the units or basis for measurement

associated with the Medication.

DESCRIPTION: This field is used to associate the correct

unit or measurement with Medication is being

specified.

1. MODIFY a line of code in the routine ^IBCU7 to prompt for the NDC field [#399.0304, 53], then prompt for the new UNITS/BASIS OF MEASUREMENT field [#399.0304, 52] and then prompt for the UNITS field [#399.0304, 54].

The display should be as below:

NDC NUMBER:

UNITS/BASIS OF MEASUREMENT:

QUANTITY:

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCU7 | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** |  | | | | | | | | |
| **Related Options** | None | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  |  | | | | |  | | | |
| **Data Dictionary (DD) References** | None | | | | | | | | |
| **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** | | | | | | | | | |
| IBCU7 ;ALB/AAS - INTERCEPT SCREEN INPUT OF PROCEDURE CODES ;29-OCT-91  ;;2.0;INTEGRATED BILLING;\*\*62,52,106,125,51,137,210,245,228,260,348,371,432,447,488,461,516,522\*\*;21-MAR-94;Build 11  ;;Per VA Directive 6402, this routine should not be modified.  ;  ;MAP TO DGCRU7  ; CHKX ; -interception of input x from Additional Procedure input  G:X=" " CHKXQ  I $$INPAT^IBCEF(DA(1)),'$P($G(^IBE(350.9,1,1)),"^",15),X'?1A1.2N D  G CHKXQ  . K X  . D EN^DDIOL("Site param does not allow entry of non-PTF procedures") ;Fileman error here will be: The previous error occurred when performing an action specified in a Pre-lookup transform (7.5 node).  G:'$D(^UTILITY($J,"IB")) CHKXQ  ;S M=($A($E(X,1))-64),S=+$E(X,2) Q:'$G(^UTILITY($J,"IB",M,S)) S X="`"\_+^(S)  S M=0 I X?1A1.2N S N=$G(^UTILITY($J,"IB","B",X)) S M=+N,S=+$P(N,U,2),P=X S S=$G(^UTILITY($J,"IB",M,S)) I +S S X="`"\_+S I $P(N,U,3)="N" S X=""""\_X\_"""" S $P(^UTILITY($J,"IB","B",P),U,3)="Y"  I +M,$D(DGPROCDT),DGPROCDT'=$P($G(^UTILITY($J,"IB",M,1)),"^",2) S DGPROCDT=$P(^(1),"^",2) W !!,"Procedure Date: " S Y=DGPROCDT X ^DD("DD") W Y,! CHKXQ Q  ; CODMUL ;Date oriented entry of procedure DELASK I $D(IBZ20),IBZ20,IBZ20'=$P(^DGCR(399,IBIFN,0),U,9) S %=2 W !,"SINCE THE PROCEDURE CODING METHOD HAS BEEN CHANGED, DO YOU WANT TO DELETE ALL",!,"PROCEDURE CODES IN THIS BILL"  I  D YN^DICN Q:%=-1 D:%=1 DELADD I %Y?1."?" W !!,"If you answer 'Yes', all procedure codes will be DELETED from this bill.",! G DELASK  K %,%Y,DA,IBZ20,DIK ;W !,"Procedure Entry:"  ; CODDT I $D(IBIFN),$D(^DGCR(399,IBIFN,0)),$P(^(0),U,9) S DIC("V")=$S($P(^(0),U,9)=9:"I +Y(0)=80.1",$P(^(0),U,9)=4!($P(^(0),U,9)=5):"I +Y(0)=81",1:"")  I $P($G(^DGCR(399,IBIFN,0)),"^",5)<3 S IBZTYPE=1 I $P($G(^UTILITY($J,"IB",1,1)),"^",2) S DGPROCDT=$P(^(1),"^",2) D ASKCOD  S X=$$PRCDIV^IBCU71(IBIFN) I +X W !!,$P(X,U,2),!  N Z,Z0 S Z=$G(^DGCR(399,IBIFN,"U")),Z0=$$FMTE^XLFDT($P(Z,U),"2D")\_"-"\_$$FMTE^XLFDT($P(Z,U,2),"2D")  W !,"Select PROCEDURE DATE"\_$S($TR(Z0,"-")'="":" ("\_Z0\_")",1:"")\_": " R X:DTIME G:'$T!("^"[X) CODQ D:X["?" CODHLP  S IBEX=0 D  ; Get procedure date  . I X=" ",$D(DGPROCDT),DGPROCDT?7N S Y=DGPROCDT D D^DIQ W " (",Y,")" Q  . I X=" ",+$P($G(^DGCR(399,IBIFN,"OP",0)),"^",4) S (DGPROCDT,Y)=$O(^DGCR(399,IBIFN,"OP",0)) D D^DIQ W " (",Y,")" Q  . S %DT="EXP",%DT(0)=-DT D ^%DT K %DT I Y<1 S IBEX=1 Q  . I '$$OPV2^IBCU41(Y,IBIFN,1) S IBEX=1 Q  . S:'$G(IBZTYPE) X=$$OPV^IBCU41(Y,IBIFN) S DGPROCDT=Y  I 'IBEX D ASKCOD,ADDCPT^IBCU71:$D(DGCPT)  K IBEX  G CODDT  ; ASKCOD N Z,Z0,DA,IBACT,IBQUIT,IBLNPRV  ;WCJ;2.0\*432  N IBPOPOUT  S IBPOPOUT=0 ; IB\*2.0\*447 BI  K DGCPT  S DGCPT=0,DGCPTUP=$P($G(^IBE(350.9,1,1)),"^",19),DGADDVST=0,IBFT=$P($G(^DGCR(399,IBIFN,0)),"^",19)  I '$D(^DGCR(399,IBIFN,"CP",0)) S ^DGCR(399,IBIFN,"CP",0)=U\_$$GETSPEC^IBEFUNC(399,304)  ;  F  S IBQUIT=0 D  Q:IBQUIT  . S IBPOPOUT=0  . D DICV ; restrict code type to PCM  . S DIC("A")=" Select PROCEDURE: "  . S DIC="^DGCR(399,"\_IBIFN\_",""CP"","  . S DIC(0)="AEQMNL"  . S DIC("S")="I '$D(DIV(""S""))&($P(^(0),U,2)=DGPROCDT)"  . S DIC("DR")="1///^S X=DGPROCDT"  . S DA(1)=IBIFN,DLAYGO=399  . W ! D ^DIC I Y<1 S IBQUIT=1 Q  . S IBPROCP=+Y  . ; If we just added inactive code - it must be deleted.  . S IBACT=0 ; Active flag  . I Y["ICD0" S IBACT=$$ICD0ACT^IBACSV(+$P(Y,U,2),$$BDATE^IBACSV(IBIFN))  . I Y["ICPT" S IBACT=$$CPTACT^IBACSV(+$P(Y,U,2),DGPROCDT)  . S DGCPTNEW=$P(Y,"^",3) ;Was the procedure just added?  . I DGCPTNEW,'IBACT D DELPROC Q  . I 'IBACT W !,\*7,"Warning: Procedure code is inactive on this date",!  . I DGCPTNEW,$D(^UTILITY($J,"IB")),$$INPAT^IBCEF(IBIFN),Y["ICPT(" D DATA^IBCU74(Y,.IBLNPRV)  . S DGADDVST=$S(DGCPTNEW:1,$D(DGADDVST):DGADDVST,1:0)  . N IBPRV,IBPRVO,IBPRVN  . ;  . ; Line level provider function by form type.  . ; CMS-1500 (FORM TYPE=2)  . ; RENDERING PROVIDER, REFERRING PROVIDER,  . ; and SUPERVISING PROVIDER.  . ; UB-04 (FORM TYPE=3)  . ; RENDERING PROVIDER, REFERRING PROVIDER,  . ; OPERATING PROVIDER, and OTHER OPERATING  . ; PROVIDER.  . ;  . ; Removed: Call to $$MAINPRV^IBCEU(IBIFN) is for claim  . ; level provider defaults.  . ; 1. For new line level providers we don't need  . ; or want default claim level provider  . ; (requirement).  . ; 2. We don't want to default claim level to  . ; line level provider (requirement).  . ;  . K DIC("V") ; DEM;432 - KILL DIC("V") because this was for previous variable pointer use.  . ;  . N IBPROCSV  ; DEM;432 - Variable IBPROCSV is variable to preserve value of 'Y', which is procedure code info returned by call to ^DIC.  . S IBPROCSV=Y  ; DEM;432 - Preserve value of Y for after calls to FileMan (Y = procedure code info returned by call to ^DIC).  . K DR   ;WCJ;IB\*2.0\*432  . ;  . I IBPROCSV["ICD0" S DR=".01",DIE=DIC,(IBPROCP,DA)=+Y D ^DIE Q:'$D(DA)!($D(Y)) K DR ; IB\*2.0\*461  . I IBPROCSV["ICPT" S DR=".01;16",DIE=DIC,(IBPROCP,DA)=+Y D ^DIE Q:'$D(DA)!($D(Y)) K DR ; IB\*2.0\*447 BI  . ;  . S DR=""  . ;  . ; MRD;IB\*2.0\*516 - Added line level PROCEDURE DESCRIPTION field,  . ; asked only if the procedure is an "NOC".  . I IBPROCSV["ICPT",$$NOCPROC(IBPROCSV) D  . . S DA=$P(IBPROCSV,"^") ; The line# on the bill/claim.  . . S DR=51 ; Field# for PROCEDURE DESCRIPTION  . . D ^DIE  . . Q  . ;  . D EN^IBCU7B ; DEM;432 - Call to line level provider user input.  . S Y=IBPROCSV  ; DEM;432 - Restore value of Y after calls to FileMan  . K IBPROCSV  . K DR   ;WCJ;IB\*2.0\*432  . I IBPOPOUT Q   ; IB\*2.0\*447 BI  . S DR="" I Y["ICPT" S DR="6;5//"\_$$DEFDIV(IBIFN)\_";"  . S DR=DR\_$S(IBFT=2:"8;9;17//NO;",1:"")\_3,DIE=DIC,(IBPROCP,DA)=+Y D ^DIE Q:'$D(DA)!($E($G(Y))=U)  . K DR   ;WCJ;IB\*2.0\*432  . ;  . ; MRD;IB\*2.0\*516 - Allow user to add an NDC and Units. Ask only if  . ; coding system is not ICD and this is not a prescription claim. If  . ; an NDC is entered, prompt for Units.  . I $P($G(^DGCR(399,IBIFN,0)),U,9)'=9,'$$RXLINK^IBCSC5C(IBIFN,IBPROCP) D  . . K DA  . . S DA=IBPROCP,DA(1)=IBIFN,DIE="^DGCR(399,"\_IBIFN\_",""CP"","  . . S DR="53NDC NUMBER;I X="""" S Y="""";54//1"  . . D ^DIE  . . Q  . ;  . I IBFT=3 D:'$$INPAT^IBCEF(IBIFN) ATTACH  ; DEM;432 - Prompt for Attachment Control Number.  . ; DEM;432 - Add Additional OB Minutes to DR string for call to DIE.  . S DR=$$SPCUNIT(IBIFN,IBPROCP) S:DR["15;" DR=DR\_"74Additional OB Minutes" D ^DIE ; miles/minutes/hours  . ;  . I IBFT=2 D  .. D DX^IBCU72(IBIFN,IBPROCP)  .. S X=$$ADDTNL(IBIFN,.DA)  . Q:$$INPAT^IBCEF(IBIFN) ;only outpatient bills  . ;add procedures to array for download to PCE: dgcpt(assoc clinic,cpt,'provider^first dx^modifiers',cnt)=""  . S DGPROC=$G(^DGCR(399,IBIFN,"CP",+DA,0))  . S X=$P(DGPROC,U,18)\_U\_+$G(^IBA(362.3,+$P(DGPROC,U,11),0))\_U\_$P(DGPROC,U,15)  . I 'DGCPTNEW,$P(DGPROC,"^",7)="" S DGCPTNEW=2  . I DGCPTUP,DGCPTNEW S DGCPT=DGCPT+1 I $P(DGPROC,"^",7) S DGCPT($P(DGPROC,"^",7),+DGPROC,X,DGCPT)=""  . ; add visit date to bill  . I DGADDVST S (X,DINUM)=DGPROCDT D VFILE1^IBCOPV1 K DINUM,X,DGNOADD,DGADDVST  ; Delete modifiers with only a sequence #, no code  S Z=0 F  S Z=$O(^DGCR(399,IBIFN,"CP",Z)) Q:'Z  S Z0=0 F  S Z0=$O(^DGCR(399,IBIFN,"CP",Z,"MOD",Z0)) Q:'Z0  I $P($G(^(Z0,0)),U,2)="" S DA(2)=IBIFN,DA(1)=Z,DA=Z0,DIK="^DGCR(399,"\_DA(2)\_",""CP"","\_DA(1)\_",""MOD""," D ^DIK  Q CODQ K %DT,DGPROC,DIC,DIE,DR,DGPROCDT,IBPROCP,DLAYGO  K IBFT,DGNOADD,DGADDVST,DGCPT,DGCPTUP,IBZTYPE,DGCPTNEW  Q  ; DELPROC ; Remove the selected procedure, because of inactive status (cancel selection)  W !!,\*7,"The Procedure code is inactive on ",$$DAT1^IBOUTL(DGPROCDT),"."  W !,"Please select another Procedure."  S DA(1)=IBIFN,DA=+Y,DIK="^DGCR(399,"\_IBIFN\_",""CP"","  D ^DIK  Q  ; DELADD N Z,Z0,DA,DIK,X,Y  S DA(1)=IBIFN  ;Delete references to proc on rev codes  S Z=0 F  S Z=$O(^DGCR(399,IBIFN,"RC",Z)) Q:'Z  S Z0=$G(^(Z,0)) I Z0'="",$P(Z0,U,15)!$S($P(Z0,U,10)=3:$P(Z0,U,11),1:0) S DIE="^DGCR(399,"\_DA(1)\_",""RC"",",DA=Z,DR=".11///@;.15///@"\_$S($P(Z0,U,8):"",1:";.08////1") D ^DIE  S DIK="^DGCR(399,"\_DA(1)\_",""CP""," F DA=0:0 S DA=$O(^DGCR(399,DA(1),"CP",DA)) Q:'DA  D ^DIK  S DGRVRCAL=1  Q  ; DTMES ;Message if procedure date not in date range  Q:'$D(IBIFN) Q:'$D(^DGCR(399,IBIFN,"U")) S DGNODUU=^("U")  G:X'<$P(DGNODUU,"^")&(X'>$P(DGNODUU,"^",2)) DTMESQ  W \*7,!!?3,"Date must be within STATEMENT COVERS FROM and STATEMENT COVERS TO period."  S Y=$P(DGNODUU,"^") X ^DD("DD")  W !?3,"Enter a date between ",Y," and " S Y=$P(DGNODUU,"^",2) X ^DD("DD") W Y,!  K X,Y DTMESQ K DGNODUU Q  ; CODHLP ;Display Additional Procedure codes  N I,J,Y,IBMOD  I '$O(^DGCR(399,IBIFN,"CP",0)) W !!?5,"No Codes Entered!",! Q  W ! F I=0:0 S I=$O(^DGCR(399,IBIFN,"CP",I)) Q:'I  S Y=$G(^(I,0)) S Z=$$PRCNM^IBCSCH1($P(Y,"^",1),$P(Y,"^",2)) W !?5,$E($P(Z,"^",2),1,33),?40,"- ",$P(Z,"^") D  . N IBY  . S IBY=$P(Y,U,2)  . S IBMOD=$$GETMOD^IBEFUNC(IBIFN,I,1)  . I IBMOD'="" S IBMOD="/"\_IBMOD W IBMOD  . W ?60,"Date: " S Y=IBY D DT^DIQ  W !  ;  K Z Q  ; DICV I $D(IBIFN),$D(^DGCR(399,IBIFN,0)),$P(^(0),U,9) S DIC("V")=$S($P(^(0),U,9)=9:"I +Y(0)=80.1",$P(^(0),U,9)=4!($P(^(0),U,9)=5):"I +Y(0)=81",1:"")  Q  ; DEFDIV(IBIFN) ; Find default division for bill IBIFN  Q $P($G(^DG(40.8,+$P($G(^DGCR(399,IBIFN,0)),U,22),0)),U)  ; ADDTNL(IBIFN,DA) ;  N DR,IBOK,X,Y,DIR  S IBOK=1  S DR="19T;50.09T;50.08T" D ^DIE ; WCJ;IB\*2.0\*488 Added Ts  ;I '($$FT^IBCEF(IBIFN)'=3&($$INPAT^IBCEF(IBIFN))) D ATTACH ; DEM;432 - Prompt for Attachment Control Number.  I '($$FT^IBCEF(IBIFN)=3&($$INPAT^IBCEF(IBIFN))) D ATTACH  ; DEM;432 - Prompt for Attachment Control Number.  I $D(Y) S IBOK=0 G ADDTNLQ  ;/Beginning of IB\*2.0\*488 (vd)  ;S DIR("B")="NO",DIR("A")="EDIT CMS-1500 SPECIAL PROGRAM FIELDS and BOX 19?: ",DIR("A",1)=" ",DIR(0)="YA"  ;S DIR("?",1)="Respond YES only if you need to add/edit data for chiropractic visits,"  ;S DIR("?")="EPSDT care, or if billing for HOSPICE and attending is not a hospice employee."  ;D ^DIR K DIR  ;I Y'=1 S IBOK=0 G ADDTNLQ  ;S DR="W !,"" <<EPSDT>>"";50.07;W !!,"" <<HOSPICE>>"";50.03"  S DR="50.07T;50.03T"   ;WCJ;IB\*2.0\*488 added Ts  ;/End of IB\*2.0\*488 (vd)  D ^DIE  W ! ADDTNLQ Q IBOK  ; XTRA1(Y) ;  K Y  Q  ; SPCUNIT(IBIFN,DA) ; return fields for special units if applicable, in DR form  N IB0,IBCPT,IBDR,IBCT,IBFT,DFN S IBDR=""  S IB0=$G(^DGCR(399,+$G(IBIFN),0)),IBCT=$P(IB0,U,27),IBFT=$P(IB0,U,19),DFN=$P(IB0,U,2)  S IBCPT=$G(^DGCR(399,+$G(IBIFN),"CP",+$G(DA),0)) I IBCPT'["ICPT" G SPCUNTQ  I +$$ITMUNIT^IBCRU4(+IBCPT,5,IBCT) S IBDR="15;" D SROMIN^IBCU74(IBIFN,DA) G SPCUNTQ ; minutes  I +$$ITMUNIT^IBCRU4(+IBCPT,4,IBCT) S IBDR="21;" G SPCUNTQ ; miles  I +$$ITMUNIT^IBCRU4(+IBCPT,6,IBCT) S IBDR="22//"\_$$OBSHOUR^IBCU74(DFN,$P(IBCPT,U,2))\_";" G SPCUNTQ ; hours  I +IBFT=2,$P($G(^IBE(353.2,+$P(IBCPT,U,10),0)),U,2)="ANESTHESIA" S IBDR="15;" ; minutes SPCUNTQ Q IBDR  ; ATTACH ; DEM;432 - Attachment control number.  ; Ask if user wants to enter Attachment Control Number.  N DIR,X,Y,DA,DIE,DR  S DIR("A")="Enter Attachment Control Number"  S DIR(0)="Y",DIR("B")="NO"  D ^DIR  Q:'Y  ; User chose to enter Attachment Control Number.  ; User enters Attachment Control fields.  S DA(1)=IBIFN,DA=IBPROCP  S DIE="^DGCR(399,"\_DA(1)\_",""CP"","  S DR="71Report Type;72Report Transmission Method;70Attachment Control Number"  D ^DIE  Q  ; NOCPROC(IBPROCSV) ; MRD;IB\*2.0\*516 - Function to determine if procedure is an  ; "NOC". Returns '1' if "NOC" procedure, otherwise '0'.  ;  N IBNOC,IBPROCEX,IBPROCIN,IBPROCNM,IBX  S IBNOC=0  I $G(IBPROCSV)="" G NOCPROCQ  S IBPROCIN=$P($P(IBPROCSV,U,2),";")  I IBPROCIN="" G NOCPROCQ  ;  ; If procedure code ends in '99', quit with a '1'.  ;  S IBPROCEX=$P($G(^ICPT(IBPROCIN,0)),U,1)  I $E(IBPROCEX,$L(IBPROCEX)-1,$L(IBPROCEX))=99 S IBNOC=1 G NOCPROCQ  ;  ; Pull procedure name, then check to see if it contains one of the  ; specified strings.  ;  S IBPROCNM=$P($G(^ICPT(IBPROCIN,0)),U,2)  I IBPROCNM'="",$$NOC(IBPROCNM) S IBNOC=1 G NOCPROCQ  ;  S IBX=0  F  S IBX=$O(^ICPT(IBPROCIN,"D",IBX)) Q:'IBX  D  I IBNOC=1 Q  . S IBTEXT=$G(^ICPT(IBPROCIN,"D",IBX,0))  . I $G(^ICPT(IBPROCIN,"D",IBX+1,0))'="" S IBTEXT=IBTEXT\_" "\_$G(^ICPT(IBPROCIN,"D",IBX+1,0))  . S IBNOC=$$NOC(IBTEXT)  . Q  ; NOCPROCQ ; Quit out.  Q IBNOC  ; NOC(IBTEXT) ; Quit with '1' if IBTEXT contains one of the specified strings.  ;  S IBTEXT=$TR(IBTEXT,"abcdefghijklmnopqrstuvwxyz","ABCDEFGHIJKLMNOPQRSTUVWXYZ")  ;  I IBTEXT["NOT OTHERWISE" Q 1  I IBTEXT["NOT ELSEWHERE" Q 1  I IBTEXT["NOT LISTED" Q 1  I IBTEXT["UNLISTED" Q 1  I IBTEXT["UNSPECIFIED" Q 1  I IBTEXT["UNCLASSIFIED" Q 1  I IBTEXT["NON-SPECIFIED" Q 1  I IBTEXT["NOS " Q 1  I IBTEXT["NOS;" Q 1  I IBTEXT["NOS." Q 1  I IBTEXT["NOS," Q 1  I IBTEXT["NOS/" Q 1  I IBTEXT["(NOS)" Q 1  I IBTEXT["NOC " Q 1  I IBTEXT["NOC;" Q 1  I IBTEXT["NOC." Q 1  I IBTEXT["NOC," Q 1  I IBTEXT["NOC/" Q 1  I IBTEXT["(NOC)" Q 1  ;  ; Check if last three charcters are 'NOC' or 'NOS'.  ;  S IBTEXT=$E(IBTEXT,$L(IBTEXT)-2,$L(IBTEXT))  I IBTEXT="NOC" Q 1  I IBTEXT="NOS" Q 1  ;  Q 0  ; | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCU7 ;ALB/AAS - INTERCEPT SCREEN INPUT OF PROCEDURE CODES ;29-OCT-91  ;;2.0;INTEGRATED BILLING;\*\*62,52,106,125,51,137,210,245,228,260,348,371,432,447,488,461,516,522,577\*\*;21-MAR-94;Build 16  ;;Per VA Directive 6402, this routine should not be modified.  ;  ;MAP TO DGCRU7  ; CHKX ; -interception of input x from Additional Procedure input  G:X=" " CHKXQ  I $$INPAT^IBCEF(DA(1)),'$P($G(^IBE(350.9,1,1)),"^",15),X'?1A1.2N D  G CHKXQ  . K X  . D EN^DDIOL("Site param does not allow entry of non-PTF procedures") ;Fileman error here will be: The previous error occurred when performing an action specified in a Pre-lookup transform (7.5 node).  G:'$D(^UTILITY($J,"IB")) CHKXQ  ;S M=($A($E(X,1))-64),S=+$E(X,2) Q:'$G(^UTILITY($J,"IB",M,S)) S X="`"\_+^(S)  S M=0 I X?1A1.2N S N=$G(^UTILITY($J,"IB","B",X)) S M=+N,S=+$P(N,U,2),P=X S S=$G(^UTILITY($J,"IB",M,S)) I +S S X="`"\_+S I $P(N,U,3)="N" S X=""""\_X\_"""" S $P(^UTILITY($J,"IB","B",P),U,3)="Y"  I +M,$D(DGPROCDT),DGPROCDT'=$P($G(^UTILITY($J,"IB",M,1)),"^",2) S DGPROCDT=$P(^(1),"^",2) W !!,"Procedure Date: " S Y=DGPROCDT X ^DD("DD") W Y,! CHKXQ Q  ; CODMUL ;Date oriented entry of procedure DELASK I $D(IBZ20),IBZ20,IBZ20'=$P(^DGCR(399,IBIFN,0),U,9) S %=2 W !,"SINCE THE PROCEDURE CODING METHOD HAS BEEN CHANGED, DO YOU WANT TO DELETE ALL",!,"PROCEDURE CODES IN THIS BILL"  I  D YN^DICN Q:%=-1 D:%=1 DELADD I %Y?1."?" W !!,"If you answer 'Yes', all procedure codes will be DELETED from this bill.",! G DELASK  K %,%Y,DA,IBZ20,DIK ;W !,"Procedure Entry:"  ; CODDT I $D(IBIFN),$D(^DGCR(399,IBIFN,0)),$P(^(0),U,9) S DIC("V")=$S($P(^(0),U,9)=9:"I +Y(0)=80.1",$P(^(0),U,9)=4!($P(^(0),U,9)=5):"I +Y(0)=81",1:"")  I $P($G(^DGCR(399,IBIFN,0)),"^",5)<3 S IBZTYPE=1 I $P($G(^UTILITY($J,"IB",1,1)),"^",2) S DGPROCDT=$P(^(1),"^",2) D ASKCOD  S X=$$PRCDIV^IBCU71(IBIFN) I +X W !!,$P(X,U,2),!  N Z,Z0 S Z=$G(^DGCR(399,IBIFN,"U")),Z0=$$FMTE^XLFDT($P(Z,U),"2D")\_"-"\_$$FMTE^XLFDT($P(Z,U,2),"2D")  W !,"Select PROCEDURE DATE"\_$S($TR(Z0,"-")'="":" ("\_Z0\_")",1:"")\_": " R X:DTIME G:'$T!("^"[X) CODQ D:X["?" CODHLP  S IBEX=0 D  ; Get procedure date  . I X=" ",$D(DGPROCDT),DGPROCDT?7N S Y=DGPROCDT D D^DIQ W " (",Y,")" Q  . I X=" ",+$P($G(^DGCR(399,IBIFN,"OP",0)),"^",4) S (DGPROCDT,Y)=$O(^DGCR(399,IBIFN,"OP",0)) D D^DIQ W " (",Y,")" Q  . S %DT="EXP",%DT(0)=-DT D ^%DT K %DT I Y<1 S IBEX=1 Q  . I '$$OPV2^IBCU41(Y,IBIFN,1) S IBEX=1 Q  . S:'$G(IBZTYPE) X=$$OPV^IBCU41(Y,IBIFN) S DGPROCDT=Y  I 'IBEX D ASKCOD,ADDCPT^IBCU71:$D(DGCPT)  K IBEX  G CODDT  ; ASKCOD N Z,Z0,DA,IBACT,IBQUIT,IBLNPRV  ;WCJ;2.0\*432  N IBPOPOUT  S IBPOPOUT=0 ; IB\*2.0\*447 BI  K DGCPT  S DGCPT=0,DGCPTUP=$P($G(^IBE(350.9,1,1)),"^",19),DGADDVST=0,IBFT=$P($G(^DGCR(399,IBIFN,0)),"^",19)  I '$D(^DGCR(399,IBIFN,"CP",0)) S ^DGCR(399,IBIFN,"CP",0)=U\_$$GETSPEC^IBEFUNC(399,304)  ;  F  S IBQUIT=0 D  Q:IBQUIT  . S IBPOPOUT=0  . D DICV ; restrict code type to PCM  . S DIC("A")=" Select PROCEDURE: "  . S DIC="^DGCR(399,"\_IBIFN\_",""CP"","  . S DIC(0)="AEQMNL"  . S DIC("S")="I '$D(DIV(""S""))&($P(^(0),U,2)=DGPROCDT)"  . S DIC("DR")="1///^S X=DGPROCDT"  . S DA(1)=IBIFN,DLAYGO=399  . W ! D ^DIC I Y<1 S IBQUIT=1 Q  . S IBPROCP=+Y  . ; If we just added inactive code - it must be deleted.  . S IBACT=0 ; Active flag  . I Y["ICD0" S IBACT=$$ICD0ACT^IBACSV(+$P(Y,U,2),$$BDATE^IBACSV(IBIFN))  . I Y["ICPT" S IBACT=$$CPTACT^IBACSV(+$P(Y,U,2),DGPROCDT)  . S DGCPTNEW=$P(Y,"^",3) ;Was the procedure just added?  . I DGCPTNEW,'IBACT D DELPROC Q  . I 'IBACT W !,\*7,"Warning: Procedure code is inactive on this date",!  . I DGCPTNEW,$D(^UTILITY($J,"IB")),$$INPAT^IBCEF(IBIFN),Y["ICPT(" D DATA^IBCU74(Y,.IBLNPRV)  . S DGADDVST=$S(DGCPTNEW:1,$D(DGADDVST):DGADDVST,1:0)  . N IBPRV,IBPRVO,IBPRVN  . ;  . ; Line level provider function by form type.  . ; CMS-1500 (FORM TYPE=2)  . ; RENDERING PROVIDER, REFERRING PROVIDER,  . ; and SUPERVISING PROVIDER.  . ; UB-04 (FORM TYPE=3)  . ; RENDERING PROVIDER, REFERRING PROVIDER,  . ; OPERATING PROVIDER, and OTHER OPERATING  . ; PROVIDER.  . ;  . ; Removed: Call to $$MAINPRV^IBCEU(IBIFN) is for claim  . ; level provider defaults.  . ; 1. For new line level providers we don't need  . ; or want default claim level provider  . ; (requirement).  . ; 2. We don't want to default claim level to  . ; line level provider (requirement).  . ;  . K DIC("V") ; DEM;432 - KILL DIC("V") because this was for previous variable pointer use.  . ;  . N IBPROCSV  ; DEM;432 - Variable IBPROCSV is variable to preserve value of 'Y', which is procedure code info returned by call to ^DIC.  . S IBPROCSV=Y  ; DEM;432 - Preserve value of Y for after calls to FileMan (Y = procedure code info returned by call to ^DIC).  . K DR   ;WCJ;IB\*2.0\*432  . ;  . I IBPROCSV["ICD0" S DR=".01",DIE=DIC,(IBPROCP,DA)=+Y D ^DIE Q:'$D(DA)!($D(Y)) K DR ; IB\*2.0\*461  . I IBPROCSV["ICPT" S DR=".01;16",DIE=DIC,(IBPROCP,DA)=+Y D ^DIE Q:'$D(DA)!($D(Y)) K DR ; IB\*2.0\*447 BI  . ;  . S DR=""  . ;  . ; MRD;IB\*2.0\*516 - Added line level PROCEDURE DESCRIPTION field,  . ; asked only if the procedure is an "NOC".  . I IBPROCSV["ICPT",$$NOCPROC(IBPROCSV) D  . . S DA=$P(IBPROCSV,"^") ; The line# on the bill/claim.  . . S DR=51 ; Field# for PROCEDURE DESCRIPTION  . . D ^DIE  . . Q  . ;  . D EN^IBCU7B ; DEM;432 - Call to line level provider user input.  . S Y=IBPROCSV  ; DEM;432 - Restore value of Y after calls to FileMan  . K IBPROCSV  . K DR   ;WCJ;IB\*2.0\*432  . I IBPOPOUT Q   ; IB\*2.0\*447 BI  . S DR="" I Y["ICPT" S DR="6;5//"\_$$DEFDIV(IBIFN)\_";"  . S DR=DR\_$S(IBFT=2:"8;9;17//NO;",1:"")\_3,DIE=DIC,(IBPROCP,DA)=+Y D ^DIE Q:'$D(DA)!($E($G(Y))=U)  . K DR   ;WCJ;IB\*2.0\*432  . ;  . ; MRD;IB\*2.0\*516 - Allow user to add an NDC and Units. Ask only if  . ; coding system is not ICD and this is not a prescription claim. If  . ; an NDC is entered, prompt for Units.  . I $P($G(^DGCR(399,IBIFN,0)),U,9)'=9,'$$RXLINK^IBCSC5C(IBIFN,IBPROCP) D  . . K DA  . . S DA=IBPROCP,DA(1)=IBIFN,DIE="^DGCR(399,"\_IBIFN\_",""CP"","  . . ; vd/Beginning IB\*2\*577 - Added the prompt for Unit/Basis of Measurement.  . . ; S DR="53NDC NUMBER;I X="""" S Y="""";54//1"  . . S DR="53NDC NUMBER;I X="""" S Y="""";52//UN;54QUANTITY//1"  ;Prompt for NDC, UN & amt.  . . ; vd/Ending IB\*2\*577  . . D ^DIE  . . Q  . ;  . I IBFT=3 D:'$$INPAT^IBCEF(IBIFN) ATTACH  ; DEM;432 - Prompt for Attachment Control Number.  . ; DEM;432 - Add Additional OB Minutes to DR string for call to DIE.  . S DR=$$SPCUNIT(IBIFN,IBPROCP) S:DR["15;" DR=DR\_"74Additional OB Minutes" D ^DIE ; miles/minutes/hours  . ;  . I IBFT=2 D  .. D DX^IBCU72(IBIFN,IBPROCP)  .. S X=$$ADDTNL(IBIFN,.DA)  . Q:$$INPAT^IBCEF(IBIFN) ;only outpatient bills  . ;add procedures to array for download to PCE: dgcpt(assoc clinic,cpt,'provider^first dx^modifiers',cnt)=""  . S DGPROC=$G(^DGCR(399,IBIFN,"CP",+DA,0))  . S X=$P(DGPROC,U,18)\_U\_+$G(^IBA(362.3,+$P(DGPROC,U,11),0))\_U\_$P(DGPROC,U,15)  . I 'DGCPTNEW,$P(DGPROC,"^",7)="" S DGCPTNEW=2  . I DGCPTUP,DGCPTNEW S DGCPT=DGCPT+1 I $P(DGPROC,"^",7) S DGCPT($P(DGPROC,"^",7),+DGPROC,X,DGCPT)=""  . ; add visit date to bill  . I DGADDVST S (X,DINUM)=DGPROCDT D VFILE1^IBCOPV1 K DINUM,X,DGNOADD,DGADDVST  ; Delete modifiers with only a sequence #, no code  S Z=0 F  S Z=$O(^DGCR(399,IBIFN,"CP",Z)) Q:'Z  S Z0=0 F  S Z0=$O(^DGCR(399,IBIFN,"CP",Z,"MOD",Z0)) Q:'Z0  I $P($G(^(Z0,0)),U,2)="" S DA(2)=IBIFN,DA(1)=Z,DA=Z0,DIK="^DGCR(399,"\_DA(2)\_",""CP"","\_DA(1)\_",""MOD""," D ^DIK  Q CODQ K %DT,DGPROC,DIC,DIE,DR,DGPROCDT,IBPROCP,DLAYGO  K IBFT,DGNOADD,DGADDVST,DGCPT,DGCPTUP,IBZTYPE,DGCPTNEW  Q  ; DELPROC ; Remove the selected procedure, because of inactive status (cancel selection)  W !!,\*7,"The Procedure code is inactive on ",$$DAT1^IBOUTL(DGPROCDT),"."  W !,"Please select another Procedure."  S DA(1)=IBIFN,DA=+Y,DIK="^DGCR(399,"\_IBIFN\_",""CP"","  D ^DIK  Q  ; DELADD N Z,Z0,DA,DIK,X,Y  S DA(1)=IBIFN  ;Delete references to proc on rev codes  S Z=0 F  S Z=$O(^DGCR(399,IBIFN,"RC",Z)) Q:'Z  S Z0=$G(^(Z,0)) I Z0'="",$P(Z0,U,15)!$S($P(Z0,U,10)=3:$P(Z0,U,11),1:0) S DIE="^DGCR(399,"\_DA(1)\_",""RC"",",DA=Z,DR=".11///@;.15///@"\_$S($P(Z0,U,8):"",1:";.08////1") D ^DIE  S DIK="^DGCR(399,"\_DA(1)\_",""CP""," F DA=0:0 S DA=$O(^DGCR(399,DA(1),"CP",DA)) Q:'DA  D ^DIK  S DGRVRCAL=1  Q  ; DTMES ;Message if procedure date not in date range  Q:'$D(IBIFN) Q:'$D(^DGCR(399,IBIFN,"U")) S DGNODUU=^("U")  G:X'<$P(DGNODUU,"^")&(X'>$P(DGNODUU,"^",2)) DTMESQ  W \*7,!!?3,"Date must be within STATEMENT COVERS FROM and STATEMENT COVERS TO period."  S Y=$P(DGNODUU,"^") X ^DD("DD")  W !?3,"Enter a date between ",Y," and " S Y=$P(DGNODUU,"^",2) X ^DD("DD") W Y,!  K X,Y DTMESQ K DGNODUU Q  ; CODHLP ;Display Additional Procedure codes  N I,J,Y,IBMOD  I '$O(^DGCR(399,IBIFN,"CP",0)) W !!?5,"No Codes Entered!",! Q  W ! F I=0:0 S I=$O(^DGCR(399,IBIFN,"CP",I)) Q:'I  S Y=$G(^(I,0)) S Z=$$PRCNM^IBCSCH1($P(Y,"^",1),$P(Y,"^",2)) W !?5,$E($P(Z,"^",2),1,33),?40,"- ",$P(Z,"^") D  . N IBY  . S IBY=$P(Y,U,2)  . S IBMOD=$$GETMOD^IBEFUNC(IBIFN,I,1)  . I IBMOD'="" S IBMOD="/"\_IBMOD W IBMOD  . W ?60,"Date: " S Y=IBY D DT^DIQ  W !  ;  K Z Q  ; DICV I $D(IBIFN),$D(^DGCR(399,IBIFN,0)),$P(^(0),U,9) S DIC("V")=$S($P(^(0),U,9)=9:"I +Y(0)=80.1",$P(^(0),U,9)=4!($P(^(0),U,9)=5):"I +Y(0)=81",1:"")  Q  ; DEFDIV(IBIFN) ; Find default division for bill IBIFN  Q $P($G(^DG(40.8,+$P($G(^DGCR(399,IBIFN,0)),U,22),0)),U)  ; ADDTNL(IBIFN,DA) ;  N DR,IBOK,X,Y,DIR  S IBOK=1  S DR="19T;50.09T;50.08T" D ^DIE ; WCJ;IB\*2.0\*488 Added Ts  ;I '($$FT^IBCEF(IBIFN)'=3&($$INPAT^IBCEF(IBIFN))) D ATTACH ; DEM;432 - Prompt for Attachment Control Number.  I '($$FT^IBCEF(IBIFN)=3&($$INPAT^IBCEF(IBIFN))) D ATTACH  ; DEM;432 - Prompt for Attachment Control Number.  I $D(Y) S IBOK=0 G ADDTNLQ  ;/Beginning of IB\*2.0\*488 (vd)  ;S DIR("B")="NO",DIR("A")="EDIT CMS-1500 SPECIAL PROGRAM FIELDS and BOX 19?: ",DIR("A",1)=" ",DIR(0)="YA"  ;S DIR("?",1)="Respond YES only if you need to add/edit data for chiropractic visits,"  ;S DIR("?")="EPSDT care, or if billing for HOSPICE and attending is not a hospice employee."  ;D ^DIR K DIR  ;I Y'=1 S IBOK=0 G ADDTNLQ  ;S DR="W !,"" <<EPSDT>>"";50.07;W !!,"" <<HOSPICE>>"";50.03"  S DR="50.07T;50.03T"   ;WCJ;IB\*2.0\*488 added Ts  ;/End of IB\*2.0\*488 (vd)  D ^DIE  W ! ADDTNLQ Q IBOK  ; XTRA1(Y) ;  K Y  Q  ; SPCUNIT(IBIFN,DA) ; return fields for special units if applicable, in DR form  N IB0,IBCPT,IBDR,IBCT,IBFT,DFN S IBDR=""  S IB0=$G(^DGCR(399,+$G(IBIFN),0)),IBCT=$P(IB0,U,27),IBFT=$P(IB0,U,19),DFN=$P(IB0,U,2)  S IBCPT=$G(^DGCR(399,+$G(IBIFN),"CP",+$G(DA),0)) I IBCPT'["ICPT" G SPCUNTQ  I +$$ITMUNIT^IBCRU4(+IBCPT,5,IBCT) S IBDR="15;" D SROMIN^IBCU74(IBIFN,DA) G SPCUNTQ ; minutes  I +$$ITMUNIT^IBCRU4(+IBCPT,4,IBCT) S IBDR="21;" G SPCUNTQ ; miles  I +$$ITMUNIT^IBCRU4(+IBCPT,6,IBCT) S IBDR="22//"\_$$OBSHOUR^IBCU74(DFN,$P(IBCPT,U,2))\_";" G SPCUNTQ ; hours  I +IBFT=2,$P($G(^IBE(353.2,+$P(IBCPT,U,10),0)),U,2)="ANESTHESIA" S IBDR="15;" ; minutes SPCUNTQ Q IBDR  ; ATTACH ; DEM;432 - Attachment control number.  ; Ask if user wants to enter Attachment Control Number.  N DIR,X,Y,DA,DIE,DR  S DIR("A")="Enter Attachment Control Number"  S DIR(0)="Y",DIR("B")="NO"  D ^DIR  Q:'Y  ; User chose to enter Attachment Control Number.  ; User enters Attachment Control fields.  S DA(1)=IBIFN,DA=IBPROCP  S DIE="^DGCR(399,"\_DA(1)\_",""CP"","  S DR="71Report Type;72Report Transmission Method;70Attachment Control Number"  D ^DIE  Q  ; NOCPROC(IBPROCSV) ; MRD;IB\*2.0\*516 - Function to determine if procedure is an  ; "NOC". Returns '1' if "NOC" procedure, otherwise '0'.  ;  N IBNOC,IBPROCEX,IBPROCIN,IBPROCNM,IBX  S IBNOC=0  I $G(IBPROCSV)="" G NOCPROCQ  S IBPROCIN=$P($P(IBPROCSV,U,2),";")  I IBPROCIN="" G NOCPROCQ  ;  ; If procedure code ends in '99', quit with a '1'.  ;  S IBPROCEX=$P($G(^ICPT(IBPROCIN,0)),U,1)  I $E(IBPROCEX,$L(IBPROCEX)-1,$L(IBPROCEX))=99 S IBNOC=1 G NOCPROCQ  ;  ; Pull procedure name, then check to see if it contains one of the  ; specified strings.  ;  S IBPROCNM=$P($G(^ICPT(IBPROCIN,0)),U,2)  I IBPROCNM'="",$$NOC(IBPROCNM) S IBNOC=1 G NOCPROCQ  ;  S IBX=0  F  S IBX=$O(^ICPT(IBPROCIN,"D",IBX)) Q:'IBX  D  I IBNOC=1 Q  . S IBTEXT=$G(^ICPT(IBPROCIN,"D",IBX,0))  . I $G(^ICPT(IBPROCIN,"D",IBX+1,0))'="" S IBTEXT=IBTEXT\_" "\_$G(^ICPT(IBPROCIN,"D",IBX+1,0))  . S IBNOC=$$NOC(IBTEXT)  . Q  ; NOCPROCQ ; Quit out.  Q IBNOC  ; NOC(IBTEXT) ; Quit with '1' if IBTEXT contains one of the specified strings.  ;  S IBTEXT=$TR(IBTEXT,"abcdefghijklmnopqrstuvwxyz","ABCDEFGHIJKLMNOPQRSTUVWXYZ")  ;  I IBTEXT["NOT OTHERWISE" Q 1  I IBTEXT["NOT ELSEWHERE" Q 1  I IBTEXT["NOT LISTED" Q 1  I IBTEXT["UNLISTED" Q 1  I IBTEXT["UNSPECIFIED" Q 1  I IBTEXT["UNCLASSIFIED" Q 1  I IBTEXT["NON-SPECIFIED" Q 1  I IBTEXT["NOS " Q 1  I IBTEXT["NOS;" Q 1  I IBTEXT["NOS." Q 1  I IBTEXT["NOS," Q 1  I IBTEXT["NOS/" Q 1  I IBTEXT["(NOS)" Q 1  I IBTEXT["NOC " Q 1  I IBTEXT["NOC;" Q 1  I IBTEXT["NOC." Q 1  I IBTEXT["NOC," Q 1  I IBTEXT["NOC/" Q 1  I IBTEXT["(NOC)" Q 1  ;  ; Check if last three charcters are 'NOC' or 'NOS'.  ;  S IBTEXT=$E(IBTEXT,$L(IBTEXT)-2,$L(IBTEXT))  I IBTEXT="NOC" Q 1  I IBTEXT="NOS" Q 1  ;  Q 0 | | | | | | | | | |

1. ADD two new triggers to the NDC field [#399.0304, 53] that require the UNITS/BASIS OF MEASUREMENT field [#399.0304, 52] and the UNITS field [#399.0304, 54] to be populated if there is an NDC number present.

399.0304,53 NDC 1;7 FREE TEXT

INPUT TRANSFORM: K:$L(X)>13!($L(X)<13)!'(X?5N1"-"4N1"-"2N) X

LAST EDITED: APR 13, 2017

HELP-PROMPT: Enter a National Drug Code in a 5-4-2 format

(nnnnn-nnnn-nn) if required on a

non-prescription claim.

DESCRIPTION: Enter a National Drug Code in a 5-4-2 format

(nnnnn-nnnn-nn) if required on a

non-prescription claim.

TECHNICAL DESCR: Enter a National Drug Code in a 5-4-2 format

(nnnnn-nnnn-nn) if required on a

non-prescription claim.

CROSS-REFERENCE: ^^TRIGGER^399.0304^52

1)= Q

2)= X ^DD(399.0304,53,1,1,2.3) I X S X=DIV S Y(

1)=$S($D(^DGCR(399,D0,"CP",D1,1)):^(1),1:""),Y(

1)=$S($D(^DGCR(399,D0,"CP",D1,2)):^(2),1:"") S

X=$P(Y(1),U,1),X=X S DIU=X K Y S X="" X ^DD(399

.0304,53,1,1,2.4)

2.3)= K DIV S DIV=X,D0=DA(1),DIV(0)=D0,D1=DA,DI

V(1)=D1 S Y(0)=X S Y(1)=$S($D(^DGCR(399,D0,"CP"

,D1,1)):^(1),1:"") S X=$P(Y(1),U,7)=""

2.4)= S DIH=$G(^DGCR(399,DIV(0),"CP",DIV(1),2))

,DIV=X S $P(^(2),U,1)=DIV,DIH=399.0304,DIG=52 D

^DICR

CREATE VALUE)= NO EFFECT

DELETE CONDITION)= NDC=""

DELETE VALUE)= @

FIELD)= UNITS/BASIS OF MEASUREMENT

When the NDC Code is removed, the UNITS/BASIS

OF MEASUREMENT field should be removed as well.

CROSS-REFERENCE: ^^TRIGGER^399.0304^54

1)= Q

2)= X ^DD(399.0304,53,1,2,2.3) I X S X=DIV S Y(

1)=$S($D(^DGCR(399,D0,"CP",D1,1)):^(1),1:"") S

X=$P(Y(1),U,8),X=X S DIU=X K Y S X="" S DIH=$G(

^DGCR(399,DIV(0),"CP",DIV(1),1)),DIV=X S $P(^(1

),U,8)=DIV,DIH=399.0304,DIG=54 D ^DICR

2.3)= K DIV S DIV=X,D0=DA(1),DIV(0)=D0,D1=DA,DI

V(1)=D1 S Y(0)=X S Y(1)=$S($D(^DGCR(399,D0,"CP"

,D1,1)):^(1),1:"") S X=$P(Y(1),U,7)=""

CREATE VALUE)= NO EFFECT

DELETE CONDITION)= NDC=""

DELETE VALUE)= @

FIELD)= UNITS

When the NDC Code is removed, the UNITS field

should be removed as well.

1. ADD a new validation code to check claim lines and confirm that if the NDC field is populated, then the UNITS and UNITS/BASIS OF MEASUREMENT fields are populated as well.

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCBB1 | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** |  | | | | | | | | |
| **Related Options** | None | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  |  | | | | |  | | | |
| **Data Dictionary (DD) References** | None | | | | | | | | |
| **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** | | | | | | | | | |
| IBCBB1 ;ALB/AAS - CONTINUATION OF EDIT CHECK ROUTINE ;2-NOV-89  ;;2.0;INTEGRATED BILLING;\*\*27,52,80,93,106,51,151,148,153,137,232,280,155,320,343,349,363,371,395,384,432,447,488,554\*\*;21-MAR-94;Build 16  ;Per VA Directive 6402, this routine should not be modified.  ;  ; \*\*\* Begin IB\*2.0\*488 VD (Issue 46 RBN)  N I  S I=""  S X=+$G(^DGCR(399,IBIFN,"MP"))  I 'X,$$MCRWNR^IBEFUNC(+$$CURR^IBCEF2(IBIFN)) S X=+$$CURR^IBCEF2(IBIFN)  I X,+$G(^DIC(36,X,3)) S I=$P(^(3),U,$S($$FT^IBCEF(IBIFN)=2:2,1:4))  S I=$$UP^XLFSTR(I)  I (I'=""&(I["PRNT")&($G(IBER)'["IB488")) D   . S IBER=$G(IBER)\_"IB488;"  ;  ; Cause an error if FORCED TO PRINT TO CLEARINGHOUSE  I $P($G(^DGCR(399,IBIFN,"TX")),U,8)=2 D  . S IBER=$G(IBER)\_"IB489;"  ;  ; Cause a fatal error if the claim has no procedures & is NOT a UB-04 Inpatient claim.  I +$O(^DGCR(399,IBIFN,"CP",0))=0 D  .I $$INPAT^IBCEF(IBIFN,1),$$INSPRF^IBCEF(IBIFN) Q   ; inpatient UB-04 check  .I '$$INPAT^IBCEF(IBIFN,1),$$INSPRF^IBCEF(IBIFN) D  Q      ; Outpatient Institutional Claim.  ..I IBER["IB352" Q  ..S IBER=IBER\_"IB352;"  .;  .; Professional claim  .I IBER["IB353" Q  .S IBER=IBER\_"IB353;"  .Q  ; \*\*\* End IB\*2.0\*488 -- VD  ;  ;MAP TO DGCRBB1  ; % ;Bill Status  N Z,Z0,Z1,IBFT  I $S(+IBST=0:1,1:"^1^2^3^4^7^"'[(U\_IBST\_U)) S IBER=IBER\_"IB045;"  ;  ;Statement Covers From  I IBFDT="" S IBER=IBER\_"IB061;"  I IBFDT]"",IBFDT'?7N&(IBFDT'?7N1".".N) S IBER=IBER\_"IB061;"  I IBFDT>IBTDT S IBER=IBER\_"IB061;" ; from must be on or before the to date   S IBFFY=$$FY^IBOUTL(IBFDT)  ; if inpat - from date must not be prior to admit date.  I $$INPAT^IBCEF(IBIFN,1),(IBFDT<($P($G(^DGPT(+$P(IBND0,U,8),0)),U,2)\1)) S IBER=IBER\_"IB061;"  ;  ;Statement Covers To  I IBTDT="" S IBER=IBER\_"IB062;"  I IBTDT]"",IBTDT'?7N&(IBTDT'?7N1".".N) S IBER=IBER\_"IB062;"  I IBTDT>DT!(IBTDT<IBFDT) S IBER=IBER\_"IB062;"  ; to date must not be >than today's date  S IBTFY=$$FY^IBOUTL(IBTDT)  ;  ;Total Charges  ; IB\*2.0\*447/TAZ Removed this error so that zero dollar revenue codes can process on the 837  ;I +IBTC'>0!(+IBTC'=IBTC) S IBER=IBER\_"IB064;"  ;  ;Billable charges for secondary claim  I $$MCRONBIL^IBEFUNC(IBIFN)&(($P(IBNDU1,U,1)-$P(IBNDU1,U,2))'>0) S IBER=IBER\_"IB094;"  ;Fiscal Year 1  S IBFFY=$$FY^IBOUTL(IBFDT)  ;  ;Check provider link for current user, enterer, reviewer and Authorizor  I '$D(^VA(200,DUZ,0)) S IBER=IBER\_"IB048;"  I IBEU]"",'$D(^VA(200,IBEU,0)) S IBER=IBER\_"IB048;"  I IBRU]"",'$D(^VA(200,IBRU,0)) S IBER=IBER\_"IB060;"  I IBAU]"",'$D(^VA(200,IBAU,0)) S IBER=IBER\_"IB041;"  ;  I IBER="",+$$STA^PRCAFN(IBIFN)=104 S IBER=IBER\_"IB040;"  ; If ins bill, must have valid COB sequence  I $P(IBND0,U,11)="i",$S($P(IBND0,U,21)="":1,1:"PST"'[$P(IBND0,U,21)) S IBER=IBER\_"IB324;"  ;  ; Check for valid sec provider id for current ins  S Z=0 F  S Z=$O(^DGCR(399,IBIFN,"PRV",Z)) Q:'Z  S Z0=$G(^(Z,0)),Z1=+$$COBN^IBCEF(IBIFN) I $P(Z0,U,4+Z1)'="",$P(Z0,U,11+Z1)'="" D  . I '$$SECIDCK^IBCEF74(IBIFN,Z1,$P(Z0,U,11+Z1),Z) D WARN^IBCBB11("Prov secondary id type for the "\_$P("PRIMARY^SECONDARY^TERTIARY",U,Z1)\_" "\_$$EXTERNAL^DILFD(399.0222,.01,,+Z0)\_" is invalid/won't transmit")  ; Check NPIs  D NPICHK^IBCBB11  ;  ; Check multiple rx NPIs  D RXNPI^IBCBB11(IBIFN)  ;  ; Check taxonomies  D TAXCHK^IBCBB11  ;  ; Check for Physician Name  K IBXDATA D F^IBCEF("N-ATT/REND PHYSICIAN NAME",,,IBIFN)  ; IB\*2.0\*432 - CMS1500 no longer needs a claim level rendering  S IBFT=$$FT^IBCEF(IBIFN)  I IBFT'=2,$P($G(IBXDATA),U)="" S IBER=IBER\_"IB303;"  ;  N FUNCTION,IBINS  ; IB\*2.0\*432 - CMS1500 no longer needs a claim level rendering  ;S FUNCTION=$S($$FT^IBCEF(IBIFN)=3:4,1:3)  S FUNCTION=$S(IBFT=3:4,1:3)  I IBFT'=2,IBER'["IB303;" D  . F IBINS=1:1:3 D  .. S Z=$$GETTYP^IBCEP2A(IBIFN,IBINS)  .. I Z,$P(Z,U,2) D  ; Rendering/attending prov secondary id required  ... N IBID,IBOK,Q0  ... D PROVINF^IBCEF74(IBIFN,IBINS,.IBID,1,"C") ; check all as though they were current  ... S IBOK=0  ... S Q0=0 F  S Q0=$O(IBID(1,FUNCTION,Q0)) Q:'Q0  I $P(IBID(1,FUNCTION,Q0),U,9)=+Z S IBOK=1 Q  ... I 'IBOK S IBER=IBER\_$S(IBINS=1:"IB236;",IBINS=2:"IB237;",IBINS=3:"IB238;",1:"")  ;  ; Patch 432 enh5:The IB system shall no longer prevent users from authorizing(fatal error message)a claim because the system cannot find the providersSSNorEIN  ; D PRIIDCHK^IBCBB11  ;  N IBM,IBM1  S IBM=$G(^DGCR(399,IBIFN,"M"))  S IBM1=$G(^DGCR(399,IBIFN,"M1"))  I $P(IBM,U),$P($G(^DIC(36,$P(IBM,U),4)),U,6),$P(IBM1,U,2)="" S IBER=IBER\_"IB244;"  I $P(IBM,U,2),$P($G(^DIC(36,$P(IBM,U,2),4)),U,6),$P(IBM1,U,3)="" S IBER=IBER\_"IB245;"  I $P(IBM,U,3),$P($G(^DIC(36,$P(IBM,U,3),4)),U,6),$P(IBM1,U,4)="" S IBER=IBER\_"IB246;"  ;  ; If outside facility, check for ID and qualifier in 355.93  ; 5/15/06 - esg - hard error IB243 turned into warning message instead  S Z=$P($G(^DGCR(399,IBIFN,"U2")),U,10)  I Z D  . I $P($G(^IBA(355.93,Z,0)),U,9)=""!($P($G(^IBA(355.93,Z,0)),U,13)="") D  .. N Z1,Z2  .. S Z1="Missing Lab or Facility Primary ID for non-VA facility, "  .. S Z2=$$EXTERNAL^DILFD(399,232,,Z)  .. I $L(Z2)'>19 D WARN^IBCBB11(Z1\_Z2) Q  .. D WARN^IBCBB11(Z1),WARN^IBCBB11(" "\_Z2)  .. Q  . Q  ;  ; Must be one and only one division on bill  S IBZ=$$MULTDIV^IBCBB11(IBIFN,IBND0)  ; I IBZ S IBER=IBER\_$S(IBZ=1:"IB095;",IBZ=2:"IB104;",1:"IB105;")  ; Allow multi-divisional for OP instutional claims  I IBZ,$$INPAT^IBCEF(IBIFN)!'($$INSPRF^IBCEF(IBIFN)) S IBER=IBER\_$S(IBZ=1:"IB095;",IBZ=2:"IB104;",1:"IB105;")  ; Still need error msg on OP Institutional if No Default division  I IBZ=3,'$$INPAT^IBCEF(IBIFN),$$INSPRF^IBCEF(IBIFN) S IBER=IBER\_"IB105;"  ; Division address must be defined in institution file  I $P(IBND0,U,22) D  . N Z,Z0,Z1  . S Z0=$G(^DIC(4,+$P($G(^DG(40.8,+$P(IBND0,U,22),0)),U,7),0))  . S Z1=$G(^DIC(4,+$P($G(^DG(40.8,+$P(IBND0,U,22),0)),U,7),1))  . I $P(Z0,U,2)="" S IBER=IBER\_"IB097;" Q  . F Z=1,3,4 I $P(Z1,U,Z)="" S IBER=IBER\_"IB097;" Q  ;  ; IB\*2.0\*432 Check ambulance addresses, COB Non-covered amt. & Attachment Control  I $$AMBCK^IBCBB11(IBIFN)=1 S IBER=IBER\_"IB329;"  I $$COBAMT^IBCBB11(IBIFN)=1 S IBER=IBER\_"IB330;"  I $$TMCK^IBCBB11(IBIFN)=1 S IBER=IBER\_"IB331;"  I $$ACCK^IBCBB11(IBIFN)=1 S IBER=IBER\_"IB332;"  I $$COBMRA^IBCBB11(IBIFN)=1 S IBER=IBER\_"IB342;"  I $$COBSEC^IBCBB11(IBIFN)=1 S IBER=IBER\_"IB343;"  ;  ;CHAMPVA Rate Type and Primary Insurance Carriers Type of Coverage must match  S (IBRTCHV,IBPICHV)=0  I $P($G(^DGCR(399.3,+IBAT,0)),U,1)="CHAMPVA" S IBRTCHV=1  I $P($G(^IBE(355.2,+$P($G(^DIC(36,+IBNDMP,0)),U,13),0)),U,1)="CHAMPVA" S IBPICHV=1  I (+IBRTCHV!+IBPICHV)&('IBRTCHV!'IBPICHV) S IBER=IBER\_"IB085;"  ;  ;Non-VA bill must use FEE REIMB INS rate type; FEE REIMB INS rate type can only be used for Non-VA bill  ;IB\*2.0\*554/DRF 10/9/2015  ;N IBNVART,IBNVAST  ;S (IBNVART,IBNVAST)=0  ;I $P($G(^DGCR(399.3,+IBAT,0)),U,1)="FEE REIMB INS" S IBNVART=1  ;S IBNVAST=$$NONVAFLG(IBIFN)  ;I IBNVART,'IBNVAST S IBER=IBER\_"IB360;" ;Non-VA rate type used for bill that is not Non-VA  ;I 'IBNVART,IBNVAST S IBER=IBER\_"IB361;" ;Non-VA rate type not used for bill that is Non-VA  ;  N IBZPRC,IBZPRCUB  D F^IBCEF("N-ALL PROCEDURES","IBZPRC",,IBIFN)  ; Procedure Clinic is required for Surgical Procedures Outpt Facility Charges  I +$P(IBND0,U,27)'=2,$$BILLRATE^IBCRU3(IBAT,IBCL,IBEVDT,"RC OUTPATIENT") D  . N Z,Z0,Z1,ZE S (ZE,Z)=0 F  S Z=$O(^DGCR(399,IBIFN,"CP",Z)) Q:'Z  D  I +ZE S IBER=IBER\_"IB320;" Q  .. S Z0=$G(^DGCR(399,IBIFN,"CP",Z,0)),Z1=+Z0 I Z0'[";ICPT(" Q  .. I '((Z1'<10000)&(Z1'>69999))&'((Z1'<93501)&(Z1'>93533)) Q  .. I '$P(Z0,U,7) S ZE=1  ;  ; Extract procedures for UB-04  D F^IBCEF("N-UB-04 PROCEDURES","IBZPRCUB",,IBIFN)  ; Does this bill have ANY prescriptions associated with it?  ; Must bill prescriptions separately from other charges  ;  ; DEM;432 - Call line level provider edit checks.  D LNPROV^IBCBB12(IBIFN) ; DEM;432 - If there are line provider edits, then routine LNPROV^IBCBB12(IBIFN) updates IBER string.  ; DEM;432 - Call to Other Operating/Operating Provider edit checks.  I $$OPPROVCK^IBCBB12(IBIFN)=1 S IBER=IBER\_"IB337;"  ; DEM;432  ; DEM;432 - Line level Attachment Control edits.  I $$LNTMCK^IBCBB11(IBIFN)=1 S IBER=IBER\_"IB331;"  ; DEM;432  I $$LNACCK^IBCBB11(IBIFN)=1 S IBER=IBER\_"IB332;"  ; DEM;432  ;  I $$ISRX^IBCEF1(IBIFN) D  . N IBZ,IBRXDEF  . S IBRXDEF=$P($G(^IBE(350.9,1,1)),U,30),IBZ=0  . F  S IBZ=$O(IBZPRCUB(IBZ)) Q:'IBZ  I IBZPRCUB(IBZ),+$P(IBZPRCUB(IBZ),U)'=IBRXDEF S IBER=IBER\_"IB102;" Q  . K IBZ  ;  ; Check that COB sequences are not skipped  K Z  F Z=1:1:3 S:+$G(^DGCR(399,IBIFN,"I"\_Z)) Z(Z)=""  F Z=0:1:2 S Z0=$O(Z(Z)) Q:'Z0  I Z0'=(Z+1) S IBER=IBER\_"IB322;" Q  K Z  ; HD64676 IB\*2\*371 - OK for payer sequence to be blank when the Rate  ; Type is either Interagency or Sharing Agreement  I $P($G(^DGCR(399,IBIFN,0)),U,21)="",$P($G(^DGCR(399,IBIFN,0)),U,7)'=4,$P($G(^DGCR(399,IBIFN,0)),U,7)'=9 S IBER=IBER\_"IB323;"  K IBXDATA D F^IBCEF("N-PROCEDURE CODING METHD",,,IBIFN)  ; Coding method should agree with types of procedure codes  S IBOK=$S('$O(IBZPRC(0))!(IBXDATA=""):1,1:0)  I 'IBOK S IBOK=1,IBZ=0 F  S IBZ=$O(IBZPRC(IBZ)) Q:'IBZ  I IBZPRC(IBZ),$P(IBZPRC(IBZ),U)'[$S(IBXDATA=9:"ICD",1:"ICP") S IBOK=0 Q  I 'IBOK D WARN^IBCBB11("Coding Method does not agree with all procedure codes found on bill")  D EDITMRA^IBCBB3(.IBQUIT,.IBER,IBIFN,IBFT)  Q:$G(IBQUIT)  ;  ;Other things that could be added: Rev Code - calculating charges  ; Diagnosis Coding, if MT copay - check for other co-payments  ;  I $P(IBNDTX,U,8),$$REQMRA^IBEFUNC(IBIFN) S IBER=IBER\_"IB121;"   ; can't force MRAs to print  I $P(IBNDTX,U,8)!$P(IBNDTX,U,9) D  . Q:$P(IBNDTX,U,8)=2 ; Don't want to do this for option 2 any more.  . D WARN^IBCBB11($S($$REQMRA^IBEFUNC(IBIFN)&($P(IBNDTX,U,9)):"MRA Secondary ",1:"")\_"Bill has been forced to print "\_$S($P(IBNDTX,U,8)=1!($P(IBNDTX,U,9)=1):"locally",1:"at clearinghouse"))  N IBXZ,IBIZ F IBIZ=12,13,14 S IBXZ=$P(IBNDM,U,IBIZ) I +IBXZ S IBXZ=$P($G(^DPT(DFN,.312,IBXZ,0)),U,18) I +IBXZ S IBXZ=$G(^IBA(355.3,+IBXZ,0)) I +$P(IBXZ,U,12) D  . D WARN^IBCBB11($P($G(^DIC(36,+IBXZ,0)),U,1)\_" requires Amb Care Certification")  ;  D VALNDC^IBCBB11(IBIFN,DFN) ;validate NDC#  ;  ;Build AR array if no errors and MRA not needed or already rec'd  I IBER="",$S($$NEEDMRA^IBEFUNC(IBIFN)!($$REQMRA^IBEFUNC(IBIFN)):0,1:1) D ARRAY  ;  ;Check ROI  N ROIERR  S ROIERR=0 I $P($G(^DGCR(399,IBIFN,"U")),U,5)=1,+$P($G(^DGCR(399,IBIFN,"U")),U,7)=0 S ROIERR=1 ; screen 7 sensitive record and no ROI  I $$ROICHK^IBCBB11(IBIFN,DFN,+IBNDMP) S ROIERR=1 ; check file for sensitive Rx and missing ROI  I ROIERR S IBER=IBER\_"IB328;"  ;  ;Verify Line Charges Match Claim Total Charge. IB\*2.0\*447 BI  I +$$GET1^DIQ(399,IBIFN\_",",201)'=+$$IBLNTOT^IBCBB13(IBIFN) S IBER=IBER\_"IB344;"  ;  ;Test for valid EIN/SY ID Values. IB\*2.0\*447 BI  I $$IBSYEI^IBCBB13(IBIFN) S IBER=IBER\_"IB345;"  ;  ;Test for a missing ICN. IB\*2.0\*447 BI  I $$IBMICN^IBCBB13(IBIFN) S IBER=IBER\_"IB346;"  ;  ;Test for a ZERO charge amounts. IB\*2.0\*447 BI  I $$IBRCCHK^IBCBB13(IBIFN) D WARN^IBCBB11("Claim contains revenue codes with no associated charges.")  ;  ;Test for missing "Patient reason for visit". IB\*2.0\*447 BI  I $$FT^IBCEF(IBIFN)=3,'$$INPAT^IBCEF(IBIFN),$$IBPRV3^IBCBB13(IBIFN) S IBER=IBER\_"IB347;"  ;  ;Test for missing Payer ID. IB\*2.0\*447 BI  ;I $$IBMPID^IBCBB13(IBIFN) S IBER=IBER\_"IB348;"  ;Changed Error to Warning. IB\*2.0\*447 TAZ  I $$IBMPID^IBCBB13(IBIFN) D WARN^IBCBB11("Not all payers have Payer IDs.")  ;  ;Test for missing "Priority (Type) of Admission" for UB-04. IB\*2.0\*447 BI  I $$FT^IBCEF(IBIFN)=3,$$GET1^DIQ(399,IBIFN\_",",158)="" S IBER=IBER\_"IB349;"  ; END ;Don't kill IBIFN, IBER, DFN  I $O(^TMP($J,"BILL-WARN",0)),$G(IBER)="" S IBER="WARN" ;Warnings only  K IBBNO,IBEVDT,IBLOC,IBCL,IBTF,IBAT,IBWHO,IBST,IBFDT,IBTDT,IBTC,IBFY,IBFY1,IBAU,IBRU,IBEU,IBARTP,IBFYC,IBMRA,IBTOB,IBTOB12,IBNDU2,IBNDUF3,IBNDUF31,IBNDTX  K IBNDS,IBND0,IBNDU,IBNDM,IBNDMP,IBNDU1,IBFFY,IBTFY,IBFT,IBRTCHV,IBPICHV,IBXDATA,IBOK  I $D(IBER),IBER="" W !,"No Errors found for National edits"  Q  ; ARRAY ;Build PRCASV(array)  N IBCOBN,X  K PRCASV  Q:$$MCRWNR^IBEFUNC(+$$CURR^IBCEF2(IBIFN))  S IBCOBN=$$COBN^IBCEF(IBIFN)  S X=IBIFN  S PRCASV("BDT")=DT,PRCASV("ARREC")=IBIFN  S PRCASV("APR")=DUZ  S PRCASV("PAT")=DFN,PRCASV("CAT")=$P(^DGCR(399.3,IBAT,0),"^",6)  I IBWHO="i" S PRCASV("DEBTOR")=+IBNDMP\_";DIC(36,"  S PRCASV("DEBTOR")=$S(IBWHO="p":DFN\_";DPT(",IBWHO="o":$P(IBNDM,"^",11)\_";DIC(4,",IBWHO="i":PRCASV("DEBTOR"),1:"")  S PRCASV("CARE")=$E($$TOB^IBCEF1(IBIFN),1,2)  S PRCASV("FY")=$$FY^IBOUTL(DT)\_U\_($P(IBNDU1,U)-$P(IBNDU1,U,2))  ;S PRCASV("FY")=$P(IBNDU1,U,9)\_U\_$S($P(IBNDU1,U,2)]"":($P(IBNDU1,U,10)-$P(IBNDU1,U,2)),1:$P(IBNDU1,U,10))\_$S($P(IBNDU1,U,11)]"":U\_$P(IBNDU1,U,11)\_U\_$P(IBNDU1,U,12),1:"") PLUS I IBWHO="i",$P(IBNDM,"^",2),$D(^DIC(36,$P(IBNDM,"^",2),0)) S PRCASV("2NDINS")=$P(IBNDM,"^",2)  I IBWHO="i",$P(IBNDM,"^",3),$D(^DIC(36,$P(IBNDM,"^",3),0)) S PRCASV("3RDINS")=$P(IBNDM,"^",3)  ;  N IBX S IBX=$P(IBND0,U,21),IBX=$S(IBX="P":"I1",IBX="S":"I2",IBX="T":"I3",1:"") Q:IBX=""  N IBNDI1  Q:'$D(^DGCR(399,IBIFN,IBX)) S IBNDI1=^(IBX)  S:$P(IBNDI1,"^",3)]"" PRCASV("GPNO")=$P(IBNDI1,"^",3)  S:$P(IBNDI1,"^",15)]"" PRCASV("GPNM")=$P(IBNDI1,"^",15)  S:$P(IBNDI1,"^",17)]"" PRCASV("INPA")=$P(IBNDI1,"^",17)  S:$P(IBNDI1,"^",2)]"" PRCASV("IDNO")=$P(IBNDI1,"^",2),PRCASV("INID")=PRCASV("IDNO")  ; Check that this is a secondary or tertiary bill and insurance for previous  ; COB sequence is Medicare WNR and MRA is active --> send data elements to AR  I IBCOBN>1,$$WNRBILL^IBEFUNC(IBIFN,IBCOBN-1),$$EDIACTV^IBCEF4(2) D MRA  Q  ; MRA N IBEOB S IBEOB=0  ;  K PRCASV("MEDURE"),PRCASV("MEDCA")  ; Get EOB data  F  S IBEOB=$O(^IBM(361.1,"B",IBIFN,IBEOB)) Q:'IBEOB  D  . D MRACALC^IBCEMU2(IBEOB,IBIFN,1,.PRCASV)  Q  ;MRA  ;  ;; PREGNANCY DX CODES: V22\*\*-V24\*\*, V27\*\*-V28\*\*, 630\*\*-677\*\*  ;; FLU SHOTS PROCEDURE CODES: 90724, G0008, 90732, G0009  ; NONVAFLG(IBIFN) ; Check if Non-VA bill  ; Function returns 1 if Non-VA bill  ; IB\*2.0\*554/DRF 10/9/2015  N FLAG,PTF  S FLAG=0  I $P($G(^DGCR(399,IBIFN,"U2")),U,10)]"" S FLAG=1 ;Non-VA provider defined  S PTF=$P($G(^DGCR(399,IBIFN,0)),U,8)  I PTF,$P($G(^DGPT(PTF,0)),U,4)=1 S FLAG=1 ;PTF entry indicates Non-VA  Q FLAG | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCBB1 ;ALB/AAS - CONTINUATION OF EDIT CHECK ROUTINE ;2-NOV-89  ;;2.0;INTEGRATED BILLING;\*\*27,52,80,93,106,51,151,148,153,137,232,280,155,320,343,349,363,371,395,384,432,447,488,554,577\*\*;21-MAR-94;Build 16  ;Per VA Directive 6402, this routine should not be modified.  ;  ; \*\*\* Begin IB\*2.0\*488 VD (Issue 46 RBN)  N I  S I=""  S X=+$G(^DGCR(399,IBIFN,"MP"))  I 'X,$$MCRWNR^IBEFUNC(+$$CURR^IBCEF2(IBIFN)) S X=+$$CURR^IBCEF2(IBIFN)  I X,+$G(^DIC(36,X,3)) S I=$P(^(3),U,$S($$FT^IBCEF(IBIFN)=2:2,1:4))  S I=$$UP^XLFSTR(I)  I (I'=""&(I["PRNT")&($G(IBER)'["IB488")) D   . S IBER=$G(IBER)\_"IB488;"  ;  ; Cause an error if FORCED TO PRINT TO CLEARINGHOUSE  I $P($G(^DGCR(399,IBIFN,"TX")),U,8)=2 D  . S IBER=$G(IBER)\_"IB489;"  ;  ; Cause a fatal error if the claim has no procedures & is NOT a UB-04 Inpatient claim.  I +$O(^DGCR(399,IBIFN,"CP",0))=0 D  .I $$INPAT^IBCEF(IBIFN,1),$$INSPRF^IBCEF(IBIFN) Q   ; inpatient UB-04 check  .I '$$INPAT^IBCEF(IBIFN,1),$$INSPRF^IBCEF(IBIFN) D  Q      ; Outpatient Institutional Claim.  ..I IBER["IB352" Q  ..S IBER=IBER\_"IB352;"  .;  .; Professional claim  .I IBER["IB353" Q  .S IBER=IBER\_"IB353;"  .Q  ; \*\*\* End IB\*2.0\*488 -- VD  ;  ;MAP TO DGCRBB1  ; % ;Bill Status  N Z,Z0,Z1,IBFT  I $S(+IBST=0:1,1:"^1^2^3^4^7^"'[(U\_IBST\_U)) S IBER=IBER\_"IB045;"  ;  ;Statement Covers From  I IBFDT="" S IBER=IBER\_"IB061;"  I IBFDT]"",IBFDT'?7N&(IBFDT'?7N1".".N) S IBER=IBER\_"IB061;"  I IBFDT>IBTDT S IBER=IBER\_"IB061;" ; from must be on or before the to date   S IBFFY=$$FY^IBOUTL(IBFDT)  ; if inpat - from date must not be prior to admit date.  I $$INPAT^IBCEF(IBIFN,1),(IBFDT<($P($G(^DGPT(+$P(IBND0,U,8),0)),U,2)\1)) S IBER=IBER\_"IB061;"  ;  ;Statement Covers To  I IBTDT="" S IBER=IBER\_"IB062;"  I IBTDT]"",IBTDT'?7N&(IBTDT'?7N1".".N) S IBER=IBER\_"IB062;"  I IBTDT>DT!(IBTDT<IBFDT) S IBER=IBER\_"IB062;"  ; to date must not be >than today's date  S IBTFY=$$FY^IBOUTL(IBTDT)  ;  ;Total Charges  ; IB\*2.0\*447/TAZ Removed this error so that zero dollar revenue codes can process on the 837  ;I +IBTC'>0!(+IBTC'=IBTC) S IBER=IBER\_"IB064;"  ;  ;Billable charges for secondary claim  I $$MCRONBIL^IBEFUNC(IBIFN)&(($P(IBNDU1,U,1)-$P(IBNDU1,U,2))'>0) S IBER=IBER\_"IB094;"  ;Fiscal Year 1  S IBFFY=$$FY^IBOUTL(IBFDT)  ;  ;Check provider link for current user, enterer, reviewer and Authorizor  I '$D(^VA(200,DUZ,0)) S IBER=IBER\_"IB048;"  I IBEU]"",'$D(^VA(200,IBEU,0)) S IBER=IBER\_"IB048;"  I IBRU]"",'$D(^VA(200,IBRU,0)) S IBER=IBER\_"IB060;"  I IBAU]"",'$D(^VA(200,IBAU,0)) S IBER=IBER\_"IB041;"  ;  I IBER="",+$$STA^PRCAFN(IBIFN)=104 S IBER=IBER\_"IB040;"  ; If ins bill, must have valid COB sequence  I $P(IBND0,U,11)="i",$S($P(IBND0,U,21)="":1,1:"PST"'[$P(IBND0,U,21)) S IBER=IBER\_"IB324;"  ;  ; Check for valid sec provider id for current ins  S Z=0 F  S Z=$O(^DGCR(399,IBIFN,"PRV",Z)) Q:'Z  S Z0=$G(^(Z,0)),Z1=+$$COBN^IBCEF(IBIFN) I $P(Z0,U,4+Z1)'="",$P(Z0,U,11+Z1)'="" D  . I '$$SECIDCK^IBCEF74(IBIFN,Z1,$P(Z0,U,11+Z1),Z) D WARN^IBCBB11("Prov secondary id type for the "\_$P("PRIMARY^SECONDARY^TERTIARY",U,Z1)\_" "\_$$EXTERNAL^DILFD(399.0222,.01,,+Z0)\_" is invalid/won't transmit")  ; Check NPIs  D NPICHK^IBCBB11  ;  ; Check multiple rx NPIs  D RXNPI^IBCBB11(IBIFN)  ;  ; Check taxonomies  D TAXCHK^IBCBB11  ;  ; Check for Physician Name  K IBXDATA D F^IBCEF("N-ATT/REND PHYSICIAN NAME",,,IBIFN)  ; IB\*2.0\*432 - CMS1500 no longer needs a claim level rendering  S IBFT=$$FT^IBCEF(IBIFN)  I IBFT'=2,$P($G(IBXDATA),U)="" S IBER=IBER\_"IB303;"  ;  N FUNCTION,IBINS  ; IB\*2.0\*432 - CMS1500 no longer needs a claim level rendering  ;S FUNCTION=$S($$FT^IBCEF(IBIFN)=3:4,1:3)  S FUNCTION=$S(IBFT=3:4,1:3)  I IBFT'=2,IBER'["IB303;" D  . F IBINS=1:1:3 D  .. S Z=$$GETTYP^IBCEP2A(IBIFN,IBINS)  .. I Z,$P(Z,U,2) D  ; Rendering/attending prov secondary id required  ... N IBID,IBOK,Q0  ... D PROVINF^IBCEF74(IBIFN,IBINS,.IBID,1,"C") ; check all as though they were current  ... S IBOK=0  ... S Q0=0 F  S Q0=$O(IBID(1,FUNCTION,Q0)) Q:'Q0  I $P(IBID(1,FUNCTION,Q0),U,9)=+Z S IBOK=1 Q  ... I 'IBOK S IBER=IBER\_$S(IBINS=1:"IB236;",IBINS=2:"IB237;",IBINS=3:"IB238;",1:"")  ;  ; Patch 432 enh5:The IB system shall no longer prevent users from authorizing(fatal error message)a claim because the system cannot find the providersSSNorEIN  ; D PRIIDCHK^IBCBB11  ;  N IBM,IBM1  S IBM=$G(^DGCR(399,IBIFN,"M"))  S IBM1=$G(^DGCR(399,IBIFN,"M1"))  I $P(IBM,U),$P($G(^DIC(36,$P(IBM,U),4)),U,6),$P(IBM1,U,2)="" S IBER=IBER\_"IB244;"  I $P(IBM,U,2),$P($G(^DIC(36,$P(IBM,U,2),4)),U,6),$P(IBM1,U,3)="" S IBER=IBER\_"IB245;"  I $P(IBM,U,3),$P($G(^DIC(36,$P(IBM,U,3),4)),U,6),$P(IBM1,U,4)="" S IBER=IBER\_"IB246;"  ;  ; If outside facility, check for ID and qualifier in 355.93  ; 5/15/06 - esg - hard error IB243 turned into warning message instead  S Z=$P($G(^DGCR(399,IBIFN,"U2")),U,10)  I Z D  . I $P($G(^IBA(355.93,Z,0)),U,9)=""!($P($G(^IBA(355.93,Z,0)),U,13)="") D  .. N Z1,Z2  .. S Z1="Missing Lab or Facility Primary ID for non-VA facility, "  .. S Z2=$$EXTERNAL^DILFD(399,232,,Z)  .. I $L(Z2)'>19 D WARN^IBCBB11(Z1\_Z2) Q  .. D WARN^IBCBB11(Z1),WARN^IBCBB11(" "\_Z2)  .. Q  . Q  ;  ; Must be one and only one division on bill  S IBZ=$$MULTDIV^IBCBB11(IBIFN,IBND0)  ; I IBZ S IBER=IBER\_$S(IBZ=1:"IB095;",IBZ=2:"IB104;",1:"IB105;")  ; Allow multi-divisional for OP instutional claims  I IBZ,$$INPAT^IBCEF(IBIFN)!'($$INSPRF^IBCEF(IBIFN)) S IBER=IBER\_$S(IBZ=1:"IB095;",IBZ=2:"IB104;",1:"IB105;")  ; Still need error msg on OP Institutional if No Default division  I IBZ=3,'$$INPAT^IBCEF(IBIFN),$$INSPRF^IBCEF(IBIFN) S IBER=IBER\_"IB105;"  ; Division address must be defined in institution file  I $P(IBND0,U,22) D  . N Z,Z0,Z1  . S Z0=$G(^DIC(4,+$P($G(^DG(40.8,+$P(IBND0,U,22),0)),U,7),0))  . S Z1=$G(^DIC(4,+$P($G(^DG(40.8,+$P(IBND0,U,22),0)),U,7),1))  . I $P(Z0,U,2)="" S IBER=IBER\_"IB097;" Q  . F Z=1,3,4 I $P(Z1,U,Z)="" S IBER=IBER\_"IB097;" Q  ;  ; IB\*2.0\*432 Check ambulance addresses, COB Non-covered amt. & Attachment Control  I $$AMBCK^IBCBB11(IBIFN)=1 S IBER=IBER\_"IB329;"  I $$COBAMT^IBCBB11(IBIFN)=1 S IBER=IBER\_"IB330;"  I $$TMCK^IBCBB11(IBIFN)=1 S IBER=IBER\_"IB331;"  I $$ACCK^IBCBB11(IBIFN)=1 S IBER=IBER\_"IB332;"  I $$COBMRA^IBCBB11(IBIFN)=1 S IBER=IBER\_"IB342;"  I $$COBSEC^IBCBB11(IBIFN)=1 S IBER=IBER\_"IB343;"  ;  ;CHAMPVA Rate Type and Primary Insurance Carriers Type of Coverage must match  S (IBRTCHV,IBPICHV)=0  I $P($G(^DGCR(399.3,+IBAT,0)),U,1)="CHAMPVA" S IBRTCHV=1  I $P($G(^IBE(355.2,+$P($G(^DIC(36,+IBNDMP,0)),U,13),0)),U,1)="CHAMPVA" S IBPICHV=1  I (+IBRTCHV!+IBPICHV)&('IBRTCHV!'IBPICHV) S IBER=IBER\_"IB085;"  ;  ;Non-VA bill must use FEE REIMB INS rate type; FEE REIMB INS rate type can only be used for Non-VA bill  ;IB\*2.0\*554/DRF 10/9/2015  ;N IBNVART,IBNVAST  ;S (IBNVART,IBNVAST)=0  ;I $P($G(^DGCR(399.3,+IBAT,0)),U,1)="FEE REIMB INS" S IBNVART=1  ;S IBNVAST=$$NONVAFLG(IBIFN)  ;I IBNVART,'IBNVAST S IBER=IBER\_"IB360;" ;Non-VA rate type used for bill that is not Non-VA  ;I 'IBNVART,IBNVAST S IBER=IBER\_"IB361;" ;Non-VA rate type not used for bill that is Non-VA  ;  N IBZPRC,IBZPRCUB  D F^IBCEF("N-ALL PROCEDURES","IBZPRC",,IBIFN)  ; Procedure Clinic is required for Surgical Procedures Outpt Facility Charges  I +$P(IBND0,U,27)'=2,$$BILLRATE^IBCRU3(IBAT,IBCL,IBEVDT,"RC OUTPATIENT") D  . N Z,Z0,Z1,ZE S (ZE,Z)=0 F  S Z=$O(^DGCR(399,IBIFN,"CP",Z)) Q:'Z  D  I +ZE S IBER=IBER\_"IB320;" Q  .. S Z0=$G(^DGCR(399,IBIFN,"CP",Z,0)),Z1=+Z0 I Z0'[";ICPT(" Q  .. I '((Z1'<10000)&(Z1'>69999))&'((Z1'<93501)&(Z1'>93533)) Q  .. I '$P(Z0,U,7) S ZE=1  ;  ; Extract procedures for UB-04  D F^IBCEF("N-UB-04 PROCEDURES","IBZPRCUB",,IBIFN)  ; Does this bill have ANY prescriptions associated with it?  ; Must bill prescriptions separately from other charges  ;  ; DEM;432 - Call line level provider edit checks.  D LNPROV^IBCBB12(IBIFN) ; DEM;432 - If there are line provider edits, then routine LNPROV^IBCBB12(IBIFN) updates IBER string.  ; DEM;432 - Call to Other Operating/Operating Provider edit checks.  I $$OPPROVCK^IBCBB12(IBIFN)=1 S IBER=IBER\_"IB337;"  ; DEM;432  ; DEM;432 - Line level Attachment Control edits.  I $$LNTMCK^IBCBB11(IBIFN)=1 S IBER=IBER\_"IB331;"  ; DEM;432  I $$LNACCK^IBCBB11(IBIFN)=1 S IBER=IBER\_"IB332;"  ; DEM;432  ;  ; vd/Beginning of IB\*2\*577 - Validate Line Level NDC edits.  I $$LNNDCCK^IBCBB11(IBIFN)=1 S IBER=IBER\_"IB360;"  ;IB\*2\*577  ; vd/End of IB\*2\*577  I $$ISRX^IBCEF1(IBIFN) D  . N IBZ,IBRXDEF  . S IBRXDEF=$P($G(^IBE(350.9,1,1)),U,30),IBZ=0  . F  S IBZ=$O(IBZPRCUB(IBZ)) Q:'IBZ  I IBZPRCUB(IBZ),+$P(IBZPRCUB(IBZ),U)'=IBRXDEF S IBER=IBER\_"IB102;" Q  . K IBZ  ;  ; Check that COB sequences are not skipped  K Z  F Z=1:1:3 S:+$G(^DGCR(399,IBIFN,"I"\_Z)) Z(Z)=""  F Z=0:1:2 S Z0=$O(Z(Z)) Q:'Z0  I Z0'=(Z+1) S IBER=IBER\_"IB322;" Q  K Z  ; HD64676 IB\*2\*371 - OK for payer sequence to be blank when the Rate  ; Type is either Interagency or Sharing Agreement  I $P($G(^DGCR(399,IBIFN,0)),U,21)="",$P($G(^DGCR(399,IBIFN,0)),U,7)'=4,$P($G(^DGCR(399,IBIFN,0)),U,7)'=9 S IBER=IBER\_"IB323;"  K IBXDATA D F^IBCEF("N-PROCEDURE CODING METHD",,,IBIFN)  ; Coding method should agree with types of procedure codes  S IBOK=$S('$O(IBZPRC(0))!(IBXDATA=""):1,1:0)  I 'IBOK S IBOK=1,IBZ=0 F  S IBZ=$O(IBZPRC(IBZ)) Q:'IBZ  I IBZPRC(IBZ),$P(IBZPRC(IBZ),U)'[$S(IBXDATA=9:"ICD",1:"ICP") S IBOK=0 Q  I 'IBOK D WARN^IBCBB11("Coding Method does not agree with all procedure codes found on bill")  D EDITMRA^IBCBB3(.IBQUIT,.IBER,IBIFN,IBFT)  Q:$G(IBQUIT)  ;  ;Other things that could be added: Rev Code - calculating charges  ; Diagnosis Coding, if MT copay - check for other co-payments  ;  I $P(IBNDTX,U,8),$$REQMRA^IBEFUNC(IBIFN) S IBER=IBER\_"IB121;"   ; can't force MRAs to print  I $P(IBNDTX,U,8)!$P(IBNDTX,U,9) D  . Q:$P(IBNDTX,U,8)=2 ; Don't want to do this for option 2 any more.  . D WARN^IBCBB11($S($$REQMRA^IBEFUNC(IBIFN)&($P(IBNDTX,U,9)):"MRA Secondary ",1:"")\_"Bill has been forced to print "\_$S($P(IBNDTX,U,8)=1!($P(IBNDTX,U,9)=1):"locally",1:"at clearinghouse"))  N IBXZ,IBIZ F IBIZ=12,13,14 S IBXZ=$P(IBNDM,U,IBIZ) I +IBXZ S IBXZ=$P($G(^DPT(DFN,.312,IBXZ,0)),U,18) I +IBXZ S IBXZ=$G(^IBA(355.3,+IBXZ,0)) I +$P(IBXZ,U,12) D  . D WARN^IBCBB11($P($G(^DIC(36,+IBXZ,0)),U,1)\_" requires Amb Care Certification")  ;  D VALNDC^IBCBB11(IBIFN,DFN) ;validate NDC#  ;  ;Build AR array if no errors and MRA not needed or already rec'd  I IBER="",$S($$NEEDMRA^IBEFUNC(IBIFN)!($$REQMRA^IBEFUNC(IBIFN)):0,1:1) D ARRAY  ;  ;Check ROI  N ROIERR  S ROIERR=0 I $P($G(^DGCR(399,IBIFN,"U")),U,5)=1,+$P($G(^DGCR(399,IBIFN,"U")),U,7)=0 S ROIERR=1 ; screen 7 sensitive record and no ROI  I $$ROICHK^IBCBB11(IBIFN,DFN,+IBNDMP) S ROIERR=1 ; check file for sensitive Rx and missing ROI  I ROIERR S IBER=IBER\_"IB328;"  ;  ;Verify Line Charges Match Claim Total Charge. IB\*2.0\*447 BI  I +$$GET1^DIQ(399,IBIFN\_",",201)'=+$$IBLNTOT^IBCBB13(IBIFN) S IBER=IBER\_"IB344;"  ;  ;Test for valid EIN/SY ID Values. IB\*2.0\*447 BI  I $$IBSYEI^IBCBB13(IBIFN) S IBER=IBER\_"IB345;"  ;  ;Test for a missing ICN. IB\*2.0\*447 BI  I $$IBMICN^IBCBB13(IBIFN) S IBER=IBER\_"IB346;"  ;  ;Test for a ZERO charge amounts. IB\*2.0\*447 BI  I $$IBRCCHK^IBCBB13(IBIFN) D WARN^IBCBB11("Claim contains revenue codes with no associated charges.")  ;  ;Test for missing "Patient reason for visit". IB\*2.0\*447 BI  I $$FT^IBCEF(IBIFN)=3,'$$INPAT^IBCEF(IBIFN),$$IBPRV3^IBCBB13(IBIFN) S IBER=IBER\_"IB347;"  ;  ;Test for missing Payer ID. IB\*2.0\*447 BI  ;I $$IBMPID^IBCBB13(IBIFN) S IBER=IBER\_"IB348;"  ;Changed Error to Warning. IB\*2.0\*447 TAZ  I $$IBMPID^IBCBB13(IBIFN) D WARN^IBCBB11("Not all payers have Payer IDs.")  ;  ;Test for missing "Priority (Type) of Admission" for UB-04. IB\*2.0\*447 BI  I $$FT^IBCEF(IBIFN)=3,$$GET1^DIQ(399,IBIFN\_",",158)="" S IBER=IBER\_"IB349;"  ; END ;Don't kill IBIFN, IBER, DFN  I $O(^TMP($J,"BILL-WARN",0)),$G(IBER)="" S IBER="WARN" ;Warnings only  K IBBNO,IBEVDT,IBLOC,IBCL,IBTF,IBAT,IBWHO,IBST,IBFDT,IBTDT,IBTC,IBFY,IBFY1,IBAU,IBRU,IBEU,IBARTP,IBFYC,IBMRA,IBTOB,IBTOB12,IBNDU2,IBNDUF3,IBNDUF31,IBNDTX  K IBNDS,IBND0,IBNDU,IBNDM,IBNDMP,IBNDU1,IBFFY,IBTFY,IBFT,IBRTCHV,IBPICHV,IBXDATA,IBOK  I $D(IBER),IBER="" W !,"No Errors found for National edits"  Q  ; ARRAY ;Build PRCASV(array)  N IBCOBN,X  K PRCASV  Q:$$MCRWNR^IBEFUNC(+$$CURR^IBCEF2(IBIFN))  S IBCOBN=$$COBN^IBCEF(IBIFN)  S X=IBIFN  S PRCASV("BDT")=DT,PRCASV("ARREC")=IBIFN  S PRCASV("APR")=DUZ  S PRCASV("PAT")=DFN,PRCASV("CAT")=$P(^DGCR(399.3,IBAT,0),"^",6)  I IBWHO="i" S PRCASV("DEBTOR")=+IBNDMP\_";DIC(36,"  S PRCASV("DEBTOR")=$S(IBWHO="p":DFN\_";DPT(",IBWHO="o":$P(IBNDM,"^",11)\_";DIC(4,",IBWHO="i":PRCASV("DEBTOR"),1:"")  S PRCASV("CARE")=$E($$TOB^IBCEF1(IBIFN),1,2)  S PRCASV("FY")=$$FY^IBOUTL(DT)\_U\_($P(IBNDU1,U)-$P(IBNDU1,U,2))  ;S PRCASV("FY")=$P(IBNDU1,U,9)\_U\_$S($P(IBNDU1,U,2)]"":($P(IBNDU1,U,10)-$P(IBNDU1,U,2)),1:$P(IBNDU1,U,10))\_$S($P(IBNDU1,U,11)]"":U\_$P(IBNDU1,U,11)\_U\_$P(IBNDU1,U,12),1:"") PLUS I IBWHO="i",$P(IBNDM,"^",2),$D(^DIC(36,$P(IBNDM,"^",2),0)) S PRCASV("2NDINS")=$P(IBNDM,"^",2)  I IBWHO="i",$P(IBNDM,"^",3),$D(^DIC(36,$P(IBNDM,"^",3),0)) S PRCASV("3RDINS")=$P(IBNDM,"^",3)  ;  N IBX S IBX=$P(IBND0,U,21),IBX=$S(IBX="P":"I1",IBX="S":"I2",IBX="T":"I3",1:"") Q:IBX=""  N IBNDI1  Q:'$D(^DGCR(399,IBIFN,IBX)) S IBNDI1=^(IBX)  S:$P(IBNDI1,"^",3)]"" PRCASV("GPNO")=$P(IBNDI1,"^",3)  S:$P(IBNDI1,"^",15)]"" PRCASV("GPNM")=$P(IBNDI1,"^",15)  S:$P(IBNDI1,"^",17)]"" PRCASV("INPA")=$P(IBNDI1,"^",17)  S:$P(IBNDI1,"^",2)]"" PRCASV("IDNO")=$P(IBNDI1,"^",2),PRCASV("INID")=PRCASV("IDNO")  ; Check that this is a secondary or tertiary bill and insurance for previous  ; COB sequence is Medicare WNR and MRA is active --> send data elements to AR  I IBCOBN>1,$$WNRBILL^IBEFUNC(IBIFN,IBCOBN-1),$$EDIACTV^IBCEF4(2) D MRA  Q  ; MRA N IBEOB S IBEOB=0  ;  K PRCASV("MEDURE"),PRCASV("MEDCA")  ; Get EOB data  F  S IBEOB=$O(^IBM(361.1,"B",IBIFN,IBEOB)) Q:'IBEOB  D  . D MRACALC^IBCEMU2(IBEOB,IBIFN,1,.PRCASV)  Q  ;MRA  ;  ;; PREGNANCY DX CODES: V22\*\*-V24\*\*, V27\*\*-V28\*\*, 630\*\*-677\*\*  ;; FLU SHOTS PROCEDURE CODES: 90724, G0008, 90732, G0009  ; NONVAFLG(IBIFN) ; Check if Non-VA bill  ; Function returns 1 if Non-VA bill  ; IB\*2.0\*554/DRF 10/9/2015  N FLAG,PTF  S FLAG=0  I $P($G(^DGCR(399,IBIFN,"U2")),U,10)]"" S FLAG=1 ;Non-VA provider defined  S PTF=$P($G(^DGCR(399,IBIFN,0)),U,8)  I PTF,$P($G(^DGPT(PTF,0)),U,4)=1 S FLAG=1 ;PTF entry indicates Non-VA  Q FLAG | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCBB11 | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** |  | | | | | | | | |
| **Related Options** | None | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  |  | | | | |  | | | |
| **Data Dictionary (DD) References** | None | | | | | | | | |
| **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** | | | | | | | | | |
| IBCBB11 ;ALB/AAS/OIFO-BP/PIJ - CONTINUATION OF EDIT CHECK ROUTINE ;12 Jun 2006 3:45 PM  ;;2.0;INTEGRATED BILLING;\*\*51,343,363,371,395,392,401,384,400,436,432,516,550\*\*;21-MAR-94;Build 25  ;;Per VA Directive 6402, this routine should not be modified.  ; WARN(IBDISP) ; Set warning in global  ; DISP = warning text to display  ;  N Z  S Z=+$O(^TMP($J,"BILL-WARN",""),-1)  I Z=0 S ^TMP($J,"BILL-WARN",1)=$J("",5)\_"\*\*Warnings\*\*:",Z=1  S Z=Z+1,^TMP($J,"BILL-WARN",Z)=$J("",5)\_IBDISP  Q  ; MULTDIV(IBIFN,IBND0) ; Check for multiple divisions on a bill ien IBIFN  ; IBND0 = 0-node of bill  ;  ; Function returns 1 if more than 1 division found on bill  N Z,Z0,Z1,MULT  S MULT=0,Z1=$P(IBND0,U,22)  I Z1 D  . S Z=0 F  S Z=$O(^DGCR(399,IBIFN,"RC",Z)) Q:'Z  S Z0=$P(^(Z,0),U,7) I Z0,Z0'=Z1 S MULT=1 Q  . S Z=0 F  S Z=$O(^DGCR(399,IBIFN,"CP",Z)) Q:'Z  S Z0=$P(^(Z,0),U,6) I Z0,Z0'=Z1 S MULT=2 Q  I 'Z1 S MULT=3  Q MULT  ;  ;; PREGNANCY DX CODES: V22\*\*-V24\*\*, V27\*\*-V28\*\*, 630\*\*-677\*\*  ;; FLU SHOTS PROCEDURE CODES: 90724, G0008, 90732, G0009  ; NPICHK ; Check for required NPIs  N IBNPIS,IBNONPI,IBNPIREQ,Z,IBNFI,IBTF,IBWC,IBXSAVE,IBPRV,IBLINE  ;\*\*\* pij start IB\*20\*436 \*\*\*  N IBRATYPE,IBLEGAL  S (IBRATYPE,IBLEGAL)=""  S IBRATYPE=$P($G(^DGCR(399,IBIFN,0)),U,7)  ; Legal types for this use.  ; 7=NO FAULT INS.  ; 10=TORT FEASOR  ; 11=WORKERS' COMP.  S IBNFI=$O(^DGCR(399.3,"B","NO FAULT INS.",0)) S:'IBNFI IBNFI=7  S IBTF=$O(^DGCR(399.3,"B","TORT FEASOR",0)) S:'IBTF IBTF=10  S IBWC=$O(^DGCR(399.3,"B","WORKERS' COMP.",0)) S:'IBWC IBWC=11  ;  I IBRATYPE=IBNFI!(IBRATYPE=IBTF)!(IBRATYPE=IBWC) D  . ; One of the legal types - force local print  . S IBLEGAL=1  ;\*\*\* pij end \*\*\*  S IBNPIREQ=$$NPIREQ^IBCEP81(DT) ; Check if NPI is required  ; Check providers  ; IB\*2.0\*432 changed the NPI check to the new Provider Array  ;S IBNPIS=$$PROVNPI^IBCEF73A(IBIFN,.IBNONPI)  D ALLIDS^IBCEFP(IBIFN,.IBXSAVE,1)  S IBPRV=""  F  S IBPRV=$O(IBXSAVE("PROVINF",IBIFN,"C",1,IBPRV)) Q:'IBPRV  D  . I $P($G(IBXSAVE("PROVINF",IBIFN,"C",1,IBPRV,0)),U,4)="" S IBNONPI(IBPRV)=""  S IBLINE=""  F  S IBLINE=$O(IBXSAVE("L-PROV",IBIFN,IBLINE)) Q:'IBLINE  D  . S IBPRV=""  . F  S IBPRV=$O(IBXSAVE("L-PROV",IBIFN,IBLINE,"C",1,IBPRV)) Q:IBPRV=""  D  .. I $P($G(IBXSAVE("L-PROV",IBIFN,IBLINE,"C",1,IBPRV,0)),U,4)="" S IBNONPI(IBPRV)=""  I $D(IBNONPI) S IBPRV="" F  S IBPRV=$O(IBNONPI(IBPRV)) Q:'IBPRV  D  . S IBER=IBER\_"IB"\_(140+IBPRV)\_";" Q  ; If required, set error IB\*2\*516  ; Check organizations  S IBNONPI=""  S IBNPIS=$$ORGNPI^IBCEF73A(IBIFN,.IBNONPI)  I $L(IBNONPI) F Z=1:1:$L(IBNONPI,U) D  . S IBER=IBER\_$P("IB339;^IB340;^IB341;",U,$P(IBNONPI,U,Z)) ; DEM;432 Added NPI errors.  Q  ; TAXCHK ; Check for required taxonomies  N IBDT,IBLINE,IBNOTAX,IBPRV,IBTAXS,IBXSAVE,Z  ;  ; MRD;IB\*2.0\*516 - This check is now moot; 'today' is always on or  ; after May 23, 2008, so taxonomy codes are always required  ; for certain providers.  ;S IBTAXREQ=$$TAXREQ^IBCEP81(DT) ; Check if taxonomy is required  ;  ; Check providers  ; IB\*2.0\*432 changed the Taxonomy check to the new Provider Array  ;S IBTAXS=$$PROVTAX^IBCEF73A(IBIFN,.IBNOTAX)  D ALLIDS^IBCEFP(IBIFN,.IBXSAVE,1)  S IBPRV=""  F  S IBPRV=$O(IBXSAVE("PROVINF",IBIFN,"C",1,IBPRV)) Q:'IBPRV  D  . I $G(IBXSAVE("PROVINF",IBIFN,"C",1,IBPRV,"TAXONOMY"))="" S IBNOTAX(IBPRV)=""  . Q  ;  S IBLINE=""  F  S IBLINE=$O(IBXSAVE("L-PROV",IBIFN,IBLINE)) Q:'IBLINE  D  . S IBPRV=""  . F  S IBPRV=$O(IBXSAVE("L-PROV",IBIFN,IBLINE,"C",1,IBPRV)) Q:IBPRV=""  D  . . I $G(IBXSAVE("L-PROV",IBIFN,IBLINE,"C",1,IBPRV,"TAXONOMY"))="" S IBNOTAX(IBPRV)=""  . . Q  . Q  ;  ; IB251 = Referring provider taxonomy missing.  ; IB253 = Rendering provider taxonomy missing.  ; IB254 = Attending provider taxonomy missing.  ;  I $D(IBNOTAX) S IBPRV="" F  S IBPRV=$O(IBNOTAX(IBPRV)) Q:'IBPRV  D  . ; Only Referring, Rendering and Attending are currently sent to the payer  . ;I IBTAXREQ,"134"[IBPRV S IBER=IBER\_"IB"\_(250+IBPRV)\_";" Q ; MRD;IB\*2.0\*516 - Always required.  . I "134"[IBPRV S IBER=IBER\_"IB"\_(250+IBPRV)\_";" Q  ; If required, set error and quit  . D WARN("Taxonomy for the "\_$P("referring^operating^rendering^attending^supervising^^^^other",U,IBPRV)\_" provider has no value") ; Else, set warning  . Q  ;  ; Check organizations. The function ORGTAX will set IBNOTAX to be a  ; list of entities missing taxonomy codes, if any (n, n^m, n^m^p,  ; where each 1 is service facility, 2 is non-VA service facility and  ; 3 is billing provider.  ;  S IBNOTAX=""  S IBTAXS=$$ORGTAX^IBCEF73A(IBIFN,.IBNOTAX)  I $L(IBNOTAX) F Z=1:1:$L(IBNOTAX,U) D  . ; IB167 = Billing Provider taxonomy missing.  . ;I IBTAXREQ,$P(IBNOTAX,U,Z)=3 S IBER=IBER\_"IB167;" Q ; MRD;IB\*2.0\*516 - Always required.  . I $P(IBNOTAX,U,Z)=3 S IBER=IBER\_"IB167;" Q  . ; MRD;IB\*2.0\*516 - Remove warning message for missing taxonomy code for lab or facility.  . ; D WARN("Taxonomy for the "\_$P("Service Facility^Non-VA Service Facility^Billing Provider",U,$P(IBNOTAX,U,Z))\_" has no value") ; Else, set warning  . Q  ;  Q  ; VALNDC(IBIFN,IBDFN) ; IB\*2\*363 - validate NDC# between PRESCRIPTION file (#52)  ; and IB BILL/CLAIMS PRESCRIPTION REFILL file (#362.4)  ; input - IBIFN = internal entry number of the billing record in the BILL/CLAIMS file (#399)  ; IBDFN = internal entry number of patient record in the PATIENT file (#2)  N IBX,IBRXCOL  ; call program that determines if NDC differences exist  D VALNDC^IBEFUNC3(IBIFN,IBDFN,.IBRXCOL)  Q:'$D(IBRXCOL)  ; at least one RX on the IB record has an NDC discrepancy   S IBX=0 F  S IBX=$O(IBRXCOL(IBX)) Q:'IBX  D WARN("NDC# on Bill does not equal the NDC# on Rx "\_IBRXCOL(IBX))  Q  ; PRIIDCHK ; Check for required Pimarary ID (SSN/EIN)  ; If the provider is on the claim, he must have one  ;   N IBI,IBZ  I $$TXMT^IBCEF4(IBIFN) D  . D F^IBCEF("N-ALL ATT/REND PROV SSN/EI","IBZ",,IBIFN)  . S IBI="" F  S IBI=$O(^DGCR(399,IBIFN,"PRV","B",IBI)) Q:IBI=""  D  .. I $P(IBZ,U,IBI)="" S IBER=IBER\_$S(IBI=1:"IB151;",IBI=2:"IB152;",IBI=3!(IBI=4):"IB321;",IBI=5:"IB153;",IBI=9:"IB154;",1:"")  Q  ; RXNPI(IBIFN) ; check for multiple pharmacy npi's on the same bill  N IBORG,IBRXNPI,IBX,IBY  S IBORG=$$RXSITE^IBCEF73A(IBIFN,.IBORG)  S IBX=0 F  S IBX=$O(IBORG(IBX)) Q:'IBX  S IBY=0 F  S IBY=$O(IBORG(IBX,IBY)) Q:'IBY  S IBRXNPI(+IBORG(IBX,IBY))=""  S (IBX,IBY)=0 F  S IBX=$O(IBRXNPI(IBX)) Q:'IBX  S IBY=IBY+1  I IBY>1 D WARN("Bill has prescriptions resulting from "\_IBY\_" different NPI locations")  Q  ; ROICHK(IBIFN,IBDFN,IBINS) ; IB\*2.0\*384 - check prescriptions that contain the  ; SENSITIVE DIAGNOSIS DRUG field #87 in the DRUG File #50 set to 1 against  ; the Claims Tracking ROI file (#356.25) to see if an ROI is on file  ; input - IBIFN = IEN of the Bill/Claims file (#399)  ; IBDFN = IEN of the patient  ; IBINS = IEN of the payer insurance company (#36)  ; OUTPUT - 0 = no error   ; 1 = a prescription is sensitive and there is no ROI on file  ;  N IBX,IBY0,IBRXIEN,IBDT,IBDRUG,ROIQ  S ROIQ=0  S IBX=0 F  S IBX=$O(^IBA(362.4,"C",IBIFN,IBX)) Q:'IBX  D  .S IBY0=^IBA(362.4,IBX,0),IBRXIEN=$P(IBY0,U,5) I 'IBRXIEN Q  .S IBDT=$P(IBY0,U,3),IBDRUG=$P(IBY0,U,4)  .D ZERO^IBRXUTL(IBDRUG)  .I $$SENS^IBNCPDR(IBDRUG) D  ; Sensitive Diagnosis Drug - check for ROI  .. I $$ROI^IBNCPDR4(IBDFN,IBDRUG,IBINS,IBDT) Q  ;ROI is on file  .. D WARN("ROI not on file for prescription "\_$$RXAPI1^IBNCPUT1(IBRXIEN,.01,"E"))  .. S ROIQ=1 ROICHKQ ;  K ^TMP($J,"IBDRUG")  Q ROIQ  ; AMBCK(IBIFN) ; IB\*2.0\*432 - if ambulance location defined, address must be defined  ; if there is anything entered in any of the address fields (either p/up or drop/off fields), than there needs to be:   ; Address 1, State and ZIP unless the State is not a US state or possession, then zip code is not needed (CMS1500 only)  ; input - IBIFN = IEN of the Bill/Claims file (#399)  ; OUTPUT - 0 = no error   ; 1 = Error  ;  N IBPAMB,IBDAMB,IBAMBR,IBCK  S IBAMBR=0  Q:$$INSPRF^IBCEF(IBIFN)'=0 IBAMBR  S IBPAMB=$G(^DGCR(399,IBIFN,"U5")),IBDAMB=$G(^DGCR(399,IBIFN,"U6"))  S IBCK(5)=$$NOPUNCT^IBCEF($P(IBPAMB,U,2,6),1),IBCK(6)=$$NOPUNCT^IBCEF($P(IBDAMB,U,1,6),1)  I IBCK(5)="",IBCK(6)="" Q IBAMBR  ; at this point we know that at least one ambulance field has data, so check to see if all have data  I IBCK(5)'="" F I=2,4,5 I $P(IBPAMB,U,I)="" S IBAMBR=1  I IBCK(6)'="" F I=1,2,4,5 I $P(IBDAMB,U,I)="" S IBAMBR=1  Q:IBAMBR=1 IBAMBR  ; now check zip code. OK to be null if state is not a US Posession  F I="IBPAMB","IBDAMB" I $P(I,U,5)'="",$P($G(^DIC(5,$P(I,U,5),0)),U,6)=1,$P(I,U,6)="" S IBAMBR=1  Q IBAMBR  ; COBAMT(IBIFN) ; IB\*2.0\*432 - IF there is a COB amt. it must equal the Total Claim Charge Amount  ; input - IBIFN = IEN of the Bill/Claims file (#399)  ; OUTPUT - 0 = no error   ; 1 = Error  ;  Q:IBIFN="" 0  Q:$P($G(^DGCR(399,IBIFN,"U4")),U)="" 0  Q:+$P($G(^DGCR(399,IBIFN,"U1")),U)'=+$P($G(^DGCR(399,IBIFN,"U4")),U) 1  Q 0  ; COBMRA(IBIFN) ; IB\*2.0\*432 - If there is a 'COB total non-covered amount' (File#399, Field#260),   ; Primary Insurance must be Medicare that never went to Medicare, and this must be a 2ndary or tertiary claim  ; input - IBIFN = IEN of the Bill/Claims file (#399)  ; OUTPUT - 0 = no error   ; 1 = Error  ;  N IBP  Q:IBIFN="" 0  Q:$P($G(^DGCR(399,IBIFN,"U4")),U)="" 0  S IBP=$P($G(^DGCR(399,IBIFN,"M1")),U,5) S:IBP="" IBP=IBIFN  I $$WNRBILL^IBEFUNC(IBIFN,1),$P($G(^DGCR(399,IBP,"S")),U,7)="",$$COBN^IBCEF(IBIFN)>1 Q 0  Q 1  ; COBSEC(IBIFN) ; IB\*2.0\*432 - If there is NOT a 'COB total non-covered amount' (File#399, Field#260),   ; and Primary Insurance is Medicare that never went to Medicare, 2ndary or tertiary claim cannot be set to transmit  ; input - IBIFN = IEN of the Bill/Claims file (#399)  ; OUTPUT - 0 = no error   ; 1 = Error  ;  N IBP  Q:IBIFN="" 0  Q:$P($G(^DGCR(399,IBIFN,"U4")),U)'="" 0  Q:$$COBN^IBCEF(IBIFN)<2 0  S IBP=$P($G(^DGCR(399,IBIFN,"M1")),U,5) S:IBP="" IBP=IBIFN  I $$WNRBILL^IBEFUNC(IBIFN,1),$P($G(^DGCR(399,IBP,"S")),U,7)="",$P($G(^DGCR(399,IBIFN,"TX")),U,8)'=1 Q 1  Q 0  ; TMCK(IBIFN) ; IB\*2.0\*432 - Attachment Control Number - REQUIRED when Transmission Method = BM, EL, EM, or FT  ; input - IBIFN = IEN of the Bill/Claims file (#399)  ; OUTPUT - 0 = no error   ; 1 = Error  ;  N IBAC  Q:IBIFN="" 0  F I=1,3 S IBAC(I)=$P($G(^DGCR(399,IBIFN,"U8")),U,I)  Q:IBAC(3)="" 0  Q:IBAC(1)'="" 0  Q:IBAC(3)="AA" 0  Q 1  ; ACCK(IBIFN) ; IB\*2.0\*432 If any of the loop info is present, then Report Type & Transmission Method req'd  ; input - IBIFN = IEN of the Bill/Claims file (#399)  ; OUTPUT - 0 = no error   ; 1 = Error  ;  N IBAC  Q:IBIFN="" 0  F I=1:1:3 S IBAC(I)=$P($G(^DGCR(399,IBIFN,"U8")),U,I)  ; All fields null, no error  I IBAC(1)="",IBAC(2)="",IBAC(3)="" Q 0  ; Both required fields complete, no error  I IBAC(2)'="",IBAC(3)'="" Q 0  ; At this point, one of the 2 required fields has data and one does not, so error  Q 1  ; LNTMCK(IBIFN) ; DEM;IB\*2.0\*432 - (Line Level) Attachment Control Number - REQUIRED when Transmission Method = BM, EL, EM, or FT  ; input - IBIFN = IEN of the Bill/Claims file (#399)  ; OUTPUT - IBLNERR = 0 = no error   ; IBLNERR = 1 = Error  ;  N IBAC,IBPROCP,I,IBLNERR  S IBLNERR=0 ; DEM;432 - Initialize error flag IBLNERR to '0' for no errors.  Q:IBIFN="" IBLNERR  S IBPROCP=0 F  S IBPROCP=$O(^DGCR(399,IBIFN,"CP",IBPROCP)) Q:'IBPROCP  D  Q:IBLNERR  . Q:'($D(^DGCR(399,IBIFN,"CP",IBPROCP,0))#10) ; DEM;432 - Node '0' is procedure node.  . Q:'($D(^DGCR(399,IBIFN,"CP",IBPROCP,1))#10) ; DEM;432 - Node '1' is line level Attachment Control fields.  . F I=1,3 S IBAC(I)=$P(^DGCR(399,IBIFN,"CP",IBPROCP,1),U,I)  . I IBAC(3)="" S IBLNERR=0 Q  . I IBAC(1)'="" S IBLNERR=0 Q  . I (IBAC(3)="AA") S IBLNERR=0 Q  . S IBLNERR=1  . Q  ;  Q IBLNERR  ; LNACCK(IBIFN) ; DEM;IB\*2.0\*432 (Line Level) If any of the loop info is present, then Report Type & Transmission Method req'd  ; input - IBIFN = IEN of the Bill/Claims file (#399)  ; OUTPUT - IBLNERR = 0 = no error   ; IBLNERR = 1 = Error  ;  N IBAC,IBPROCP,I,IBLNERR  S IBLNERR=0 ; DEM;432 - Initialize error flag IBLNERR to '0' for no errors.  Q:IBIFN="" IBLNERR  S IBPROCP=0 F  S IBPROCP=$O(^DGCR(399,IBIFN,"CP",IBPROCP)) Q:'IBPROCP  D  Q:IBLNERR  . Q:'($D(^DGCR(399,IBIFN,"CP",IBPROCP,0))#10) ; DEM;432 - Node '0' is procedure node.  . Q:'($D(^DGCR(399,IBIFN,"CP",IBPROCP,1))#10) ; DEM;432 - Node '1' is line level Attachment Control fields.  . F I=1:1:3 S IBAC(I)=$P(^DGCR(399,IBIFN,"CP",IBPROCP,1),U,I)  . ; All fields null, no error  . I IBAC(1)="",IBAC(2)="",IBAC(3)="" S IBLNERR=0 Q  . ; Both required fields complete, no error  . I IBAC(2)'="",IBAC(3)'="" S IBLNERR=0 Q  . ; At this point, one of the 2 required fields has data and one does not, so error  . S IBLNERR=1  . Q  ;  Q IBLNERR | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCBB11 ;ALB/AAS/OIFO-BP/PIJ - CONTINUATION OF EDIT CHECK ROUTINE ;12 Jun 2006 3:45 PM  ;;2.0;INTEGRATED BILLING;\*\*51,343,363,371,395,392,401,384,400,436,432,516,550,577\*\*;21-MAR-94;Build 16  ;;Per VA Directive 6402, this routine should not be modified.  ; WARN(IBDISP) ; Set warning in global  ; DISP = warning text to display  ;  N Z  S Z=+$O(^TMP($J,"BILL-WARN",""),-1)  I Z=0 S ^TMP($J,"BILL-WARN",1)=$J("",5)\_"\*\*Warnings\*\*:",Z=1  S Z=Z+1,^TMP($J,"BILL-WARN",Z)=$J("",5)\_IBDISP  Q  ; MULTDIV(IBIFN,IBND0) ; Check for multiple divisions on a bill ien IBIFN  ; IBND0 = 0-node of bill  ;  ; Function returns 1 if more than 1 division found on bill  N Z,Z0,Z1,MULT  S MULT=0,Z1=$P(IBND0,U,22)  I Z1 D  . S Z=0 F  S Z=$O(^DGCR(399,IBIFN,"RC",Z)) Q:'Z  S Z0=$P(^(Z,0),U,7) I Z0,Z0'=Z1 S MULT=1 Q  . S Z=0 F  S Z=$O(^DGCR(399,IBIFN,"CP",Z)) Q:'Z  S Z0=$P(^(Z,0),U,6) I Z0,Z0'=Z1 S MULT=2 Q  I 'Z1 S MULT=3  Q MULT  ;  ;; PREGNANCY DX CODES: V22\*\*-V24\*\*, V27\*\*-V28\*\*, 630\*\*-677\*\*  ;; FLU SHOTS PROCEDURE CODES: 90724, G0008, 90732, G0009  ; NPICHK ; Check for required NPIs  N IBNPIS,IBNONPI,IBNPIREQ,Z,IBNFI,IBTF,IBWC,IBXSAVE,IBPRV,IBLINE  ;\*\*\* pij start IB\*20\*436 \*\*\*  N IBRATYPE,IBLEGAL  S (IBRATYPE,IBLEGAL)=""  S IBRATYPE=$P($G(^DGCR(399,IBIFN,0)),U,7)  ; Legal types for this use.  ; 7=NO FAULT INS.  ; 10=TORT FEASOR  ; 11=WORKERS' COMP.  S IBNFI=$O(^DGCR(399.3,"B","NO FAULT INS.",0)) S:'IBNFI IBNFI=7  S IBTF=$O(^DGCR(399.3,"B","TORT FEASOR",0)) S:'IBTF IBTF=10  S IBWC=$O(^DGCR(399.3,"B","WORKERS' COMP.",0)) S:'IBWC IBWC=11  ;  I IBRATYPE=IBNFI!(IBRATYPE=IBTF)!(IBRATYPE=IBWC) D  . ; One of the legal types - force local print  . S IBLEGAL=1  ;\*\*\* pij end \*\*\*  S IBNPIREQ=$$NPIREQ^IBCEP81(DT) ; Check if NPI is required  ; Check providers  ; IB\*2.0\*432 changed the NPI check to the new Provider Array  ;S IBNPIS=$$PROVNPI^IBCEF73A(IBIFN,.IBNONPI)  D ALLIDS^IBCEFP(IBIFN,.IBXSAVE,1)  S IBPRV=""  F  S IBPRV=$O(IBXSAVE("PROVINF",IBIFN,"C",1,IBPRV)) Q:'IBPRV  D  . I $P($G(IBXSAVE("PROVINF",IBIFN,"C",1,IBPRV,0)),U,4)="" S IBNONPI(IBPRV)=""  S IBLINE=""  F  S IBLINE=$O(IBXSAVE("L-PROV",IBIFN,IBLINE)) Q:'IBLINE  D  . S IBPRV=""  . F  S IBPRV=$O(IBXSAVE("L-PROV",IBIFN,IBLINE,"C",1,IBPRV)) Q:IBPRV=""  D  .. I $P($G(IBXSAVE("L-PROV",IBIFN,IBLINE,"C",1,IBPRV,0)),U,4)="" S IBNONPI(IBPRV)=""  I $D(IBNONPI) S IBPRV="" F  S IBPRV=$O(IBNONPI(IBPRV)) Q:'IBPRV  D  . S IBER=IBER\_"IB"\_(140+IBPRV)\_";" Q  ; If required, set error IB\*2\*516  ; Check organizations  S IBNONPI=""  S IBNPIS=$$ORGNPI^IBCEF73A(IBIFN,.IBNONPI)  I $L(IBNONPI) F Z=1:1:$L(IBNONPI,U) D  . S IBER=IBER\_$P("IB339;^IB340;^IB341;",U,$P(IBNONPI,U,Z)) ; DEM;432 Added NPI errors.  Q  ; TAXCHK ; Check for required taxonomies  N IBDT,IBLINE,IBNOTAX,IBPRV,IBTAXS,IBXSAVE,Z  ;  ; MRD;IB\*2.0\*516 - This check is now moot; 'today' is always on or  ; after May 23, 2008, so taxonomy codes are always required  ; for certain providers.  ;S IBTAXREQ=$$TAXREQ^IBCEP81(DT) ; Check if taxonomy is required  ;  ; Check providers  ; IB\*2.0\*432 changed the Taxonomy check to the new Provider Array  ;S IBTAXS=$$PROVTAX^IBCEF73A(IBIFN,.IBNOTAX)  D ALLIDS^IBCEFP(IBIFN,.IBXSAVE,1)  S IBPRV=""  F  S IBPRV=$O(IBXSAVE("PROVINF",IBIFN,"C",1,IBPRV)) Q:'IBPRV  D  . I $G(IBXSAVE("PROVINF",IBIFN,"C",1,IBPRV,"TAXONOMY"))="" S IBNOTAX(IBPRV)=""  . Q  ;  S IBLINE=""  F  S IBLINE=$O(IBXSAVE("L-PROV",IBIFN,IBLINE)) Q:'IBLINE  D  . S IBPRV=""  . F  S IBPRV=$O(IBXSAVE("L-PROV",IBIFN,IBLINE,"C",1,IBPRV)) Q:IBPRV=""  D  . . I $G(IBXSAVE("L-PROV",IBIFN,IBLINE,"C",1,IBPRV,"TAXONOMY"))="" S IBNOTAX(IBPRV)=""  . . Q  . Q  ;  ; IB251 = Referring provider taxonomy missing.  ; IB253 = Rendering provider taxonomy missing.  ; IB254 = Attending provider taxonomy missing.  ;  I $D(IBNOTAX) S IBPRV="" F  S IBPRV=$O(IBNOTAX(IBPRV)) Q:'IBPRV  D  . ; Only Referring, Rendering and Attending are currently sent to the payer  . ;I IBTAXREQ,"134"[IBPRV S IBER=IBER\_"IB"\_(250+IBPRV)\_";" Q ; MRD;IB\*2.0\*516 - Always required.  . I "134"[IBPRV S IBER=IBER\_"IB"\_(250+IBPRV)\_";" Q  ; If required, set error and quit  . D WARN("Taxonomy for the "\_$P("referring^operating^rendering^attending^supervising^^^^other",U,IBPRV)\_" provider has no value") ; Else, set warning  . Q  ;  ; Check organizations. The function ORGTAX will set IBNOTAX to be a  ; list of entities missing taxonomy codes, if any (n, n^m, n^m^p,  ; where each 1 is service facility, 2 is non-VA service facility and  ; 3 is billing provider.  ;  S IBNOTAX=""  S IBTAXS=$$ORGTAX^IBCEF73A(IBIFN,.IBNOTAX)  I $L(IBNOTAX) F Z=1:1:$L(IBNOTAX,U) D  . ; IB167 = Billing Provider taxonomy missing.  . ;I IBTAXREQ,$P(IBNOTAX,U,Z)=3 S IBER=IBER\_"IB167;" Q ; MRD;IB\*2.0\*516 - Always required.  . I $P(IBNOTAX,U,Z)=3 S IBER=IBER\_"IB167;" Q  . ; MRD;IB\*2.0\*516 - Remove warning message for missing taxonomy code for lab or facility.  . ; D WARN("Taxonomy for the "\_$P("Service Facility^Non-VA Service Facility^Billing Provider",U,$P(IBNOTAX,U,Z))\_" has no value") ; Else, set warning  . Q  ;  Q  ; VALNDC(IBIFN,IBDFN) ; IB\*2\*363 - validate NDC# between PRESCRIPTION file (#52)  ; and IB BILL/CLAIMS PRESCRIPTION REFILL file (#362.4)  ; input - IBIFN = internal entry number of the billing record in the BILL/CLAIMS file (#399)  ; IBDFN = internal entry number of patient record in the PATIENT file (#2)  N IBX,IBRXCOL  ; call program that determines if NDC differences exist  D VALNDC^IBEFUNC3(IBIFN,IBDFN,.IBRXCOL)  Q:'$D(IBRXCOL)  ; at least one RX on the IB record has an NDC discrepancy   S IBX=0 F  S IBX=$O(IBRXCOL(IBX)) Q:'IBX  D WARN("NDC# on Bill does not equal the NDC# on Rx "\_IBRXCOL(IBX))  Q  ; PRIIDCHK ; Check for required Pimarary ID (SSN/EIN)  ; If the provider is on the claim, he must have one  ;   N IBI,IBZ  I $$TXMT^IBCEF4(IBIFN) D  . D F^IBCEF("N-ALL ATT/REND PROV SSN/EI","IBZ",,IBIFN)  . S IBI="" F  S IBI=$O(^DGCR(399,IBIFN,"PRV","B",IBI)) Q:IBI=""  D  .. I $P(IBZ,U,IBI)="" S IBER=IBER\_$S(IBI=1:"IB151;",IBI=2:"IB152;",IBI=3!(IBI=4):"IB321;",IBI=5:"IB153;",IBI=9:"IB154;",1:"")  Q  ; RXNPI(IBIFN) ; check for multiple pharmacy npi's on the same bill  N IBORG,IBRXNPI,IBX,IBY  S IBORG=$$RXSITE^IBCEF73A(IBIFN,.IBORG)  S IBX=0 F  S IBX=$O(IBORG(IBX)) Q:'IBX  S IBY=0 F  S IBY=$O(IBORG(IBX,IBY)) Q:'IBY  S IBRXNPI(+IBORG(IBX,IBY))=""  S (IBX,IBY)=0 F  S IBX=$O(IBRXNPI(IBX)) Q:'IBX  S IBY=IBY+1  I IBY>1 D WARN("Bill has prescriptions resulting from "\_IBY\_" different NPI locations")  Q  ; ROICHK(IBIFN,IBDFN,IBINS) ; IB\*2.0\*384 - check prescriptions that contain the  ; SENSITIVE DIAGNOSIS DRUG field #87 in the DRUG File #50 set to 1 against  ; the Claims Tracking ROI file (#356.25) to see if an ROI is on file  ; input - IBIFN = IEN of the Bill/Claims file (#399)  ; IBDFN = IEN of the patient  ; IBINS = IEN of the payer insurance company (#36)  ; OUTPUT - 0 = no error   ; 1 = a prescription is sensitive and there is no ROI on file  ;  N IBX,IBY0,IBRXIEN,IBDT,IBDRUG,ROIQ  S ROIQ=0  S IBX=0 F  S IBX=$O(^IBA(362.4,"C",IBIFN,IBX)) Q:'IBX  D  .S IBY0=^IBA(362.4,IBX,0),IBRXIEN=$P(IBY0,U,5) I 'IBRXIEN Q  .S IBDT=$P(IBY0,U,3),IBDRUG=$P(IBY0,U,4)  .D ZERO^IBRXUTL(IBDRUG)  .I $$SENS^IBNCPDR(IBDRUG) D  ; Sensitive Diagnosis Drug - check for ROI  .. I $$ROI^IBNCPDR4(IBDFN,IBDRUG,IBINS,IBDT) Q  ;ROI is on file  .. D WARN("ROI not on file for prescription "\_$$RXAPI1^IBNCPUT1(IBRXIEN,.01,"E"))  .. S ROIQ=1 ROICHKQ ;  K ^TMP($J,"IBDRUG")  Q ROIQ  ; AMBCK(IBIFN) ; IB\*2.0\*432 - if ambulance location defined, address must be defined  ; if there is anything entered in any of the address fields (either p/up or drop/off fields), than there needs to be:   ; Address 1, State and ZIP unless the State is not a US state or possession, then zip code is not needed (CMS1500 only)  ; input - IBIFN = IEN of the Bill/Claims file (#399)  ; OUTPUT - 0 = no error   ; 1 = Error  ;  N IBPAMB,IBDAMB,IBAMBR,IBCK  S IBAMBR=0  Q:$$INSPRF^IBCEF(IBIFN)'=0 IBAMBR  S IBPAMB=$G(^DGCR(399,IBIFN,"U5")),IBDAMB=$G(^DGCR(399,IBIFN,"U6"))  S IBCK(5)=$$NOPUNCT^IBCEF($P(IBPAMB,U,2,6),1),IBCK(6)=$$NOPUNCT^IBCEF($P(IBDAMB,U,1,6),1)  I IBCK(5)="",IBCK(6)="" Q IBAMBR  ; at this point we know that at least one ambulance field has data, so check to see if all have data  I IBCK(5)'="" F I=2,4,5 I $P(IBPAMB,U,I)="" S IBAMBR=1  I IBCK(6)'="" F I=1,2,4,5 I $P(IBDAMB,U,I)="" S IBAMBR=1  Q:IBAMBR=1 IBAMBR  ; now check zip code. OK to be null if state is not a US Posession  F I="IBPAMB","IBDAMB" I $P(I,U,5)'="",$P($G(^DIC(5,$P(I,U,5),0)),U,6)=1,$P(I,U,6)="" S IBAMBR=1  Q IBAMBR  ; COBAMT(IBIFN) ; IB\*2.0\*432 - IF there is a COB amt. it must equal the Total Claim Charge Amount  ; input - IBIFN = IEN of the Bill/Claims file (#399)  ; OUTPUT - 0 = no error   ; 1 = Error  ;  Q:IBIFN="" 0  Q:$P($G(^DGCR(399,IBIFN,"U4")),U)="" 0  Q:+$P($G(^DGCR(399,IBIFN,"U1")),U)'=+$P($G(^DGCR(399,IBIFN,"U4")),U) 1  Q 0  ; COBMRA(IBIFN) ; IB\*2.0\*432 - If there is a 'COB total non-covered amount' (File#399, Field#260),   ; Primary Insurance must be Medicare that never went to Medicare, and this must be a 2ndary or tertiary claim  ; input - IBIFN = IEN of the Bill/Claims file (#399)  ; OUTPUT - 0 = no error   ; 1 = Error  ;  N IBP  Q:IBIFN="" 0  Q:$P($G(^DGCR(399,IBIFN,"U4")),U)="" 0  S IBP=$P($G(^DGCR(399,IBIFN,"M1")),U,5) S:IBP="" IBP=IBIFN  I $$WNRBILL^IBEFUNC(IBIFN,1),$P($G(^DGCR(399,IBP,"S")),U,7)="",$$COBN^IBCEF(IBIFN)>1 Q 0  Q 1  ; COBSEC(IBIFN) ; IB\*2.0\*432 - If there is NOT a 'COB total non-covered amount' (File#399, Field#260),   ; and Primary Insurance is Medicare that never went to Medicare, 2ndary or tertiary claim cannot be set to transmit  ; input - IBIFN = IEN of the Bill/Claims file (#399)  ; OUTPUT - 0 = no error   ; 1 = Error  ;  N IBP  Q:IBIFN="" 0  Q:$P($G(^DGCR(399,IBIFN,"U4")),U)'="" 0  Q:$$COBN^IBCEF(IBIFN)<2 0  S IBP=$P($G(^DGCR(399,IBIFN,"M1")),U,5) S:IBP="" IBP=IBIFN  I $$WNRBILL^IBEFUNC(IBIFN,1),$P($G(^DGCR(399,IBP,"S")),U,7)="",$P($G(^DGCR(399,IBIFN,"TX")),U,8)'=1 Q 1  Q 0  ; TMCK(IBIFN) ; IB\*2.0\*432 - Attachment Control Number - REQUIRED when Transmission Method = BM, EL, EM, or FT  ; input - IBIFN = IEN of the Bill/Claims file (#399)  ; OUTPUT - 0 = no error   ; 1 = Error  ;  N IBAC  Q:IBIFN="" 0  F I=1,3 S IBAC(I)=$P($G(^DGCR(399,IBIFN,"U8")),U,I)  Q:IBAC(3)="" 0  Q:IBAC(1)'="" 0  Q:IBAC(3)="AA" 0  Q 1  ; ACCK(IBIFN) ; IB\*2.0\*432 If any of the loop info is present, then Report Type & Transmission Method req'd  ; input - IBIFN = IEN of the Bill/Claims file (#399)  ; OUTPUT - 0 = no error   ; 1 = Error  ;  N IBAC  Q:IBIFN="" 0  F I=1:1:3 S IBAC(I)=$P($G(^DGCR(399,IBIFN,"U8")),U,I)  ; All fields null, no error  I IBAC(1)="",IBAC(2)="",IBAC(3)="" Q 0  ; Both required fields complete, no error  I IBAC(2)'="",IBAC(3)'="" Q 0  ; At this point, one of the 2 required fields has data and one does not, so error  Q 1  ; LNTMCK(IBIFN) ; DEM;IB\*2.0\*432 - (Line Level) Attachment Control Number - REQUIRED when Transmission Method = BM, EL, EM, or FT  ; input - IBIFN = IEN of the Bill/Claims file (#399)  ; OUTPUT - IBLNERR = 0 = no error   ; IBLNERR = 1 = Error  ;  N IBAC,IBPROCP,I,IBLNERR  S IBLNERR=0 ; DEM;432 - Initialize error flag IBLNERR to '0' for no errors.  Q:IBIFN="" IBLNERR  S IBPROCP=0 F  S IBPROCP=$O(^DGCR(399,IBIFN,"CP",IBPROCP)) Q:'IBPROCP  D  Q:IBLNERR  . Q:'($D(^DGCR(399,IBIFN,"CP",IBPROCP,0))#10) ; DEM;432 - Node '0' is procedure node.  . Q:'($D(^DGCR(399,IBIFN,"CP",IBPROCP,1))#10) ; DEM;432 - Node '1' is line level Attachment Control fields.  . F I=1,3 S IBAC(I)=$P(^DGCR(399,IBIFN,"CP",IBPROCP,1),U,I)  . I IBAC(3)="" S IBLNERR=0 Q  . I IBAC(1)'="" S IBLNERR=0 Q  . I (IBAC(3)="AA") S IBLNERR=0 Q  . S IBLNERR=1  . Q  ;  Q IBLNERR  ; LNACCK(IBIFN) ; DEM;IB\*2.0\*432 (Line Level) If any of the loop info is present, then Report Type & Transmission Method req'd  ; input - IBIFN = IEN of the Bill/Claims file (#399)  ; OUTPUT - IBLNERR = 0 = no error   ; IBLNERR = 1 = Error  ;  N IBAC,IBPROCP,I,IBLNERR  S IBLNERR=0 ; DEM;432 - Initialize error flag IBLNERR to '0' for no errors.  Q:IBIFN="" IBLNERR  S IBPROCP=0 F  S IBPROCP=$O(^DGCR(399,IBIFN,"CP",IBPROCP)) Q:'IBPROCP  D  Q:IBLNERR  . Q:'($D(^DGCR(399,IBIFN,"CP",IBPROCP,0))#10) ; DEM;432 - Node '0' is procedure node.  . Q:'($D(^DGCR(399,IBIFN,"CP",IBPROCP,1))#10) ; DEM;432 - Node '1' is line level Attachment Control fields.  . F I=1:1:3 S IBAC(I)=$P(^DGCR(399,IBIFN,"CP",IBPROCP,1),U,I)  . ; All fields null, no error  . I IBAC(1)="",IBAC(2)="",IBAC(3)="" S IBLNERR=0 Q  . ; Both required fields complete, no error  . I IBAC(2)'="",IBAC(3)'="" S IBLNERR=0 Q  . ; At this point, one of the 2 required fields has data and one does not, so error  . S IBLNERR=1  . Q  ;  Q IBLNERR  ;  ;vd/Beginning of IB\*2\*577 - Validate Line Level for NDC LNNDCCK(IBIFN) ;IB\*2\*577 (Line Level) The Units and Units/Basis of Measurement fields are required if the NDC field is populated.  ; INPUT - IBIFN = IEN of the Bill/Claims file (#399)  ; OUTPUT - IBLNERR = 0 = no error  ; IBLNERR = 1 = Error  ;  N IBAC,IBPROCP,I,IBLNERR  S IBLNERR=0 ; IB\*2\*577 - Initialize error flag IBLNERR to '0' for no errors.  Q:IBIFN="" IBLNERR  S IBPROCP=0 F  S IBPROCP=$O(^DGCR(399,IBIFN,"CP",IBPROCP)) Q:'IBPROCP  D  Q:IBLNERR  . Q:($$GET1^DIQ(399.0304,IBPROCP\_","\_IBIFN\_",","NDC","I")="") ; IB\*2\*577 - No NDC Code  . ; If there is an NDC Code, then the UNITS and UNITS/BASIS OF MEASUREMENT are Required.  . I $$GET1^DIQ(399.0304,IBPROCP\_","\_IBIFN\_",","UNITS/BASIS OF MEASUREMENT","I")="" S IBLNERR=1 Q  . I $$GET1^DIQ(399.0304,IBPROCP\_","\_IBIFN\_",","UNITS","I")="" S IBLNERR=1 Q  ;Units (Quantity) is required if there is an NDC Code.  . Q  ;  Q IBLNERR  ;vd/End of IB\*2\*577 | | | | | | | | | |

1. MODIFY [#364.7, 941] to properly include the UNITS/BASIS OF MEASUREMENT in the “PRF” (Professional) Segment of the 837, piece 25.

PIECE 25 Unit or Basis of Measurement Code 364.6[979]

364.7[941] 364.5[5] Length=2 Max Lines=0

>Constant Value: ""

Change the following code from this:

>K IBXDATA N Z S Z=0 F S Z=$O(IBXSAVE("OUTPT",Z)) Q:'Z I

$P(IBXSAVE("OUTPT",Z),U,16)'="" S IBXDATA(Z)="UN"

To this:

>K IBXDATA N Z S Z=0 F S Z=$O(IBXSAVE("OUTPT",Z)) Q:'Z I

$P(IBXSAVE("OUTPT",Z),U,16)'="" S IBXDATA(Z)=

$P(IBXSAVE(“OUTPT”,Z),U,17) I IBXDATA(Z)=”” S IBXDATA(Z)=”UN”

1. MODIFY [#364.7, 1950] to properly include the UNITS/BASIS OF MEASUREMENT in the “INS” (Institutional) Segment of the 837, piece 17.

PIECE 17 Unit or Basis of Measurement Code 364.6[2237]

364.7[1950] 364.5[5] Length=2 Max Lines=0

>Constant Value: ""

Change the following code from this:

>K IBXDATA N Z S Z=0 F S Z=$O(IBXSAVE("INPT",Z)) Q:'Z I

$P(IBXSAVE("INPT",Z),U,12)'="" S IBXDATA(Z)="UN"

To this:

>K IBXDATA N Z S Z=0 F S Z=$O(IBXSAVE("INPT",Z)) Q:'Z I

$P(IBXSAVE(“INPT”,Z),U,12)’=”” S IBXDATA(Z)=

$P(IBXSAVE(“INPT”,Z),U,13) I IBXDATA(Z)=”” S IBXDATA(Z)=”UN”

1. ADD a new error code (“IB360”) to the Error Code file [#350.8] for the line level check on NDC number and the UNITS/BASIS OF MEASUREMENT and the UNITS fields.

The Error Message should be: “Units & Units/Basis of Measurement are Required if NDC Code exists.”

(Refer to the INC3508 module of code in the new ^IBY577PR Pre-Install Routine.)

1. ADD the 364.7 file to the build for all of the updated OUTPUT FORMATTER entries.

Edit a Build PAGE 2 OF 5

Name: **IB\*2.0\*???**  TYPE: **SINGLE PACKAGE**

-------------------------------------------------------------------------------

File List (Name or Number)

 DD Export Options 

 

 File: IB FORM FIELD CONTENT 

 

 Send Full or Partial DD...: **FULL**  

 

Update the Data Dictionary: **NO**  Send Security Code: **NO**  

 

Screen to Determine DD Update 

 

 

 Data Comes With File...: **YES** 



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name: **IB\*2.0\*???**  TYPE: **SINGLE PACKAGE**

-------------------------------------------------------------------------------

File List (Name or Number)

 DD Export Options 

 Data Export Options 

 

 Site's Data: OVERWRITE 

 

 Resolve Pointers: **YES** May User Override Data Update: **NO**  

 

 Data List: 

 

 Screen to Select Data 

 **I $$INCLUDE^IBY???PR(7,Y)**  





\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. MODIFY the ^IBCEF11 and ^IBCF23A routines to update the 837 and printed CMS 1500 for the new UNITS/BASIS OF MEASUREMENT field.

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCEF11 | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** |  | | | | | | | | |
| **Related Options** | None | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  |  | | | | |  | | | |
| **Data Dictionary (DD) References** | None | | | | | | | | |
| **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** | | | | | | | | | |
| IBCEF11 ;ALB/TMP - FORMATTER SPECIFIC BILL FUNCTIONS - CONT ;30-JAN-96  ;;2.0;INTEGRATED BILLING;\*\*51,137,155,309,335,348,349,371,432,447,473,516\*\*;21-MAR-94;Build 123  ;;Per VA Directive 6402, this routine should not be modified.  ; BOX24D(A,IB) ; Returns the lines for boxes 19-24 of the CMS-1500 display  ; IB = flag is 1 if only box 24 is needed  Q $S('$G(IB):"36",1:"44")\_"^55"  ; RCBOX() ; Returns the lines for revenue code boxes of the UB-04 display  Q "19^41"  ; OUTPT(IBIFN,IBPRINT) ; Returns an array of service line data from  ; CMS-1500 box 24. Output is in IBXDATA(n)  ; IBPRINT = print flag 1: return print fields  ; 0: return EDI fields  ; Uses diagnosis array ^TMP("IBXSAVE",$J,"DX",IBIFN,DIAG CODE)=SEQ #  ; if it already exists. If not, it builds it from N-DIAGNOSES element  ;  ; For EDI call: Returns IBXDATA(n)=  ; begin date(YYYYMMDD) ^ end date(YYYYMMDD) ^ pos ^ tos ^  ; proc code/revenue code - if no procedure (not the pointers) ^  ; type of code ^ dx pointer(s ) ^ unit charge ^ units ^ modifiers separated by ;  ; ^ purchased charge amount ^ anesthesia minutes ^ emergency indicator ^  ; lab-type service flag ^ NDC ^ Units  ;  ; Also Returns IBXDATA(IBI,"COB",COB,m) with COB data for each line  ; item found in an accepted EOB for the bill and = the reference  ; line in the first '^' piece followed by the '0' node data of file  ; 361.115 (LINE LEVEL ADJUSTMENTS)  ; COB = COB sequence # of adjustment's ins co, m = seq #  ; -- AND --  ; IBXDATA(IBI,"COB",COB,m,z,p)=  ; the data on the '0' node for each subordinate entry of file  ; 361.11511 (REASONS) (Only first 3 pieces for 837 output)  ; z = group code, sometimes preceeded by a space p = seq #  ;  ; For Print call: Returns begin date(DDMMYYYY)^end date(DDMMYYYY) or  ; null if equal to begin date^pos^tos^bedsection name(if no procedure)  ; or procedure code(not the pointer)^ ... refer to EDI call results  ; Also, IBXDATA(n,"TEXT")=the text to print on first line of box 24,  ; If no procedure code, returns IBXDATA(n,"A")=rev code abbrev  ;  ; For both calls, returns IBXDATA(n,item type,item ptr)=""  ; -- AND --  ; IBXDATA(n,"RX")=RX#^drug name^NDC^refill #^(re)fill date^qty^days  ; ^chrge^ien of file 362.4^NDC format  ; If line references a prescription  ; -- AND --  ; If no revenue code for a prescription, returns IBXDATA(n,"ARX")=""  ; -- AND --  ; IBXDATA(n,"AUX")='AUX' node of the procedure entry  ;  ; Also returns IBXDATA(n,"CPLNK") = soft link to corresponding entry in PROCEDURES multiple of file 399  ;  N IB,IBI,IBJ,IBFLD,IBDXI,IBXIEN,Z,IBXTRA,IBRX,IBRX0,IBRX1,Z0,Z1  ;  K ^TMP($J,"IBITEM")  S ^TMP($J,"IBITEM")=""  ; Build diagnosis array if not already built  I $O(^TMP("IBXSAVE",$J,"DX",IBIFN,""))="",$O(^IBA(362.3,"AIFN"\_IBIFN,"")) D  .N Z,IBXDATA D F^IBCEF("N-DIAGNOSES",,,IBIFN)  .S Z="" F  S Z=$O(IBXDATA(Z)) K:$O(IBXDATA(0))=""&(Z="") IBXDATA Q:Z=""  S:$P(IBXDATA(Z),U,2) ^TMP("IBXSAVE",$J,"DX",IBIFN,$P(IBXDATA(Z),U,2))=Z  ;  S IB(0)=$G(^DGCR(399,IBIFN,0)),IB("U")=$G(^("U")),IB("U1")=$G(^("U1"))  S IBI="" F  S IBI=$O(^TMP("IBXSAVE",$J,"DX",IBIFN,IBI)) Q:IBI=""  S IBDXI(IBI)=^(IBI)  I '$G(IBPRINT) D RVCE^IBCF23(IBIFN,IBIFN)  I $G(IBPRINT) D RVCE^IBCF23(,IBIFN)  ; Returns IBFLD(24) = begin date ^ end date ^ pos ^ tos ^  ; proc/bedsection/revenue code ^ dx pointer ^ unit charge ^  ; units ^ modifiers ^ purchased charge amount ^ anesthesia minutes ^  ; emergency indicator ^ soft pointer to PROCEDURES multiple in file 399 ^  ; NDC ^ Units  ; IBFLD(24,n,type,item)=""  ; IBFLD(24,n\_"A") = revenue code abbreviation if no procedure  ; IBFLD(24,n,"AUX") = 'AUX' node of line item   ; IBFLD(24,n,"RX") = soft pointer to file 362.4 from 'item' fld  ; (can be null)  ;  D SET^IBCSC5A(IBIFN,.IBRX) ;prescriptions  ; IBRX1(ien 362.4)=RX#^drug ien^NDC^refil #^(re)fil date^qty^days^chrge  I IBRX S IBRX="" F  S IBRX=$O(IBRX(IBRX)) Q:IBRX=""  S IBRX0=0 F  S IBRX0=$O(IBRX(IBRX,IBRX0)) Q:'IBRX0  D  . N IBRXH  . S IBRXH=IBRX(IBRX,IBRX0)  . ; \*\*IB\*2.0\*432\*\* added \_U\_$P(IBRXH,U,9) (Rx Date) to Output Formatter  . S IBRX1(+IBRXH)=IBRX\_U\_$P(IBRXH,U,2)\_U\_$P(IBRXH,U,5)\_U\_$P(IBRXH,U,7)\_U\_IBRX0\_U\_$P(IBRXH,U,4)\_U\_$P(IBRXH,U,3)\_U\_$P(IBRXH,U,6)\_U\_+IBRXH\_U\_$P(IBRXH,U,8)\_U\_$P(IBRXH,U,9)  K IBRX  ;  ; for EDI, remove any $0 line items from the IBFLD array before   ; dropping down into the next loop (IB\*2\*371)  ; Start IB\*2.0\*447 BI - Code removed to allow 0 dollars to print.  ;I '$G(IBPRINT) D  ;. NEW IBZ,IBI,Z  ;. M IBZ=IBFLD K IBFLD  ;. S (IBI,Z)=0  ;. F S IBI=$O(IBZ(24,IBI)) Q:IBI'=+IBI D  ;.. I $P(IBZ(24,IBI),U,7)\*$P(IBZ(24,IBI),U,8)'>0 Q  ;.. S Z=Z+1  ;.. M IBFLD(24,Z)=IBZ(24,IBI)  ;.. S IBFLD(24)=Z  ;.. Q  ;. Q  ; End IB\*2.0\*447 BI  ;  S IBI=0  F  S IBI=$O(IBFLD(24,IBI)) Q:IBI'=+IBI  D  . S IBRX1=0  . S IBXDATA(IBI)=$P(IBFLD(24,IBI),U)\_U\_$P(IBFLD(24,IBI),U,$S($P(IBFLD(24,IBI),U,2)=""&'$G(IBPRINT):1,1:2))  . S $P(IBXDATA(IBI),U,3,5)=$P(IBFLD(24,IBI),U,3,5)  . S $P(IBXDATA(IBI),U,6)=$S($D(IBFLD(24,IBI\_"X")):"CJ",1:"HC")  . S $P(IBXDATA(IBI),U,7,13)=$P(IBFLD(24,IBI),U,6,12)  . S $P(IBXDATA(IBI),U,14)=+$$ISLAB(IBXDATA(IBI))  . ; MRD;IB\*2.0\*516 - Added NDC and Units to line level of claim,  . ; pieces 14 & 15 of IBFLD, pieces 15 & 16 of IBXDATA. Print  . ; in Box 24 by setting in IBXDATA(IBI,"TEXT").  . S $P(IBXDATA(IBI),U,15,16)=$P(IBFLD(24,IBI),U,14,15)  . I $P(IBFLD(24,IBI),U,14)'="" S IBXDATA(IBI,"TEXT")="N4"\_$P(IBFLD(24,IBI),U,14)\_" UN"\_$P(IBFLD(24,IBI),U,15)  . ;  . I $D(IBFLD(24,IBI,"RX")) D  ;Rx  .. S IBRX1=1  .. I $P($G(IBFLD(24,IBI,"AUX")),U,8)'="" S $P(IBFLD(24,IBI,"AUX"),U,8)="",$P(IBFLD(24,IBI,"AUX"),U,9)=""  ;No free text allowed for rx's  .. I $D(IBRX1(+IBFLD(24,IBI,"RX"))) D  Q  ;Soft link exists  ...D ZERO^IBRXUTL(+$P(IBRX1(+IBFLD(24,IBI,"RX")),U,2))  ... S IBXDATA(IBI,"RX")=IBRX1(+IBFLD(24,IBI,"RX")),$P(IBXDATA(IBI,"RX"),U,2)=$E($G(^TMP($J,"IBDRUG",+$P(IBRX1(+IBFLD(24,IBI,"RX")),U,2),.01)),1,30)  ... K IBRX1(+IBFLD(24,IBI,"RX"))  ... ; No soft link - must find the first Rx with the same charge  .. S IBRX="" F  S IBRX=$O(IBRX1(IBRX)) Q:'IBRX  I +$P(IBRX1(IBRX),U,8)=+$P(IBXDATA(IBI),U,8) D  Q  ... D ZERO^IBRXUTL(+$P(IBRX1(IBRX),U,2))  ... S IBXDATA(IBI,"RX")=IBRX1(IBRX),$P(IBXDATA(IBI,"RX"),U,2)=$E($G(^TMP($J,"IBDRUG",+$P(IBRX1(IBRX),U,2),.01)),1,30) K IBRX1(IBRX) Q  ... Q  .. Q  . ;  . ; MRD;IB\*2.0\*516 - If additional service line comments to appear in  . ; Box 24, concatenate to front if something (NDC) is already there.  . I $G(IBFLD(24,IBI,"AUX"))'="" D  .. I $G(IBPRINT),$P(IBFLD(24,IBI,"AUX"),U,8)'="" D  ... I $G(IBXDATA(IBI,"TEXT"))'="" S IBXDATA(IBI,"TEXT")=$E($P(IBFLD(24,IBI,"AUX"),U,8)\_" "\_IBXDATA(IBI,"TEXT"),1,59)  ... E  S IBXDATA(IBI,"TEXT")=$P(IBFLD(24,IBI,"AUX"),U,8)  ... S $P(IBFLD(24,IBI,"AUX"),U,8)=""  ... Q  .. S IBXDATA(IBI,"AUX")=IBFLD(24,IBI,"AUX")  .. Q  . ;  . I $G(IBPRINT) D  .. ; START IB\*2.0\*447 BI ZERO DOLLAR CHANGES  .. ; I '$P(IBXDATA(IBI),U,8),'$G(IBXDATA(IBI,"RX")) D Q  .. I $P(IBXDATA(IBI),U,8)="",'$G(IBXDATA(IBI,"RX")) D  Q  ... ; END IB\*2.0\*447 BI ZERO DOLLAR CHANGES  ... I $G(IBNOSHOW) Q    ; don't show errors/warnings  ... S IBXDATA(IBI,"TEXT")="Warning:\*\* REV CODE UNITS < #PROCEDURES, THEY MUST BE ="  ... I $D(IBXDATA(IBI,"AUX")) S $P(IBXDATA(IBI,"AUX"),U,9)=""  ... Q  .. ;  .. I $G(IBFLD(24,IBI\_"A"))'="" D  Q  ... S IBXDATA(IBI,"A")=IBFLD(24,IBI\_"A")  ... I $G(IBNOSHOW) Q    ; don't show errors/warnings  ... S IBXDATA(IBI,"TEXT")="Warning:\*\* REV CODE UNITS > #PROCEDURES, THEY MUST BE=: "\_IBFLD(24,IBI\_"A")  ... I $D(IBXDATA(IBI,"AUX")) S $P(IBXDATA(IBI,"AUX"),U,9)=""  ... Q  .. ;  .. S IBRX=$G(IBXDATA(IBI,"RX"))  .. I IBRX'="" D  ;Format Rx detail  ... N Z  ... S Z=$P(IBRX,U)  ... S Z=$S(Z'="":"Rx#"\_Z\_" ",1:"RX: ")  ... S IBXDATA(IBI,"TEXT")=Z\_$S($P(IBRX,U,3)'="":"NDC: "\_$P(IBRX,U,3),1:"NOC: "\_$P(IBRX,U,2))\_" Qty: "\_$P(IBRX,U,6)\_" Days: "\_$P(IBRX,U,7)  ... S $P(IBXDATA(IBI,"AUX"),U,9)="N4"   ; service line comment qualifier for RX's  ... Q  .. Q  . S IBXDATA(IBI,"CPLNK")=$P(IBFLD(24,IBI),U,13)  . I '$G(IBPRINT) D COBLINE^IBCEU6(IBIFN,IBI,.IBXDATA,,.IBXTRA)  . Q  ;  I $G(IBPRINT) D  . S IBRX=0 F  S IBRX=$O(IBRX1(IBRX)) Q:'IBRX  D  .. S IBI=+$O(IBXDATA(""),-1)+1  .. S IBXDATA(IBI)=$$DATE($P(IBRX1(IBRX),U,5))  .. S IBXDATA(IBI,"TEXT")="\*\*\*\* ERROR - NO PROC LINK TO REV CODE FOR DRUG: RX#: "\_$P(IBRX1(IBRX),U)\_" NDC #: "\_$P(IBRX1(IBRX),U,3)  .. I $D(IBXDATA(IBI,"AUX")) S $P(IBXDATA(IBI,"AUX"),U,9)=""  .. S IBXDATA(IBI,"ARX")=""  .. D ZERO^IBRXUTL(+$P(IBRX1(IBRX),U,2))  .. S IBXDATA(IBI,"RX")=IBRX1(IBRX),$P(IBXDATA(IBI,"RX"),U,2)=$E($G(^TMP($J,"IBDRUG",+$P(IBRX1(IBRX),U,2),.01)),1,30) K IBRX1(IBRX)  .. Q  . Q  ;  I '$G(IBPRINT),$D(IBXTRA) D COMBO^IBCEU2(.IBXDATA,.IBXTRA,0) ;Handle bundled/unbundled lines  K ^TMP($J,"IBDRUG")  Q  ; ISLAB(LDATA) ; Returns 0/1 if line item data indicates the item is a lab (1)  ; 'LAB' is defined here as type of service = 5  Q $E($P(LDATA,U,4))="5"  ; FMT(DATA,DLEN,FLEN) ; Returns a string in DATA with a max length of DLEN  ; and a field length of FLEN  Q $E($E(DATA,1,DLEN)\_$J("",FLEN),1,FLEN)  ; DATE(X,DEL) ; Returns FM date in X as MMxDDxYYYY where x=DEL  S DEL=$G(DEL)  S X=$$DATE^IBCF2(X,1,1)  I X'="" S X=$E(X,1,2)\_DEL\_$E(X,3,4)\_DEL\_$E(X,5,8)  Q X  ; BATCH() ; Sets up record for and stores/returns the next batch number  N NUM,FAC,DO,DD,DLAYGO,DIC,X,Y  ;Keep latest batch number for view/print edi bill extract data option  I $D(IBVNUM) S NUM=IBVNUM G BATCHQ  ;Check for batch resubmit - if yes, use same number as original batch  I $P($G(^TMP("IBRESUBMIT",$J)),U,3)=1 S NUM=$P(^($J),U) G BATCHQ  L +^IBA(364.1,0):5 I '$T Q 0  S FAC=+$P($$SITE^VASITE(),U,3),NUM=$O(^IBA(364.1,"B",""),-1)  I $D(^IBA(364.1,+NUM,0)),$P(^(0),U,2)="" F  D  Q:'NUM!($P($G(^IBA(364.1,+NUM,0)),U,2)'="")  . I $D(^IBA(364.1,NUM,0)) S DA=NUM,DIK="^IBA(364.1," D ^DIK  . S NUM=$O(^IBA(364.1,"B",""),-1)  F  S NUM=$S($P(NUM,FAC,2)'="":NUM+1,1:FAC\_"0000001") Q:'$D(^IBA(364.1,"B",NUM))  K DO,DD S DIC="^IBA(364.1,",DLAYGO=364.1,DIC(0)="L",X=NUM D FILE^DICN K DD,DO I Y'>0 S NUM=0  L -^IBA(364.1,0) BATCHQ Q NUM  ; GETLDAT(IBXIEN) ; Extract data for 837 transmission LDAT record  ; IBXIEN - ien in file 399  ; Sets up IBXSAVE("LDAT",n) array:  ; Attachment report type ^ Attachment report transmission code ^ Attachment control number ^   ; OB Anesthesia Additional Units ^ Purchase Service Provider ID ^ Purchase Service Amount ^  N CPIEN,FTYPE,IBXDATA,IDS,IBIDS,NODE1,PSAMNT,PSPID,Z,PCE1,LINE  I '+$G(IBXIEN) Q  K IBXSAVE("LDAT")  S FTYPE=$$FT^IBCEF(IBXIEN)  I FTYPE=2 D OUTPT(IBXIEN,0)  I FTYPE=3 D HOS^IBCEF2(IBXIEN)  D ALLIDS^IBCEFP(IBXIEN,.IDS,1)  S (PSPID,PSAMNT)=""  ; IB\*2.0\*473/TAZ - Convert PROVIDER code to function call to PSID^IBCEFP  I $$SUB1OK^IBCEP8A(IBXIEN),(FTYPE=2) D  . D PSID^IBCEFP(IBXIEN,.IDS,.IBIDS)  . S PSPID=$G(IBIDS(0)) I PSPID="" S PSPID=$P($G(IBIDS(1)),U,1)  ;IB\*2.0\*473/TAZ - END  S Z=0 F  S Z=$O(IBXDATA(Z)) Q:'Z  D  . S CPIEN=+$G(IBXDATA(Z,"CPLNK")) ;I 'CPIEN Q  . I FTYPE=2,$$SUB1OK^IBCEP8A(IBXIEN) S PSAMNT=$$DOLLAR^IBCEFG1($P($G(IBXDATA(Z)),U,11))  . S (PCE1,NODE1)=""  . I CPIEN D  . . S NODE1=$G(^DGCR(399,IBXIEN,"CP",CPIEN,1))  . . S PCE1=$$GET1^DIQ(399.0304,CPIEN\_","\_IBXIEN\_",",71)  . . Q  . ; MRD;IB\*2.0\*516 - Added addl. procedure description as piece 7   . ; of IBXSAVE, which will exist only if the procedure ends in '99'  . ; or is an 'NOC/NOS' procedure.  . S IBXSAVE("LDAT",Z)=PCE1\_U\_$P(NODE1,U,3)\_U\_$P(NODE1,U)\_U\_$P(NODE1,U,5)\_U\_$G(PSPID)\_U\_$G(PSAMNT)\_U\_$P(NODE1,U,4)  . Q  Q | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCEF11 ;ALB/TMP - FORMATTER SPECIFIC BILL FUNCTIONS - CONT ;30-JAN-96  ;;2.0;INTEGRATED BILLING;\*\*51,137,155,309,335,348,349,371,432,447,473,516,577\*\*;21-MAR-94;Build 16  ;;Per VA Directive 6402, this routine should not be modified.  ; BOX24D(A,IB) ; Returns the lines for boxes 19-24 of the CMS-1500 display  ; IB = flag is 1 if only box 24 is needed  Q $S('$G(IB):"36",1:"44")\_"^55"  ; RCBOX() ; Returns the lines for revenue code boxes of the UB-04 display  Q "19^41"  ; OUTPT(IBIFN,IBPRINT) ; Returns an array of service line data from  ; CMS-1500 box 24. Output is in IBXDATA(n)  ; IBPRINT = print flag 1: return print fields  ; 0: return EDI fields  ; Uses diagnosis array ^TMP("IBXSAVE",$J,"DX",IBIFN,DIAG CODE)=SEQ #  ; if it already exists. If not, it builds it from N-DIAGNOSES element  ;  ; For EDI call: Returns IBXDATA(n)=  ; begin date(YYYYMMDD) ^ end date(YYYYMMDD) ^ pos ^ tos ^  ; proc code/revenue code - if no procedure (not the pointers) ^  ; type of code ^ dx pointer(s ) ^ unit charge ^ units ^ modifiers separated by ;  ; ^ purchased charge amount ^ anesthesia minutes ^ emergency indicator ^  ; lab-type service flag ^ NDC ^ Units/Quantity ^ Unit/Basis of Measurement (vd/IB\*2\*577)  ;  ; Also Returns IBXDATA(IBI,"COB",COB,m) with COB data for each line  ; item found in an accepted EOB for the bill and = the reference  ; line in the first '^' piece followed by the '0' node data of file  ; 361.115 (LINE LEVEL ADJUSTMENTS)  ; COB = COB sequence # of adjustment's ins co, m = seq #  ; -- AND --  ; IBXDATA(IBI,"COB",COB,m,z,p)=  ; the data on the '0' node for each subordinate entry of file  ; 361.11511 (REASONS) (Only first 3 pieces for 837 output)  ; z = group code, sometimes preceeded by a space p = seq #  ;  ; For Print call: Returns begin date(DDMMYYYY)^end date(DDMMYYYY) or  ; null if equal to begin date^pos^tos^bedsection name(if no procedure)  ; or procedure code(not the pointer)^ ... refer to EDI call results  ; Also, IBXDATA(n,"TEXT")=the text to print on first line of box 24,  ; If no procedure code, returns IBXDATA(n,"A")=rev code abbrev  ;  ; For both calls, returns IBXDATA(n,item type,item ptr)=""  ; -- AND --  ; IBXDATA(n,"RX")=RX#^drug name^NDC^refill #^(re)fill date^qty^days  ; ^chrge^ien of file 362.4^NDC format  ; If line references a prescription  ; -- AND --  ; If no revenue code for a prescription, returns IBXDATA(n,"ARX")=""  ; -- AND --  ; IBXDATA(n,"AUX")='AUX' node of the procedure entry  ;  ; Also returns IBXDATA(n,"CPLNK") = soft link to corresponding entry in PROCEDURES multiple of file 399  ;  N IB,IBI,IBJ,IBFLD,IBDXI,IBXIEN,Z,IBXTRA,IBRX,IBRX0,IBRX1,Z0,Z1  ;  K ^TMP($J,"IBITEM")  S ^TMP($J,"IBITEM")=""  ; Build diagnosis array if not already built  I $O(^TMP("IBXSAVE",$J,"DX",IBIFN,""))="",$O(^IBA(362.3,"AIFN"\_IBIFN,"")) D  .N Z,IBXDATA D F^IBCEF("N-DIAGNOSES",,,IBIFN)  .S Z="" F  S Z=$O(IBXDATA(Z)) K:$O(IBXDATA(0))=""&(Z="") IBXDATA Q:Z=""  S:$P(IBXDATA(Z),U,2) ^TMP("IBXSAVE",$J,"DX",IBIFN,$P(IBXDATA(Z),U,2))=Z  ;  S IB(0)=$G(^DGCR(399,IBIFN,0)),IB("U")=$G(^("U")),IB("U1")=$G(^("U1"))  S IBI="" F  S IBI=$O(^TMP("IBXSAVE",$J,"DX",IBIFN,IBI)) Q:IBI=""  S IBDXI(IBI)=^(IBI)  I '$G(IBPRINT) D RVCE^IBCF23(IBIFN,IBIFN)  I $G(IBPRINT) D RVCE^IBCF23(,IBIFN)  ; Returns IBFLD(24) = begin date ^ end date ^ pos ^ tos ^  ; proc/bedsection/revenue code ^ dx pointer ^ unit charge ^  ; units ^ modifiers ^ purchased charge amount ^ anesthesia minutes ^  ; emergency indicator ^ soft pointer to PROCEDURES multiple in file 399 ^  ; NDC ^ Units  ; IBFLD(24,n,type,item)=""  ; IBFLD(24,n\_"A") = revenue code abbreviation if no procedure  ; IBFLD(24,n,"AUX") = 'AUX' node of line item   ; IBFLD(24,n,"RX") = soft pointer to file 362.4 from 'item' fld  ; (can be null)  ;  D SET^IBCSC5A(IBIFN,.IBRX) ;prescriptions  ; IBRX1(ien 362.4)=RX#^drug ien^NDC^refil #^(re)fil date^qty^days^chrge  I IBRX S IBRX="" F  S IBRX=$O(IBRX(IBRX)) Q:IBRX=""  S IBRX0=0 F  S IBRX0=$O(IBRX(IBRX,IBRX0)) Q:'IBRX0  D  . N IBRXH  . S IBRXH=IBRX(IBRX,IBRX0)  . ; \*\*IB\*2.0\*432\*\* added \_U\_$P(IBRXH,U,9) (Rx Date) to Output Formatter  . S IBRX1(+IBRXH)=IBRX\_U\_$P(IBRXH,U,2)\_U\_$P(IBRXH,U,5)\_U\_$P(IBRXH,U,7)\_U\_IBRX0\_U\_$P(IBRXH,U,4)\_U\_$P(IBRXH,U,3)\_U\_$P(IBRXH,U,6)\_U\_+IBRXH\_U\_$P(IBRXH,U,8)\_U\_$P(IBRXH,U,9)  K IBRX  ;  ; for EDI, remove any $0 line items from the IBFLD array before   ; dropping down into the next loop (IB\*2\*371)  ; Start IB\*2.0\*447 BI - Code removed to allow 0 dollars to print.  ;I '$G(IBPRINT) D  ;. NEW IBZ,IBI,Z  ;. M IBZ=IBFLD K IBFLD  ;. S (IBI,Z)=0  ;. F S IBI=$O(IBZ(24,IBI)) Q:IBI'=+IBI D  ;.. I $P(IBZ(24,IBI),U,7)\*$P(IBZ(24,IBI),U,8)'>0 Q  ;.. S Z=Z+1  ;.. M IBFLD(24,Z)=IBZ(24,IBI)  ;.. S IBFLD(24)=Z  ;.. Q  ;. Q  ; End IB\*2.0\*447 BI  ;  S IBI=0  F  S IBI=$O(IBFLD(24,IBI)) Q:IBI'=+IBI  D  . S IBRX1=0  . S IBXDATA(IBI)=$P(IBFLD(24,IBI),U)\_U\_$P(IBFLD(24,IBI),U,$S($P(IBFLD(24,IBI),U,2)=""&'$G(IBPRINT):1,1:2))  . S $P(IBXDATA(IBI),U,3,5)=$P(IBFLD(24,IBI),U,3,5)  . S $P(IBXDATA(IBI),U,6)=$S($D(IBFLD(24,IBI\_"X")):"CJ",1:"HC")  . S $P(IBXDATA(IBI),U,7,13)=$P(IBFLD(24,IBI),U,6,12)  . S $P(IBXDATA(IBI),U,14)=+$$ISLAB(IBXDATA(IBI))  . ; MRD;IB\*2.0\*516 - Added NDC and Units to line level of claim,  . ; pieces 14 & 15 of IBFLD, pieces 15 & 16 of IBXDATA. Print  . ; in Box 24 by setting in IBXDATA(IBI,"TEXT").  . ;S $P(IBXDATA(IBI),U,15,16)=$P(IBFLD(24,IBI),U,14,15)  . ;I $P(IBFLD(24,IBI),U,14)'="" S IBXDATA(IBI,"TEXT")="N4"\_$P(IBFLD(24,IBI),U,14)\_" UN"\_$P(IBFLD(24,IBI),U,15)  . ; vd/IB\*2\*577 - Added Unit/Basis of Measurement to line level of claim,  . ; piece 16 of IBFLD, piece 17 of IBXDATA.  . ; Print in Box 24 by setting in IBXDATA(IBI,"TEXT").  . S $P(IBXDATA(IBI),U,15,17)=$P(IBFLD(24,IBI),U,14,16)  . I $P(IBFLD(24,IBI),U,14)'="" S IBXDATA(IBI,"TEXT")="N4"\_$P(IBFLD(24,IBI),U,14)\_" "\_$P(IBFLD(24,IBI),U,16)\_$P(IBFLD(24,IBI),U,15)  . ;  . I $D(IBFLD(24,IBI,"RX")) D  ;Rx  .. S IBRX1=1  .. I $P($G(IBFLD(24,IBI,"AUX")),U,8)'="" S $P(IBFLD(24,IBI,"AUX"),U,8)="",$P(IBFLD(24,IBI,"AUX"),U,9)=""  ;No free text allowed for rx's  .. I $D(IBRX1(+IBFLD(24,IBI,"RX"))) D  Q  ;Soft link exists  ...D ZERO^IBRXUTL(+$P(IBRX1(+IBFLD(24,IBI,"RX")),U,2))  ... S IBXDATA(IBI,"RX")=IBRX1(+IBFLD(24,IBI,"RX")),$P(IBXDATA(IBI,"RX"),U,2)=$E($G(^TMP($J,"IBDRUG",+$P(IBRX1(+IBFLD(24,IBI,"RX")),U,2),.01)),1,30)  ... K IBRX1(+IBFLD(24,IBI,"RX"))  ... ; No soft link - must find the first Rx with the same charge  .. S IBRX="" F  S IBRX=$O(IBRX1(IBRX)) Q:'IBRX  I +$P(IBRX1(IBRX),U,8)=+$P(IBXDATA(IBI),U,8) D  Q  ... D ZERO^IBRXUTL(+$P(IBRX1(IBRX),U,2))  ... S IBXDATA(IBI,"RX")=IBRX1(IBRX),$P(IBXDATA(IBI,"RX"),U,2)=$E($G(^TMP($J,"IBDRUG",+$P(IBRX1(IBRX),U,2),.01)),1,30) K IBRX1(IBRX) Q  ... Q  .. Q  . ;  . ; MRD;IB\*2.0\*516 - If additional service line comments to appear in  . ; Box 24, concatenate to front if something (NDC) is already there.  . I $G(IBFLD(24,IBI,"AUX"))'="" D  .. I $G(IBPRINT),$P(IBFLD(24,IBI,"AUX"),U,8)'="" D  ... I $G(IBXDATA(IBI,"TEXT"))'="" S IBXDATA(IBI,"TEXT")=$E($P(IBFLD(24,IBI,"AUX"),U,8)\_" "\_IBXDATA(IBI,"TEXT"),1,59)  ... E  S IBXDATA(IBI,"TEXT")=$P(IBFLD(24,IBI,"AUX"),U,8)  ... S $P(IBFLD(24,IBI,"AUX"),U,8)=""  ... Q  .. S IBXDATA(IBI,"AUX")=IBFLD(24,IBI,"AUX")  .. Q  . ;  . I $G(IBPRINT) D  .. ; START IB\*2.0\*447 BI ZERO DOLLAR CHANGES  .. ; I '$P(IBXDATA(IBI),U,8),'$G(IBXDATA(IBI,"RX")) D Q  .. I $P(IBXDATA(IBI),U,8)="",'$G(IBXDATA(IBI,"RX")) D  Q  ... ; END IB\*2.0\*447 BI ZERO DOLLAR CHANGES  ... I $G(IBNOSHOW) Q    ; don't show errors/warnings  ... S IBXDATA(IBI,"TEXT")="Warning:\*\* REV CODE UNITS < #PROCEDURES, THEY MUST BE ="  ... I $D(IBXDATA(IBI,"AUX")) S $P(IBXDATA(IBI,"AUX"),U,9)=""  ... Q  .. ;  .. I $G(IBFLD(24,IBI\_"A"))'="" D  Q  ... S IBXDATA(IBI,"A")=IBFLD(24,IBI\_"A")  ... I $G(IBNOSHOW) Q    ; don't show errors/warnings  ... S IBXDATA(IBI,"TEXT")="Warning:\*\* REV CODE UNITS > #PROCEDURES, THEY MUST BE=: "\_IBFLD(24,IBI\_"A")  ... I $D(IBXDATA(IBI,"AUX")) S $P(IBXDATA(IBI,"AUX"),U,9)=""  ... Q  .. ;  .. S IBRX=$G(IBXDATA(IBI,"RX"))  .. I IBRX'="" D  ;Format Rx detail  ... N Z  ... S Z=$P(IBRX,U)  ... S Z=$S(Z'="":"Rx#"\_Z\_" ",1:"RX: ")  ... S IBXDATA(IBI,"TEXT")=Z\_$S($P(IBRX,U,3)'="":"NDC: "\_$P(IBRX,U,3),1:"NOC: "\_$P(IBRX,U,2))\_" Qty: "\_$P(IBRX,U,6)\_" Days: "\_$P(IBRX,U,7)  ... S $P(IBXDATA(IBI,"AUX"),U,9)="N4"   ; service line comment qualifier for RX's  ... Q  .. Q  . S IBXDATA(IBI,"CPLNK")=$P(IBFLD(24,IBI),U,13)  . I '$G(IBPRINT) D COBLINE^IBCEU6(IBIFN,IBI,.IBXDATA,,.IBXTRA)  . Q  ;  I $G(IBPRINT) D  . S IBRX=0 F  S IBRX=$O(IBRX1(IBRX)) Q:'IBRX  D  .. S IBI=+$O(IBXDATA(""),-1)+1  .. S IBXDATA(IBI)=$$DATE($P(IBRX1(IBRX),U,5))  .. S IBXDATA(IBI,"TEXT")="\*\*\*\* ERROR - NO PROC LINK TO REV CODE FOR DRUG: RX#: "\_$P(IBRX1(IBRX),U)\_" NDC #: "\_$P(IBRX1(IBRX),U,3)  .. I $D(IBXDATA(IBI,"AUX")) S $P(IBXDATA(IBI,"AUX"),U,9)=""  .. S IBXDATA(IBI,"ARX")=""  .. D ZERO^IBRXUTL(+$P(IBRX1(IBRX),U,2))  .. S IBXDATA(IBI,"RX")=IBRX1(IBRX),$P(IBXDATA(IBI,"RX"),U,2)=$E($G(^TMP($J,"IBDRUG",+$P(IBRX1(IBRX),U,2),.01)),1,30) K IBRX1(IBRX)  .. Q  . Q  ;  I '$G(IBPRINT),$D(IBXTRA) D COMBO^IBCEU2(.IBXDATA,.IBXTRA,0) ;Handle bundled/unbundled lines  K ^TMP($J,"IBDRUG")  Q  ; ISLAB(LDATA) ; Returns 0/1 if line item data indicates the item is a lab (1)  ; 'LAB' is defined here as type of service = 5  Q $E($P(LDATA,U,4))="5"  ; FMT(DATA,DLEN,FLEN) ; Returns a string in DATA with a max length of DLEN  ; and a field length of FLEN  Q $E($E(DATA,1,DLEN)\_$J("",FLEN),1,FLEN)  ; DATE(X,DEL) ; Returns FM date in X as MMxDDxYYYY where x=DEL  S DEL=$G(DEL)  S X=$$DATE^IBCF2(X,1,1)  I X'="" S X=$E(X,1,2)\_DEL\_$E(X,3,4)\_DEL\_$E(X,5,8)  Q X  ; BATCH() ; Sets up record for and stores/returns the next batch number  N NUM,FAC,DO,DD,DLAYGO,DIC,X,Y  ;Keep latest batch number for view/print edi bill extract data option  I $D(IBVNUM) S NUM=IBVNUM G BATCHQ  ;Check for batch resubmit - if yes, use same number as original batch  I $P($G(^TMP("IBRESUBMIT",$J)),U,3)=1 S NUM=$P(^($J),U) G BATCHQ  L +^IBA(364.1,0):5 I '$T Q 0  S FAC=+$P($$SITE^VASITE(),U,3),NUM=$O(^IBA(364.1,"B",""),-1)  I $D(^IBA(364.1,+NUM,0)),$P(^(0),U,2)="" F  D  Q:'NUM!($P($G(^IBA(364.1,+NUM,0)),U,2)'="")  . I $D(^IBA(364.1,NUM,0)) S DA=NUM,DIK="^IBA(364.1," D ^DIK  . S NUM=$O(^IBA(364.1,"B",""),-1)  F  S NUM=$S($P(NUM,FAC,2)'="":NUM+1,1:FAC\_"0000001") Q:'$D(^IBA(364.1,"B",NUM))  K DO,DD S DIC="^IBA(364.1,",DLAYGO=364.1,DIC(0)="L",X=NUM D FILE^DICN K DD,DO I Y'>0 S NUM=0  L -^IBA(364.1,0) BATCHQ Q NUM  ; GETLDAT(IBXIEN) ; Extract data for 837 transmission LDAT record  ; IBXIEN - ien in file 399  ; Sets up IBXSAVE("LDAT",n) array:  ; Attachment report type ^ Attachment report transmission code ^ Attachment control number ^   ; OB Anesthesia Additional Units ^ Purchase Service Provider ID ^ Purchase Service Amount ^  N CPIEN,FTYPE,IBXDATA,IDS,IBIDS,NODE1,PSAMNT,PSPID,Z,PCE1,LINE  I '+$G(IBXIEN) Q  K IBXSAVE("LDAT")  S FTYPE=$$FT^IBCEF(IBXIEN)  I FTYPE=2 D OUTPT(IBXIEN,0)  I FTYPE=3 D HOS^IBCEF2(IBXIEN)  D ALLIDS^IBCEFP(IBXIEN,.IDS,1)  S (PSPID,PSAMNT)=""  ; IB\*2.0\*473/TAZ - Convert PROVIDER code to function call to PSID^IBCEFP  I $$SUB1OK^IBCEP8A(IBXIEN),(FTYPE=2) D  . D PSID^IBCEFP(IBXIEN,.IDS,.IBIDS)  . S PSPID=$G(IBIDS(0)) I PSPID="" S PSPID=$P($G(IBIDS(1)),U,1)  ;IB\*2.0\*473/TAZ - END  S Z=0 F  S Z=$O(IBXDATA(Z)) Q:'Z  D  . S CPIEN=+$G(IBXDATA(Z,"CPLNK")) ;I 'CPIEN Q  . I FTYPE=2,$$SUB1OK^IBCEP8A(IBXIEN) S PSAMNT=$$DOLLAR^IBCEFG1($P($G(IBXDATA(Z)),U,11))  . S (PCE1,NODE1)=""  . I CPIEN D  . . S NODE1=$G(^DGCR(399,IBXIEN,"CP",CPIEN,1))  . . S PCE1=$$GET1^DIQ(399.0304,CPIEN\_","\_IBXIEN\_",",71)  . . Q  . ; MRD;IB\*2.0\*516 - Added addl. procedure description as piece 7   . ; of IBXSAVE, which will exist only if the procedure ends in '99'  . ; or is an 'NOC/NOS' procedure.  . S IBXSAVE("LDAT",Z)=PCE1\_U\_$P(NODE1,U,3)\_U\_$P(NODE1,U)\_U\_$P(NODE1,U,5)\_U\_$G(PSPID)\_U\_$G(PSAMNT)\_U\_$P(NODE1,U,4)  . Q  Q | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCF23A | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** |  | | | | | | | | |
| **Related Options** | None | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  |  | | | | |  | | | |
| **Data Dictionary (DD) References** | None | | | | | | | | |
| **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** | | | | | | | | | |
| IBCF23A ;ALB/ARH - HCFA 1500 19-90 DATA - Split from IBCF23 ;12-JUN-93  ;;2.0;INTEGRATED BILLING;\*\*51,432,516,547\*\*;21-MAR-94;Build 119  ;;Per VA Directive 6402, this routine should not be modified.  ;  ; $$INSTALDT^XPDUTL(IBPATCH,.IBARY) - ICR 10141  ; B24 ; set individual entries in print array, external format  ; IBAUX = additional data for EDI output  ; IBRXF = array of RX procedures  N IBX,Z,IBD1,IBD2,IBCPLINK  S IBI=IBI+1,IBPROC=$P(IBSS,U,2),IBD1=$$DATE^IBCF23(IBDT1),IBD2=$S(IBDT1'=IBDT2:$$DATE^IBCF23(IBDT2),1:"")  I '$D(IBXIEN) S IBD1=$E(IBD1,5,8)\_$E(IBD1,1,4),IBD2=$E(IBD2,5,8)\_$E(IBD2,1,4)  S IBFLD(24,IBI)=IBD1\_U\_IBD2\_U\_$P($G(^IBE(353.1,+$P(IBSS,U,6),0)),U)\_U\_$P($G(^IBE(353.2,+$P(IBSS,U,7),0)),U)  I +IBPROC D  . S IBFLD(24,IBI)=IBFLD(24,IBI)\_U\_$P($$PRCD^IBCEF1(IBPROC,1),U,2) S:$P(IBPROC,";",2)'["ICPT" IBFLD(24,IBI\_"X")=""  I 'IBPROC S IBFLD(24,IBI)=IBFLD(24,IBI)\_U\_$S('$D(IBXIEN):IBPROC,1:+IBREV),IBFLD(24,IBI\_"A")=$P($G(^DGCR(399.2,+IBREV,0)),U,2)  I $D(IBRXF),IBCHARG="" S IBFLD(24,IBI\_"A")=$P($G(^DGCR(399.2,+IBREV,0)),U,2)  S IBFLD(24,IBI)=IBFLD(24,IBI)\_U\_$P(IBSS,U,5)\_U\_IBCHARG\_U\_IBUNIT\_U\_$P(IBSS,U,8)\_U\_$G(IBPCHG)\_U\_$G(IBMIN)\_U\_$G(IBEMG)  I $D(IBSS("L")) S Z=0 F  S Z=$O(IBSS("L",Z)) Q:'Z  S IBFLD(24,IBI,$P(IBSS("L",Z),U),$P(IBSS("L",Z),U,2))=$G(IBFLD(24,IBI,$P(IBSS("L",Z),U),$P(IBSS("L",Z),U,2)))+1  S:$TR($G(IBAUX),U)'="" IBFLD(24,IBI,"AUX")=$G(IBAUX)  S:$D(IBRXF) IBFLD(24,IBI,"RX")=IBRXF  K IBPROC,IBSS("L")  S IBCPLINK=$P(IBSS,U,$L(IBSS,U))  S IBFLD(24,IBI)=IBFLD(24,IBI)\_U\_IBCPLINK  ; MRD;IB\*2.0\*516 - Added NDC and Units to line level of claim.  I IBCPLINK'="" S $P(IBFLD(24,IBI),U,14,15)=$TR($P($G(^DGCR(399,IBIFN,"CP",IBCPLINK,1)),U,7,8),"-")  Q  ; AUXOK(IBSS,IBSS1) ; Check all other flds are the same to combine procs  ; IBSS = subscript of IBCP to check for dups to combine - pass by ref  ; IBSS(IBSS,"AUX-X",n) = all the previously extracted line items for the  ; same set of basic data, but having different "AUX" data  ; IBSS1 = the "AUX" data of the current IBCP entry  ;  ; Returns entry # in IBSS array if match found, or 0 if no match  ; Set the IBSS "AUX-X" node for no match  N Z,Z0  S Z=0 F  S Z=$O(IBSS(IBSS,"AUX-X",Z)) Q:'Z  I IBSS1=IBSS(IBSS,"AUX-X",Z) Q  I 'Z S Z0=+$O(IBSS(IBSS,"AUX-X",""),-1)+1,IBSS(IBSS,"AUX-X",Z0)=IBSS1  Q +Z  ; PRC ; Extract procedure data for HCFA 1500  ; IBRC(IBSS) = #rev codes with same billing criteria (IBSS)  ; IBLINK('CP' ien,'RC' ien) = IBSS including modifiers,rx seq in pc 7,8  ; IBLINK1(IBSS, 'RC' ien) = auto (1)^ 'CP' ien (soft link)  ;  ; proc array w/chrg  N IBPR,IBP  S IBI=0 F  S IBI=$O(^DGCR(399,IBIFN,"CP",IBI)) Q:'IBI  S IBLN=^(IBI,0),IBAUXLN=$G(^("AUX")) D  . N Z,Z0,Z1,Q1  . S IBPDT=$P(IBLN,U,2)  . S IBSS=$$IBSS(IBI,.IBDXI,IBLN)  . S IBPO=$S($P(IBLN,U,4):+$P(IBLN,U,4),1:IBI+1000) ;Set print order  . S IBCP(IBPO)=IBPDT\_"^"\_IBSS,IBCP(IBPO,"AUX")=IBAUXLN  . S IBCP(IBPO,"LNK")=IBI  . ; Rx  . N IBZ,IBITEM  . S IBZ=$S($P(IBSS,U):$P(IBSS,U),1:"")  . I IBZ'="",$D(IBLINKRX(IBZ,IBI)) D  Q:IBCHARG'=""  .. S IBPO1=IBPO  .. S IBITEM=+$O(IBLINKRX(IBZ,IBI,0)),IBRV=$G(IBLINKRX(IBZ,IBI,IBITEM))  .. Q:$S(IBRV="":1,1:'$G(IBRC(IBRV)))  .. S IBCHARG=$P(IBRV,U,6),IBRC(IBRV)=IBRC(IBRV)-1  .. S $P(IBCP(IBPO1),U,9)=IBCHARG,IBCP(IBPO1,"RX")=IBITEM K IBLINKRX(IBZ,IBI,IBITEM)  . ; find chrgs directly linked to proc  . S IBK=0 F  S IBK=$O(IBLINK(IBI,IBK)) Q:'IBK  S IBRV1=IBLINK(IBI,IBK),IBRV=$P(IBRV1,U,1,6) I +IBRC(IBRV1) D  .. S IBCHARG=$P(IBRV,U,6),IBRC(IBRV1)=IBRC(IBRV1)-1  .. I IBCHARG'="" S $P(IBSS,U,8)=IBCHARG,IBCP(IBPO)=IBPDT\_"^"\_IBSS,IBPO=IBPO+.1  ;  ; add chrgs associated with a proc (not a direct link)  ; find chrg associated with proc, if any (match proc,div,+/-basc)  K IBP(0)  F IBP=3,2 Q:$D(IBP(0)) S IBPO="" F  S IBPO=$O(IBCP(IBPO)) Q:'IBPO  I $P(IBCP(IBPO),U,9)="" D  . S IBSS=$P(IBCP(IBPO),U,2,9)  . S IBCHARG="",(IBRV,IBSS)=$P(IBSS,U,1,IBP) F  S IBRV=$O(IBRC(IBRV)) Q:$P(IBRV,U,1,IBP)'=IBSS  S IBP(0)=0 I +IBRC(IBRV) D  Q  .. S IBCHARG=$P(IBRV,U,6),IBRC(IBRV)=IBRC(IBRV)-1  .. I IBRC(IBRV) S Z=0 F  S Z=$O(IBCP(IBPO,Z)) Q:'Z  S IBRC(IBRV)=IBRC(IBRV)-1  . S $P(IBCP(IBPO),U,9)=IBCHARG  . I IBCHARG'="" S Z=$O(IBLINK1(IBRV,0)) I Z S IBCP(IBPO,"L",Z)=IBLINK1(IBRV,Z) K IBLINK1(IBRV,Z)  ;  ; add chrgs not associated with a proc to first proc with no chrg  ; Aggggh!!! TP  S IBPO="" F  S IBPO=$O(IBCP(IBPO)) Q:'IBPO  I $P(IBCP(IBPO),U,9)="" D  . S IBCHARG="",IBRV="^" F  S IBRV=$O(IBRC(IBRV)) Q:IBRV=""!+IBRV  I +IBRC(IBRV) D  Q  .. S IBCHARG=$P(IBRV,U,6),IBRC(IBRV)=IBRC(IBRV)-1  .. S Z=$O(IBLINK1(IBRV,0)) I Z S IBCP(IBPO,"L",Z)=IBLINK1(IBRV,Z) K IBLINK1(IBRV,Z)  . S $P(IBCP(IBPO),U,9)=IBCHARG  ;  Q IBSS(IBI,IBDXI,IBLN) ; Creates index sequence for procedure  N IBPC,IBJ,IBSS,IBLPI,IBX,IBLPAR  S (IBPC,IBLPI)=0  F IBJ=1,6,5,0,9,10 S IBPC=IBPC+1 S:IBJ $P(IBSS,U,IBPC,IBPC+1)=($P(IBLN,U,IBJ)\_U)  S $P(IBSS,U,7)=($$GETMOD^IBEFUNC(IBIFN,IBI)\_U) ;Modifiers  ;IB\*547/TAZ - IBDXI not defined, use internal DX pointer  I '$G(IBNWPTCH) F IBJ=11:1:14 I $P(IBLN,U,IBJ) S $P(IBSS,U,4)=$P(IBSS,U,4)\_$S(IBJ>11:",",1:"")\_$G(IBDXI(+$P(IBLN,U,IBJ))) ; dx  I $G(IBNWPTCH) F IBJ=11:1:14 S IBX=$P(IBLN,U,IBJ) I IBX S $P(IBSS,U,4)=$P(IBSS,U,4)\_$S(IBJ>11:",",1:"")\_$G(IBDXI(IBX),IBX) ; dx  S $P(IBSS,U,10)=$P(IBLN,U,16),$P(IBSS,U,9)=$P(IBLN,U,19),$P(IBSS,U,11)=+$P(IBLN,U,17)  G:'$G(IBNWPTCH) IBSSX  ;IB\*547/TAZ - Add additional fields for roll-up compare  S $P(IBSS,U,21)=$$GET1^DIQ(399.0304,IBI\_","\_IBIFN\_",","ASSOCIATED CLINIC","I")  S $P(IBSS,U,22)=$$GET1^DIQ(399.0304,IBI\_","\_IBIFN\_",","TYPE OF SERVICE","I")  S $P(IBSS,U,23)=$$GET1^DIQ(399.0304,IBI\_","\_IBIFN\_",","ATTACHMENT CONTROL NUMBER","I")  S $P(IBSS,U,24)=$$GET1^DIQ(399.0304,IBI\_","\_IBIFN\_",","NDC","I")  S $P(IBSS,U,25)=$$GET1^DIQ(399.0304,IBI\_","\_IBIFN\_",","PROCEDURE DESCRIPTION","I")  S $P(IBSS,U,26)=$$GET1^DIQ(399.0304,IBI\_","\_IBIFN\_",","ADDITIONAL OB MINUTES","I")  ;Add Provider info in pieces 41-49  M IBLPAR=^DGCR(399,IBIFN,"CP",IBI,"LNPRV")  F  S IBLPI=$O(IBLPAR(IBLPI)) Q:'IBLPI  S IBX=IBLPAR(IBLPI,0),$P(IBSS,U,40+IBX)=$TR(IBX,"^","~")  K IBLPAR IBSSX ;  Q IBSS  ; IBNWPTCH(IBIFN,IBPATCH) ;  ;Checks the date the primary claim was 1st transmitted and returns 1 if the transmitted date is after the patch  ;referenced in variable IBPATCH was released. This allows the MRA/EOBs returning to roll up procedures the same  ;way as they went out. Otherwise the order changes and the MRA/EOB won't match up.  ;  N IBARY,IBIDT,IBPFN,IBEFN,IBBN,IBX,IBBDT  S IBX=0  I $$INSTALDT^XPDUTL(IBPATCH,.IBARY) D   ;ICR 10141  . S IBX=1  . S IBIDT=$O(IBARY(""))  . ; Get Primary Bill Number. This will insure COB data is consistent across all bills.  . S IBPFN=$$GET1^DIQ(399,IBIFN\_",","PRIMARY BILL #","I") I 'IBPFN S IBPFN=IBIFN  . ; Find 1st Accepted Entry (A1, A2, or Z) of Primary Bill in EDI TRANSMIT BILL FILE (364) to determine Batch Number  . S (IBEFN,IBBN)=0 F  S IBEFN=$O(^IBA(364,"B",IBPFN,IBEFN)) Q:'IBEFN  D  I IBBN Q  .. I ",A1,A2,Z,"'[(","\_$$GET1^DIQ(364,IBEFN\_",","TRANSMISSION STATUS","I")\_",") Q  .. S IBBN=$$GET1^DIQ(364,IBEFN\_",","BATCH NUMBER","I")  . ;Retrieve the date the batch was 1st sent. If IBBN="" IBBDT will be null  . S IBBDT=$$GET1^DIQ(364.1,$$GET1^DIQ(364,IBBN\_",","BATCH NUMBER","I")\_",","DATE FIRST SENT","I")  . I IBBDT,(IBBDT<IBIDT) S IBX=0  Q IBX | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCF23A ;ALB/ARH - HCFA 1500 19-90 DATA - Split from IBCF23 ;12-JUN-93  ;;2.0;INTEGRATED BILLING;\*\*51,432,516,547,577\*\*;21-MAR-94;Build 16  ;;Per VA Directive 6402, this routine should not be modified.  ;  ; $$INSTALDT^XPDUTL(IBPATCH,.IBARY) - ICR 10141  ; B24 ; set individual entries in print array, external format  ; IBAUX = additional data for EDI output  ; IBRXF = array of RX procedures  N IBX,Z,IBD1,IBD2,IBCPLINK  S IBI=IBI+1,IBPROC=$P(IBSS,U,2),IBD1=$$DATE^IBCF23(IBDT1),IBD2=$S(IBDT1'=IBDT2:$$DATE^IBCF23(IBDT2),1:"")  I '$D(IBXIEN) S IBD1=$E(IBD1,5,8)\_$E(IBD1,1,4),IBD2=$E(IBD2,5,8)\_$E(IBD2,1,4)  S IBFLD(24,IBI)=IBD1\_U\_IBD2\_U\_$P($G(^IBE(353.1,+$P(IBSS,U,6),0)),U)\_U\_$P($G(^IBE(353.2,+$P(IBSS,U,7),0)),U)  I +IBPROC D  . S IBFLD(24,IBI)=IBFLD(24,IBI)\_U\_$P($$PRCD^IBCEF1(IBPROC,1),U,2) S:$P(IBPROC,";",2)'["ICPT" IBFLD(24,IBI\_"X")=""  I 'IBPROC S IBFLD(24,IBI)=IBFLD(24,IBI)\_U\_$S('$D(IBXIEN):IBPROC,1:+IBREV),IBFLD(24,IBI\_"A")=$P($G(^DGCR(399.2,+IBREV,0)),U,2)  I $D(IBRXF),IBCHARG="" S IBFLD(24,IBI\_"A")=$P($G(^DGCR(399.2,+IBREV,0)),U,2)  S IBFLD(24,IBI)=IBFLD(24,IBI)\_U\_$P(IBSS,U,5)\_U\_IBCHARG\_U\_IBUNIT\_U\_$P(IBSS,U,8)\_U\_$G(IBPCHG)\_U\_$G(IBMIN)\_U\_$G(IBEMG)  I $D(IBSS("L")) S Z=0 F  S Z=$O(IBSS("L",Z)) Q:'Z  S IBFLD(24,IBI,$P(IBSS("L",Z),U),$P(IBSS("L",Z),U,2))=$G(IBFLD(24,IBI,$P(IBSS("L",Z),U),$P(IBSS("L",Z),U,2)))+1  S:$TR($G(IBAUX),U)'="" IBFLD(24,IBI,"AUX")=$G(IBAUX)  S:$D(IBRXF) IBFLD(24,IBI,"RX")=IBRXF  K IBPROC,IBSS("L")  S IBCPLINK=$P(IBSS,U,$L(IBSS,U))  S IBFLD(24,IBI)=IBFLD(24,IBI)\_U\_IBCPLINK  ; MRD;IB\*2.0\*516 - Added NDC and Units to line level of claim.  ;I IBCPLINK'="" S $P(IBFLD(24,IBI),U,14,15)=$TR($P($G(^DGCR(399,IBIFN,"CP",IBCPLINK,1)),U,7,8),"-")  ; vd/Beginning of IB\*2\*577 - Added Unit/Basis of Measurment to line level of claim.  I IBCPLINK'="" S $P(IBFLD(24,IBI),U,14,16)=$TR($P($G(^DGCR(399,IBIFN,"CP",IBCPLINK,1)),U,7,8),"-")\_U\_$P($G(^DGCR(399,IBIFN,"CP",IBCPLINK,2)),U)  ; vd/End of IB\*2\*577  Q  ; AUXOK(IBSS,IBSS1) ; Check all other flds are the same to combine procs  ; IBSS = subscript of IBCP to check for dups to combine - pass by ref  ; IBSS(IBSS,"AUX-X",n) = all the previously extracted line items for the  ; same set of basic data, but having different "AUX" data  ; IBSS1 = the "AUX" data of the current IBCP entry  ;  ; Returns entry # in IBSS array if match found, or 0 if no match  ; Set the IBSS "AUX-X" node for no match  N Z,Z0  S Z=0 F  S Z=$O(IBSS(IBSS,"AUX-X",Z)) Q:'Z  I IBSS1=IBSS(IBSS,"AUX-X",Z) Q  I 'Z S Z0=+$O(IBSS(IBSS,"AUX-X",""),-1)+1,IBSS(IBSS,"AUX-X",Z0)=IBSS1  Q +Z  ; PRC ; Extract procedure data for HCFA 1500  ; IBRC(IBSS) = #rev codes with same billing criteria (IBSS)  ; IBLINK('CP' ien,'RC' ien) = IBSS including modifiers,rx seq in pc 7,8  ; IBLINK1(IBSS, 'RC' ien) = auto (1)^ 'CP' ien (soft link)  ;  ; proc array w/chrg  N IBPR,IBP  S IBI=0 F  S IBI=$O(^DGCR(399,IBIFN,"CP",IBI)) Q:'IBI  S IBLN=^(IBI,0),IBAUXLN=$G(^("AUX")) D  . N Z,Z0,Z1,Q1  . S IBPDT=$P(IBLN,U,2)  . S IBSS=$$IBSS(IBI,.IBDXI,IBLN)  . S IBPO=$S($P(IBLN,U,4):+$P(IBLN,U,4),1:IBI+1000) ;Set print order  . S IBCP(IBPO)=IBPDT\_"^"\_IBSS,IBCP(IBPO,"AUX")=IBAUXLN  . S IBCP(IBPO,"LNK")=IBI  . ; Rx  . N IBZ,IBITEM  . S IBZ=$S($P(IBSS,U):$P(IBSS,U),1:"")  . I IBZ'="",$D(IBLINKRX(IBZ,IBI)) D  Q:IBCHARG'=""  .. S IBPO1=IBPO  .. S IBITEM=+$O(IBLINKRX(IBZ,IBI,0)),IBRV=$G(IBLINKRX(IBZ,IBI,IBITEM))  .. Q:$S(IBRV="":1,1:'$G(IBRC(IBRV)))  .. S IBCHARG=$P(IBRV,U,6),IBRC(IBRV)=IBRC(IBRV)-1  .. S $P(IBCP(IBPO1),U,9)=IBCHARG,IBCP(IBPO1,"RX")=IBITEM K IBLINKRX(IBZ,IBI,IBITEM)  . ; find chrgs directly linked to proc  . S IBK=0 F  S IBK=$O(IBLINK(IBI,IBK)) Q:'IBK  S IBRV1=IBLINK(IBI,IBK),IBRV=$P(IBRV1,U,1,6) I +IBRC(IBRV1) D  .. S IBCHARG=$P(IBRV,U,6),IBRC(IBRV1)=IBRC(IBRV1)-1  .. I IBCHARG'="" S $P(IBSS,U,8)=IBCHARG,IBCP(IBPO)=IBPDT\_"^"\_IBSS,IBPO=IBPO+.1  ;  ; add chrgs associated with a proc (not a direct link)  ; find chrg associated with proc, if any (match proc,div,+/-basc)  K IBP(0)  F IBP=3,2 Q:$D(IBP(0)) S IBPO="" F  S IBPO=$O(IBCP(IBPO)) Q:'IBPO  I $P(IBCP(IBPO),U,9)="" D  . S IBSS=$P(IBCP(IBPO),U,2,9)  . S IBCHARG="",(IBRV,IBSS)=$P(IBSS,U,1,IBP) F  S IBRV=$O(IBRC(IBRV)) Q:$P(IBRV,U,1,IBP)'=IBSS  S IBP(0)=0 I +IBRC(IBRV) D  Q  .. S IBCHARG=$P(IBRV,U,6),IBRC(IBRV)=IBRC(IBRV)-1  .. I IBRC(IBRV) S Z=0 F  S Z=$O(IBCP(IBPO,Z)) Q:'Z  S IBRC(IBRV)=IBRC(IBRV)-1  . S $P(IBCP(IBPO),U,9)=IBCHARG  . I IBCHARG'="" S Z=$O(IBLINK1(IBRV,0)) I Z S IBCP(IBPO,"L",Z)=IBLINK1(IBRV,Z) K IBLINK1(IBRV,Z)  ;  ; add chrgs not associated with a proc to first proc with no chrg  ; Aggggh!!! TP  S IBPO="" F  S IBPO=$O(IBCP(IBPO)) Q:'IBPO  I $P(IBCP(IBPO),U,9)="" D  . S IBCHARG="",IBRV="^" F  S IBRV=$O(IBRC(IBRV)) Q:IBRV=""!+IBRV  I +IBRC(IBRV) D  Q  .. S IBCHARG=$P(IBRV,U,6),IBRC(IBRV)=IBRC(IBRV)-1  .. S Z=$O(IBLINK1(IBRV,0)) I Z S IBCP(IBPO,"L",Z)=IBLINK1(IBRV,Z) K IBLINK1(IBRV,Z)  . S $P(IBCP(IBPO),U,9)=IBCHARG  ;  Q IBSS(IBI,IBDXI,IBLN) ; Creates index sequence for procedure  N IBPC,IBJ,IBSS,IBLPI,IBX,IBLPAR  S (IBPC,IBLPI)=0  F IBJ=1,6,5,0,9,10 S IBPC=IBPC+1 S:IBJ $P(IBSS,U,IBPC,IBPC+1)=($P(IBLN,U,IBJ)\_U)  S $P(IBSS,U,7)=($$GETMOD^IBEFUNC(IBIFN,IBI)\_U) ;Modifiers  ;IB\*547/TAZ - IBDXI not defined, use internal DX pointer  I '$G(IBNWPTCH) F IBJ=11:1:14 I $P(IBLN,U,IBJ) S $P(IBSS,U,4)=$P(IBSS,U,4)\_$S(IBJ>11:",",1:"")\_$G(IBDXI(+$P(IBLN,U,IBJ))) ; dx  I $G(IBNWPTCH) F IBJ=11:1:14 S IBX=$P(IBLN,U,IBJ) I IBX S $P(IBSS,U,4)=$P(IBSS,U,4)\_$S(IBJ>11:",",1:"")\_$G(IBDXI(IBX),IBX) ; dx  S $P(IBSS,U,10)=$P(IBLN,U,16),$P(IBSS,U,9)=$P(IBLN,U,19),$P(IBSS,U,11)=+$P(IBLN,U,17)  G:'$G(IBNWPTCH) IBSSX  ;IB\*547/TAZ - Add additional fields for roll-up compare  S $P(IBSS,U,21)=$$GET1^DIQ(399.0304,IBI\_","\_IBIFN\_",","ASSOCIATED CLINIC","I")  S $P(IBSS,U,22)=$$GET1^DIQ(399.0304,IBI\_","\_IBIFN\_",","TYPE OF SERVICE","I")  S $P(IBSS,U,23)=$$GET1^DIQ(399.0304,IBI\_","\_IBIFN\_",","ATTACHMENT CONTROL NUMBER","I")  S $P(IBSS,U,24)=$$GET1^DIQ(399.0304,IBI\_","\_IBIFN\_",","NDC","I")  S $P(IBSS,U,25)=$$GET1^DIQ(399.0304,IBI\_","\_IBIFN\_",","PROCEDURE DESCRIPTION","I")  S $P(IBSS,U,26)=$$GET1^DIQ(399.0304,IBI\_","\_IBIFN\_",","ADDITIONAL OB MINUTES","I")  ;Add Provider info in pieces 41-49  M IBLPAR=^DGCR(399,IBIFN,"CP",IBI,"LNPRV")  F  S IBLPI=$O(IBLPAR(IBLPI)) Q:'IBLPI  S IBX=IBLPAR(IBLPI,0),$P(IBSS,U,40+IBX)=$TR(IBX,"^","~")  K IBLPAR IBSSX ;  Q IBSS  ; IBNWPTCH(IBIFN,IBPATCH) ;  ;Checks the date the primary claim was 1st transmitted and returns 1 if the transmitted date is after the patch  ;referenced in variable IBPATCH was released. This allows the MRA/EOBs returning to roll up procedures the same  ;way as they went out. Otherwise the order changes and the MRA/EOB won't match up.  ;  N IBARY,IBIDT,IBPFN,IBEFN,IBBN,IBX,IBBDT  S IBX=0  I $$INSTALDT^XPDUTL(IBPATCH,.IBARY) D   ;ICR 10141  . S IBX=1  . S IBIDT=$O(IBARY(""))  . ; Get Primary Bill Number. This will insure COB data is consistent across all bills.  . S IBPFN=$$GET1^DIQ(399,IBIFN\_",","PRIMARY BILL #","I") I 'IBPFN S IBPFN=IBIFN  . ; Find 1st Accepted Entry (A1, A2, or Z) of Primary Bill in EDI TRANSMIT BILL FILE (364) to determine Batch Number  . S (IBEFN,IBBN)=0 F  S IBEFN=$O(^IBA(364,"B",IBPFN,IBEFN)) Q:'IBEFN  D  I IBBN Q  .. I ",A1,A2,Z,"'[(","\_$$GET1^DIQ(364,IBEFN\_",","TRANSMISSION STATUS","I")\_",") Q  .. S IBBN=$$GET1^DIQ(364,IBEFN\_",","BATCH NUMBER","I")  . ;Retrieve the date the batch was 1st sent. If IBBN="" IBBDT will be null  . S IBBDT=$$GET1^DIQ(364.1,$$GET1^DIQ(364,IBBN\_",","BATCH NUMBER","I")\_",","DATE FIRST SENT","I")  . I IBBDT,(IBBDT<IBIDT) S IBX=0  Q IBX | | | | | | | | | |

1. MODIFY the ^IBCEF22 routines to update the printed UB-04 for the new UNITS/BASIS OF MEASUREMENT field.

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCEF22 | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** |  | | | | | | | | |
| **Related Options** | None | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  |  | | | | |  | | | |
| **Data Dictionary (DD) References** | None | | | | | | | | |
| **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** | | | | | | | | | |
| IBCEF22 ;ALB/TMP - FORMATTER SPECIFIC BILL FUNCTIONS ;06-FEB-96  ;;2.0;INTEGRATED BILLING;\*\*51,137,135,155,309,349,389,432,488,516\*\*;21-MAR-94;Build 123  ;;Per VA Directive 6402, this routine should not be modified.  ;  ; OVERFLOW FROM ROUTINE IBCEF2 HOS(IBIFN) ; Extract rev codes for episode billed on a UB-04 into IBXDATA  ; IBIFN = bill ien  ; Format: IBXDATA(n) =  ; rev cd ptr ^ CPT CODE ptr ^ unit chg ^ units ^ tot charge  ; ^ tot uncov ^ FL49 value  ; ^ ien of rev code multiple entry(s) (separated by ";")  ; ^ modifiers specific to rev code/proc (separated by ",")  ; ^ rev code date, if it can be determined by a corresponding proc  ; ^ NDC from "CP" node of claim ^ Units from "CP" node  ;  ; Also Returns IBXDATA(IBI,"COB",COB,m) with COB data for each line  ; item found in an accepted EOB for the bill and = the reference  ; line in the first '^' piece followed by the '0' node of file  ; 361.115 (LINE LEVEL ADJUSTMENTS)  ; COB = COB seq # of adjustment's ins co, m = seq #  ; -- AND --  ; IBXDATA(IBI,"COB",COB,m,z,p)=  ; the '0' node for each subordinate entry of file  ; 361.11511 (REASONS) (Only first 3 pieces for 837)  ; z = group code, sometimes preceeded by a space p = seq #  ;  ; -- AND --  ; IBXDATA(n,"CPLNK") = soft link to corresponding entry in PROCEDURES multiple of file 399  ;  N IBDA,IBCOMB,IBINPAT,IBLN,IBX,IBY,IBZ,IBS,IBSS,IBXTRA,IBX1,IBXS,IBP,IBPO,IBP1,IBDEF,Z,Z0,Z1,ZX,QQ,IBMOD,LST  S IBINPAT=$$INPAT^IBCEF(IBIFN,1)  I 'IBINPAT D F^IBCEF("N-STATEMENT COVERS FROM DATE","IBZ",,IBIFN)  S IBDEF=$G(IBZ),LST=""  ;  ; Loop through lines of claim beneath ^DGCR(399,IBIFN,"CP") and build  ; the array IBP to be used below.  ; IBP(Procedure ^ Modifiers, Print Order, Line#) = Procedure Date  ;  S IBDA=0 F  S IBDA=$O(^DGCR(399,IBIFN,"CP",IBDA)) Q:'IBDA  S IBZ=$G(^(IBDA,0)) I IBZ D  . S IBP(+$P(IBZ,U)\_U\_$$GETMOD^IBEFUNC(IBIFN,IBDA,1),$S($P(IBZ,U,4):$P(IBZ,U,4),1:999),IBDA)=$P(IBZ,U,2)  ;  ; Loop through the revenue codes beneath ^DGCR(399,IBIFN,"RC") and  ; build the array IBX to be used below.  ; IBX(" "\_Revenue Code, Print Order, Revenue Line#) =  ; ^DGCR(399.2, Revenue Code IEN, 0)  ; IBX(" "\_Revenue Code, Print Order, Revenue Line#, "DT") = Procedure Date  ; IBX(" "\_Revenue Code, Print Order, Revenue Line#, "MOD") = Modifiers  ;  S IBDA=0 F  S IBDA=$O(^DGCR(399,IBIFN,"RC",IBDA)) Q:'IBDA  S IBZ=$G(^(IBDA,0)) I IBZ S IBMOD="" D  . S IBX=$G(^DGCR(399.2,+IBZ,0)),IBX1="",IBPO=0  . ; Auto-added procedure charge  . I $P(IBZ,U,10)=4,$P(IBZ,U,11) D  ; Soft link to proc  .. S Z=$G(^DGCR(399,IBIFN,"CP",$P(IBZ,U,11),0))  .. Q:Z=""  .. S ZX=+Z\_U\_$$GETMOD^IBEFUNC(IBIFN,$P(IBZ,U,11),1)  .. Q:'$O(IBP(ZX,0))&'$O(IBP1(ZX,0))  .. I $P(IBZ,U,6) Q:$S($P(Z,U)'["ICPT":1,1:+$P(Z,U)'=$P(IBZ,U,6))  .. S Z0=$S($D(IBP(ZX)):$O(IBP(ZX,0)),1:$O(IBP1(ZX,0)))  .. S:'Z0 Z0=999  .. Q:'$D(IBP(ZX,+Z0,$P(IBZ,U,11)))&'$D(IBP1(ZX,+Z0,$P(IBZ,U,11)))  .. I '$D(IBP1(ZX,+Z0,$P(IBZ,U,11))) S IBP1(ZX,+Z0,$P(IBZ,U,11))=IBP(ZX,+Z0,$P(IBZ,U,11)) K IBP(ZX,+Z0,$P(IBZ,U,11))  .. S IBX1=$P(Z,U,2),IBPO=+Z0,IBMOD=$P(ZX,U,2)  . ; Manually added charge with a procedure  . I $P(IBZ,U,6),$S($P(IBZ,U,10)=4:'$P(IBZ,U,11),1:1),+$O(IBP($P(IBZ,U,6)))=$P(IBZ,U,6) D  .. ; No direct link, but a proc exists on rev code and in procedure mult without and then with modifiers  .. S ZX=$O(IBP($P(IBZ,U,6)))  .. F QQ=1,2 Q:IBPO  S Z="" F  S Z=$O(IBP(ZX,Z),-1) Q:'Z!(IBPO) S Z0=0 F  S Z0=$O(IBP(ZX,Z,Z0)) Q:'Z0  S Z1=$G(^DGCR(399,IBIFN,"CP",Z0,0)) D  Q:IBPO  ... ; Ignore if not a CPT or a modifier exists and this is first pass  ... S IBMOD=$$GETMOD^IBEFUNC(IBIFN,Z0,1)  ... Q:$S($P(Z1,U)'["ICPT":1,QQ=1:IBMOD'="",1:0)  ... S IBPO=+$P(Z1,U,4),IBX1=$P(Z1,U,2)  ... K IBP(+Z1\_U\_IBMOD,Z,Z0)  . ;  . I IBX'="" D  ; revenue code is valid  .. S LST=$S(LST="":900,1:LST+1)  .. F Z=LST:1 S Z0=$S(IBPO:IBPO,$D(IBX(" "\_$P(IBX,U),Z)):0,1:Z) I Z0 S (LST,IBPO)=Z0 Q  .. S IBX(" "\_$P(IBX,U),IBPO,IBDA)=IBX,IBX(" "\_$P(IBX,U),IBPO,IBDA,"DT")=$S(IBX1:IBX1,1:IBDEF),IBX(" "\_$P(IBX,U),IBPO,IBDA,"MOD")=IBMOD  ;  ; Loop through revenue codes in IBX and build the array IBX1.  ;  S IBS="" F  S IBS=$O(IBX(IBS)) Q:IBS=""  S IBPO=0 F  S IBPO=$O(IBX(IBS,IBPO)) Q:'IBPO  D  . S IBDA=0 F  S IBDA=$O(IBX(IBS,IBPO,IBDA)) Q:'IBDA  S IBX=$G(IBX(IBS,IBPO,IBDA)),IBZ=$G(^DGCR(399,IBIFN,"RC",IBDA,0)) I IBX'="" D  .. ;S IBXS=$P(IBZ,U,2)\_U\_$P(IBZ,U,6)\_U\_$G(IBX(IBS,IBPO,IBDA,"MOD"))  .. S IBXS=U\_$P(IBZ,U,6)\_U\_$G(IBX(IBS,IBPO,IBDA,"MOD")) ;combine same proc and modifiers regardless of rate  .. S:IBPO'<900&'$$ACCRV($P(IBS," ",2))&$S(IBINPAT:$P(IBZ,U,6),1:1) IBCOMB(IBS,IBXS,IBPO)=IBDA  .. S:'$D(IBX1(IBS,IBPO,IBXS,1)) IBX1(IBS,IBPO,IBXS,1)=IBX,IBX1(IBS,IBPO,IBXS,2)=IBZ  .. S $P(IBX1(IBS,IBPO,IBXS),U)=$P($G(IBX1(IBS,IBPO,IBXS)),U)+$P(IBZ,U,3)  .. S $P(IBX1(IBS,IBPO,IBXS),U,2)=$P($G(IBX1(IBS,IBPO,IBXS)),U,2)+$P(IBZ,U,4)  .. S IBX1(IBS,IBPO,IBXS,"DT")=$G(IBX(IBS,IBPO,IBDA,"DT")),IBX1(IBS,IBPO,IBXS,"IEN")=$G(IBX1(IBS,IBPO,IBXS,"IEN"))\_$S($G(IBX1(IBS,IBPO,IBXS,"IEN")):";",1:"")\_IBDA  ;  S IBS="" F  S IBS=$O(IBX1(IBS)) Q:IBS=""  S IBPO=899 F  S IBPO=$O(IBX1(IBS,IBPO)) Q:'IBPO  D  ; Check to combine like rev codes without print order  . N Q,Q0,Q1,Z,Z0,Z1,Z2,IBZ1,IBZ2  . S Z=""  . N IBACC  . F  S Z=$O(IBX1(IBS,IBPO,Z)) Q:Z=""  S Q=IBPO F  S Q=$O(IBCOMB(IBS,Z,Q)) Q:'Q  I Q'=IBPO S IBZ1=$G(IBX1(IBS,IBPO,Z,1)),IBZ2=$G(IBX1(IBS,IBPO,Z,2)) D  .. Q:$G(IBX1(IBS,IBPO,Z,1))'=$G(IBX1(IBS,Q,Z,1))  .. S Q1=1,IBACC=$$ACCRV(+$P(IBS," ",2))  .. F Q0=1,5:1:7,10:1:13,15 D  Q:'Q1  ... I IBACC Q:Q0=5!(Q0>6)  ... I (Q0=11!(Q0=15))&($P($G(IBX1(IBS,Q,Z,2)),U,10)=3) Q  ... I Q0=5,'IBINPAT Q  ... I $P($G(IBX1(IBS,IBPO,Z,2)),U,Q0)'=$P($G(IBX1(IBS,Q,Z,2)),U,Q0) S Q1=0  .. Q:'Q1  .. S $P(IBX1(IBS,IBPO,Z,2),U,3)=$P(IBX1(IBS,IBPO,Z,2),U,3)+$P(IBX1(IBS,Q,Z,2),U,3)  .. S $P(IBX1(IBS,IBPO,Z,2),U,4)=$P(IBX1(IBS,IBPO,Z,2),U,4)+$P(IBX1(IBS,Q,Z,2),U,4)  .. S $P(IBX1(IBS,IBPO,Z,2),U,9)=$P(IBX1(IBS,IBPO,Z,2),U,9)+$P(IBX1(IBS,Q,Z,2),U,9)  .. S IBX1(IBS,IBPO,Z)=$P(IBX1(IBS,IBPO,Z,2),U,3)\_U\_$P(IBX1(IBS,IBPO,Z,2),U,4)  .. S IBX1(IBS,IBPO,Z,"IEN")=IBX1(IBS,IBPO,Z,"IEN")\_";"\_IBX1(IBS,Q,Z,"IEN")  .. K IBX1(IBS,Q,Z)  ;  D SPLIT  ; 488 ; baa  ;  ; Loop through IBX1 and build the array IBXDATA. Everything in the  ; array IBXDATA comes from the array IBX1.  ;  S IBS="",IBLN=0  F  S IBS=$O(IBX1(IBS)) Q:IBS=""  S IBPO=0 F  S IBPO=$O(IBX1(IBS,IBPO)) Q:'IBPO  S IBSS="" F  S IBSS=$O(IBX1(IBS,IBPO,IBSS)) Q:IBSS=""  D  . S IBX=$G(IBX1(IBS,IBPO,IBSS,1)),IBZ=$G(IBX1(IBS,IBPO,IBSS,2))  . S IBLN=$G(IBLN)+1,IBXDATA(IBLN)=$P(IBX,U)\_U\_$P(IBZ,U,6)\_U\_$P(IBZ,U,2)\_U\_+IBX1(IBS,IBPO,IBSS)\_U\_+$P(IBX1(IBS,IBPO,IBSS),U,2),$P(IBXDATA(IBLN),U,10)=$G(IBX1(IBS,IBPO,IBSS,"DT"))  . S $P(IBXDATA(IBLN),U,6)=$P(IBZ,U,9),$P(IBXDATA(IBLN),U,7)=$P(IBZ,U,13),$P(IBXDATA(IBLN),U,8)=$G(IBX1(IBS,IBPO,IBSS,"IEN")),$P(IBXDATA(IBLN),U,9)=$P($P(IBSS,U,3),",",1,2)  . S IBXDATA(IBLN,"CPLNK")=$$RC2CP(IBIFN,$P($P(IBXDATA(IBLN),U,8),";"))  . ;  . ; MRD;IB\*2.0\*516 - Added NDC and Units to line level of claim.  . I IBXDATA(IBLN,"CPLNK") S $P(IBXDATA(IBLN),U,11,12)=$TR($P($G(^DGCR(399,IBIFN,"CP",IBXDATA(IBLN,"CPLNK"),1)),U,7,8),"-")  . ;  . ; Extract line lev COB data for sec or tert bill  . I $$COBN^IBCEF(IBIFN)>1 D COBLINE^IBCEU6(IBIFN,IBLN,.IBXDATA,,.IBXTRA) I $D(IBXTRA) D COMBO^IBCEU2(.IBXDATA,.IBXTRA,1) ;Handle bundled/unbundled  ;  I $D(^IBA(362.4,"AIFN"\_IBIFN))!$D(^IBA(362.5,"AIFN"\_IBIFN)) D  . N IBARRAY,IBX,IBZ,IBRX,IBLCNT  . S IBLCNT=0  . ; Print prescriptions, prosthetics on front of UB-04  . D SET^IBCSC5A(IBIFN,.IBARRAY)  . I $P(IBARRAY,U,2) D  .. S IBX=+$P(IBARRAY,U,2)+2  .. S IBLCNT=IBLCNT+1,IBXSAVE("RX-UB-04",IBLCNT)=""  .. S IBLCNT=IBLCNT+1,IBXSAVE("RX-UB-04",IBLCNT)="PRESCRIPTION REFILLS:",IBLCNT=2  .. S IBX=0 F  S IBX=$O(IBARRAY(IBX)) Q:IBX=""  S IBY=0 F  S IBY=$O(IBARRAY(IBX,IBY)) Q:'IBY  S IBRX=IBARRAY(IBX,IBY) D  ... D ZERO^IBRXUTL(+$P(IBRX,U,2))  ... S IBLCNT=IBLCNT+1,IBXSAVE("RX-UB-04",IBLCNT)=IBX\_$J(" ",(11-$L(IBX)))\_" "\_$J($S($P(IBRX,U,6):"$"\_$FN($P(IBRX,U,6),",",2),1:""),10)\_" "\_$J($$FMTE^XLFDT(IBY,2),8)\_" "\_$G(^TMP($J,"IBDRUG",+$P(IBRX,U,2),.01))  ... S IBZ=$S(+$P(IBRX,U,4):"QTY: "\_$P(IBRX,U,4)\_" ",1:"")\_$S(+$P(IBRX,U,3):"for "\_$P(IBRX,U,3)\_" days supply ",1:"") I IBZ'="" S IBLCNT=IBLCNT+1,IBXSAVE("RX-UB-04",IBLCNT)=$J(" ",35)\_IBZ  ... S IBZ=$S($P(IBRX,U,5)'="":"NDC #: "\_$P(IBRX,U,5),1:"") I IBZ'="" S IBLCNT=IBLCNT+1,IBXSAVE("RX-UB-04",IBLCNT)=$J(" ",35)\_IBZ  ... K ^TMP($J,"IBDRUG")  ... Q  . ;  . D SET^IBCSC5B(IBIFN,.IBARRAY)  . I $P(IBARRAY,U,2) D  .. S IBLCNT=0  .. S IBX=+$P(IBARRAY,U,2)+2  .. S IBLCNT=IBLCNT+1,IBXSAVE("PROS-UB-04",IBLCNT)=""  .. S IBLCNT=IBLCNT+1,IBXSAVE("PROS-UB-04",IBLCNT)="PROSTHETIC REFILLS:",IBLCNT=2  .. S IBX=0 F  S IBX=$O(IBARRAY(IBX)) Q:IBX=""  S IBY=0 F  S IBY=$O(IBARRAY(IBX,IBY)) Q:'IBY  D  ... S IBLCNT=IBLCNT+1,IBXSAVE("PROS-UB-04",IBLCNT)=$$FMTE^XLFDT(IBX,2)\_" "\_$J($S($P(IBARRAY(IBX,IBY),U,2):"$"\_$FN($P(IBARRAY(IBX,IBY),U,2),",",2),1:""),10)\_" "\_$E($$PINB^IBCSC5B(+IBARRAY(IBX,IBY)),1,54)  Q  ; ACCRV(X) ; Returns 1 if X is an accomodation RC, 0 if not  Q ((X'<100&(X'>219))!(X=224))  ; RC2CP(IBIFN,IBRCIEN) ; returns "CP" multiple pointer that corresponds to a given "RC" multiple pointer in file 399  ; IBIFN - ien in file 399, top level  ; IBRCIEN, ien in sub-file 399.042 (REVENUE CODE)  ;  ; returns pointer to sub-file 399.0304 (PROCEDURES) or 0 if no valid pointer can be found.  ;  N IBRC0,IBCPIEN  I +IBIFN'>0 Q 0  I +IBRCIEN'>0 Q 0  S IBRC0=$G(^DGCR(399,IBIFN,"RC",IBRCIEN,0)),IBCPIEN=0  I $P(IBRC0,U,10)=4 S IBCPIEN=+$P(IBRC0,U,11) ; type = CPT  I $P(IBRC0,U,10)=3 S IBCPIEN=+$P(IBRC0,U,15) ; type = RX  I 'IBCPIEN D  . S IBRC=$P(IBRC0,U,6)  . N IBCPTIEN S IBCPTIEN=IBRC  . F  S IBCPTIEN=$O(^DGCR(399,IBIFN,"CP","B",IBCPTIEN)) Q:(+IBCPTIEN'=IBRC)!IBCPIEN  D  .. N OK,Z S OK="",Z=""  .. S Z=$O(^DGCR(399,IBIFN,"CP","B",IBCPTIEN,Z)) Q:'Z!OK  D  ... N CNTR S CNTR=0  ... F  S CNTR=$O(IBXDATA(CNTR)) Q:'CNTR!'OK  D  .... I $G(IBXDATA(CNTR,"CPLNK"))=Z S OK=0 Q  ... I OK="" S OK=1,IBCPIEN=Z  I IBCPIEN,'$D(^DGCR(399,IBIFN,"CP",IBCPIEN)) S IBCPIEN=0  Q IBCPIEN  ; SPLIT    ; Split codes into multiple lines as needed => baa ; 488  ; The max line $ amount for a printed claim is less than the max line $ amount for an electronically transmitted claim.  ; However, since the clearinghouse can drop to print for a myriad of reasons at any time, the lines may need to be split  ; so they can all fit on a printed claim line just in case. In addition, since some claims are sent to primary payers as   ; electronic claims but printed for secondary claims, the lines numbers need to be the same going out to ensure the   ; COB data is correct applied (previous payments adj, etc are applied to the correct line.)  N IBS,IBSS,DATA,CHRG,UNTS,TOT,LNS,MOD,CPT,LNK,RLNK,IBSS1,LTOT,LUNT,REC,LST,FST  S IBS="",IBLN=0  F  S IBS=$O(IBX1(IBS)) Q:IBS=""  D  . S LST=$O(IBX1(IBS,""),-1) ;we have to go through each level so must reset for each  . S LNK=0  . F  S LNK=$O(IBX1(IBS,LNK)) Q:('LNK!(LNK>LST)) S IBSS="" F  S IBSS=$O(IBX1(IBS,LNK,IBSS)) Q:IBSS=""  D  .. S DATA=IBX1(IBS,LNK,IBSS,2)  .. S CHRG=$P(DATA,U,2)  .. S UNTS=$P(DATA,U,3)  .. I UNTS=1 Q  ; if only one unit can't split  .. S TOT=UNTS\*CHRG  .. I TOT<=9999999.99 Q  ; if the total is less tham max we don't need to split  .. S LNS=TOT\9999999.99  .. S MOD=TOT#9999999.99  .. I MOD S LNS=LNS+1  .. I CHRG>4999999.995 S LNS=UNTS  ; if the charge is greater than half the mas can't put more than one on a line.  .. S LUNT=UNTS\LNS  .. S MOD=UNTS#LNS  .. I MOD S LUNT=LUNT+1  .. F L=1:1:LNS D  ... N Q  ... S Q=$O(IBX1(IBS,""),-1)+1  ... I L=1 S Q=LNK  ... M IBX1(IBS,Q,IBSS)=IBX1(IBS,LNK,IBSS)  ... S $P(IBX1(IBS,Q,IBSS,2),U,3)=LUNT,$P(IBX1(IBS,Q,IBSS,2),U,4)=LUNT\*CHRG  ... S $P(IBX1(IBS,Q,IBSS),U,1)=LUNT,$P(IBX1(IBS,Q,IBSS),U,2)=LUNT\*CHRG  ... I L>1 S $P(IBX1(IBS,Q,IBSS,2),U,9)=""  ... S UNTS=UNTS-LUNT,LUNT=$S(UNTS>LUNT:LUNT,1:UNTS)  Q | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCEF22 ;ALB/TMP - FORMATTER SPECIFIC BILL FUNCTIONS ;06-FEB-96  ;;2.0;INTEGRATED BILLING;\*\*51,137,135,155,309,349,389,432,488,516,577\*\*;21-MAR-94;Build 16  ;;Per VA Directive 6402, this routine should not be modified.  ;  ; OVERFLOW FROM ROUTINE IBCEF2 HOS(IBIFN) ; Extract rev codes for episode billed on a UB-04 into IBXDATA  ; IBIFN = bill ien  ; Format: IBXDATA(n) =  ; rev cd ptr ^ CPT CODE ptr ^ unit chg ^ units ^ tot charge  ; ^ tot uncov ^ FL49 value  ; ^ ien of rev code multiple entry(s) (separated by ";")  ; ^ modifiers specific to rev code/proc (separated by ",")  ; ^ rev code date, if it can be determined by a corresponding proc  ; ^ NDC from "CP" node of claim ^ Units/Quantity from "CP" node - vd/IB\*2\*577  ; ^ Units/Basis of Measurement for Drugs - vd/IB\*2\*577  ;  ; Also Returns IBXDATA(IBI,"COB",COB,m) with COB data for each line  ; item found in an accepted EOB for the bill and = the reference  ; line in the first '^' piece followed by the '0' node of file  ; 361.115 (LINE LEVEL ADJUSTMENTS)  ; COB = COB seq # of adjustment's ins co, m = seq #  ; -- AND --  ; IBXDATA(IBI,"COB",COB,m,z,p)=  ; the '0' node for each subordinate entry of file  ; 361.11511 (REASONS) (Only first 3 pieces for 837)  ; z = group code, sometimes preceeded by a space p = seq #  ;  ; -- AND --  ; IBXDATA(n,"CPLNK") = soft link to corresponding entry in PROCEDURES multiple of file 399  ;  N IBDA,IBCOMB,IBINPAT,IBLN,IBX,IBY,IBZ,IBS,IBSS,IBXTRA,IBX1,IBXS,IBP,IBPO,IBP1,IBDEF,Z,Z0,Z1,ZX,QQ,IBMOD,LST  S IBINPAT=$$INPAT^IBCEF(IBIFN,1)  I 'IBINPAT D F^IBCEF("N-STATEMENT COVERS FROM DATE","IBZ",,IBIFN)  S IBDEF=$G(IBZ),LST=""  ;  ; Loop through lines of claim beneath ^DGCR(399,IBIFN,"CP") and build  ; the array IBP to be used below.  ; IBP(Procedure ^ Modifiers, Print Order, Line#) = Procedure Date  ;  S IBDA=0 F  S IBDA=$O(^DGCR(399,IBIFN,"CP",IBDA)) Q:'IBDA  S IBZ=$G(^(IBDA,0)) I IBZ D  . S IBP(+$P(IBZ,U)\_U\_$$GETMOD^IBEFUNC(IBIFN,IBDA,1),$S($P(IBZ,U,4):$P(IBZ,U,4),1:999),IBDA)=$P(IBZ,U,2)  ;  ; Loop through the revenue codes beneath ^DGCR(399,IBIFN,"RC") and  ; build the array IBX to be used below.  ; IBX(" "\_Revenue Code, Print Order, Revenue Line#) =  ; ^DGCR(399.2, Revenue Code IEN, 0)  ; IBX(" "\_Revenue Code, Print Order, Revenue Line#, "DT") = Procedure Date  ; IBX(" "\_Revenue Code, Print Order, Revenue Line#, "MOD") = Modifiers  ;  S IBDA=0 F  S IBDA=$O(^DGCR(399,IBIFN,"RC",IBDA)) Q:'IBDA  S IBZ=$G(^(IBDA,0)) I IBZ S IBMOD="" D  . S IBX=$G(^DGCR(399.2,+IBZ,0)),IBX1="",IBPO=0  . ; Auto-added procedure charge  . I $P(IBZ,U,10)=4,$P(IBZ,U,11) D  ; Soft link to proc  .. S Z=$G(^DGCR(399,IBIFN,"CP",$P(IBZ,U,11),0))  .. Q:Z=""  .. S ZX=+Z\_U\_$$GETMOD^IBEFUNC(IBIFN,$P(IBZ,U,11),1)  .. Q:'$O(IBP(ZX,0))&'$O(IBP1(ZX,0))  .. I $P(IBZ,U,6) Q:$S($P(Z,U)'["ICPT":1,1:+$P(Z,U)'=$P(IBZ,U,6))  .. S Z0=$S($D(IBP(ZX)):$O(IBP(ZX,0)),1:$O(IBP1(ZX,0)))  .. S:'Z0 Z0=999  .. Q:'$D(IBP(ZX,+Z0,$P(IBZ,U,11)))&'$D(IBP1(ZX,+Z0,$P(IBZ,U,11)))  .. I '$D(IBP1(ZX,+Z0,$P(IBZ,U,11))) S IBP1(ZX,+Z0,$P(IBZ,U,11))=IBP(ZX,+Z0,$P(IBZ,U,11)) K IBP(ZX,+Z0,$P(IBZ,U,11))  .. S IBX1=$P(Z,U,2),IBPO=+Z0,IBMOD=$P(ZX,U,2)  . ; Manually added charge with a procedure  . I $P(IBZ,U,6),$S($P(IBZ,U,10)=4:'$P(IBZ,U,11),1:1),+$O(IBP($P(IBZ,U,6)))=$P(IBZ,U,6) D  .. ; No direct link, but a proc exists on rev code and in procedure mult without and then with modifiers  .. S ZX=$O(IBP($P(IBZ,U,6)))  .. F QQ=1,2 Q:IBPO  S Z="" F  S Z=$O(IBP(ZX,Z),-1) Q:'Z!(IBPO) S Z0=0 F  S Z0=$O(IBP(ZX,Z,Z0)) Q:'Z0  S Z1=$G(^DGCR(399,IBIFN,"CP",Z0,0)) D  Q:IBPO  ... ; Ignore if not a CPT or a modifier exists and this is first pass  ... S IBMOD=$$GETMOD^IBEFUNC(IBIFN,Z0,1)  ... Q:$S($P(Z1,U)'["ICPT":1,QQ=1:IBMOD'="",1:0)  ... S IBPO=+$P(Z1,U,4),IBX1=$P(Z1,U,2)  ... K IBP(+Z1\_U\_IBMOD,Z,Z0)  . ;  . I IBX'="" D  ; revenue code is valid  .. S LST=$S(LST="":900,1:LST+1)  .. F Z=LST:1 S Z0=$S(IBPO:IBPO,$D(IBX(" "\_$P(IBX,U),Z)):0,1:Z) I Z0 S (LST,IBPO)=Z0 Q  .. S IBX(" "\_$P(IBX,U),IBPO,IBDA)=IBX,IBX(" "\_$P(IBX,U),IBPO,IBDA,"DT")=$S(IBX1:IBX1,1:IBDEF),IBX(" "\_$P(IBX,U),IBPO,IBDA,"MOD")=IBMOD  ;  ; Loop through revenue codes in IBX and build the array IBX1.  ;  S IBS="" F  S IBS=$O(IBX(IBS)) Q:IBS=""  S IBPO=0 F  S IBPO=$O(IBX(IBS,IBPO)) Q:'IBPO  D  . S IBDA=0 F  S IBDA=$O(IBX(IBS,IBPO,IBDA)) Q:'IBDA  S IBX=$G(IBX(IBS,IBPO,IBDA)),IBZ=$G(^DGCR(399,IBIFN,"RC",IBDA,0)) I IBX'="" D  .. ;S IBXS=$P(IBZ,U,2)\_U\_$P(IBZ,U,6)\_U\_$G(IBX(IBS,IBPO,IBDA,"MOD"))  .. S IBXS=U\_$P(IBZ,U,6)\_U\_$G(IBX(IBS,IBPO,IBDA,"MOD")) ;combine same proc and modifiers regardless of rate  .. S:IBPO'<900&'$$ACCRV($P(IBS," ",2))&$S(IBINPAT:$P(IBZ,U,6),1:1) IBCOMB(IBS,IBXS,IBPO)=IBDA  .. S:'$D(IBX1(IBS,IBPO,IBXS,1)) IBX1(IBS,IBPO,IBXS,1)=IBX,IBX1(IBS,IBPO,IBXS,2)=IBZ  .. S $P(IBX1(IBS,IBPO,IBXS),U)=$P($G(IBX1(IBS,IBPO,IBXS)),U)+$P(IBZ,U,3)  .. S $P(IBX1(IBS,IBPO,IBXS),U,2)=$P($G(IBX1(IBS,IBPO,IBXS)),U,2)+$P(IBZ,U,4)  .. S IBX1(IBS,IBPO,IBXS,"DT")=$G(IBX(IBS,IBPO,IBDA,"DT")),IBX1(IBS,IBPO,IBXS,"IEN")=$G(IBX1(IBS,IBPO,IBXS,"IEN"))\_$S($G(IBX1(IBS,IBPO,IBXS,"IEN")):";",1:"")\_IBDA  ;  S IBS="" F  S IBS=$O(IBX1(IBS)) Q:IBS=""  S IBPO=899 F  S IBPO=$O(IBX1(IBS,IBPO)) Q:'IBPO  D  ; Check to combine like rev codes without print order  . N Q,Q0,Q1,Z,Z0,Z1,Z2,IBZ1,IBZ2  . S Z=""  . N IBACC  . F  S Z=$O(IBX1(IBS,IBPO,Z)) Q:Z=""  S Q=IBPO F  S Q=$O(IBCOMB(IBS,Z,Q)) Q:'Q  I Q'=IBPO S IBZ1=$G(IBX1(IBS,IBPO,Z,1)),IBZ2=$G(IBX1(IBS,IBPO,Z,2)) D  .. Q:$G(IBX1(IBS,IBPO,Z,1))'=$G(IBX1(IBS,Q,Z,1))  .. S Q1=1,IBACC=$$ACCRV(+$P(IBS," ",2))  .. F Q0=1,5:1:7,10:1:13,15 D  Q:'Q1  ... I IBACC Q:Q0=5!(Q0>6)  ... I (Q0=11!(Q0=15))&($P($G(IBX1(IBS,Q,Z,2)),U,10)=3) Q  ... I Q0=5,'IBINPAT Q  ... I $P($G(IBX1(IBS,IBPO,Z,2)),U,Q0)'=$P($G(IBX1(IBS,Q,Z,2)),U,Q0) S Q1=0  .. Q:'Q1  .. S $P(IBX1(IBS,IBPO,Z,2),U,3)=$P(IBX1(IBS,IBPO,Z,2),U,3)+$P(IBX1(IBS,Q,Z,2),U,3)  .. S $P(IBX1(IBS,IBPO,Z,2),U,4)=$P(IBX1(IBS,IBPO,Z,2),U,4)+$P(IBX1(IBS,Q,Z,2),U,4)  .. S $P(IBX1(IBS,IBPO,Z,2),U,9)=$P(IBX1(IBS,IBPO,Z,2),U,9)+$P(IBX1(IBS,Q,Z,2),U,9)  .. S IBX1(IBS,IBPO,Z)=$P(IBX1(IBS,IBPO,Z,2),U,3)\_U\_$P(IBX1(IBS,IBPO,Z,2),U,4)  .. S IBX1(IBS,IBPO,Z,"IEN")=IBX1(IBS,IBPO,Z,"IEN")\_";"\_IBX1(IBS,Q,Z,"IEN")  .. K IBX1(IBS,Q,Z)  ;  D SPLIT  ; 488 ; baa  ;  ; Loop through IBX1 and build the array IBXDATA. Everything in the  ; array IBXDATA comes from the array IBX1.  ;  S IBS="",IBLN=0  F  S IBS=$O(IBX1(IBS)) Q:IBS=""  S IBPO=0 F  S IBPO=$O(IBX1(IBS,IBPO)) Q:'IBPO  S IBSS="" F  S IBSS=$O(IBX1(IBS,IBPO,IBSS)) Q:IBSS=""  D  . S IBX=$G(IBX1(IBS,IBPO,IBSS,1)),IBZ=$G(IBX1(IBS,IBPO,IBSS,2))  . S IBLN=$G(IBLN)+1,IBXDATA(IBLN)=$P(IBX,U)\_U\_$P(IBZ,U,6)\_U\_$P(IBZ,U,2)\_U\_+IBX1(IBS,IBPO,IBSS)\_U\_+$P(IBX1(IBS,IBPO,IBSS),U,2),$P(IBXDATA(IBLN),U,10)=$G(IBX1(IBS,IBPO,IBSS,"DT"))  . S $P(IBXDATA(IBLN),U,6)=$P(IBZ,U,9),$P(IBXDATA(IBLN),U,7)=$P(IBZ,U,13),$P(IBXDATA(IBLN),U,8)=$G(IBX1(IBS,IBPO,IBSS,"IEN")),$P(IBXDATA(IBLN),U,9)=$P($P(IBSS,U,3),",",1,2)  . S IBXDATA(IBLN,"CPLNK")=$$RC2CP(IBIFN,$P($P(IBXDATA(IBLN),U,8),";"))  . ;  . ; MRD;IB\*2.0\*516 - Added NDC and Units to line level of claim.  . ;I IBXDATA(IBLN,"CPLNK") S $P(IBXDATA(IBLN),U,11,12)=$TR($P($G(^DGCR(399,IBIFN,"CP",IBXDATA(IBLN,"CPLNK"),1)),U,7,8),"-")  . ; VAD;IB\*2.0\*577 - Added Unit/Basis of Measurement to line level of claim.  . I IBXDATA(IBLN,"CPLNK") D  . . S $P(IBXDATA(IBLN),U,11,13)=$TR($P($G(^DGCR(399,IBIFN,"CP",IBXDATA(IBLN,"CPLNK"),1)),U,7,8),"-")\_U\_$P($G(^DGCR(399,IBIFN,"CP",IBXDATA(IBLN,"CPLNK"),2)),U)  . . I +$P(IBXDATA(IBLN),U,12) S $P(IBXDATA(IBLN),U,12)=$S($P(IBXDATA(IBLN),U,12)#1:+$J($P(IBXDATA(IBLN),U,12),0,3),1:$P(IBXDATA(IBLN),U,12))  . ;  . ; Extract line lev COB data for sec or tert bill  . I $$COBN^IBCEF(IBIFN)>1 D COBLINE^IBCEU6(IBIFN,IBLN,.IBXDATA,,.IBXTRA) I $D(IBXTRA) D COMBO^IBCEU2(.IBXDATA,.IBXTRA,1) ;Handle bundled/unbundled  ;  I $D(^IBA(362.4,"AIFN"\_IBIFN))!$D(^IBA(362.5,"AIFN"\_IBIFN)) D  . N IBARRAY,IBX,IBZ,IBRX,IBLCNT  . S IBLCNT=0  . ; Print prescriptions, prosthetics on front of UB-04  . D SET^IBCSC5A(IBIFN,.IBARRAY)  . I $P(IBARRAY,U,2) D  .. S IBX=+$P(IBARRAY,U,2)+2  .. S IBLCNT=IBLCNT+1,IBXSAVE("RX-UB-04",IBLCNT)=""  .. S IBLCNT=IBLCNT+1,IBXSAVE("RX-UB-04",IBLCNT)="PRESCRIPTION REFILLS:",IBLCNT=2  .. S IBX=0 F  S IBX=$O(IBARRAY(IBX)) Q:IBX=""  S IBY=0 F  S IBY=$O(IBARRAY(IBX,IBY)) Q:'IBY  S IBRX=IBARRAY(IBX,IBY) D  ... D ZERO^IBRXUTL(+$P(IBRX,U,2))  ... S IBLCNT=IBLCNT+1,IBXSAVE("RX-UB-04",IBLCNT)=IBX\_$J(" ",(11-$L(IBX)))\_" "\_$J($S($P(IBRX,U,6):"$"\_$FN($P(IBRX,U,6),",",2),1:""),10)\_" "\_$J($$FMTE^XLFDT(IBY,2),8)\_" "\_$G(^TMP($J,"IBDRUG",+$P(IBRX,U,2),.01))  ... S IBZ=$S(+$P(IBRX,U,4):"QTY: "\_$P(IBRX,U,4)\_" ",1:"")\_$S(+$P(IBRX,U,3):"for "\_$P(IBRX,U,3)\_" days supply ",1:"") I IBZ'="" S IBLCNT=IBLCNT+1,IBXSAVE("RX-UB-04",IBLCNT)=$J(" ",35)\_IBZ  ... S IBZ=$S($P(IBRX,U,5)'="":"NDC #: "\_$P(IBRX,U,5),1:"") I IBZ'="" S IBLCNT=IBLCNT+1,IBXSAVE("RX-UB-04",IBLCNT)=$J(" ",35)\_IBZ  ... K ^TMP($J,"IBDRUG")  ... Q  . ;  . D SET^IBCSC5B(IBIFN,.IBARRAY)  . I $P(IBARRAY,U,2) D  .. S IBLCNT=0  .. S IBX=+$P(IBARRAY,U,2)+2  .. S IBLCNT=IBLCNT+1,IBXSAVE("PROS-UB-04",IBLCNT)=""  .. S IBLCNT=IBLCNT+1,IBXSAVE("PROS-UB-04",IBLCNT)="PROSTHETIC REFILLS:",IBLCNT=2  .. S IBX=0 F  S IBX=$O(IBARRAY(IBX)) Q:IBX=""  S IBY=0 F  S IBY=$O(IBARRAY(IBX,IBY)) Q:'IBY  D  ... S IBLCNT=IBLCNT+1,IBXSAVE("PROS-UB-04",IBLCNT)=$$FMTE^XLFDT(IBX,2)\_" "\_$J($S($P(IBARRAY(IBX,IBY),U,2):"$"\_$FN($P(IBARRAY(IBX,IBY),U,2),",",2),1:""),10)\_" "\_$E($$PINB^IBCSC5B(+IBARRAY(IBX,IBY)),1,54)  Q  ; ACCRV(X) ; Returns 1 if X is an accomodation RC, 0 if not  Q ((X'<100&(X'>219))!(X=224))  ; RC2CP(IBIFN,IBRCIEN) ; returns "CP" multiple pointer that corresponds to a given "RC" multiple pointer in file 399  ; IBIFN - ien in file 399, top level  ; IBRCIEN, ien in sub-file 399.042 (REVENUE CODE)  ;  ; returns pointer to sub-file 399.0304 (PROCEDURES) or 0 if no valid pointer can be found.  ;  N IBRC0,IBCPIEN  I +IBIFN'>0 Q 0  I +IBRCIEN'>0 Q 0  S IBRC0=$G(^DGCR(399,IBIFN,"RC",IBRCIEN,0)),IBCPIEN=0  I $P(IBRC0,U,10)=4 S IBCPIEN=+$P(IBRC0,U,11) ; type = CPT  I $P(IBRC0,U,10)=3 S IBCPIEN=+$P(IBRC0,U,15) ; type = RX  I 'IBCPIEN D  . S IBRC=$P(IBRC0,U,6)  . N IBCPTIEN S IBCPTIEN=IBRC  . F  S IBCPTIEN=$O(^DGCR(399,IBIFN,"CP","B",IBCPTIEN)) Q:(+IBCPTIEN'=IBRC)!IBCPIEN  D  .. N OK,Z S OK="",Z=""  .. S Z=$O(^DGCR(399,IBIFN,"CP","B",IBCPTIEN,Z)) Q:'Z!OK  D  ... N CNTR S CNTR=0  ... F  S CNTR=$O(IBXDATA(CNTR)) Q:'CNTR!'OK  D  .... I $G(IBXDATA(CNTR,"CPLNK"))=Z S OK=0 Q  ... I OK="" S OK=1,IBCPIEN=Z  I IBCPIEN,'$D(^DGCR(399,IBIFN,"CP",IBCPIEN)) S IBCPIEN=0  Q IBCPIEN  ; SPLIT    ; Split codes into multiple lines as needed => baa ; 488  ; The max line $ amount for a printed claim is less than the max line $ amount for an electronically transmitted claim.  ; However, since the clearinghouse can drop to print for a myriad of reasons at any time, the lines may need to be split  ; so they can all fit on a printed claim line just in case. In addition, since some claims are sent to primary payers as   ; electronic claims but printed for secondary claims, the lines numbers need to be the same going out to ensure the   ; COB data is correct applied (previous payments adj, etc are applied to the correct line.)  N IBS,IBSS,DATA,CHRG,UNTS,TOT,LNS,MOD,CPT,LNK,RLNK,IBSS1,LTOT,LUNT,REC,LST,FST  S IBS="",IBLN=0  F  S IBS=$O(IBX1(IBS)) Q:IBS=""  D  . S LST=$O(IBX1(IBS,""),-1) ;we have to go through each level so must reset for each  . S LNK=0  . F  S LNK=$O(IBX1(IBS,LNK)) Q:('LNK!(LNK>LST)) S IBSS="" F  S IBSS=$O(IBX1(IBS,LNK,IBSS)) Q:IBSS=""  D  .. S DATA=IBX1(IBS,LNK,IBSS,2)  .. S CHRG=$P(DATA,U,2)  .. S UNTS=$P(DATA,U,3)  .. I UNTS=1 Q  ; if only one unit can't split  .. S TOT=UNTS\*CHRG  .. I TOT<=9999999.99 Q  ; if the total is less tham max we don't need to split  .. S LNS=TOT\9999999.99  .. S MOD=TOT#9999999.99  .. I MOD S LNS=LNS+1  .. I CHRG>4999999.995 S LNS=UNTS  ; if the charge is greater than half the mas can't put more than one on a line.  .. S LUNT=UNTS\LNS  .. S MOD=UNTS#LNS  .. I MOD S LUNT=LUNT+1  .. F L=1:1:LNS D  ... N Q  ... S Q=$O(IBX1(IBS,""),-1)+1  ... I L=1 S Q=LNK  ... M IBX1(IBS,Q,IBSS)=IBX1(IBS,LNK,IBSS)  ... S $P(IBX1(IBS,Q,IBSS,2),U,3)=LUNT,$P(IBX1(IBS,Q,IBSS,2),U,4)=LUNT\*CHRG  ... S $P(IBX1(IBS,Q,IBSS),U,1)=LUNT,$P(IBX1(IBS,Q,IBSS),U,2)=LUNT\*CHRG  ... I L>1 S $P(IBX1(IBS,Q,IBSS,2),U,9)=""  ... S UNTS=UNTS-LUNT,LUNT=$S(UNTS>LUNT:LUNT,1:UNTS)  Q | | | | | | | | | |

1. MODIFY the acceptable format of the UNITS field (#399.0304, 54) so that it’s format can be “99999999999.999”.

Change the field definition for the UNITS field FROM:

399.0304,54 UNITS 1;8 NUMBER

INPUT TRANSFORM: K:+X'=X!(X>9999999999)!(X<0)!(X?.E1"."9N.N) X

LAST EDITED: APR 20, 2017

HELP-PROMPT: Enter the number of units, a number between 1

and 9999999999 with up to 9 decimal digits.

DESCRIPTION: Enter the number of units of the

non-prescription medication administered.

TECHNICAL DESCR: Enter the number of units of the

non-prescription medication administered.

TO:

399.0304,54 UNITS 1;8 NUMBER

INPUT TRANSFORM: K:+X'=X!(X>99999999999)!(X<0)!((X[".")&

(X'?1.11N1"."1.3N)) X

LAST EDITED: APR 20, 2017

HELP-PROMPT: Enter a number between 0 and 9999999999 with

up to 3 decimal digits.

DESCRIPTION: Enter the number of units of the

non-prescription medication administered.

TECHNICAL DESCR: The number entered must be greater than zero

and have a format of 99999999999 and up to 3

decimal digits.

NOTES: XXXX--CAN'T BE ALTERED EXCEPT BY PROGRAMMER

TRIGGERED by the NDC field of the PROCEDURES

sub-field of the BILL/CLAIMS File

1. The FORMAT CODE of INS-16 and PRF-23 for the 837 Transmission (in the IB FORM FIELD CONTENT file [#364.7], entries 939 and 1949) needs to be changed remove the decimal point “.” from the output formatter.

* To modify INS-16 for the 837 Transmission (in the IB FORM FIELD CONTENT file [#364.7], entry 1949) the following change is to take place to remove the decimal point “.” from the output formatter. To do this the FORMAT CODE needed to change:

FROM:

PIECE 16 National Drug Unit Count 364.6[2236] **364.7[1949]** 364.5[5]

Length=15

Max Lines=0

>Constant Value: ""

>K IBXDATA N Z S Z=0 F S Z=$O(IBXSAVE("INPT",Z)) Q:'Z I $P(IBXSAVE(“INPT”,

Z),U,12)'="" S IBXDATA(Z)=$P(IBXSAVE("INPT",Z),U,12)

TO:

>K IBXDATA N Z S Z=0 F S Z=$O(IBXSAVE("INPT",Z)) Q:'Z I $P(IBXSAVE("INPT",

Z),U,12)'="" S IBXDATA(Z)=+$TR($J($P(IBXSAVE("INPT",Z),U,12),11,3)," .")

* To modify PRF-23 for the 837 Transmission (in the IB FORM FIELD CONTENT file [#364.7], entry 939) the following change is to take place to remove the decimal point “.” from the output formatter. To do this the FORMAT CODE needed to change:

FROM:

PIECE 23 National Drug Unit Count 364.6[977] 364.7[939] 364.5[5]

Length=15

Max Lines=0

>Constant Value: ""

>K IBXDATA N Z S Z=0 F S Z=$O(IBXSAVE("OUTPT",Z)) Q:'Z I $P(IBXSAVE("OUTPT",Z),

U,16)'="" S IBXDATA(Z)=$P(IBXSAVE("OUTPT",Z),U,16)

TO:

>K IBXDATA N Z S Z=0 F S Z=$O(IBXSAVE("OUTPT",Z)) Q:'Z I

$P(IBXSAVE("OUTPT",Z),U,16)'="" S IBXDATA(Z)=+$TR($J($P(IBXSAVE("OUTPT",Z),

U,16),11,3)," .")

1. CREATE the New ^IBY577PR Pre-Install Routine

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBY577PR | | | | | | | | |
| **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)** | | | | | | | | | |
| IBY577PR ;ALB/VD - Pre-Installation for IB patch 577 ; 4/06/17 4:33pm  ;;2.0;INTEGRATED BILLING;\*\*577\*\*;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  ; INC3508(Y) ; function to determine if entry in IB ERROR file (350.8) should be included in the build  ; Y - ien to file  N DATA,ENTRY,LN,OK,TAG  S OK=0,ENTRY=U\_$P($G(^IBE(350.8,Y,0)),U,3)\_U  F LN=2:1 S TAG="ENT3508+"\_LN,DATA=$P($T(@TAG),";;",2) Q:DATA=""  I $F(DATA,ENTRY) S OK=1 Q  Q OK  ; INCLUDE(FILE,Y) ; function to determine if O.F. entry should be included in the build  ; FILE=5,6,7 indicating file 364.x  ; Y=ien to file  NEW OK,LN,TAG,DATA  S OK=0  F LN=2:1 S TAG="ENT"\_FILE\_"+"\_LN,DATA=$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:  ;;^195^254^259^269^324^325^  ; Note: Must have beginning and ending up-carat  ;  ;-----------------------------------------------------------------------  ; 364.5 entries modified:  ; ENT5 ; OF entries in file 364.5 to be included  ;  ;;  ;  ;-----------------------------------------------------------------------  ; 364.6 entries modified:  ; ENT6 ; O.F. entries in file 364.6 to be included  ;  ;;  ;  ;-----------------------------------------------------------------------  ; 364.7 entries modified:  ;  ; 939 - PRF.23  ; 941 - PRF.25  ; 1949 - INS.16  ; 1950 - INS.17  ; ENT7 ; O.F. entries in file 364.7 to be included  ;  ;;^939^941^1949^1950^  ;  ;-----------------------------------------------------------------------  ; 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)  ;  ;;  ;  ;----------------------------------------------------------------------- ENT3508 ; Add New IB Error Codes to File 350.8  ;  ;;^IB360^  ; | | | | | | | | | |