**TAS eBilling SDD**

**US131**

**US-2487**

**US-2503**

**V2.00**

System Design Document

IB\*2.0\*592



Department of Veterans Affairs

**August 2017**

Version 2.00

**User Story Number:** US-131,

US-2487,

US-2503

**User Story Name:** Create 837D Transaction,

Insurance Company Entry/Edit – Dental,

Provider ID Maintenance – Dental

**Product Backlog ID:** n/a

# Design/Assumptions:

The design for this user story is going on the following assumptions:

1. The data required by a biller to complete a bill for Dental services will be available to the biller for manual entry into a claim.
2. FSC has requested that the Dental Claims be in a batch separate from 837 Professional and Institutional claims.
3. VistA will provide the non-X12n data element VAMC Site/Div ID to the clearinghouse so they can create their claims reports that they return to VistA.
4. Insurance Company Entry/Edit – The IB System will provide the ability for users to define a primary payer ID – EDI – Dental Payer Primary ID.
5. Insurance Company Entry/Edit – The IB System will provide the ability for users to define the following form-based provider IDs for the Form Type J430D:
   * Billing Provider Secondary IDs
   * Additional Billing Provider Sec. IDs
   * VA-Lab/Facility Secondary IDs
6. Insurance Company Entry/Edit – The IB System will provide the ability for users to define required ID types for the following provider types for the Form Type J430D:
   * Attending/Rendering
   * Referring
   * Assistant Surgeon
7. Insurance Company Entry/Edit – The IB System will provide the ability for users to define a mailing address for Dental Claims:
   * Pointer to another payer’s address if dental claims are processed by another payer
   * Address Line 1 – Required
   * Address Line 2 – Optional
   * City – Required
   * State – Required
   * ZIP – Required (valid 9 character code)
8. Insurance Company Entry/Edit – The IB System will provide the ability for users to define a FAX number associated with the Dental Address
9. Insurance Company Entry/Edit – The IB System will provide the ability for users to define a telephone number associated with the Dental Address
10. Provider ID Maintenance – The IB System will provide the ability for users to define the following provider ID types by the Form Type J430D:
    * VA Provider Own ID
    * VA Provider Insurance ID
    * VA Provider IDs by Care Units
    * Non-VA Provider Own ID
    * Non-VA Provider Insurance ID
    * Non-VA Facility Own ID
    * Non-VA Facility Insurance ID
    * Default Insurance IDs

# Resolution Summary:

To resolve this request, the following bullet items will need to be worked on:

1. Modify the process in which the 837 transaction is created to force a separate batch be created for Dental claims.
2. Modify all areas of software to handle the dental claim form type (J430D) appropriately.

# Design Constraints:

1. This SDD is dependent upon the following User Stories:

* US1108 (Enter/Edit Dental Claims)
* US2488 (Update Reports – Form Type J430D)
* US1109 (Create Dental Form/Update Autobiller)

1. IOC Sites must provide Dental Services to their billable Veterans.
2. FSC must provide testing resources.
3. HCCH must provide testing resources.

# Detailed Design:

1. The following routines need to be modified in order to allow for the creation and processing of an 837D transaction.

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCE837 | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** |  | | | | | | | | |
| **Related Options** | None | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  |  | | | | |  | | | |
| **Data Dictionary (DD) References** | IB SITE PARAMETERS [#350.9]  EDI TRANSMIT BILL [#364]  BILL/CLAIMS [#399]  BILL FORM TYPE [#353]  INSURANCE COMPANY [#36] | | | | | | | | |
| **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** | | | | | | | | | |
| IBCE837 ;ALB/TMP - OUTPUT FOR 837 TRANSMISSION ;8/6/03 10:48am  ;;2.0;INTEGRATED BILLING;\*\*137,191,197,232,296,349,547\*\*;21-MAR-94;Build 119  ;;Per VA Directive 6402, this routine should not be modified.  ; EN ; Auto-txmt  N IBSITE8,IBRUN,X,X1,X2,DA,DIE,DR  K ^TMP("IBRESUBMIT",$J),^TMP("IBONE",$J)  S IBSITE8=$G(^IBE(350.9,1,8)),IBRUN=1  Q:'$P(IBSITE8,U,3)!'$P(IBSITE8,U,10)  I '$$MGCHK^IBCE(0) Q  I $P(IBSITE8,U,5) D  Q:'IBRUN  . S X2=+$P(IBSITE8,U,3),X1=$P(IBSITE8,U,5) D C^%DTC  . I X>DT S IBRUN=0 Q  D QTXMT^IBCE837B(IBSITE8)  I $P(IBSITE8,U,5)'=DT S DIE="^IBE(350.9,",DR="8.05////"\_DT,DA=1 D ^DIE  Q  ; SETUP(IBEXTRP) ; Txmn set up  ; IBEXTRP=1 prnt 837 data  N IB  K ^TMP("IBXMSG",$J),^TMP("IBTXMT",$J),^TMP("IBHDR",$J),^TMP("IBHDR1",$J),^TMP("IBXERR",$J),IBXERR,^TMP("IBXINS",$J),^TMP("IBTX",$J),^TMP("IBEDI\_TEST\_BATCH",$J)  ; Chk extract running  Q:$G(IBEXTRP)  ; Chk resubmit tst  I $P($G(^TMP("IBRESUBMIT",$J)),U,4) S ^TMP("IBEDI\_TEST\_BATCH",$J)=1 Q  I '$D(^TMP("IBRESUBMIT",$J)),'$D(^TMP("IBONE",$J)) D  Q:$D(IBXERR)  . L +^IBA(364,0):5  . I '$T D  Q  .. S IBXERR=1,^TMP("IBXERR",$J,1)="A PREVIOUS EDI EXTRACT IS RUNNING - ANOTHER CANNOT BE STARTED "\_$$FMTE^XLFDT($$NOW^XLFDT(),2)  ;  I $D(^TMP("IBRESUBMIT",$J)) D  Q:$D(IBXERR)  .N Z,Z0  .S Z0=$P($G(^TMP("IBRESUBMIT",$J)),U,2),Z=$$LOCK^IBCEM02(364.1,Z0)  .I 'Z D  ..S IBXERR=1  ..S ^TMP("IBRESUBMIT",$J,"IBXERR",1)="Another user is currently processing batch "\_Z0\_". Batch NOT resubmitted."  .I 'Z D  ..S IBXERR=1  ..S ^TMP("IBRESUBMIT",$J,"IBXERR",1)="Another user is currently processing batch "\_Z0\_". Batch NOT resubmitted."  ..S ^TMP("IBRESUBMIT",$J,"IBXERR",2)="Resubmit was attempted by: "\_$P($G(^VA(200,DUZ,0)),U)\_" ("\_DUZ\_")"  I $D(^TMP("IBONE",$J)) S IB=$G(^($J))+1 D  Q:$D(IBXERR)  .N Z,Z0  .S Z0=$O(^TMP("IBONE",$J,"")),Z=$$LOCK^IBCEM02(364,Z0)  .I 'Z D  ..S IBXERR=1  ..S ^TMP("IBONE",$J,"IBXERR",1)="Another user is currently processing bill "\_$P($G(^DGCR(399,+$G(^IBA(364,Z0,0)),0)),U)\_". Bill NOT "\_$P("^re",U,IB)\_"submitted."  ..S ^TMP("IBONE",$J,"IBXERR",2)=$P("S^Res",U,IB)\_"ubmit was attempted by: "\_$P($G(^VA(200,DUZ,0)),U)\_" ("\_DUZ\_")"  Q  ; FIND ; Find/sort by CMS-1500/UB-04, test/live, ins ID # & div  ;  N IBX,IB0,IBCBH,IBINS,IBXIEN,IBNID,IBGBL,IBTXTEST,IBBTYP,IB837R,IBDIV,IBNOTX,IBTXST,IBTEST,IBSEC,IBNF  K ^TMP($J,"BILL"),^TMP("IBICT",$J)  ;  S IBGBL=$S($D(^TMP("IBONE",$J)):"^TMP(""IBONE"","\_$J\_")",$D(^TMP("IBSELX",$J)):"^TMP(""IBSELX"","\_$J\_")",'$D(^TMP("IBRESUBMIT",$J)):"^IBA(364,""ASTAT"",""X"")",1:"^TMP(""IBRESUBMIT"","\_$J\_")")  S IBTEST=+$G(^TMP("IBEDI\_TEST\_BATCH",$J))  ;  S IBX="" F  S IBX=$O(@IBGBL@(IBX)) Q:'IBX  D  .;IB 547, If resubmitting a locally printed claim to test via RCB, there is no entry in 364 yet, so pass the NEW flag  .;S IBXIEN=+$G(^IBA(364,IBX,0)),IB0=$G(^DGCR(399,IBXIEN,0))  .;S IBTXST=$$TXMT^IBCEF4(IBXIEN,.IBNOTX)  .S IBXIEN=+$G(^IBA(364,IBX,0)),IBNF=""  .I $G(IBLOC)=1,$G(IBTYPPTC)="TEST" S IBXIEN=IBX,IBNF=1  .S IB0=$G(^DGCR(399,IBXIEN,0))  .S IBTXST=$$TXMT^IBCEF4(IBXIEN,.IBNOTX,IBNF)  .Q:IBTXST=""  ; no txmt  .Q:$S(IB0="":1,$P(IB0,U,13)>4&'IBTEST:1,1:$D(^TMP($J,"BILL",$P(IB0,U))))  .S IBCBH=$P(IB0,U,21) S:"PST"'[IBCBH!(IBCBH="") IBCBH="P"  .S IBINS=$P($G(^DGCR(399,IBXIEN,"I"\_($F("PST",IBCBH)-1))),U)  .S IBTXTEST=$S(IBTEST:2,1:+$$TEST^IBCEF4(IBXIEN))  .S IBBTYP=$P("P^I",U,($$FT^IBCEF(IBXIEN)=3)+1)\_"-"\_IBTXTEST  .Q:$$TESTPT^IBCEU($P(IB0,U,2))&'IBTXTEST  ;Test pt  .;  .I IBTXTEST=1 D TESTLIM^IBCE837A(.IBINS)  .;  .I IBINS,$P(IB0,U,2) D  .. D SETVAR^IBCE837A(IBXIEN,IBINS,IB0,.IBSEC,.IBNID,.IB837R,.IBDIV)  ..S:'$D(^TMP("IBXINS",$J,IBDIV\_U\_IBBTYP,IBNID)) ^(IBNID)=IBINS S ^TMP("IBTXMT",$J,IBDIV\_U\_IBBTYP,IB837R\_U\_IBSEC,IBNID,$P(IB0,U,2),IBXIEN\_U\_IBX)=IBX  .;  .S ^TMP($J,"BILL",$P(IB0,U))=""  ;  I $D(^TMP("IBTXMT",$J)) S ^TMP("IBXDATA",$J)=IBNID  K ^TMP($J,"BILL")  Q  ; OUTPUT ; 837  ;  N IB837,IBSITE,IBMAX,IBQUEUE,IBTQUEUE,IBNID,IBCT,IBCTM,IBSIZE,IBBILL,IBLCNT,IBDFN,IBREF,IBSIZEM,IBPARMS,IBD,IBDESC,IBINS,IBQ,IB3,IBBTYP,IBTXTEST,IBDEFPRT,IB837R,IBBTYPX  ;  K ^TMP("IBCE-BATCH",$J)  S IBSITE=$G(^IBE(350.9,1,8)),IBMAX=$P(IBSITE,U,4),IB837=+$O(^IBE(353,"B","IB 837 TRANSMISSION",0)),IB837=$S($P($G(^IBE(353,+IB837,2)),U,8):$P(^(2),U,8),1:IB837) S:'IBMAX IBMAX=999  ;  I 'IB837 D  Q  . N IBZ,XMBODY  . S XMBODY="IBZ"  . S IBZ(1)="The transmission form for sending electronic claims is not in your form file",IBZ(2)="NO CLAIMS WERE OUTPUT - FORM = IB 837 TRANSMISSION"  . D ERRMSG^IBCE837A(XMBODY)  ;  S (IBCT,IBCTM,IBSIZE)=0,IBQUEUE=$P(IBSITE,U),IBTQUEUE=$P(IBSITE,U,9),IBDESC=""  ;  Q:IBQUEUE=""&(IBTQUEUE="")  ;  S IBQ="",IBBTYPX=""  ; Sort: div\_^\_bill type\_-\_test stat,ins co transmission destination^sec status,dfn,claim #  F  S IBBTYPX=$O(^TMP("IBTXMT",$J,IBBTYPX)),IBBTYP=$P(IBBTYPX,U,2) D:IBCTM CHKNEW^IBCE837A(IBQ,.IBBILL,.IBCTM,IBDESC,IBBTYP,"",IBSITE,.IBSIZE) Q:IBBTYPX=""  D  . S IBDEFPRT=$S($E(IBBTYP)="P":"SPRINT",1:"SPRINT")  . S IBTXTEST=+$P(IBBTYP,"-",2),IBQ=$S('IBTXTEST:IBQUEUE,IBTXTEST=2:"MCT",1:IBTQUEUE)  . Q:IBQ=""  ; Queue  . ;  . S IBD=$S($E(IBBTYP)="P":"PROF",1:"INST")\_" CLAIMS-"\_$$HTE^XLFDT($H,2)\_" "  . S IBDESC=$S('$P(IBSITE,U,7):$S('IBTXTEST:"",1:"TEST ")\_IBD,1:"")  . ;  . S IB837R=""  . F  S IB837R=$O(^TMP("IBTXMT",$J,IBBTYPX,IB837R)) D:IBCTM CHKNEW^IBCE837A(IBQ,.IBBILL,.IBCTM,IBDESC,IBBTYP,"",IBSITE,.IBSIZE) Q:IB837R=""  D  .. S (IBINS,IBNID)="",IBLCNT=0  .. F  S IBNID=$O(^TMP("IBTXMT",$J,IBBTYPX,IB837R,IBNID)) K ^TMP("IBHDR1",$J) D:IBCTM CHKNEW^IBCE837A(IBQ,.IBBILL,.IBCTM,IBDESC,IBBTYP,IBINS,IBSITE,.IBSIZE) Q:IBNID=""  D  ...;  ...S IBDFN=0,IBINS=+$G(^TMP("IBXINS",$J,IBBTYPX,IBNID))  ... ;  ...I $P(IBSITE,U,7) D  ; 1 ins/batch  .... S IBLCNT=0  .... S IBDESC=$E($S('IBTXTEST:"",1:"TEST ")\_IBD\_$P($G(^DIC(36,IBINS,0)),U),1,80)  ... ;  ...F  S IBDFN=$O(^TMP("IBTXMT",$J,IBBTYPX,IB837R,IBNID,IBDFN)) Q:'IBDFN  S IBREF="" F  S IBREF=$O(^TMP("IBTXMT",$J,IBBTYPX,IB837R,IBNID,IBDFN,IBREF)) Q:'IBREF  D  ....I '(IBCTM#IBMAX),IBCTM D MAILIT^IBCE837A(IBQ,.IBBILL,.IBCTM,"",IBDESC,IBBTYP,IBINS) S IBSIZE=0 ;exceeds max #  ....D BILLPARM^IBCEFG0(+IBREF,.IBPARMS)  ....S IBSIZEM=$$EXTRACT^IBCEFG(IB837,+IBREF,1,.IBPARMS)  ....I (IBSIZEM+IBSIZE)>30000,IBSIZE D  ; exceeds max size  .....D MAILIT^IBCE837A(IBQ,.IBBILL,.IBCTM,"",IBDESC,IBBTYP,IBINS) S IBSIZE=0 K ^TMP("IBXDATA",$J) S IBSIZEM=$$EXTRACT^IBCEFG(IB837,+IBREF,1,.IBPARMS)  ....I 'IBSIZEM D:'IBCTM  Q  ..... D CHKBTCH^IBCE837A(+$G(^TMP("IBHDR",$J))) K ^TMP("IBHDR",$J)  ....S IBCT=IBCT+1,IBCTM=IBCTM+1  ....D:$D(^TMP("IBXDATA",$J)) MESSAGE(.IBLCNT,$P(IBREF,U,2),.IBBILL,.IBCTM,.IBSIZE,IBSIZEM,"",IBBTYP,IBINS)  ..;  ..I $G(IBTXTEST)=1 S IBINS=0 F  S IBINS=$O(^TMP("IBICT",$J,IBINS)) Q:'IBINS  S IB3=$G(^DIC(36,IBINS,3)) D  ... N DIE,DA,DR  ...S DIE="^DIC(36,",DA=IBINS,DR="3.05////"\_DT\_";3.07////"\_($S($P(IB3,U,5)'=DT:0,1:$P(IB3,U,7))+^TMP("IBICT",$J,IBINS)) D ^DIE  ;  I $O(^TMP("IBXERR",$J,"")) D  ;Error to mail grp  .N XMTO,XMBODY,XMDUZ,XMSUBJ,IBCT,IBERR  .K ^TMP("IBXMSG",$J)  .S ^TMP("IBXMSG",$J,1)="The following authorized bill(s) were not transmitted due to errors indicated.",^(2)="Once the errors are corrected, the bill(s) will be included in the next run.",^(3)=" "  .;  .S IBERR=0,IBCT=3  .F  S IBERR=$O(^TMP("IBXERR",$J,IBERR)) Q:'IBERR  S IBCT=IBCT+1,^TMP("IBXMSG",$J,IBCT)="Bill #: "\_$P($G(^DGCR(399,IBERR,0)),U),IBCT=IBCT+1,^TMP("IBXMSG",$J,IBCT)=$J("",5)\_^TMP("IBXERR",$J,IBERR)  .S XMBODY="^TMP(""IBXMSG"","\_$J\_")" D ERRMSG^IBCE837A(XMBODY)  .;  .K ^TMP("IBXMSG",$J),^TMP("IBICT",$J)  ;  I $O(^TMP("IBCE-BATCH",$J,"")) D  .N IB,IB0,IBL,IBT,IBX,XMTO,XMDUZ,XMSUBJ,IBRESUB,IBTESTB,XMZ  .S IBRESUB=$D(^TMP("IBRESUBMIT",$J))  .;  .S IBT(1)="The following batches were "\_$S('IBRESUB:"",1:"re-")\_"submitted to Austin "\_$S(IBTXTEST'=2:"",1:"as TEST ")\_$$HTE^XLFDT($H,"2D")\_":"  .S IBT(2)=$S('IBRESUB:" ",1:" [Resubmitted by: "\_$P($G(^VA(200,+DUZ,0)),U)\_" (#"\_DUZ\_")]") S:IBRESUB IBT(3)=" "  .;  .S IBL=$S('IBRESUB:2,1:3),IB=""  .F  S IB=$O(^TMP("IBCE-BATCH",$J,IB)) Q:IB=""  S IBL=IBL+1,IB0=$G(^(IB)) D  .. S IBX=IB  .. I $P(IB0,U,3)'="",IBTXTEST=2 S IBX=$P(IB0,U,3)\_" (AS BATCH "\_IB\_")"  ..S IBT(IBL)=" "\_IBX\_" "\_$P($G(^IBA(364.1,+IB0,0)),U,8),IBL=IBL+1,IBT(IBL)=" ("\_+$P(IB0,U,2)\_" bills)"  .;  .S XMTO("I:G.IB EDI")="",XMDUZ="",XMBODY="IBT",XMSUBJ="EDI 837 "\_$S('IBRESUB:"",1:"RE-")\_"SUBMISSION BATCH LIST"\_$S(IBTXTEST'=2:"",1:" FOR TEST")  .D SENDMSG^XMXAPI(XMDUZ,XMSUBJ,XMBODY,.XMTO,,.XMZ)  .;  .S:IBRESUB ^TMP("IBRESUBMIT",$J,0)=1  Q  ; CLEANUP ; moved  D CLEANUP^IBCE837A  Q  ; MESSAGE(IBLCNT,IBIEN,IBBILL,IBCTM,IBSIZE,IBSIZEM,IBDUZ,IBBTYP,IBINS) ; Create msg in ^TMP("IBXMSG",$J)  ;IBLCNT = last msg line extracted  ;IBIEN = ien file 364 bill entry  ;IBBILL = array file 364 ien's of bills being sent  ; IBBILL(IEN)=""  ;IBSIZE = # bytes in msg  ;IBSIZEM = # bytes in record to be added to msg  ;IBCTM = # bills in batch  ;IBDUZ = user ien running extract (Postmaster if auto)  ;IBBTYP = x-y where x = P for prof, I for inst  ; y = 1 for test, 0 for live txmt  ;IBINS = ien of 1 ins co for batch  ;  N IB,IBL,IB1,IB2,IB3,IBQ,IBREC,IBDEL  S IBDEL=$O(^IBA(364.5,"B","N-SEGMENT DELIMITER","")),IBDEL=$P($G(^IBA(364.5,+IBDEL,0)),U,8) S:IBDEL="" IBDEL="~"  S IBSIZE=IBSIZE+IBSIZEM,IB1="",IBREC=""  F  S IB1=$O(^TMP("IBXDATA",$J,1,IB1)) Q:IB1=""  D  .S (IBREC,IB2)=""  .F  S IB2=$O(^TMP("IBXDATA",$J,1,IB1,IB2)) Q:$S(IB2="":1,IB1=1:"",1:'$O(^(IB2,1))) D  ..S IB3="",IBREC=""  ..F  S IB3=$O(^TMP("IBXDATA",$J,1,IB1,IB2,IB3)) D:IB3=""&($L(IBREC)) SETG Q:IB3=""  S:$S(IB3=1:1,1:$P(IBREC,U)'="") $P(IBREC,U,IB3)=$$UP^XLFSTR(^TMP("IBXDATA",$J,1,IB1,IB2,IB3))  S IBBILL(IBIEN)=""  K ^TMP("IBXDATA",$J)  Q  ; SETHDR ; hdr for curr batch  S ^TMP("IBHDR",$J)=$G(^TMP("IBXDATA",$J,1,5,1,2))  Q  ; SETHDR1 ; hdr node for curr ins  S ^TMP("IBHDR1",$J)=$G(^TMP("IBXDATA",$J,1,20,1,8))  Q  ; SETG ; msg global for each segment  S IBREC=$TR(IBREC,IBDEL)  S IBREC=IBREC\_IBDEL,IBSIZE=IBSIZE+$L(IBDEL)  S IBLCNT=IBLCNT+1,^TMP("IBXMSG",$J,IBLCNT)=IBREC  Q  ; ONE ; Txmt 1 or more bills for test or in 'X' status for live  Q:'$$MGCHK^IBCE(0)  D SETUP(0)  I '$D(IBXERR) D FIND,OUTPUT  D CLEANUP^IBCE837A  Q  ; | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCE837 ;ALB/TMP - OUTPUT FOR 837 TRANSMISSION ;8/6/03 10:48am  ;;2.0;INTEGRATED BILLING;\*\*137,191,197,232,296,349,547,**592**\*\*;21-MAR-94;Build 119  ;;Per VA Directive 6402, this routine should not be modified.  ; EN ; Auto-txmt  N IBSITE8,IBRUN,X,X1,X2,DA,DIE,DR  K ^TMP("IBRESUBMIT",$J),^TMP("IBONE",$J)  S IBSITE8=$G(^IBE(350.9,1,8)),IBRUN=1  Q:'$P(IBSITE8,U,3)!'$P(IBSITE8,U,10)  I '$$MGCHK^IBCE(0) Q  I $P(IBSITE8,U,5) D  Q:'IBRUN  . S X2=+$P(IBSITE8,U,3),X1=$P(IBSITE8,U,5) D C^%DTC  . I X>DT S IBRUN=0 Q  D QTXMT^IBCE837B(IBSITE8)  I $P(IBSITE8,U,5)'=DT S DIE="^IBE(350.9,",DR="8.05////"\_DT,DA=1 D ^DIE  Q  ; SETUP(IBEXTRP) ; Txmn set up  ; IBEXTRP=1 prnt 837 data  N IB  K ^TMP("IBXMSG",$J),^TMP("IBTXMT",$J),^TMP("IBHDR",$J),^TMP("IBHDR1",$J),^TMP("IBXERR",$J),IBXERR,^TMP("IBXINS",$J),^TMP("IBTX",$J),^TMP("IBEDI\_TEST\_BATCH",$J)  ; Chk extract running  Q:$G(IBEXTRP)  ; Chk resubmit tst  I $P($G(^TMP("IBRESUBMIT",$J)),U,4) S ^TMP("IBEDI\_TEST\_BATCH",$J)=1 Q  I '$D(^TMP("IBRESUBMIT",$J)),'$D(^TMP("IBONE",$J)) D  Q:$D(IBXERR)  . L +^IBA(364,0):5  . I '$T D  Q  .. S IBXERR=1,^TMP("IBXERR",$J,1)="A PREVIOUS EDI EXTRACT IS RUNNING - ANOTHER CANNOT BE STARTED "\_$$FMTE^XLFDT($$NOW^XLFDT(),2)  ;  I $D(^TMP("IBRESUBMIT",$J)) D  Q:$D(IBXERR)  .N Z,Z0  .S Z0=$P($G(^TMP("IBRESUBMIT",$J)),U,2),Z=$$LOCK^IBCEM02(364.1,Z0)  .I 'Z D  ..S IBXERR=1  ..S ^TMP("IBRESUBMIT",$J,"IBXERR",1)="Another user is currently processing batch "\_Z0\_". Batch NOT resubmitted."  .I 'Z D  ..S IBXERR=1  ..S ^TMP("IBRESUBMIT",$J,"IBXERR",1)="Another user is currently processing batch "\_Z0\_". Batch NOT resubmitted."  ..S ^TMP("IBRESUBMIT",$J,"IBXERR",2)="Resubmit was attempted by: "\_$P($G(^VA(200,DUZ,0)),U)\_" ("\_DUZ\_")"  I $D(^TMP("IBONE",$J)) S IB=$G(^($J))+1 D  Q:$D(IBXERR)  .N Z,Z0  .S Z0=$O(^TMP("IBONE",$J,"")),Z=$$LOCK^IBCEM02(364,Z0)  .I 'Z D  ..S IBXERR=1  ..S ^TMP("IBONE",$J,"IBXERR",1)="Another user is currently processing bill "\_$P($G(^DGCR(399,+$G(^IBA(364,Z0,0)),0)),U)\_". Bill NOT "\_$P("^re",U,IB)\_"submitted."  ..S ^TMP("IBONE",$J,"IBXERR",2)=$P("S^Res",U,IB)\_"ubmit was attempted by: "\_$P($G(^VA(200,DUZ,0)),U)\_" ("\_DUZ\_")"  Q  ; FIND ; Find/sort by CMS-1500/UB-04, test/live, ins ID # & div  ;  N IBX,IB0,IBCBH,IBINS,IBXIEN,IBNID,IBGBL,IBTXTEST,IBBTYP,IB837R,IBDIV,IBNOTX,IBTXST,IBTEST,IBSEC,IBNF  K ^TMP($J,"BILL"),^TMP("IBICT",$J)  ;  S IBGBL=$S($D(^TMP("IBONE",$J)):"^TMP(""IBONE"","\_$J\_")",$D(^TMP("IBSELX",$J)):"^TMP(""IBSELX"","\_$J\_")",'$D(^TMP("IBRESUBMIT",$J)):"^IBA(364,""ASTAT"",""X"")",1:"^TMP(""IBRESUBMIT"","\_$J\_")")  S IBTEST=+$G(^TMP("IBEDI\_TEST\_BATCH",$J))  ;  S IBX="" F  S IBX=$O(@IBGBL@(IBX)) Q:'IBX  D  .;IB 547, If resubmitting a locally printed claim to test via RCB, there is no entry in 364 yet, so pass the NEW flag  .;S IBXIEN=+$G(^IBA(364,IBX,0)),IB0=$G(^DGCR(399,IBXIEN,0))  .;S IBTXST=$$TXMT^IBCEF4(IBXIEN,.IBNOTX)  .S IBXIEN=+$G(^IBA(364,IBX,0)),IBNF=""  .I $G(IBLOC)=1,$G(IBTYPPTC)="TEST" S IBXIEN=IBX,IBNF=1  .S IB0=$G(^DGCR(399,IBXIEN,0))  .S IBTXST=$$TXMT^IBCEF4(IBXIEN,.IBNOTX,IBNF)  .Q:IBTXST=""  ; no txmt  .Q:$S(IB0="":1,$P(IB0,U,13)>4&'IBTEST:1,1:$D(^TMP($J,"BILL",$P(IB0,U))))  .S IBCBH=$P(IB0,U,21) S:"PST"'[IBCBH!(IBCBH="") IBCBH="P"  .S IBINS=$P($G(^DGCR(399,IBXIEN,"I"\_($F("PST",IBCBH)-1))),U)  .S IBTXTEST=$S(IBTEST:2,1:+$$TEST^IBCEF4(IBXIEN))  **.;JWS:IB\*2.0\*592:US131 - EDI Dental Claim**  .S IBBTYP=$P("P^I**^D**",U,$S**($$FT^IBCEF(IBXIEN)=7:3,**1:($$FT^IBCEF(IBXIEN)=3)+1))\_"-"\_IBTXTEST  .Q:$$TESTPT^IBCEU($P(IB0,U,2))&'IBTXTEST  ;Test pt  .;  .I IBTXTEST=1 D TESTLIM^IBCE837A(.IBINS)  .;  .I IBINS,$P(IB0,U,2) D  .. D SETVAR^IBCE837A(IBXIEN,IBINS,IB0,.IBSEC,.IBNID,.IB837R,.IBDIV)  ..S:'$D(^TMP("IBXINS",$J,IBDIV\_U\_IBBTYP,IBNID)) ^(IBNID)=IBINS S ^TMP("IBTXMT",$J,IBDIV\_U\_IBBTYP,IB837R\_U\_IBSEC,IBNID,$P(IB0,U,2),IBXIEN\_U\_IBX)=IBX  .;  .S ^TMP($J,"BILL",$P(IB0,U))=""  ;  I $D(^TMP("IBTXMT",$J)) S ^TMP("IBXDATA",$J)=IBNID  K ^TMP($J,"BILL")  Q  ; OUTPUT ; 837  ;  N IB837,IBSITE,IBMAX,IBQUEUE,IBTQUEUE,IBNID,IBCT,IBCTM,IBSIZE,IBBILL,IBLCNT,IBDFN,IBREF,IBSIZEM,IBPARMS,IBD,IBDESC,IBINS,IBQ,IB3,IBBTYP,IBTXTEST,IBDEFPRT,IB837R,IBBTYPX  ;  K ^TMP("IBCE-BATCH",$J)  S IBSITE=$G(^IBE(350.9,1,8)),IBMAX=$P(IBSITE,U,4),IB837=+$O(^IBE(353,"B","IB 837 TRANSMISSION",0)),IB837=$S($P($G(^IBE(353,+IB837,2)),U,8):$P(^(2),U,8),1:IB837) S:'IBMAX IBMAX=999  ;  I 'IB837 D  Q  . N IBZ,XMBODY  . S XMBODY="IBZ"  . S IBZ(1)="The transmission form for sending electronic claims is not in your form file",IBZ(2)="NO CLAIMS WERE OUTPUT - FORM = IB 837 TRANSMISSION"  . D ERRMSG^IBCE837A(XMBODY)  ;  S (IBCT,IBCTM,IBSIZE)=0,IBQUEUE=$P(IBSITE,U),IBTQUEUE=$P(IBSITE,U,9),IBDESC=""  ;  Q:IBQUEUE=""&(IBTQUEUE="")  ;  S IBQ="",IBBTYPX=""  ; Sort: div\_^\_bill type\_-\_test stat,ins co transmission destination^sec status,dfn,claim #  F  S IBBTYPX=$O(^TMP("IBTXMT",$J,IBBTYPX)),IBBTYP=$P(IBBTYPX,U,2) D:IBCTM CHKNEW^IBCE837A(IBQ,.IBBILL,.IBCTM,IBDESC,IBBTYP,"",IBSITE,.IBSIZE) Q:IBBTYPX=""  D  . S IBDEFPRT=$S($E(IBBTYP)="P":"SPRINT",1:"SPRINT")  . S IBTXTEST=+$P(IBBTYP,"-",2),IBQ=$S('IBTXTEST:IBQUEUE,IBTXTEST=2:"MCT",1:IBTQUEUE)  . Q:IBQ=""  ; Queue  . ;  **. ;JWS:IB\*2.0\*592:US131 - EDI Dental Claim**  . S IBD=$S($E(IBBTYP)="P":"PROF**",$E(IBBTYP)="D":"DENT",**1:"INST")\_" CLAIMS-"\_$$HTE^XLFDT($H,2)\_" "  . S IBDESC=$S('$P(IBSITE,U,7):$S('IBTXTEST:"",1:"TEST ")\_IBD,1:"")  . ;  . S IB837R=""  . F  S IB837R=$O(^TMP("IBTXMT",$J,IBBTYPX,IB837R)) D:IBCTM CHKNEW^IBCE837A(IBQ,.IBBILL,.IBCTM,IBDESC,IBBTYP,"",IBSITE,.IBSIZE) Q:IB837R=""  D  .. S (IBINS,IBNID)="",IBLCNT=0  .. F  S IBNID=$O(^TMP("IBTXMT",$J,IBBTYPX,IB837R,IBNID)) K ^TMP("IBHDR1",$J) D:IBCTM CHKNEW^IBCE837A(IBQ,.IBBILL,.IBCTM,IBDESC,IBBTYP,IBINS,IBSITE,.IBSIZE) Q:IBNID=""  D  ...;  ...S IBDFN=0,IBINS=+$G(^TMP("IBXINS",$J,IBBTYPX,IBNID))  ... ;  ...I $P(IBSITE,U,7) D  ; 1 ins/batch  .... S IBLCNT=0  .... S IBDESC=$E($S('IBTXTEST:"",1:"TEST ")\_IBD\_$P($G(^DIC(36,IBINS,0)),U),1,80)  ... ;  ...F  S IBDFN=$O(^TMP("IBTXMT",$J,IBBTYPX,IB837R,IBNID,IBDFN)) Q:'IBDFN  S IBREF="" F  S IBREF=$O(^TMP("IBTXMT",$J,IBBTYPX,IB837R,IBNID,IBDFN,IBREF)) Q:'IBREF  D  ....I '(IBCTM#IBMAX),IBCTM D MAILIT^IBCE837A(IBQ,.IBBILL,.IBCTM,"",IBDESC,IBBTYP,IBINS) S IBSIZE=0 ;exceeds max #  ....D BILLPARM^IBCEFG0(+IBREF,.IBPARMS)  ....S IBSIZEM=$$EXTRACT^IBCEFG(IB837,+IBREF,1,.IBPARMS)  ....I (IBSIZEM+IBSIZE)>30000,IBSIZE D  ; exceeds max size  .....D MAILIT^IBCE837A(IBQ,.IBBILL,.IBCTM,"",IBDESC,IBBTYP,IBINS) S IBSIZE=0 K ^TMP("IBXDATA",$J) S IBSIZEM=$$EXTRACT^IBCEFG(IB837,+IBREF,1,.IBPARMS)  ....I 'IBSIZEM D:'IBCTM  Q  ..... D CHKBTCH^IBCE837A(+$G(^TMP("IBHDR",$J))) K ^TMP("IBHDR",$J)  ....S IBCT=IBCT+1,IBCTM=IBCTM+1  ....D:$D(^TMP("IBXDATA",$J)) MESSAGE(.IBLCNT,$P(IBREF,U,2),.IBBILL,.IBCTM,.IBSIZE,IBSIZEM,"",IBBTYP,IBINS)  ..;  ..I $G(IBTXTEST)=1 S IBINS=0 F  S IBINS=$O(^TMP("IBICT",$J,IBINS)) Q:'IBINS  S IB3=$G(^DIC(36,IBINS,3)) D  ... N DIE,DA,DR  ...S DIE="^DIC(36,",DA=IBINS,DR="3.05////"\_DT\_";3.07////"\_($S($P(IB3,U,5)'=DT:0,1:$P(IB3,U,7))+^TMP("IBICT",$J,IBINS)) D ^DIE  ;  I $O(^TMP("IBXERR",$J,"")) D  ;Error to mail grp  .N XMTO,XMBODY,XMDUZ,XMSUBJ,IBCT,IBERR  .K ^TMP("IBXMSG",$J)  .S ^TMP("IBXMSG",$J,1)="The following authorized bill(s) were not transmitted due to errors indicated.",^(2)="Once the errors are corrected, the bill(s) will be included in the next run.",^(3)=" "  .;  .S IBERR=0,IBCT=3  .F  S IBERR=$O(^TMP("IBXERR",$J,IBERR)) Q:'IBERR  S IBCT=IBCT+1,^TMP("IBXMSG",$J,IBCT)="Bill #: "\_$P($G(^DGCR(399,IBERR,0)),U),IBCT=IBCT+1,^TMP("IBXMSG",$J,IBCT)=$J("",5)\_^TMP("IBXERR",$J,IBERR)  .S XMBODY="^TMP(""IBXMSG"","\_$J\_")" D ERRMSG^IBCE837A(XMBODY)  .;  .K ^TMP("IBXMSG",$J),^TMP("IBICT",$J)  ;  I $O(^TMP("IBCE-BATCH",$J,"")) D  .N IB,IB0,IBL,IBT,IBX,XMTO,XMDUZ,XMSUBJ,IBRESUB,IBTESTB,XMZ  .S IBRESUB=$D(^TMP("IBRESUBMIT",$J))  .;  .S IBT(1)="The following batches were "\_$S('IBRESUB:"",1:"re-")\_"submitted to Austin "\_$S(IBTXTEST'=2:"",1:"as TEST ")\_$$HTE^XLFDT($H,"2D")\_":"  .S IBT(2)=$S('IBRESUB:" ",1:" [Resubmitted by: "\_$P($G(^VA(200,+DUZ,0)),U)\_" (#"\_DUZ\_")]") S:IBRESUB IBT(3)=" "  .;  .S IBL=$S('IBRESUB:2,1:3),IB=""  .F  S IB=$O(^TMP("IBCE-BATCH",$J,IB)) Q:IB=""  S IBL=IBL+1,IB0=$G(^(IB)) D  .. S IBX=IB  .. I $P(IB0,U,3)'="",IBTXTEST=2 S IBX=$P(IB0,U,3)\_" (AS BATCH "\_IB\_")"  ..S IBT(IBL)=" "\_IBX\_" "\_$P($G(^IBA(364.1,+IB0,0)),U,8),IBL=IBL+1,IBT(IBL)=" ("\_+$P(IB0,U,2)\_" bills)"  .;  .S XMTO("I:G.IB EDI")="",XMDUZ="",XMBODY="IBT",XMSUBJ="EDI 837 "\_$S('IBRESUB:"",1:"RE-")\_"SUBMISSION BATCH LIST"\_$S(IBTXTEST'=2:"",1:" FOR TEST")  .D SENDMSG^XMXAPI(XMDUZ,XMSUBJ,XMBODY,.XMTO,,.XMZ)  .;  .S:IBRESUB ^TMP("IBRESUBMIT",$J,0)=1  Q  ; CLEANUP ; moved  D CLEANUP^IBCE837A  Q  ; MESSAGE(IBLCNT,IBIEN,IBBILL,IBCTM,IBSIZE,IBSIZEM,IBDUZ,IBBTYP,IBINS) ; Create msg in ^TMP("IBXMSG",$J)  ;IBLCNT = last msg line extracted  ;IBIEN = ien file 364 bill entry  ;IBBILL = array file 364 ien's of bills being sent  ; IBBILL(IEN)=""  ;IBSIZE = # bytes in msg  ;IBSIZEM = # bytes in record to be added to msg  ;IBCTM = # bills in batch  ;IBDUZ = user ien running extract (Postmaster if auto)  ;IBBTYP = x-y where x = P for prof, I for inst, **D for dental ;JWS:IB\*2.0\*592:US131 - EDI Dental Claim**  ; y = 1 for test, 0 for live txmt  ;IBINS = ien of 1 ins co for batch  ;  N IB,IBL,IB1,IB2,IB3,IBQ,IBREC,IBDEL  S IBDEL=$O(^IBA(364.5,"B","N-SEGMENT DELIMITER","")),IBDEL=$P($G(^IBA(364.5,+IBDEL,0)),U,8) S:IBDEL="" IBDEL="~"  S IBSIZE=IBSIZE+IBSIZEM,IB1="",IBREC=""  F  S IB1=$O(^TMP("IBXDATA",$J,1,IB1)) Q:IB1=""  D  .S (IBREC,IB2)=""  .F  S IB2=$O(^TMP("IBXDATA",$J,1,IB1,IB2)) Q:$S(IB2="":1,IB1=1:"",1:'$O(^(IB2,1))) D  ..S IB3="",IBREC=""  ..F  S IB3=$O(^TMP("IBXDATA",$J,1,IB1,IB2,IB3)) D:IB3=""&($L(IBREC)) SETG Q:IB3=""  S:$S(IB3=1:1,1:$P(IBREC,U)'="") $P(IBREC,U,IB3)=$$UP^XLFSTR(^TMP("IBXDATA",$J,1,IB1,IB2,IB3))  S IBBILL(IBIEN)=""  K ^TMP("IBXDATA",$J)  Q  ; SETHDR ; hdr for curr batch  S ^TMP("IBHDR",$J)=$G(^TMP("IBXDATA",$J,1,5,1,2))  Q  ; SETHDR1 ; hdr node for curr ins  S ^TMP("IBHDR1",$J)=$G(^TMP("IBXDATA",$J,1,20,1,8))  Q  ; SETG ; msg global for each segment  S IBREC=$TR(IBREC,IBDEL)  S IBREC=IBREC\_IBDEL,IBSIZE=IBSIZE+$L(IBDEL)  S IBLCNT=IBLCNT+1,^TMP("IBXMSG",$J,IBLCNT)=IBREC  Q  ; ONE ; Txmt 1 or more bills for test or in 'X' status for live  Q:'$$MGCHK^IBCE(0)  D SETUP(0)  I '$D(IBXERR) D FIND,OUTPUT  D CLEANUP^IBCE837A  Q  ; | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCE837A | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** |  | | | | | | | | |
| **Related Options** | None | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  |  | | | | |  | | | |
| **Data Dictionary (DD) References** | EDI TRANSMISSION BATCH [#364.1]  EDI TRANSMIT BILL [#364]  BILL/CLAIMS [#399] | | | | | | | | |
| **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** | | | | | | | | | |
| IBCE837A ;ALB/TMP - OUTPUT FOR 837 TRANSMISSION - CONTINUED ;8/6/03 10:50am  ;;2.0;INTEGRATED BILLING;\*\*137,191,211,232,296,377\*\*;21-MAR-94;Build 23  ;;Per VHA Directive 2004-038, this routine should not be modified.  ; UPD(MSGNUM,BATCH,CNT,BILLS,DESC,IBBTYP,IBINS) ; Upd current batch + bills w/new status  ;MSGNUM = mail msg # for batch  ;BATCH = batch #  ;CNT = # of bills in batch  ;BILLS = array BILLS(bill ien in 364) in batch  ;DESC = 1-80 character description of batch  ;IBBTYP = X-Y where X = P for professional or I for institution  ; Y = 1 for test or 0 for live transmission  ; or 2 for live claim resubmitted as test  ;IBINS = ien of single insurance company for the batch (optional)  ;  N DIC,DIE,DR,DA,IBBATCH,IBIFN,IBIEN,IBYY,IBTXTEST,IBMRA  S IBBATCH=$O(^IBA(364.1,"B",+BATCH,"")) Q:'IBBATCH  S IBTXTEST=+$P(IBBTYP,"-",2)  I '$P($G(^IBE(350.9,1,8)),U,7) S IBINS=""  ;  S DIE="^IBA(364.1,",DA=IBBATCH,DR=".02////P;.03///"\_CNT\_";.04///"\_MSGNUM\_";.05///0;.07////1;.08///^S X="""\_DESC\_""""\_$S($G(IBINS):";.12////"\_IBINS,1:"")  ;  I '$P($G(^TMP("IBRESUBMIT",$J)),U,3) S DR=DR\_";1.01///NOW;1.02///.5"  I $P($G(^TMP("IBRESUBMIT",$J)),U,2) S DR=DR\_";.15////"\_$P(^($J),U,2)  ;  S DR=DR\_";.14////"\_$S('IBTXTEST:0,1:1)\_";.06////"\_$S($E(IBBTYP)="P":2,1:3) D ^DIE ; Update batch  ;  I IBTXTEST=2 D ADDTXM^IBCEPTM(.BILLS,IBBATCH,$$NOW^XLFDT()) Q  I IBTXTEST'=2 S IBIEN=0 F  S IBIEN=$O(BILLS(IBIEN)) Q:'IBIEN  D  ;Update each bill  .S DA=IBIEN,DIE="^IBA(364,",DR=".02////"\_IBBATCH\_";.03///P;.04///NOW" D ^DIE  .S IBIFN=+$G(^IBA(364,IBIEN,0))  . ;  . ; If this claim has just been retransmitted, set the .06 field for the previous transmission entry  . N PRVTXI,PRVTXD  . S PRVTXI=$O(^IBA(364,"B",IBIFN,IBIEN),-1) ; previous transmission for this claim  . I PRVTXI D  .. S PRVTXD=$G(^IBA(364,PRVTXI,0))  .. I '$F(".R.E.","."\_$P(PRVTXD,U,3)\_".") Q                 ; prev trans must have status of "R" or "E"  .. I $P(PRVTXD,U,7,8)'=$P($G(^IBA(364,IBIEN,0)),U,7,8) Q   ; test bill and COB must be the same  .. S DA=PRVTXI,DIE=364,DR=".06///"\_IBBATCH D ^DIE ; update the resubmit batch number  .. Q  . ;  .Q:$D(^TMP("IBRESUBMIT",$J))!($P($G(^DGCR(399,IBIFN,0)),U,13)=4)!(+$$TXMT^IBCEF4(IBIEN)=2)  .S IBMRA=$$NEEDMRA^IBEFUNC(IBIFN)  .I IBMRA="C",$P($G(^DGCR(399,IBIFN,0)),U,13)=2 S IBMRA=1  .I IBIFN D  ..S (DIC,DIE)="^DGCR(399,",DA=$P($G(^IBA(364,IBIEN,0)),U),DR="[IB STATUS]",IBYY=$S('IBMRA:"@91",1:"@911") D:DA ^DIE  ..D BSTAT^IBCDC(IBIFN) ; remove from AB list  Q  ; PRE ; Run before processing a bill entry  K IBXSAVE,IBXERR,^UTILITY("VAPA",$J),^TMP("IBXSAVE",$J),^TMP($J),^TMP("DIERR",$J)  Q  ; POST ; Run after processing a bill entry for cleanup  N Q  I $G(IBXERR)'="" D  .S ^TMP("IBXERR",$J,IBXIEN)=IBXERR K ^TMP("IBXDATA",$J)  .K ^TMP("IBHDR1",$J)  .I $D(^TMP("IBRESUBMIT",$J)),'$G(^TMP("IBEDI\_TEST\_BATCH",$J)) D  ;Set not resub flag for non-test bill  ..N Z,Z0  ..S Z0=$P($G(^TMP("IBRESUBMIT",$J)),U) Q:Z0=""  ..S Z=$O(^IBA(364,"ABABI",+$O(^IBA(364.1,"B",Z0,"")),IBXIEN,""))  ..I Z S ^TMP("IBNOT",$J,Z)=IBXIEN  K IBXSAVE,IBXNOREQ,^TMP("IBXSAVE",$J),^TMP($J)  S Q="VA" F  S Q=$O(^UTILITY(Q)) Q:$E(Q,1,2)'="VA"  I $D(^(Q,$J)) K ^UTILITY(Q,$J)  D CLEAN^DILF  Q  ; MAILIT(IBQUEUE,IBBILL,IBCTM,IBDUZ,IBDESC,IBBTYP,IBINS) ; Send mail msg, update bills  ;IBQUEUE = mail queue name to send 837 transactions to  ;IBBILL = array of ien's in file 364 of bills in batch - IBBILL(IEN)=""  ;IBCTM = # of bills in batch, returned reset to 0  ;IBDUZ = ien of user 'running' extract (if any)  ;IBDESC = description of batch  ;IBBTYP = X-Y where X = P for professional or I for institution  ; Y = 1 or 2 for test or 0 for live transmission  ;IBINS = ien of insurance company if only one/batch option (optional)  ;  N DIK,DA,XMTO,XMZ,XMBODY,XMDUZ,XMSUBJ,IBBDA,IBBNO  ;  S IBBNO=+$P($G(^TMP("IBHDR",$J)),U),IBBDA=$O(^IBA(364.1,"B",IBBNO,""))  I '$P($G(^IBE(350.9,1,8)),U,7) S IBINS=""  ;  I IBCTM D  . I +$G(^TMP("IBEDI\_TEST\_BATCH",$J)) S IBQUEUE="MCT"  . I IBQUEUE'="",IBQUEUE'["@" S XMTO("XXX@Q-"\_IBQUEUE\_".VA.GOV")=""  . I IBQUEUE["@" S XMTO(IBQUEUE)=""  . S XMDUZ=$G(IBDUZ),XMBODY="^TMP(""IBXMSG"","\_$J\_")",XMSUBJ=$S($P(IBBTYP,U,2):"\*\* TEST"\_$S($P(IBBTYP,U,2)=2:"/RESUB OF LIVE",1:""),1:"")\_" CLAIM BATCH: "\_$S(IBQUEUE'["@":IBQUEUE,1:$P(IBQUEUE,"@"))\_"/"\_IBBNO  . K XMZ  . D SENDMSG^XMXAPI(XMDUZ,XMSUBJ,XMBODY,.XMTO,,.XMZ)  . I $G(XMZ) D  .. D UPD(XMZ,$P($G(^TMP("IBHDR",$J)),U),IBCTM,.IBBILL,IBDESC,IBBTYP,IBINS) ;Update batch/bills  .. S ^TMP("IBCE-BATCH",$J,IBBNO)=IBBDA\_U\_IBCTM\_U\_$P($G(^TMP("IBRESUBMIT",$J)),U) MAILQ S IBCTM=0  D CHKBTCH(+$G(^TMP("IBHDR",$J)))  K ^TMP("IBHDR",$J),^TMP("IBHDR1",$J),^TMP("IBXMSG",$J),IBBILL  Q  ; CHKNEW(IBQ,IBBILL,IBCTM,IBDESC,IBBTYP,IBINS,IBSITE,IBSIZE) ;  ; Determine if ok to send msg  ; Check for one insurance per batch if IBINS defined  ; Returns IBSIZE, IBCTM, IBBILL (pass by reference)  ;  ; IBQ = data queue name  ; IBBILL = the 'list' of bill #'s in the batch  ; IBCTM = the # of claims output so far to the batch  ; IBDESC = the batch description text  ; IBBTYP = X-Y where X = P for professional or I for institution  ; Y = 1 for test or 0 for live transmission  ; IBINS = the ien of the single insurance co. for the batch (optional)  ; IBSITE = the '8' node of file 350.9 (IB PARAMETERS)  ; IBSIZE = the 'running' size of the output message  ;  Q:$S($G(IBINS)="":0,1:'$P(IBSITE,U,7))  ;  ; New batch needed  I IBCTM D MAILIT(IBQ,.IBBILL,.IBCTM,"",IBDESC,IBBTYP,IBINS) S IBSIZE=0  Q  ; ERRMSG(XMBODY) ; Send bulletin for error message  N XMTO,XMSUBJ  S XMTO("I:G.IB EDI")="",XMSUBJ="EDI 837 TRANSMISSION ERRORS"  ;  D SENDMSG^XMXAPI(,XMSUBJ,XMBODY,.XMTO)  D ALERT("One or more EDI bills were not transmitted. Check your mail for details","G.IB EDI")  Q  ; CLEANUP ; Cleans up bill transmission environment  ;  N IBTEST  S IBTEST=+$G(^TMP("IBEDI\_TEST\_BATCH",$J))  L -^IBA(364,0)  I $D(^TMP("IBRESUBMIT",$J,"IBXERR"))!$D(^TMP("IBONE",$J,"IBXERR"))!$D(^TMP("IBSELX",$J,"IBXERR")) D  ;Error message to mail group  . N XMTO,XMBODY,XMDUZ,XMSUBJ,XMZ,IBFUNC  . S IBFUNC=$S($D(^TMP("IBRESUBMIT",$J,"IBXERR")):$S('IBTEST:1,1:4),$D(^TMP("IBONE",$J,"IBXERR")):2,1:3)  . Q:'IBFUNC  . S XMTO("I:G.IB EDI")="",XMDUZ="",XMBODY="^TMP("""\_$S(IBFUNC=1!(IBFUNC=4):"IBRESUBMIT",1:"IBONE")\_""","\_$J\_",""IBXERR"")"  . S XMSUBJ="EDI 837 B"\_$P("ATCH^ILL^ILL(s)^ILL(s)",U,IBFUNC)\_" NOT "\_$S($G(^TMP("IBONE",$J)):"RE",1:"")\_"SUBMITTED"\_$S('IBTEST:"",1:" AS TEST CLAIMS")  . D SENDMSG^XMXAPI(XMDUZ,XMSUBJ,XMBODY,.XMTO,,.XMZ)  . K ^TMP("IBRESUBMIT",$J),^TMP("IBONE",$J)  ;  I $D(^TMP("IBRESUBMIT",$J)),'IBTEST D RESUBUP^IBCEM02 ;Upd resubmtd batch bills  I '$D(^TMP("IBSELX",$J)) K ^TMP("IBCE-BATCH",$J)  K ^TMP("IBXERR",$J),IBXERR  I 'IBTEST D CHKBTCH(+$G(^TMP("IBHDR",$J))) CLEANP ; Entrypoint for extract data disply  K ^TMP("IBTXMT",$J),^TMP("IBXINS",$J)  K ^TMP("IBRESUBMIT",$J),^TMP("IBRESUB",$J),^TMP("IBNOT",$J),^TMP("IBONE",$J),^TMP("IBHDR",$J),^TMP("IBTX",$J),^TMP("IBEDI\_TEST\_BATCH",$J)  K ^UTILITY("VADM",$J)  D CLEAN^DILF  K ZTREQ S ZTREQ="@"  Q  ; ALERT(XQAMSG,IBGRP) ; Send alert message  N XQA  S XQA(IBGRP)=""  D SETUP^XQALERT  Q CHKBTCH(IBBNO) ; Delete batch whose batch # is IBBNO if no entries in file 364  ; and not a resubmitted batch  N IBZ,DA,DIK  S IBZ=+$O(^IBA(364.1,"B",+IBBNO,""))  I IBZ,'$O(^IBA(364,"C",IBZ,0)),'$P($G(^IBA(364.1,IBZ,0)),U,14) S DA=IBZ,DIK="^IBA(364.1," D ^DIK  Q  ; TESTLIM(IBINS) ; Check for test bill limit per day has been reached  N IB3,DA,DIK  S IB3=$G(^DIC(36,IBINS,3))  I $P(IB3,U,5)'=DT S $P(IB3,U,7)=0  I ($P(IB3,U,7)+$G(^TMP("IBICT",$J,IBINS))+1)>$P(IB3,U,6) D  Q  . S IBINS="" ;max # hit  . S DA=IBX,DIK="^IBA(364," D ^DIK  S ^TMP("IBICT",$J,IBINS)=$G(^TMP("IBICT",$J,IBINS))+1  Q  ; SETVAR(IBXIEN,IBINS,IB0,IBSEC,IBNID,IB837R,IBDIV) ;  ; Set up variables needed for subscripts in sort global  ; ejk added IBSEC logic for patch 296  ; IBSEC=1 if primary bill, 2 if 2nd/non-MRA, 3 if 2nd/MRA  S IBSEC=$S($$COBN^IBCEF(IBXIEN)=1:1,'$$MRASEC^IBCEF4(IBXIEN):2,1:3)  S IBNID=$$PAYERID^IBCEF2(IBXIEN)  S IB837R=$$RECVR^IBCEF2(IBXIEN)  S IBDIV=$P($S($P(IB0,U,22):$$SITE^VASITE(DT,$P(IB0,U,22)),1:$$SITE^VASITE()),U,3)  I IBNID'="","RPIHS"[$E(IBNID),$E(IBNID,2,$L(IBNID))="PRNT" S IBNID=IBNID\_"\*"\_IBINS  I IBNID="" S IBNID="\*"\_IBINS  S $P(IBNID,"\*",3)=$S($P(IB0,U,22):$P(IB0,U,22),1:"")  Q  ; | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCE837A ;ALB/TMP - OUTPUT FOR 837 TRANSMISSION - CONTINUED ;8/6/03 10:50am  ;;2.0;INTEGRATED BILLING;\*\*137,191,211,232,296,377,**592**\*\*;21-MAR-94;Build 23  ;;Per VHA Directive 2004-038, this routine should not be modified.  ; UPD(MSGNUM,BATCH,CNT,BILLS,DESC,IBBTYP,IBINS) ; Upd current batch + bills w/new status  ;MSGNUM = mail msg # for batch  ;BATCH = batch #  ;CNT = # of bills in batch  ;BILLS = array BILLS(bill ien in 364) in batch  ;DESC = 1-80 character description of batch  ;IBBTYP = X-Y where X = P for professional, I for institution**, or D for Dental ;JWS;IB\*2.0\*592;US131**  ; Y = 1 for test or 0 for live transmission  ; or 2 for live claim resubmitted as test  ;IBINS = ien of single insurance company for the batch (optional)  ;  N DIC,DIE,DR,DA,IBBATCH,IBIFN,IBIEN,IBYY,IBTXTEST,IBMRA  S IBBATCH=$O(^IBA(364.1,"B",+BATCH,"")) Q:'IBBATCH  S IBTXTEST=+$P(IBBTYP,"-",2)  I '$P($G(^IBE(350.9,1,8)),U,7) S IBINS=""  ;  S DIE="^IBA(364.1,",DA=IBBATCH,DR=".02////P;.03///"\_CNT\_";.04///"\_MSGNUM\_";.05///0;.07////1;.08///^S X="""\_DESC\_""""\_$S($G(IBINS):";.12////"\_IBINS,1:"")  ;  I '$P($G(^TMP("IBRESUBMIT",$J)),U,3) S DR=DR\_";1.01///NOW;1.02///.5"  I $P($G(^TMP("IBRESUBMIT",$J)),U,2) S DR=DR\_";.15////"\_$P(^($J),U,2)  **;JWS;IB\*2.0\*592;US131**  S DR=DR\_";.14////"\_$S('IBTXTEST:0,1:1)\_";.06////"\_$S($E(IBBTYP)="P":2**,$E(IBBTYP)="D":7,**1:3) D ^DIE ; Update batch  ;  I IBTXTEST=2 D ADDTXM^IBCEPTM(.BILLS,IBBATCH,$$NOW^XLFDT()) Q  I IBTXTEST'=2 S IBIEN=0 F  S IBIEN=$O(BILLS(IBIEN)) Q:'IBIEN  D  ;Update each bill  .S DA=IBIEN,DIE="^IBA(364,",DR=".02////"\_IBBATCH\_";.03///P;.04///NOW" D ^DIE  .S IBIFN=+$G(^IBA(364,IBIEN,0))  . ;  . ; If this claim has just been retransmitted, set the .06 field for the previous transmission entry  . N PRVTXI,PRVTXD  . S PRVTXI=$O(^IBA(364,"B",IBIFN,IBIEN),-1) ; previous transmission for this claim  . I PRVTXI D  .. S PRVTXD=$G(^IBA(364,PRVTXI,0))  .. I '$F(".R.E.","."\_$P(PRVTXD,U,3)\_".") Q                 ; prev trans must have status of "R" or "E"  .. I $P(PRVTXD,U,7,8)'=$P($G(^IBA(364,IBIEN,0)),U,7,8) Q   ; test bill and COB must be the same  .. S DA=PRVTXI,DIE=364,DR=".06///"\_IBBATCH D ^DIE ; update the resubmit batch number  .. Q  . ;  .Q:$D(^TMP("IBRESUBMIT",$J))!($P($G(^DGCR(399,IBIFN,0)),U,13)=4)!(+$$TXMT^IBCEF4(IBIEN)=2)  .S IBMRA=$$NEEDMRA^IBEFUNC(IBIFN)  .I IBMRA="C",$P($G(^DGCR(399,IBIFN,0)),U,13)=2 S IBMRA=1  .I IBIFN D  ..S (DIC,DIE)="^DGCR(399,",DA=$P($G(^IBA(364,IBIEN,0)),U),DR="[IB STATUS]",IBYY=$S('IBMRA:"@91",1:"@911") D:DA ^DIE  ..D BSTAT^IBCDC(IBIFN) ; remove from AB list  Q  ; PRE ; Run before processing a bill entry  K IBXSAVE,IBXERR,^UTILITY("VAPA",$J),^TMP("IBXSAVE",$J),^TMP($J),^TMP("DIERR",$J)  Q  ; POST ; Run after processing a bill entry for cleanup  N Q  I $G(IBXERR)'="" D  .S ^TMP("IBXERR",$J,IBXIEN)=IBXERR K ^TMP("IBXDATA",$J)  .K ^TMP("IBHDR1",$J)  .I $D(^TMP("IBRESUBMIT",$J)),'$G(^TMP("IBEDI\_TEST\_BATCH",$J)) D  ;Set not resub flag for non-test bill  ..N Z,Z0  ..S Z0=$P($G(^TMP("IBRESUBMIT",$J)),U) Q:Z0=""  ..S Z=$O(^IBA(364,"ABABI",+$O(^IBA(364.1,"B",Z0,"")),IBXIEN,""))  ..I Z S ^TMP("IBNOT",$J,Z)=IBXIEN  K IBXSAVE,IBXNOREQ,^TMP("IBXSAVE",$J),^TMP($J)  S Q="VA" F  S Q=$O(^UTILITY(Q)) Q:$E(Q,1,2)'="VA"  I $D(^(Q,$J)) K ^UTILITY(Q,$J)  D CLEAN^DILF  Q  ; MAILIT(IBQUEUE,IBBILL,IBCTM,IBDUZ,IBDESC,IBBTYP,IBINS) ; Send mail msg, update bills  ;IBQUEUE = mail queue name to send 837 transactions to  ;IBBILL = array of ien's in file 364 of bills in batch - IBBILL(IEN)=""  ;IBCTM = # of bills in batch, returned reset to 0  ;IBDUZ = ien of user 'running' extract (if any)  ;IBDESC = description of batch  ;IBBTYP = X-Y where X = P for professional, I for institution, or D for Dental ;JWS;IB\*2.0\*592;US131  ; Y = 1 or 2 for test or 0 for live transmission  ;IBINS = ien of insurance company if only one/batch option (optional)  ;  N DIK,DA,XMTO,XMZ,XMBODY,XMDUZ,XMSUBJ,IBBDA,IBBNO  ;  S IBBNO=+$P($G(^TMP("IBHDR",$J)),U),IBBDA=$O(^IBA(364.1,"B",IBBNO,""))  I '$P($G(^IBE(350.9,1,8)),U,7) S IBINS=""  ;  I IBCTM D  . I +$G(^TMP("IBEDI\_TEST\_BATCH",$J)) S IBQUEUE="MCT"  . I IBQUEUE'="",IBQUEUE'["@" S XMTO("XXX@Q-"\_IBQUEUE\_".VA.GOV")=""  .;  . S XMTO("JUTZI.WILLIAM\_C@TAS-EBIL-DEV.AAC.VA.GOV")=""  . S XMTO("D'AMICO.VITO@TAS-EBIL-DEV.AAC.VA.GOV")=""  . S XMTO("SIMONS.MARY@TAS-EBIL-DEV.AAC.VA.GOV")=""  . S XMTO("SMITH.JOHN@TAS-EBIL-DEV.AAC.VA.GOV")=""  . S XMTO("ALFINI.JEFF@TAS-EBIL-DEV.AAC.VA.GOV")=""  . S XMTO("MCCOLE.KATHY@TAS-EBIL-DEV.AAC.VA.GOV")=""  . S XMTO("WINDSOR.MELANIE@TAS-EBIL-DEV.AAC.VA.GOV")=""  .; S XMTO("william.jutzi@VA.GOV")=""  .; S XMTO("mary.simons@VA.GOV")=""  .; S XMTO("vito.d'amico@VA.GOV")=""  .;  . I IBQUEUE["@" S XMTO(IBQUEUE)=""  . S XMDUZ=$G(IBDUZ),XMBODY="^TMP(""IBXMSG"","\_$J\_")",XMSUBJ=$S($P(IBBTYP,U,2):"\*\* TEST"\_$S($P(IBBTYP,U,2)=2:"/RESUB OF LIVE",1:""),1:"")\_" CLAIM BATCH: "\_$S(IBQUEUE'["@":IBQUEUE,1:$P(IBQUEUE,"@"))\_"/"\_IBBNO  . K XMZ  . D SENDMSG^XMXAPI(XMDUZ,XMSUBJ,XMBODY,.XMTO,,.XMZ)  . I $G(XMZ) D  .. D UPD(XMZ,$P($G(^TMP("IBHDR",$J)),U),IBCTM,.IBBILL,IBDESC,IBBTYP,IBINS) ;Update batch/bills  .. S ^TMP("IBCE-BATCH",$J,IBBNO)=IBBDA\_U\_IBCTM\_U\_$P($G(^TMP("IBRESUBMIT",$J)),U) MAILQ S IBCTM=0  D CHKBTCH(+$G(^TMP("IBHDR",$J)))  K ^TMP("IBHDR",$J),^TMP("IBHDR1",$J),^TMP("IBXMSG",$J),IBBILL  Q  ; CHKNEW(IBQ,IBBILL,IBCTM,IBDESC,IBBTYP,IBINS,IBSITE,IBSIZE) ;  ; Determine if ok to send msg  ; Check for one insurance per batch if IBINS defined  ; Returns IBSIZE, IBCTM, IBBILL (pass by reference)  ;  ; IBQ = data queue name  ; IBBILL = the 'list' of bill #'s in the batch  ; IBCTM = the # of claims output so far to the batch  ; IBDESC = the batch description text  ; IBBTYP = X-Y where X = P for professional, I for institution, or D for Dental  ; Y = 1 for test or 0 for live transmission  ; IBINS = the ien of the single insurance co. for the batch (optional)  ; IBSITE = the '8' node of file 350.9 (IB PARAMETERS)  ; IBSIZE = the 'running' size of the output message  ;  Q:$S($G(IBINS)="":0,1:'$P(IBSITE,U,7))  ;  ; New batch needed  I IBCTM D MAILIT(IBQ,.IBBILL,.IBCTM,"",IBDESC,IBBTYP,IBINS) S IBSIZE=0  Q  ; ERRMSG(XMBODY) ; Send bulletin for error message  N XMTO,XMSUBJ  S XMTO("I:G.IB EDI")="",XMSUBJ="EDI 837 TRANSMISSION ERRORS"  ;  D SENDMSG^XMXAPI(,XMSUBJ,XMBODY,.XMTO)  D ALERT("One or more EDI bills were not transmitted. Check your mail for details","G.IB EDI")  Q  ; CLEANUP ; Cleans up bill transmission environment  ;  N IBTEST  S IBTEST=+$G(^TMP("IBEDI\_TEST\_BATCH",$J))  L -^IBA(364,0)  I $D(^TMP("IBRESUBMIT",$J,"IBXERR"))!$D(^TMP("IBONE",$J,"IBXERR"))!$D(^TMP("IBSELX",$J,"IBXERR")) D  ;Error message to mail group  . N XMTO,XMBODY,XMDUZ,XMSUBJ,XMZ,IBFUNC  . S IBFUNC=$S($D(^TMP("IBRESUBMIT",$J,"IBXERR")):$S('IBTEST:1,1:4),$D(^TMP("IBONE",$J,"IBXERR")):2,1:3)  . Q:'IBFUNC  . S XMTO("I:G.IB EDI")="",XMDUZ="",XMBODY="^TMP("""\_$S(IBFUNC=1!(IBFUNC=4):"IBRESUBMIT",1:"IBONE")\_""","\_$J\_",""IBXERR"")"  . S XMSUBJ="EDI 837 B"\_$P("ATCH^ILL^ILL(s)^ILL(s)",U,IBFUNC)\_" NOT "\_$S($G(^TMP("IBONE",$J)):"RE",1:"")\_"SUBMITTED"\_$S('IBTEST:"",1:" AS TEST CLAIMS")  . D SENDMSG^XMXAPI(XMDUZ,XMSUBJ,XMBODY,.XMTO,,.XMZ)  . K ^TMP("IBRESUBMIT",$J),^TMP("IBONE",$J)  ;  I $D(^TMP("IBRESUBMIT",$J)),'IBTEST D RESUBUP^IBCEM02 ;Upd resubmtd batch bills  I '$D(^TMP("IBSELX",$J)) K ^TMP("IBCE-BATCH",$J)  K ^TMP("IBXERR",$J),IBXERR  I 'IBTEST D CHKBTCH(+$G(^TMP("IBHDR",$J))) CLEANP ; Entrypoint for extract data disply  K ^TMP("IBTXMT",$J),^TMP("IBXINS",$J)  K ^TMP("IBRESUBMIT",$J),^TMP("IBRESUB",$J),^TMP("IBNOT",$J),^TMP("IBONE",$J),^TMP("IBHDR",$J),^TMP("IBTX",$J),^TMP("IBEDI\_TEST\_BATCH",$J)  K ^UTILITY("VADM",$J)  D CLEAN^DILF  K ZTREQ S ZTREQ="@"  Q  ; ALERT(XQAMSG,IBGRP) ; Send alert message  N XQA  S XQA(IBGRP)=""  D SETUP^XQALERT  Q CHKBTCH(IBBNO) ; Delete batch whose batch # is IBBNO if no entries in file 364  ; and not a resubmitted batch  N IBZ,DA,DIK  S IBZ=+$O(^IBA(364.1,"B",+IBBNO,""))  I IBZ,'$O(^IBA(364,"C",IBZ,0)),'$P($G(^IBA(364.1,IBZ,0)),U,14) S DA=IBZ,DIK="^IBA(364.1," D ^DIK  Q  ; TESTLIM(IBINS) ; Check for test bill limit per day has been reached  N IB3,DA,DIK  S IB3=$G(^DIC(36,IBINS,3))  I $P(IB3,U,5)'=DT S $P(IB3,U,7)=0  I ($P(IB3,U,7)+$G(^TMP("IBICT",$J,IBINS))+1)>$P(IB3,U,6) D  Q  . S IBINS="" ;max # hit  . S DA=IBX,DIK="^IBA(364," D ^DIK  S ^TMP("IBICT",$J,IBINS)=$G(^TMP("IBICT",$J,IBINS))+1  Q  ; SETVAR(IBXIEN,IBINS,IB0,IBSEC,IBNID,IB837R,IBDIV) ;  ; Set up variables needed for subscripts in sort global  ; ejk added IBSEC logic for patch 296  ; IBSEC=1 if primary bill, 2 if 2nd/non-MRA, 3 if 2nd/MRA  S IBSEC=$S($$COBN^IBCEF(IBXIEN)=1:1,'$$MRASEC^IBCEF4(IBXIEN):2,1:3)  S IBNID=$$PAYERID^IBCEF2(IBXIEN)  S IB837R=$$RECVR^IBCEF2(IBXIEN)  S IBDIV=$P($S($P(IB0,U,22):$$SITE^VASITE(DT,$P(IB0,U,22)),1:$$SITE^VASITE()),U,3)  I IBNID'="","RPIHS"[$E(IBNID),$E(IBNID,2,$L(IBNID))="PRNT" S IBNID=IBNID\_"\*"\_IBINS  I IBNID="" S IBNID="\*"\_IBINS  S $P(IBNID,"\*",3)=$S($P(IB0,U,22):$P(IB0,U,22),1:"")  Q  ; | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCEF1 | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** |  | | | | | | | | |
| **Related Options** | None | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  |  | | | | |  | | | |
| **Data Dictionary (DD) References** | BILL/CLAIMS [#399]  MCCR UTILITY [#399.1]  IB SITE PARAMETERS [#350.9]  IB BILL/CLAIMS PRESCRIPTION REFILL [#362.4]  IB BILL/CLAIMS PROSTHETICS [#362.5] | | | | | | | | |
| **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** | | | | | | | | | |
| IBCEF1 ;ALB/TMP - FORMATTER SPECIFIC BILL FUNCTIONS - CONT ;30-JAN-96  ;;2.0;INTEGRATED BILLING;\*\*52,124,51,137,210,155,349,371,447,547,574\*\*;21-MAR-94;Build 12  ;;Per VA Directive 6402, this routine should not be modified.  ; OCC(IBIFN,REL,TEXT) ;Sets up an arrays of occurrence codes for various cks  ;RETURNS 1^additional data for entry IBXSAVE("OCC",n) if REL or TEXT  ; parameters have been met or null if conditions not met  ;If no REL or TEXT parameters sent, just extract codes array  ; IBIFN = bill ien  ; REL = 'OCC RELATED TO' value to check for  ; TEXT = text to check for the .01 field of 399.1 entry pointed to  ; by the occurrence code  N OCC,SORT,ARR,N,DATA,CODE,CT  I '$D(IBXSAVE("OCC")),'$D(IBXSAVE("OCCS")) D  .N IBI,Z,CT1,CT2,Z0 S (IBI,CT1,CT2)=0  .F  S IBI=$O(^DGCR(399,IBIFN,"OC",IBI)) Q:'IBI  S Z=$G(^(IBI,0)) D  ..S Z0=$G(^DGCR(399.1,+Z,0))  ..Q:'$P(Z0,U,10)&'$P(Z0,U,4) ;Not an occurrence code  ..I $P(Z0,U,10) S CT2=CT2+1,IBXSAVE("OCCS",CT2)=$S($P(Z0,U,4):$P(Z0,U,2)\_U\_$P(Z,U,2),1:U)\_U\_$P(Z,U,4)\_U\_$P(Z0,U)\_U\_$P(Z0,U,9)\_U\_$P(Z,U,3)\_U\_$P(Z,U,2)  ..I '$P(Z0,U,10) S CT1=CT1+1,IBXSAVE("OCC",CT1)=$S($P(Z0,U,4):$P(Z0,U,2)\_U\_$P(Z,U,2),1:U)\_U\_U\_$P(Z0,U)\_U\_$P(Z0,U,9)\_U\_$P(Z,U,3)\_U\_$P(Z,U,2)  I '$D(IBXSAVE("OCC"))&'$D(IBXSAVE("OCCS")) S IBXSAVE("OCC")="" G OCCQ  ;  ; esg - IB\*2\*349 - order the occurrence codes  ; Build the SORT array sorted by the occ code  F ARR="OCC","OCCS" S N=0 F  S N=$O(IBXSAVE(ARR,N)) Q:'N  S DATA=$G(IBXSAVE(ARR,N)) I $P(DATA,U,1)'="" S CODE=" "\_$P(DATA,U,1),SORT(ARR,CODE,N)=DATA  ; Loop thru the SORT array and re-build the IBXSAVE array  F ARR="OCC","OCCS" K IBXSAVE(ARR) S CODE="",CT=0 F  S CODE=$O(SORT(ARR,CODE)) Q:CODE=""  S N=0 F  S N=$O(SORT(ARR,CODE,N)) Q:'N  S CT=CT+1,IBXSAVE(ARR,CT)=SORT(ARR,CODE,N)  ;  I $G(REL)'=""!($G(TEXT)'="") D OCC1("",.OCC,$G(REL),$G(TEXT)) D:'$D(OCC) OCC1("S",.OCC,$G(REL),$G(TEXT)) OCCQ Q $G(OCC)  ; OCC1(ARR,OCC,REL,TEXT) ; Search thru local array for parameters met  ; ARR = null to search OCC subscript, "S" to search OCCS subscript  N Z  S ARR="OCC"\_ARR,Z=0  F  S Z=$O(IBXSAVE(ARR,Z)) Q:'Z  D  .I $G(REL)'="",$P(IBXSAVE(ARR,Z),U,5)=REL S OCC="1"\_$S(REL=2:U\_$P(IBXSAVE(ARR,Z),U,6),1:"") Q  .I $G(TEXT)'="",$P(IBXSAVE(ARR,Z),U,4)=TEXT S OCC="1^"\_$P(IBXSAVE(ARR,Z),U,7)  Q  ; RX(IBIFN) ; Format billable prescription data for refills for 837  N Z,IBXDATA,CT  I '$D(IBXSAVE("BOX24")) D B24^IBCEF3(.IBXSAVE,IBIFN,1)  S Z="",CT=0  F  S Z=$O(IBXSAVE("BOX24",Z)) Q:Z=""  I $D(IBXSAVE("BOX24",Z,"RX")) S CT=CT+1,IBXDATA(Z)=IBXSAVE("BOX24",Z,"RX") RXQ Q CT  ; OTHPAY(IBIFN,SEQ) ; Return the other insurance payment amount for bill  ; IBIFN and payer sequence SEQ (1-3)  N AMT,IBIFN1,PRP  S IBIFN1=$P($G(^DGCR(399,IBIFN,"M1")),U,SEQ+4),PRP=0  I IBIFN1 D  . ; IB\*2.0\*547 if Medicare on bill, make sure you are pulling amt paid from correct sequence  . ; code was leaving out MRA amt on tertiary bills and cloned secondary where MRA claim# does NOT match current claim#  . ;I $$MCRWNR^IBEFUNC(+$G(^DGCR(399,IBIFN,"I"\_SEQ))) S AMT=$$MCRPAY^IBCEU0(IBIFN) Q  . I $$MCRWNR^IBEFUNC(+$G(^DGCR(399,IBIFN,"I"\_SEQ))) S AMT=$$MCRPAY^IBCEU0(IBIFN1),PRP=1 Q  . S AMT=+$$TPR^PRCAFN(IBIFN1) I AMT S PRP=1 Q  ; A/R amount  . ; IB\*2.0\*547 - moved this line because it was not getting executed if IBIFN1 was not defined, which it won't be for   . ; manually created secondary and tertiary claims. Using new flag PRP to indicate if prior payment already found.  . ; S AMT=+$P($G(^DGCR(399,IBIFN,"U2")),U,SEQ+3) ; amount on bill  S:PRP=0 AMT=+$P($G(^DGCR(399,IBIFN,"U2")),U,SEQ+3) ; amount on bill  Q $G(AMT)  ; OUTPT(IBIFN,IBPRINT) ; Moved for space  D OUTPT^IBCEF11(IBIFN,$G(IBPRINT))  Q  ; OCC92 ;Reformats IBXSAVE("OCC") and IBXSAVE("OCCS") to fit blocks on UB-04  ; Set up IBXSAVE(32-36) arrays  N IBPG,IB32,IB33,IB34,IB35,IB36,IBFL,Z,Z0,PG  S IBPG=0  F Z=32:1:36 K IBFL(Z) S IBFL(Z)=0  M IB32=IBXSAVE("OCC"),IB36=IBXSAVE("OCCS")  S IB32=$O(IB32(""),-1),IB36=$O(IB36(""),-1),PG=1  D OCC^IBCF32  F Z=32:1:36 S Z0="" F  S Z0=$O(IBFL(Z,Z0)) Q:'Z0  S IBXSAVE("OC92",Z,Z0)=$P(IBFL(Z,Z0),U,1,3)  Q  ; BATCH() ; Moved for space IB\*2\*349  Q $$BATCH^IBCEF11()  ; PROC(T,TYPE) ; Find procedure code, strip '.' Function returns result  ; T = Procedure internal entry #;file reference  ; TYPE = "CPT" for only CPT/HCPCS valid  ; "ICD" for only ICD9 valid or null for either  N Q,S  S Q="",S="^"\_$P($P(T,";",2),"(")  I $G(TYPE)="" D  . I $E(S,2,3)="IC" S Q=$P($$PRCD(T),U) Q  . I T["DIC(81.3" S Q=$$MOD^ICPTMOD(+T,"I") S Q=$S(Q>0:$P(Q,U,4),1:"")  I $G(TYPE)="CPT",$E(S,2,3)="IC" S Q=$$PRCD(T) Q  I $G(TYPE)="ICD",T["ICD0" S Q=$P($$ICD0^IBACSV(+T),U)  Q $TR(Q,".")  ; FACILITY(IBIFN) ;return the Facility (Institution pointer-#4) for a bill  ; the institution of the Bill Division (399,.22) if defined, otherwise the Facility Name (350.9,.02)  ;  N IB0,IBIN S IBIN=0  S IB0=$G(^DGCR(399,+$G(IBIFN),0)) I +$P(IB0,U,22) S IBIN=$$SITE^VASITE(+$P(IB0,U,3),+$P(IB0,U,22))  I IBIN'>0 S IBIN=+$P($G(^IBE(350.9,1,0)),U,2)  Q +IBIN  ; ISRX(IBIFN) ; Function to determine if bill is a prescription refill bill  ; Returns 0 if no Rx on bill or 1 if there is.  ;  N IBRX  I $D(^IBA(362.4,"AIFN"\_IBIFN)) S IBRX=1  Q +$G(IBRX)  ; ISPROS(IBIFN) ; Function to determine if bill is a prosthetics bill  ; Returns 0 if no Prosthetics on bill or 1 if there is.  ;  N IBPROS  I $D(^IBA(362.5,"AIFN"\_IBIFN)) S IBPROS=1  Q +$G(IBPROS)  ; FINDINS(IBIFN,IBSEQ) ; Returns the internal entry number of the insurance  ; company for bill ien IBIFN for payer sequence IBSEQ (or current if  ; IBSEQ is null)  Q $P($G(^DGCR(399,IBIFN,"I"\_$$COBN^IBCEF(IBIFN,$G(IBSEQ)))),U)  ; TOB(IBIFN) ; Returns UB-04 type of bill from data in the output formatter  N IBTOB,IBZ1,IBZ2,IBZ3  D F^IBCEF("N-UB-04 LOCATION OF CARE","IBZ1",,IBIFN)  D F^IBCEF("N-UB-04 BILL CLASSIFICATION","IBZ2",,IBIFN)  D F^IBCEF("N-UB-04 TIMEFRAME OF BILL","IBZ3",,IBIFN)  S IBTOB=IBZ1\_IBZ2\_IBZ3  Q IBTOB  ; PRCD(PRIEN,ALL,EDT) ; Function returns the code that corresponds to the variable  ; pointer data in PRIEN (ien;file)  ; ALL = if ALL=1, returns the entire $$CPT^ICPTCOD for CPT or  ; ^code^name format for ICD result  ; or null if lookup fails  ; EDT = Effective date to check (not used if +$G(ALL)=0)  N CODE,IBX  S CODE=""  ;Modified for Code Set Versioning  I PRIEN["ICPT" S IBX=$$CPT^ICPTCOD(+PRIEN,$G(EDT)) G:IBX'>0 PRCDQ S CODE=$S($G(ALL):IBX,1:$P(IBX,U,2))  I PRIEN["ICD0" S IBX=$$ICD0^IBACSV(+PRIEN,$G(EDT)) G:IBX="" PRCDQ S CODE=$S($G(ALL):U\_$P(IBX,U)\_U\_$P(IBX,U,4),1:$P(IBX,U)) PRCDQ Q CODE  ; NFT(FT,IBIFN) ; Returns 1 if bill IBIFN is not of form type FT (internal)  ; so the data element should not be required  S FT=$S($$FT^IBCEF(IBIFN)=FT:0,1:1)  Q FT  ; REQ(FT,INP,IBIFN) ; Determine if bill IBIFN is of form type FT and  ; Inpatient (I) or Outpatient (O) status INP [or either if (null)]   ;  ;Returns 1 if both conditions FT and INP match for the bill  ; or 0 if either of these conditions are not true  ; I $$REQ^IBCEF1(2,"I",1) would mean if bill entry #1 is  ; CMS-1500/inpatient the data would be required  ; I '$$REQ^IBCEF1(2,"I",1) would mean if bill entry #1 is anything but  ; CMS-1500/inpatient, the data would not be  ; required  N Z  S Z=1  S:$$NFT(FT,IBIFN) Z=0 ; Not the form type for requirement  I Z,$G(INP)'="" D  . S Z0=$$INPAT^IBCEF(IBIFN,1),INP=$G(INP)  . S Z=$S(Z0:INP="I",1:INP="O") ;Check if I/O matches required state  Q Z  ; SET1(IBIFN,A,IBZ,IBXDATA,IBXNOREQ) ; Utility to set variables for output  ; formatter for professional EDI   ; Returns values of A, IBXDATA, IBZ, IBXNOREQ  N Z,CT  S A="^TMP($J,""IBLCT"")"  S (Z,CT)=0  F  S Z=$O(IBXDATA(Z)) Q:'Z  D  ; Don't transmit 0-charges  . ;IB\*2.0\*447/TAZ - Transmit $0 charges.  . ;I $P(IBXDATA(Z),U,9),$P(IBXDATA(Z),U,8) S CT=CT+1 M IBZ(CT)=IBXDATA(Z)  . I $P(IBXDATA(Z),U,9) S CT=CT+1 M IBZ(CT)=IBXDATA(Z)  . ;IB\*2.0\*447  K IBXDATA  S IBXNOREQ='$$REQ(2,"O",IBIFN)  Q  ; CIADDR(IBXDATA,IBXSAVE,LINE,FORM) ; Format current ins co address line LINE for FORM  ; FORM = 1 for CMS-1500, 2 for UB-04  ; Called from output formatter - both IBXDATA, IBXSAVE parameters are  ; passed by reference  ;  K IBXDATA  I $G(FORM)'=1 D  . ;  . ; esg - 11/17/06 - IB\*2\*349 - UB-04 FL-38 contains the payer name  . ; and address on 4 lines within this 5 line box. All 5 lines  . ; are formatted here into the IBXDATA array. This is the  . ; address that shows through the envelope window.  . ;  . ; esg - 9/13/07 - IB\*2\*371 - Line 1 of this box contains the print  . ; status (i.e. copy, 2nd notice, 3rd notice, MRA needed).  . ;  . N Z,Z1,LM,Q,ADDR,X,IBPSTAT  . S LM=$P($G(^IBE(350.9,1,1)),U,31) ; UB address column parameter  . S Z=""  . I LM S $P(Z," ",LM)=""              ; beginning spaces indent  . S ADDR=$G(IBXSAVE("CADR")) ; address data string  . ;  . D F^IBCEF("N-PRINT BILL SUBMIT STATUS","IBPSTAT",,+$G(IBXIEN))  . S Z1=Z I Z1="" S Z1=" "     ; line 1 can't start in column 1  . S IBXDATA(1)=Z1\_$G(IBPSTAT),Q=1 ; line 1 print status  . S Q=Q+1  . S IBXDATA(Q)=Z\_$G(IBXSAVE("CADR\_NAME")) ; line 2 payer name  . S X=$P(ADDR,U,1)  . I X'="" S Q=Q+1,IBXDATA(Q)=Z\_X              ; address line 1  . S X=$P(ADDR,U,2)  . I X'="" S Q=Q+1,IBXDATA(Q)=Z\_X D            ; address line 2  .. S X=$P(ADDR,U,3)  .. I X'="" S IBXDATA(Q)=IBXDATA(Q)\_" "\_X      ; address line 3  .. Q  . S Q=Q+1 ; city,st,zip on last line  . S IBXDATA(Q)=Z\_$P(ADDR,U,4)\_", "\_$$STATE^IBCEFG1($P(ADDR,U,5))\_" "\_$P(ADDR,U,6)  . KILL IBXSAVE("CADR\_NAME"),IBXSAVE("CADR") ; cleanup  . Q  ;  I $G(FORM)=1 D           ; CMS-1500  . N CT,X,Z  . S:'$D(IBXSAVE("INDENT")) Z="",$P(Z," ",+$P($G(^IBE(350.9,1,1)),U,27)+1)="",IBXSAVE("INDENT")=Z  . S CT=0  . S X=$P(IBXSAVE("CADR"),U) S:X'="" CT=CT+1,IBXDATA(CT)=IBXSAVE("INDENT")\_X  . S X=$S($P(IBXSAVE("CADR"),U,2)'="":$P(IBXSAVE("CADR"),U,2),1:"")\_$S($P(IBXSAVE("CADR"),U,2)'="":" ",1:"")\_$P(IBXSAVE("CADR"),U,3) S:X'="" CT=CT+1,IBXDATA(CT)=IBXSAVE("INDENT")\_X  . S CT=CT+1,IBXDATA(CT)=IBXSAVE("INDENT")\_$P(IBXSAVE("CADR"),U,4)\_", "\_$$STATE^IBCEFG1($P(IBXSAVE("CADR"),U,5))\_" "\_$P(IBXSAVE("CADR"),U,6)  . Q  ;  Q  ; HHLTH(IBIFN,OUT) ; determine if claim is hospice/home health and needs episode of care date \*\*574\*\*  ; per NUBC, date the episode of care began is needed for all outpatient CMS-1500 Home Health and Hospice claims and  ; UB-04: 012x,022x,032x,034x,081x & 082x claims  ; this string is zero + the Bill Type field from screens 6&7 of enter/edit Bill: 0\_field#.24(LOC OF CARE)\_.25(BILL CLASS)\_.26(TIMEFRAME)  ; required - IBIFN = internal claim#  ; optional - OUT = optional flag to pass to INPAT^IBCEF  ; returns a 1 if date should be included on bill and a 0 if it should NOT be included on bill  ;  N IB0,IBL,IBC,IBT  Q:$G(IBIFN)="" 0  ; all inpatient claims include date  I $$INPAT^IBCEF(IBIFN,+$G(OUT))'=0 Q 1  S IB0=$G(^DGCR(399,IBIFN,0)),IBL=$P(IB0,U,24)  ; Per Lisa Duncan, all Home health must have date, not just 032x & 034x  Q:IBL=3 1  ; not home health or hospice if LOC OF CARE = 7  Q:IBL=7 0  S IBC=$P($G(^DGCR(399.1,+$P(IB0,U,25),0)),U,2)  ; not home health or hospice if BILL CLASS is 3 or a number greater than 4  Q:IBC>4 0  Q:IBC=3 0  S IBT=IBL\_IBC  ; any claim where the location of care\_bill classification combo is 12,22,32,34,81 or 82 must have date  Q:"^12^22^32^34^81^82^"[IBT 1  Q 0 | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCEF1 ;ALB/TMP - FORMATTER SPECIFIC BILL FUNCTIONS - CONT ;30-JAN-96  ;;2.0;INTEGRATED BILLING;\*\*52,124,51,137,210,155,349,371,447,547,574,592\*\*;21-MAR-94;Build 12  ;;Per VA Directive 6402, this routine should not be modified.  ; OCC(IBIFN,REL,TEXT) ;Sets up an arrays of occurrence codes for various cks  ;RETURNS 1^additional data for entry IBXSAVE("OCC",n) if REL or TEXT  ; parameters have been met or null if conditions not met  ;If no REL or TEXT parameters sent, just extract codes array  ; IBIFN = bill ien  ; REL = 'OCC RELATED TO' value to check for  ; TEXT = text to check for the .01 field of 399.1 entry pointed to  ; by the occurrence code  N OCC,SORT,ARR,N,DATA,CODE,CT  I '$D(IBXSAVE("OCC")),'$D(IBXSAVE("OCCS")) D  .N IBI,Z,CT1,CT2,Z0 S (IBI,CT1,CT2)=0  .F  S IBI=$O(^DGCR(399,IBIFN,"OC",IBI)) Q:'IBI  S Z=$G(^(IBI,0)) D  ..S Z0=$G(^DGCR(399.1,+Z,0))  ..Q:'$P(Z0,U,10)&'$P(Z0,U,4) ;Not an occurrence code  ..I $P(Z0,U,10) S CT2=CT2+1,IBXSAVE("OCCS",CT2)=$S($P(Z0,U,4):$P(Z0,U,2)\_U\_$P(Z,U,2),1:U)\_U\_$P(Z,U,4)\_U\_$P(Z0,U)\_U\_$P(Z0,U,9)\_U\_$P(Z,U,3)\_U\_$P(Z,U,2)  ..I '$P(Z0,U,10) S CT1=CT1+1,IBXSAVE("OCC",CT1)=$S($P(Z0,U,4):$P(Z0,U,2)\_U\_$P(Z,U,2),1:U)\_U\_U\_$P(Z0,U)\_U\_$P(Z0,U,9)\_U\_$P(Z,U,3)\_U\_$P(Z,U,2)  I '$D(IBXSAVE("OCC"))&'$D(IBXSAVE("OCCS")) S IBXSAVE("OCC")="" G OCCQ  ;  ; esg - IB\*2\*349 - order the occurrence codes  ; Build the SORT array sorted by the occ code  F ARR="OCC","OCCS" S N=0 F  S N=$O(IBXSAVE(ARR,N)) Q:'N  S DATA=$G(IBXSAVE(ARR,N)) I $P(DATA,U,1)'="" S CODE=" "\_$P(DATA,U,1),SORT(ARR,CODE,N)=DATA  ; Loop thru the SORT array and re-build the IBXSAVE array  F ARR="OCC","OCCS" K IBXSAVE(ARR) S CODE="",CT=0 F  S CODE=$O(SORT(ARR,CODE)) Q:CODE=""  S N=0 F  S N=$O(SORT(ARR,CODE,N)) Q:'N  S CT=CT+1,IBXSAVE(ARR,CT)=SORT(ARR,CODE,N)  ;  I $G(REL)'=""!($G(TEXT)'="") D OCC1("",.OCC,$G(REL),$G(TEXT)) D:'$D(OCC) OCC1("S",.OCC,$G(REL),$G(TEXT)) OCCQ Q $G(OCC)  ; OCC1(ARR,OCC,REL,TEXT) ; Search thru local array for parameters met  ; ARR = null to search OCC subscript, "S" to search OCCS subscript  N Z  S ARR="OCC"\_ARR,Z=0  F  S Z=$O(IBXSAVE(ARR,Z)) Q:'Z  D  .I $G(REL)'="",$P(IBXSAVE(ARR,Z),U,5)=REL S OCC="1"\_$S(REL=2:U\_$P(IBXSAVE(ARR,Z),U,6),1:"") Q  .I $G(TEXT)'="",$P(IBXSAVE(ARR,Z),U,4)=TEXT S OCC="1^"\_$P(IBXSAVE(ARR,Z),U,7)  Q  ; RX(IBIFN) ; Format billable prescription data for refills for 837  N Z,IBXDATA,CT  I '$D(IBXSAVE("BOX24")) D B24^IBCEF3(.IBXSAVE,IBIFN,1)  S Z="",CT=0  F  S Z=$O(IBXSAVE("BOX24",Z)) Q:Z=""  I $D(IBXSAVE("BOX24",Z,"RX")) S CT=CT+1,IBXDATA(Z)=IBXSAVE("BOX24",Z,"RX") RXQ Q CT  ; OTHPAY(IBIFN,SEQ) ; Return the other insurance payment amount for bill  ; IBIFN and payer sequence SEQ (1-3)  N AMT,IBIFN1,PRP  S IBIFN1=$P($G(^DGCR(399,IBIFN,"M1")),U,SEQ+4),PRP=0  I IBIFN1 D  . ; IB\*2.0\*547 if Medicare on bill, make sure you are pulling amt paid from correct sequence  . ; code was leaving out MRA amt on tertiary bills and cloned secondary where MRA claim# does NOT match current claim#  . ;I $$MCRWNR^IBEFUNC(+$G(^DGCR(399,IBIFN,"I"\_SEQ))) S AMT=$$MCRPAY^IBCEU0(IBIFN) Q  . I $$MCRWNR^IBEFUNC(+$G(^DGCR(399,IBIFN,"I"\_SEQ))) S AMT=$$MCRPAY^IBCEU0(IBIFN1),PRP=1 Q  . S AMT=+$$TPR^PRCAFN(IBIFN1) I AMT S PRP=1 Q  ; A/R amount  . ; IB\*2.0\*547 - moved this line because it was not getting executed if IBIFN1 was not defined, which it won't be for   . ; manually created secondary and tertiary claims. Using new flag PRP to indicate if prior payment already found.  . ; S AMT=+$P($G(^DGCR(399,IBIFN,"U2")),U,SEQ+3) ; amount on bill  S:PRP=0 AMT=+$P($G(^DGCR(399,IBIFN,"U2")),U,SEQ+3) ; amount on bill  Q $G(AMT)  ; OUTPT(IBIFN,IBPRINT) ; Moved for space  D OUTPT^IBCEF11(IBIFN,$G(IBPRINT))  Q  ; OCC92 ;Reformats IBXSAVE("OCC") and IBXSAVE("OCCS") to fit blocks on UB-04  ; Set up IBXSAVE(32-36) arrays  N IBPG,IB32,IB33,IB34,IB35,IB36,IBFL,Z,Z0,PG  S IBPG=0  F Z=32:1:36 K IBFL(Z) S IBFL(Z)=0  M IB32=IBXSAVE("OCC"),IB36=IBXSAVE("OCCS")  S IB32=$O(IB32(""),-1),IB36=$O(IB36(""),-1),PG=1  D OCC^IBCF32  F Z=32:1:36 S Z0="" F  S Z0=$O(IBFL(Z,Z0)) Q:'Z0  S IBXSAVE("OC92",Z,Z0)=$P(IBFL(Z,Z0),U,1,3)  Q  ; BATCH() ; Moved for space IB\*2\*349  Q $$BATCH^IBCEF11()  ; PROC(T,TYPE) ; Find procedure code, strip '.' Function returns result  ; T = Procedure internal entry #;file reference  ; TYPE = "CPT" for only CPT/HCPCS valid  ; "ICD" for only ICD9 valid or null for either  N Q,S  S Q="",S="^"\_$P($P(T,";",2),"(")  I $G(TYPE)="" D  . I $E(S,2,3)="IC" S Q=$P($$PRCD(T),U) Q  . I T["DIC(81.3" S Q=$$MOD^ICPTMOD(+T,"I") S Q=$S(Q>0:$P(Q,U,4),1:"")  I $G(TYPE)="CPT",$E(S,2,3)="IC" S Q=$$PRCD(T) Q  I $G(TYPE)="ICD",T["ICD0" S Q=$P($$ICD0^IBACSV(+T),U)  Q $TR(Q,".")  ; FACILITY(IBIFN) ;return the Facility (Institution pointer-#4) for a bill  ; the institution of the Bill Division (399,.22) if defined, otherwise the Facility Name (350.9,.02)  ;  N IB0,IBIN S IBIN=0  S IB0=$G(^DGCR(399,+$G(IBIFN),0)) I +$P(IB0,U,22) S IBIN=$$SITE^VASITE(+$P(IB0,U,3),+$P(IB0,U,22))  I IBIN'>0 S IBIN=+$P($G(^IBE(350.9,1,0)),U,2)  Q +IBIN  ; ISRX(IBIFN) ; Function to determine if bill is a prescription refill bill  ; Returns 0 if no Rx on bill or 1 if there is.  ;  N IBRX  I $D(^IBA(362.4,"AIFN"\_IBIFN)) S IBRX=1  Q +$G(IBRX)  ; ISPROS(IBIFN) ; Function to determine if bill is a prosthetics bill  ; Returns 0 if no Prosthetics on bill or 1 if there is.  ;  N IBPROS  I $D(^IBA(362.5,"AIFN"\_IBIFN)) S IBPROS=1  Q +$G(IBPROS)  ; FINDINS(IBIFN,IBSEQ) ; Returns the internal entry number of the insurance  ; company for bill ien IBIFN for payer sequence IBSEQ (or current if  ; IBSEQ is null)  Q $P($G(^DGCR(399,IBIFN,"I"\_$$COBN^IBCEF(IBIFN,$G(IBSEQ)))),U)  ; TOB(IBIFN) ; Returns UB-04 type of bill from data in the output formatter  N IBTOB,IBZ1,IBZ2,IBZ3  D F^IBCEF("N-UB-04 LOCATION OF CARE","IBZ1",,IBIFN)  D F^IBCEF("N-UB-04 BILL CLASSIFICATION","IBZ2",,IBIFN)  D F^IBCEF("N-UB-04 TIMEFRAME OF BILL","IBZ3",,IBIFN)  S IBTOB=IBZ1\_IBZ2\_IBZ3  Q IBTOB  ; PRCD(PRIEN,ALL,EDT) ; Function returns the code that corresponds to the variable  ; pointer data in PRIEN (ien;file)  ; ALL = if ALL=1, returns the entire $$CPT^ICPTCOD for CPT or  ; ^code^name format for ICD result  ; or null if lookup fails  ; EDT = Effective date to check (not used if +$G(ALL)=0)  N CODE,IBX  S CODE=""  ;Modified for Code Set Versioning  I PRIEN["ICPT" S IBX=$$CPT^ICPTCOD(+PRIEN,$G(EDT)) G:IBX'>0 PRCDQ S CODE=$S($G(ALL):IBX,1:$P(IBX,U,2))  I PRIEN["ICD0" S IBX=$$ICD0^IBACSV(+PRIEN,$G(EDT)) G:IBX="" PRCDQ S CODE=$S($G(ALL):U\_$P(IBX,U)\_U\_$P(IBX,U,4),1:$P(IBX,U)) PRCDQ Q CODE  ; NFT(FT,IBIFN) ; Returns 1 if bill IBIFN is not of form type FT (internal)  ; so the data element should not be required  S FT=$S($$FT^IBCEF(IBIFN)=FT:0,1:1)  Q FT  ; REQ(FT,INP,IBIFN) ; Determine if bill IBIFN is of form type FT and  ; Inpatient (I) or Outpatient (O) status INP [or either if (null)]   ;  ;Returns 1 if both conditions FT and INP match for the bill  ; or 0 if either of these conditions are not true  ; I $$REQ^IBCEF1(2,"I",1) would mean if bill entry #1 is  ; CMS-1500/inpatient the data would be required  ; I '$$REQ^IBCEF1(2,"I",1) would mean if bill entry #1 is anything but  ; CMS-1500/inpatient, the data would not be  ; required  N Z  S Z=1  S:$$NFT(FT,IBIFN) Z=0 ; Not the form type for requirement  I Z,$G(INP)'="" D  . S Z0=$$INPAT^IBCEF(IBIFN,1),INP=$G(INP)  . S Z=$S(Z0:INP="I",1:INP="O") ;Check if I/O matches required state  Q Z  ; SET1(IBIFN,A,IBZ,IBXDATA,IBXNOREQ) ; Utility to set variables for output  ; formatter for professional EDI   ; Returns values of A, IBXDATA, IBZ, IBXNOREQ  N Z,CT  S A="^TMP($J,""IBLCT"")"  S (Z,CT)=0  F  S Z=$O(IBXDATA(Z)) Q:'Z  D  ; Don't transmit 0-charges  . ;IB\*2.0\*447/TAZ - Transmit $0 charges.  . ;I $P(IBXDATA(Z),U,9),$P(IBXDATA(Z),U,8) S CT=CT+1 M IBZ(CT)=IBXDATA(Z)  . ;JWS;IB\*2.0\*592:US131  . I $P(IBXDATA(Z),U,9) S CT=CT+1 M IBZ(CT)=IBXDATA(Z) I $$FT^IBCEF(IBXIEN)=7,$O(IBXDATA(Z,"DEN1",0)) M IBZ(CT,"DEN1")=IBXDATA(Z,"DEN1")  . ;IB\*2.0\*447  K IBXDATA  ;JWS;IB\*2.0\*592:US131  I $$FT^IBCEF(IBXIEN)'=7 S IBXNOREQ='$$REQ(2,"O",IBIFN)  Q  ; CIADDR(IBXDATA,IBXSAVE,LINE,FORM) ; Format current ins co address line LINE for FORM  ; FORM = 1 for CMS-1500, 2 for UB-04  ; Called from output formatter - both IBXDATA, IBXSAVE parameters are  ; passed by reference  ;  K IBXDATA  I $G(FORM)'=1 D  . ;  . ; esg - 11/17/06 - IB\*2\*349 - UB-04 FL-38 contains the payer name  . ; and address on 4 lines within this 5 line box. All 5 lines  . ; are formatted here into the IBXDATA array. This is the  . ; address that shows through the envelope window.  . ;  . ; esg - 9/13/07 - IB\*2\*371 - Line 1 of this box contains the print  . ; status (i.e. copy, 2nd notice, 3rd notice, MRA needed).  . ;  . N Z,Z1,LM,Q,ADDR,X,IBPSTAT  . S LM=$P($G(^IBE(350.9,1,1)),U,31) ; UB address column parameter  . S Z=""  . I LM S $P(Z," ",LM)=""              ; beginning spaces indent  . S ADDR=$G(IBXSAVE("CADR")) ; address data string  . ;  . D F^IBCEF("N-PRINT BILL SUBMIT STATUS","IBPSTAT",,+$G(IBXIEN))  . S Z1=Z I Z1="" S Z1=" "     ; line 1 can't start in column 1  . S IBXDATA(1)=Z1\_$G(IBPSTAT),Q=1 ; line 1 print status  . S Q=Q+1  . S IBXDATA(Q)=Z\_$G(IBXSAVE("CADR\_NAME")) ; line 2 payer name  . S X=$P(ADDR,U,1)  . I X'="" S Q=Q+1,IBXDATA(Q)=Z\_X              ; address line 1  . S X=$P(ADDR,U,2)  . I X'="" S Q=Q+1,IBXDATA(Q)=Z\_X D            ; address line 2  .. S X=$P(ADDR,U,3)  .. I X'="" S IBXDATA(Q)=IBXDATA(Q)\_" "\_X      ; address line 3  .. Q  . S Q=Q+1 ; city,st,zip on last line  . S IBXDATA(Q)=Z\_$P(ADDR,U,4)\_", "\_$$STATE^IBCEFG1($P(ADDR,U,5))\_" "\_$P(ADDR,U,6)  . KILL IBXSAVE("CADR\_NAME"),IBXSAVE("CADR") ; cleanup  . Q  ;  I $G(FORM)=1 D           ; CMS-1500  . N CT,X,Z  . S:'$D(IBXSAVE("INDENT")) Z="",$P(Z," ",+$P($G(^IBE(350.9,1,1)),U,27)+1)="",IBXSAVE("INDENT")=Z  . S CT=0  . S X=$P(IBXSAVE("CADR"),U) S:X'="" CT=CT+1,IBXDATA(CT)=IBXSAVE("INDENT")\_X  . S X=$S($P(IBXSAVE("CADR"),U,2)'="":$P(IBXSAVE("CADR"),U,2),1:"")\_$S($P(IBXSAVE("CADR"),U,2)'="":" ",1:"")\_$P(IBXSAVE("CADR"),U,3) S:X'="" CT=CT+1,IBXDATA(CT)=IBXSAVE("INDENT")\_X  . S CT=CT+1,IBXDATA(CT)=IBXSAVE("INDENT")\_$P(IBXSAVE("CADR"),U,4)\_", "\_$$STATE^IBCEFG1($P(IBXSAVE("CADR"),U,5))\_" "\_$P(IBXSAVE("CADR"),U,6)  . Q  ;  Q  ; HHLTH(IBIFN,OUT) ; determine if claim is hospice/home health and needs episode of care date \*\*574\*\*  ; per NUBC, date the episode of care began is needed for all outpatient CMS-1500 Home Health and Hospice claims and  ; UB-04: 012x,022x,032x,034x,081x & 082x claims  ; this string is zero + the Bill Type field from screens 6&7 of enter/edit Bill: 0\_field#.24(LOC OF CARE)\_.25(BILL CLASS)\_.26(TIMEFRAME)  ; required - IBIFN = internal claim#  ; optional - OUT = optional flag to pass to INPAT^IBCEF  ; returns a 1 if date should be included on bill and a 0 if it should NOT be included on bill  ;  N IB0,IBL,IBC,IBT  Q:$G(IBIFN)="" 0  ; all inpatient claims include date  I $$INPAT^IBCEF(IBIFN,+$G(OUT))'=0 Q 1  S IB0=$G(^DGCR(399,IBIFN,0)),IBL=$P(IB0,U,24)  ; Per Lisa Duncan, all Home health must have date, not just 032x & 034x  Q:IBL=3 1  ; not home health or hospice if LOC OF CARE = 7  Q:IBL=7 0  S IBC=$P($G(^DGCR(399.1,+$P(IB0,U,25),0)),U,2)  ; not home health or hospice if BILL CLASS is 3 or a number greater than 4  Q:IBC>4 0  Q:IBC=3 0  S IBT=IBL\_IBC  ; any claim where the location of care\_bill classification combo is 12,22,32,34,81 or 82 must have date  Q:"^12^22^32^34^81^82^"[IBT 1  Q 0 | | | | | | | | | |

| **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** | IB BILL/CLAIMS DIAGNOSIS [#362.3]  BILL/CLAIMS [#399]  EDI TRANSMISSION BATCH [#364.1] | | | | | | | | |
| **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,**592**\*\*;21-MAR-94;Build 1  ;;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  . ; **. ;JWS;IB\*2.0\*592:US131  . I $G(IBFLD(24,IBI,"DEN"))'="" S IBXDATA(IBI,"DEN")=IBFLD(24,IBI,"DEN")  . I $O(IBFLD(24,IBI,"DEN1",0)) M IBXDATA(IBI,"DEN1")=IBFLD(24,IBI,"DEN1")  . ;end - ;JWS;IB\*2.0\*592:US131**  . 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** | IBCEF12 | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** |  | | | | | | | | |
| **Related Options** | None | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  |  | | | | |  | | | |
| **Data Dictionary (DD) References** | BILL/CLAIMS [#399]  X12 278 DENTAL NUMBERING SYSTEM [#356.022] | | | | | | | | |
| **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)** | | | | | | | | | |
| IBCEF12 ;EDE/JWS - OUTPUT FORMATTER SPECIFIC DENTAL FUNCTIONS ;30-JAN-96  ;;2.0;INTEGRATED BILLING;\*\*592\*\*;21-MAR-94;Build 123  ;;Per VA Directive 6402, this routine should not be modified.  ;  ;JWS;IB\*2.0\*592;US131 TNUM(IBIFN) ; Extract code for 364.5 field 383 N-TOOTH NUMBER  N IB,IB1  K ^TMP("IBXSAVE",$J,"TO")  S IB=0 F  S IB=$O(^DGCR(399,IBIFN,"DEN1",IB)) Q:'IB  S IB1=^(IB,0),^TMP("IBXSAVE",$J,"TO",IBIFN,IB)=IB1\_U\_"JP"  Q  ; DEN ; Output formatter Format Code for file DEN, field 2  ;JWS;IB\*2.0\*592;US131  N A,Z,Q,IBZ K IBXSAVE("OUTPT")  D SET1^IBCEF1(IBXIEN,.A,.IBZ,.IBXDATA,.IBXNOREQ)  S (Q,Z)=0 ;,Q=$G(@A)  F  S Z=$O(IBZ(Z)) S:'Z @A=Q Q:'Z  M IBXSAVE("OUTPT",Z)=IBZ(Z) S Q=Q+1,IBXDATA(Z)=Q D:Z>1 ID^IBCEF2(Z,"DEN ") D SVITM^IBCEF2(.IBXSAVE,Z)  Q  ; DEN1 ; Output formatter Format Code for file DEN1, field 2  ;JWS;IB\*2.0\*592;US131  N A,Z,Q,IBZ K IBXSAVE("OUTPT")  D SET1^IBCEF1(IBXIEN,.A,.IBZ,.IBXDATA,.IBXNOREQ)  S (Q,Z)=0 ;,Q=$G(@A)  F  S Z=$O(IBZ(Z)) S:'Z @A=Q Q:'Z  M IBXSAVE("OUTPT",Z)=IBZ(Z) S Q=Q+1,IBXDATA(Z)=Q D:Z>1 ID^IBCEF2(Z,"DEN1") D SVITM^IBCEF2(.IBXSAVE,Z)  Q  ; DEN2 ; Output formatter Format Code for file DEN2, fields 2  ;JWS;IB\*2.0\*592;US131  N A,Z,Z1,CT  D SET1^IBCEF1(IBXIEN,.A,.IBZ,.IBXDATA,.IBXNOREQ)  S (CT,Z)=0 ;,Q=$G(@A)  F  S Z=$O(IBZ(Z)) Q:'Z  D  . S Z1=0 F  S Z1=$O(IBXSAVE("OUTPT",Z,"DEN1",Z1)) Q:'Z1  D  I CT=1,$P($G(IBXSAVE("OUTPT",Z)),U,9)'=1 Q  .. S CT=CT+1 D ID^IBCEF2(CT,"DEN2")  .. S IBXDATA(CT)=Z  .. D SETGBL^IBCEFG(IBXPG,CT,2,Z,.IBXSIZE)  K IBXDATA  Q  ; DEN23 ; Output formatter format code for file DEN2, field 3 (8,186.2,1,3)  ;JWS;IB\*2.0\*592;US131  N Z,Z0,CT  S (CT,Z)=0  F  S Z=$O(IBXSAVE("OUTPT",Z)) Q:'Z  D  . S Z0=0 F  S Z0=$O(IBXSAVE("OUTPT",Z,"DEN1",Z0)) Q:'Z0  D  .. S CT=CT+1  .. S IBXDATA(CT)="JP"  .. D SETGBL^IBCEFG(IBXPG,CT,3,"JP",.IBXSIZE)  K IBXDATA  Q  ; DEN24 ; Output formatter Format Code for file DEN2, field 4  ;JWS;IB\*2.0\*592;US131  N Z,ZO,CT K IBXSAVE("DONE")  S (CT,Z)=0  F  S Z=$O(IBXSAVE("OUTPT",Z)) Q:'Z  D  . S Z0=0 F  S Z0=$O(IBXSAVE("OUTPT",Z,"DEN1",Z0)) Q:'Z0  D  .. S CT=CT+1  .. I $D(IBXSAVE("DONE",Z,Z0)) Q  .. S IBXSAVE("DONE",Z,Z0)=""  .. S IBXDATA(CT)=$$GET1^DIQ(356.022,$P(IBXSAVE("OUTPT",Z,"DEN1",Z0,0),U),.01)  .. D SETGBL^IBCEFG(IBXPG,CT,4,IBXDATA(CT),.IBXSIZE)  K IBXDATA  Q  ; DEN25 ; Output formatter Format Code for file DEN2, field 5  ;JWS;IB\*2.0\*592;US131  N Z,ZO,CT K IBXSAVE("DONE")  S (CT,Z)=0  F  S Z=$O(IBXSAVE("OUTPT",Z)) Q:'Z  D  . S Z0=0 F  S Z0=$O(IBXSAVE("OUTPT",Z,"DEN1",Z0)) Q:'Z0  D  .. S CT=CT+1  .. I $D(IBXSAVE("DONE",Z,Z0)) Q  .. S IBXSAVE("DONE",Z,Z0)=""  .. S IBXDATA(CT)=$P(IBXSAVE("OUTPT",Z,"DEN1",Z0,0),U,2)  .. D SETGBL^IBCEFG(IBXPG,CT,5,IBXDATA(CT),.IBXSIZE)  K IBXDATA  Q  ; DEN26 ; Output formatter Format Code for file DEN2, field 6  ;JWS;IB\*2.0\*592;US131  N Z,ZO,CT K IBXSAVE("DONE")  S (CT,Z)=0  F  S Z=$O(IBXSAVE("OUTPT",Z)) Q:'Z  D  . S Z0=0 F  S Z0=$O(IBXSAVE("OUTPT",Z,"DEN1",Z0)) Q:'Z0  D  .. S CT=CT+1  .. I $D(IBXSAVE("DONE",Z,Z0)) Q  .. S IBXSAVE("DONE",Z,Z0)=""  .. S IBXDATA(CT)=$P(IBXSAVE("OUTPT",Z,"DEN1",Z0,0),U,3)  .. D SETGBL^IBCEFG(IBXPG,CT,6,IBXDATA(CT),.IBXSIZE)  K IBXDATA  Q  ; DEN27 ; Output formatter Format Code for file DEN2, field 7  ;JWS;IB\*2.0\*592;US131  N Z,ZO,CT K IBXSAVE("DONE")  S (CT,Z)=0  F  S Z=$O(IBXSAVE("OUTPT",Z)) Q:'Z  D  . S Z0=0 F  S Z0=$O(IBXSAVE("OUTPT",Z,"DEN1",Z0)) Q:'Z0  D  .. S CT=CT+1  .. I $D(IBXSAVE("DONE",Z,Z0)) Q  .. S IBXSAVE("DONE",Z,Z0)=""  .. S IBXDATA(CT)=$P(IBXSAVE("OUTPT",Z,"DEN1",Z0,0),U,4)  .. D SETGBL^IBCEFG(IBXPG,CT,7,IBXDATA(CT),.IBXSIZE)  K IBXDATA  Q  ; DEN28 ; Output formatter Format Code for file DEN2, field 8  ;JWS;IB\*2.0\*592;US131  N Z,ZO,CT K IBXSAVE("DONE")  S (CT,Z)=0  F  S Z=$O(IBXSAVE("OUTPT",Z)) Q:'Z  D  . S Z0=0 F  S Z0=$O(IBXSAVE("OUTPT",Z,"DEN1",Z0)) Q:'Z0  D  .. S CT=CT+1  .. I $D(IBXSAVE("DONE",Z,Z0)) Q  .. S IBXSAVE("DONE",Z,Z0)=""  .. S IBXDATA(CT)=$P(IBXSAVE("OUTPT",Z,"DEN1",Z0,0),U,5)  .. D SETGBL^IBCEFG(IBXPG,CT,8,IBXDATA(CT),.IBXSIZE)  K IBXDATA  Q  ; DEN29 ; Output formatter Format Code for file DEN2, field 9  ;JWS;IB\*2.0\*592;US131  N Z,ZO,CT K IBXSAVE("DONE")  S (CT,Z)=0  F  S Z=$O(IBXSAVE("OUTPT",Z)) Q:'Z  D  . S Z0=0 F  S Z0=$O(IBXSAVE("OUTPT",Z,"DEN1",Z0)) Q:'Z0  D  .. S CT=CT+1  .. I $D(IBXSAVE("DONE",Z,Z0)) Q  .. S IBXSAVE("DONE",Z,Z0)=""  .. S IBXDATA(CT)=$P(IBXSAVE("OUTPT",Z,"DEN1",Z0,0),U,6)  .. D SETGBL^IBCEFG(IBXPG,CT,9,IBXDATA(CT),.IBXSIZE)  K IBXDATA  Q  ; TRANS ; Output formatter Format Code for file DN1, field 6  ;JWS;IB\*2.0\*592;US131  I $$GET1^DIQ(399,IBXIEN\_",",93)'="",$$GET1^DIQ(399,IBXIEN\_",",94)'="" K IBXDATA  Q  ; SRVDT ; Output formatter Format Code for file DEN, field 4 Service date  ;JWS;IB\*2.0\*592;US131  S IBXNOREQ=$$NFT^IBCEF1(2,IBXIEN)  N Z S Z=0  F  S Z=$O(IBXSAVE("OUTPT",Z)) Q:'Z  D  . I $P($G(IBXSAVE("OUTPT",Z,"DEN")),U,11)'="" Q  ;treatment start date  . I $P($G(IBXSAVE("OUTPT",Z,"DEN")),U,12)'="" Q  ;treatment completion date  . I $P(IBXSAVE("OUTPT",Z),U)'="" S IBXDATA(Z)=$P(IBXSAVE("OUTPT",Z),U)  . Q  Q  ; SRVDTQ ; Output formatter Format Code for file DEN, field 3 Date/Time Qualifier  ;JWS;IB\*2.0\*592;US131  N Z S Z=0  F  S Z=$O(IBXSAVE("OUTPT",Z)) Q:'Z  D  . I $P($G(IBXSAVE("OUTPT",Z,"DEN")),U,11)'="" Q  ;treatment start date  . I $P($G(IBXSAVE("OUTPT",Z,"DEN")),U,12)'="" Q  ;treatment completion date  . S IBXDATA(Z)=472  Q  ; PROC ; Output formatter Format Code for file DEN1, field 3 Procedure Count  N Z S Z=0  F  S Z=$O(IBXSAVE("OUTPT",Z)) Q:'Z  D  . S IBXDATA(Z)=$P($G(IBXSAVE("OUTPT",Z)),U,9) Q  ;number of units (default=1)  Q  ; | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCEFP | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** |  | | | | | | | | |
| **Related Options** | None | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  |  | | | | |  | | | |
| **Data Dictionary (DD) References** | BILL/CLAIMS [#399] | | | | | | | | |
| **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** | | | | | | | | | |
| IBCEFP ;ALB/TAZ - Provider ID functions ;28-OCT-10  ;;2.0;INTEGRATED BILLING;\*\*432,447,473,516\*\*;21-MAR-94;Build 123  ;;Per VA Directive 6402, this routine should not be modified.  ;  Q  ; ALLIDS(IBIFN,IBXSAVE,IBSTRIP,SEG) ; Return all of the Provider IDS  I '$D(IBSTRIP) S IBSTRIP=0  I '$D(SEG) S SEG=""  N IBXIEN,ARINFO,ARID,ARQ,IBFRMTYP,ARIEN,ARINS,Z0,DAT,I,SORT1,SORT2,SORT3,COB,IBCCOB,IBCARE,IBCURR,IBXDATA,NPI,CUROTH  ;  S IBFRMTYP=$$FT^IBCEF(IBIFN),IBFRMTYP=$S(IBFRMTYP=2:2,IBFRMTYP=3:1,1:0)  S IBCARE=$S($$ISRX^IBCEF1(IBIFN):3,1:0) ;if an Rx refill bill  S:IBCARE=0 IBCARE=$$INPAT^IBCEF(IBIFN,1) S:'IBCARE IBCARE=2 ;1-inp,2-out  S IBCURR=$$COB^IBCEF(IBIFN) ;current bill payer sequence  ;don't create anything if form type not CMS-1500 or UB  I IBFRMTYP,'+$G(IBXSAVE("PROVINF",IBIFN)) D  . N IBZ,CUROTH  . I IBFRMTYP=2 D OUTPT^IBCEF11(IBIFN,0)  . I IBFRMTYP=1 D HOS^IBCEF22(IBIFN)  . ; START IB\*2.0\*447 BI  . I IBCURR="A" D  Q  .. N IBRESARR  .. S IBLIMIT=5  .. D PROVINF(IBIFN,1,.IBRESARR,1,"C",IBFRMTYP,IBCARE,IBLIMIT,IBCURR,.IBXDATA)  .. M IBXSAVE=IBRESARR  .. S IBXSAVE("PROVINF",IBIFN)=IBIFN  . ; END IB\*2.0\*447 BI  . F CUROTH="C","O" D PROVIDER(IBIFN,CUROTH,.IBZ,IBFRMTYP,IBCARE,IBCURR,.IBXDATA) M IBXSAVE=IBZ  . S IBXSAVE("PROVINF",IBIFN)=IBIFN  . Q  ;  D LFIDS^IBCEF76(IBIFN,.IBXSAVE,IBSTRIP,SEG) ; Get the Lab/Facility IDs  S NPI=$P($$ORGNPI^IBCEF73A(IBIFN),U,1)  F CUROTH="C","O" D  . S IBXSAVE("LAB/FAC",IBIFN,CUROTH,1,0)=$S(NPI]"":"XX",1:"")\_U\_NPI  ;  S IBFRMTYP=$$FT^IBCEF(IBIFN)  S ARIEN=$S(IBFRMTYP=2:3,1:4)  S IBCCOB=$$COBN^IBCEF(IBIFN) ; Current Insurance  F COB=1:1:3 D  . S SORT1=$S(COB=IBCCOB:"C",1:"O")  . S SORT2=$S(SORT1="C":1,COB=1:1,COB=2&(IBCCOB=1):1,1:2)  . S ARINFO=$G(IBXSAVE("PROVINF",IBIFN,SORT1,SORT2,ARIEN,1))  . ;  . D BPIDS^IBCEF75(IBIFN,.IBXSAVE,SORT1,SORT2,COB,IBSTRIP,SEG)  ;  D EN^IBCEF81(.IBXSAVE)  ;  I SEG="OPR1" D  . I '$D(IBXSAVE("PROVINF",IBIFN,"C",1,3)) D  G ALLIDSQ  .. N SLC,CRED S SLC=0  .. F  S SLC=$O(IBXSAVE("L-PROV",IBIFN,SLC)) Q:'SLC  D  I $D(IBXSAVE("PROVINF",IBIFN,"C",1,3,"NAME")) Q  ... S CRED=$P($G(IBXSAVE("L-PROV",IBIFN,SLC,"C",1,3,"NAME")),U,4)  ... I CRED]"" S $P(IBXSAVE("PROVINF",IBIFN,"C",1,3,"NAME"),U,4)=CRED  ;  I SEG="LPUR",$$SUB1OK^IBCEP8A(IBIFN),$G(IBXSAVE("SLC")) D  G ALLIDSQ  . N IBCNT,OUT  . ;IB\*2.0\*473/TAZ - Moved ID lookup into seperate function.  . D PSID(IBIFN,.IBXSAVE,.IDS)  . ;IB\*2.0\*473/TAZ - END  . S $P(OUT,U,6)=IDS(0)  . S $P(OUT,U,7,8)=IDS(1) ; secondary id (1) ^ secondary id qualifier(1)  . F IBCNT=1:1:IBXSAVE("SLC") S IBXSAVE("SLPRV",IBCNT)=OUT,IBXSAVE("SLPRV",IBCNT,"SLC")=IBCNT  ;  I SEG="LOPE" D SLPRV(IBIFN,.IBXSAVE,2) G ALLIDSQ  I SEG="LOP1" D SLPRV(IBIFN,.IBXSAVE,9) G ALLIDSQ  I SEG="LREN" D SLPRV(IBIFN,.IBXSAVE,3) G ALLIDSQ  I SEG="LSUP" D SLPRV(IBIFN,.IBXSAVE,5) G ALLIDSQ  I SEG="LREF" D SLPRV(IBIFN,.IBXSAVE,1) G ALLIDSQ  ; ALLIDSQ ;  Q  ; PSID(IBIFN,IBXPROV,IBXIDS) ; Build array of either the Fac/Lab ID or Rendering Provider IDs  ;IB\*2.0\*473/TAZ - Created a function to standardize IDs in LDAT and LPUR  ; Input:  ; IBXIEN - Internal Entry Number of claim  ; IBXPROV - Provider Array  ; IBXIDS - Array for IDs  ; Output:  ; IBXIDS(0) - Primary ID  ; IBXIDS(1) - Secondary ID  ;  N LINE,PLINE,PID,SID,SIDQ  K IBXIDS  ; Get Lab/Facility IDs  S (PID,SID,SIDQ)=""  S PID=$P($G(IBXPROV("LAB/FAC",IBIFN,"C",1,0)),U,2)  F LINE=1:1 Q:'$D(IBXPROV("LAB/FAC",IBIFN,"C",1,LINE)) D  I SID'="" Q  . S SIDQ=$P($G(IBXPROV("LAB/FAC",IBIFN,"C",1,LINE)),U) I ",0B,1G,G2,"'[(","\_SIDQ\_",") S SIDQ="" Q  . S SID=$P($G(IBXPROV("LAB/FAC",IBIFN,"C",1,LINE)),U,2)  I $L(PID)!$L(SID) S IBXIDS(0)=PID,IBXIDS(1)=SID\_U\_SIDQ G PSIDQ  ; Get Claim Level Rendering Provider IDs  S PID=$P($G(IBXPROV("PROVINF",IBIFN,"C",1,3,0)),U,4) ; Get claim level Rendering Provider NPI  F LINE=1:1 Q:'$D(IBXPROV("PROVINF",IBIFN,"C",1,3,LINE)) D  I SID'="" Q  . S SIDQ=$P($G(IBXPROV("PROVINF",IBIFN,"C",1,3,LINE)),U,3) I ",0B,1G,G2,"'[(","\_SIDQ\_",") S SIDQ="" Q  . S SID=$P($G(IBXPROV("PROVINF",IBIFN,"C",1,3,LINE)),U,4)  I $L(PID)!$L(SID) S IBXIDS(0)=PID,IBXIDS(1)=SID\_U\_SIDQ G PSIDQ  ; Get Line Level Rendering Provider IDs  F PLINE=1:1 Q:'$D(IBXPROV("L-PROV",IBIFN,PLINE)) D  I $L(PID)!$L(SID) Q  . S PID=$P($G(IBXPROV("L-PROV",IBIFN,PLINE,"C",1,3,0)),U,4)  . F LINE=1:1 Q:'$D(IBXPROV("L-PROV",IBIFN,PLINE,"C",1,3,LINE)) D  I SID'="" Q  .. S SIDQ=$P($G(IBXPROV("L-PROV",IBIFN,PLINE,"C",1,3,LINE)),U,3) I ",0B,1G,G2,"'[(","\_SIDQ\_",") S SIDQ="" Q  .. S SID=$P($G(IBXPROV("L-PROV",IBIFN,PLINE,"C",1,3,LINE)),U,4)  I $L(PID)!$L(SID) S IBXIDS(0)=PID,IBXIDS(1)=SID\_U\_SIDQ PSIDQ ;  Q  ; SLPRV(IBXIEN,IBX,PRTYPE) ;Build SLPRV nodes for the line provider type record  N SLC,DATA,IBCNT,NAME,OUT  S (SLC,IBCNT)=0  F  S SLC=$O(IBX("L-PROV",IBXIEN,SLC)) Q:'SLC  D  . I '$D(IBX("L-PROV",IBXIEN,SLC,"C",1,PRTYPE)) Q  . S NAME=$G(IBX("L-PROV",IBXIEN,SLC,"C",1,PRTYPE,"NAME"))  . S OUT=$P(NAME,U,1,3)\_U\_$P(NAME,U,5)\_U\_$G(IBX("L-PROV",IBXIEN,SLC,"C",1,PRTYPE,"TAXONOMY"))\_U\_$P($G(IBX("L-PROV",IBXIEN,SLC,"C",1,PRTYPE,0)),U,4)  . F IBN=1:1 Q:'$D(IBX("L-PROV",IBXIEN,SLC,"C",1,PRTYPE,IBN)) D  .. S DATA=$G(IBX("L-PROV",IBXIEN,SLC,"C",1,PRTYPE,IBN))  .. I ",0B,1G,G2,LU,"[(","\_$P(DATA,U,3)\_",") S OUT=OUT\_U\_$P(DATA,U,4)\_U\_$P(DATA,U,3)  . S IBCNT=IBCNT+1  . S IBX("SLPRV",IBCNT)=OUT  . S IBX("SLPRV",IBCNT,"SLC")=SLC SLPRVQ ;  Q  ;  ;PROVIDER  ;Input:  ; IB399 - ien of #399  ; IBPROV:  ; "C"- to get info for CURRENT provider  ; "O"- to get info for all others (in this case the array will contain info fot two providers  ; IBRES - array for results (by reference)  ; IBFRMTYP - Form Type  ; IBCARE - Care Type  ; IBCURR - current bill payer sequence  ;  ;Output:  ; IBRES - array to get back info (by reference)  ; IBRES(IBPROV,PRNUM,PRTYPE,SEQ#)=PROV^INSUR^IDTYPE^ID^FORMTYP^CARETYP  ; where:  ; IBPROV - see input parameter  ; PRNUM: 1=primary insurance provider, 2= secondary, 3 -tretiary  ; PRTYPE: Provider type(FUNCTION)   ; SEQ# : sequence number (1st is used for ID1, 2nd - for ID2, etc)  ; PROV : provider/VARIABLEPTR  ; INSUR: Insurance PTR #36 or NONE  ; IDTYPE: ID type  ; ID: ID   ; FORMTYP: Form type 1=UB,2=1500  ; CARETYP: Care type 0=both inp/outp,1=inpatient, 2=outpatient PROVIDER(IB399,IBPROV,IBRES,IBFRMTYP,IBCARE,IBCURR,IBXDATA) ;  N IBZ,IBRESARR,IBLIMIT  S IBRESARR=""  Q:IBCURR="A"  ;PATIENT's bill IB\*2.0\*447 BI Changes IBPROV to IBCURR  I IBPROV="C" D  . S IBLIMIT=5  . D:$$ISINSUR^IBCEF71(IBCURR,IB399) PROVINF(IB399,$S(IBCURR="T":3,IBCURR="S":2,IBCURR="P":1,1:1),.IBRESARR,1,IBPROV,IBFRMTYP,IBCARE,IBLIMIT,IBCURR,.IBXDATA)  I IBPROV="O" D  . S IBLIMIT=3  . I IBCURR="P" D  .. D:$$ISINSUR^IBCEF71("S",IB399) PROVINF(IB399,2,.IBRESARR,1,IBPROV,IBFRMTYP,IBCARE,IBLIMIT,IBCURR,.IBXDATA)  .. D:$$ISINSUR^IBCEF71("T",IB399) PROVINF(IB399,3,.IBRESARR,2,IBPROV,IBFRMTYP,IBCARE,IBLIMIT,IBCURR,.IBXDATA)  . I IBCURR="S" D  .. D:$$ISINSUR^IBCEF71("P",IB399) PROVINF(IB399,1,.IBRESARR,1,IBPROV,IBFRMTYP,IBCARE,IBLIMIT,IBCURR,.IBXDATA)  .. D:$$ISINSUR^IBCEF71("T",IB399) PROVINF(IB399,3,.IBRESARR,2,IBPROV,IBFRMTYP,IBCARE,IBLIMIT,IBCURR,.IBXDATA)  . I IBCURR="T" D  .. D:$$ISINSUR^IBCEF71("P",IB399) PROVINF(IB399,1,.IBRESARR,1,IBPROV,IBFRMTYP,IBCARE,IBLIMIT,IBCURR,.IBXDATA)  .. D:$$ISINSUR^IBCEF71("S",IB399) PROVINF(IB399,2,.IBRESARR,2,IBPROV,IBFRMTYP,IBCARE,IBLIMIT,IBCURR,.IBXDATA)  M IBRES=IBRESARR  Q  ;  ;-- PROVINF --  ;Create array with prov info  ;Input:  ; IB399 - ien #399  ; IBPRNUM - 1=prim ins, 2= sec, 3 -tert  ; IBRES - for results  ; IBSORT - to sort OTHER INSURANCE data   ; if PROVINF is called for "C" mode of PROVIDER subroutine then   ; IBSORT can be any (say 1)  ; if PROVINF is called for "O" mode then can be more than set of data  ; - need to sort array to use it (like IBXDATA(1) and IBXDATA(2))  ; for mode "O" it should be 1 or 2 (see PROVIDER section)  ;IBINSTP - "C" -current ins, "O"-other  ;IBFRMTYP - Form Type  ;IBCARE - Care Type  ;IBLIMIT - Limits on Secondary  ;IBCURR - Current Insurance  ;IBXDAYA - Revenue Code Array  ;Output:  ; IBRES(PRNUM,PRTYPE,SEQ#)=PROV^INSUR^IDTYPE^ID^FORMTYP^CARETYP  ; where:(see PROVIDER) PROVINF(IB399,IBPRNUM,IBRES,IBSORT,IBINSTP,IBFRMTYP,IBCARE,IBLIMIT,IBCURR,IBXDATA) ;  I $G(IB399)="" G PROVINFQ  I $G(IBINSTP)="" G PROVINFQ  I +$G(IBSORT)=0 S IBSORT=$G(IBPRNUM)  N IBPRTYP,IBINSCO,IBPROV,IB35591,IBN,IBEXC  S IBN=0  S IBINSCO=+$P($G(^DGCR(399,IB399,"M")),"^",IBPRNUM)  S IB35591=$$CH35591^IBCEF72(IBINSCO,IBFRMTYP,IBCARE)  S IBPRTYP=0  F  S IBPRTYP=$O(^DGCR(399,IB399,"PRV","B",IBPRTYP)) Q:'IBPRTYP  D  . N Z,IB355OV,IBPROV,IBARR  . S IBPROV=$$PROVPTR(IB399,IBPRTYP,0),IBEXC=""  . Q:+IBPROV=0  . S Z=$O(^DGCR(399,IB399,"PRV","B",IBPRTYP,0)) I Z S Z=$G(^DGCR(399,IB399,"PRV",Z,0))  . D GETPRV(IBINSCO,IBFRMTYP,IBCARE,IBPROV,.IBARR,IBPRTYP,IBINSTP,Z)  . M IBRES("PROVINF",IB399,IBINSTP)=IBARR  I $D(IBRES("PROVINF",IB399,IBINSTP,IBSORT))>1 S IBRES("PROVINF",IB399,IBINSTP,IBSORT)=$S(IBPRNUM=3:"T",IBPRNUM=2:"S",1:"P")  N SLC,CPLNK  S SLC=0  F  S SLC=$O(IBXDATA(SLC)) Q:'SLC   S IBXSAVE("SLC")=+SLC D  . S CPLNK=$G(IBXDATA(SLC,"CPLNK")) I 'CPLNK Q  . S IBPRTYP=0  . F  S IBPRTYP=$O(^DGCR(399,IB399,"CP",CPLNK,"LNPRV","B",IBPRTYP)) Q:'IBPRTYP  D  .. N Z,IBPROV,IBARR  .. S IBPROV=$$PROVPTR(IB399,IBPRTYP,CPLNK),IBEXC=""  .. Q:'+IBPROV  .. S Z=$O(^DGCR(399,IB399,"CP",CPLNK,"LNPRV","B",IBPRTYP,0)) I Z S Z=$G(^DGCR(399,IB399,"CP",CPLNK,"LNPRV",Z,0))  .. D GETPRV(IBINSCO,IBFRMTYP,IBCARE,IBPROV,.IBARR,IBPRTYP,IBINSTP,Z)  .. M IBRES("L-PROV",IB399,SLC,IBINSTP)=IBARR  . I $D(IBRES("L-PROV",IB399,SLC,IBINSTP,IBSORT))>1 S IBRES("L-PROV",IB399,SLC,IBINSTP,IBSORT)=$S(IBPRNUM=3:"T",IBPRNUM=2:"S",1:"P")  ; PROVINFQ ;Exit PROVINF  Q  ; GETPRV(IBINSCO,IBFRMTYP,IBCARE,IBPROV,IBRES,IBPRTYP,IBINSTP,IBD) ;  I "CO"'[$G(IBINSTP) G GETPRVQ  N IBRETARR,IBNPI,IBN,IBMRAND,IB355OV S IBRETARR=0,IB355OV=""  D PRACT^IBCEF71(IBINSCO,IBFRMTYP,IBCARE,IBPROV,.IBRETARR,IBPRTYP,$G(IBINSTP))  I $P(IBD,U,IBPRNUM+4)'="",$P(IBD,U,IBPRNUM+11)'="" S IB355OV=$P(IBD,U,IBPRNUM+4)\_U\_$P(IBD,U,IBPRNUM+11)  S IBN=0,IBMRAND=$$MCRONBIL^IBEFUNC(IB399)  ;Calculate MEDICARE (WNR) specific provider qualifier and ID for CMS-1500 secondary claims  I "34"[$G(IBPRTYP),$G(IBFRMTYP)=2,IBMRAND S IB355OV=$$MCR24K^IBCEU3(IB399,IBPROV)\_"^12"  I $P(IB355OV,U,2) D  . I $$CHCKSEC^IBCEF73(IBFRMTYP,IBPRTYP,$G(IBINSTP),$P($G(^IBE(355.97,+$P(IB355OV,U,2),0)),U,3)) D  .. S IBEXC=$P(IB355OV,U,2),IBN=IBN+1,IBRES(IBSORT,IBPRTYP,IBN)="OVERRIDE^"\_IBINSCO\_U\_$P($G(^IBE(355.97,+IBEXC,0)),U,3)\_U\_$P(IB355OV,U)\_"^^^^^"\_+IBEXC  I IB35591'="",IBEXC'=$P(IB35591,U,3) I $$CHCKSEC^IBCEF73(IBFRMTYP,IBPRTYP,$G(IBINSTP),$P(IB35591,"^")) D  . S IBN=IBN+1,IBRES(IBSORT,IBPRTYP,IBN)="DEFAULT^"\_IBINSCO\_"^"\_IB35591\_"^^",$P(IBRES(IBSORT,IBPRTYP,IBN),U,9)=$P(IB35591,U,3)  S IBNPI=$$NPI^IBCEFP1(IBPROV)  D SORT^IBCEF77(IBSORT,IBPRTYP,IB399,.IBRETARR,.IBRES,IBN,IBEXC,IBPRNUM,IBLIMIT)  S IBRES(IBSORT,IBPRTYP,0)="PRIMARY"\_U\_U\_$$STRIP^IBCEF76($S(IBNPI]"":"XX",1:"")\_U\_IBNPI,1,U,IBSTRIP)  F IBN=1:1 Q:'$D(IBRES(IBSORT,IBPRTYP,IBN)) S $P(IBRES(IBSORT,IBPRTYP,IBN),U,3,4)=$$STRIP^IBCEF76($P(IBRES(IBSORT,IBPRTYP,IBN),U,3,4),1,U,IBSTRIP)  S IBRES(IBSORT,IBPRTYP,"NAME")=$$NAME^IBCEFP1(IBPROV,IBIFN,$P(IBD,U,3),$P(IBD,U,8))  S IBRES(IBSORT,IBPRTYP,"ENTITY TYPE")=$S(IBPROV'["355.93,":1,$P($G(^IBA(355.93,+IBPROV,0)),U,2)=2:1,1:2)  S IBRES(IBSORT,IBPRTYP,"TAXONOMY")=$$TAXON^IBCEFP1(IBPROV,$P(IBD,U,15))  S IBRES(IBSORT,IBPRTYP,"COBID")=$$COBID^IBCEFP1(IB399,IBPRTYP,IBMRAND,IBD)  S IBRES(IBSORT,IBPRTYP)=IBPROV GETPRVQ ;  Q  ; PROVPTR(IBIEN399,IBFUNC,IBCP) ; Retrieve Provider Pointer from appropriate file  N IBN,RSLT  S IBCP=+$G(IBCP)  I 'IBCP D  . S IBN=$O(^DGCR(399,IBIEN399,"PRV","B",IBFUNC,0))  . I +IBN=0 S RSLT=0 Q  . S RSLT=$P($G(^DGCR(399,IBIEN399,"PRV",+IBN,0)),U,2)  I IBCP D  . S IBN=$O(^DGCR(399,IBIEN399,"CP",IBCP,"LNPRV","B",IBFUNC,0))  . I +IBN=0 S RSLT=0 Q  . S RSLT=$P($G(^DGCR(399,IBIEN399,"CP",IBCP,"LNPRV",+IBN,0)),U,2)  Q RSLT  ;  ;Input:  ;IBXIEN - Internal Entry Number for the current bill/claim  ;IBXSAVE - Array for returning the data  ;  ;Output:  ;IBXSAVE - Data Array AMB(IBXIEN,IBXSAVE) ; Gather Ambulance Data for AMB Record(s) - IB\*2.0\*447/TAZ  N NODE,CODE,CNT,IBXDATA  K IBXSAVE("AMB")  F NODE="U5","U6","U7" S IBXDATA=$G(^DGCR(399,IBXIEN,NODE)) I $TR(IBXDATA,U)'="" S IBXSAVE("AMB",NODE)=IBXDATA  S CODE="",CNT=0  F  S CODE=$O(^DGCR(399,IBXIEN,"U9","B",CODE)) Q:'CODE  D  . S IBXDATA=$P($G(^IBE(353.5,CODE,0)),U,1) I IBXDATA="" Q  . S CNT=CNT+1,IBXSAVE("AMB","U9",CNT)=IBXDATA  Q  ; SNDS2(IBXDATA,PIECE) ;Determine if a SUB2 record is necessary.  ; Input: IBXDATA  ; May contain data from field 232 of file 399.  ; Output: IBXDATA  ; Returns Output for piece 2 or 3 or 1 for any other piece (like 1.5)  ;Any time that ONE of the following criteria is met we should send a SUB2 record  ; 1. Incoming IBXDATA is not null SEND - Non-VA facility in field 232 of file 399  ; 2. If the service facility is a VA Institution in file 4 or a non-VA facility in file 355.93 SEND  ; 3. Not a switchback payer $$SENDSF^IBCEF79(IBXIEN)'=0 SEND  ;  ; MRD;IB\*2.0\*516 - Due to fields being marked for deletion, the  ; function $$SENDSF^IBCEF79 will always return '1'. Refer to  ; that function and INSFLGS^^IBCEF79 for more information.  ;  I IBXDATA="" D  . N Z  . S Z=$P($$B^IBCEF79(IBXIEN),U,3)  . ;S Z1=$$SENDSF^IBCEF79(IBXIEN)  . ;S IBXDATA=$S(Z="":0,'Z1:0,1:1)  . S IBXDATA=$S(Z="":0,1:1)  . Q  I 'IBXDATA S IBXDATA=""  I IBXDATA'="" S IBXDATA=$S(PIECE=2:77,PIECE=3:2,1:1)  Q IBXDATA | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCEFP ;ALB/TAZ - Provider ID functions ;28-OCT-10  ;;2.0;INTEGRATED BILLING;\*\*432,447,473,516,**592**\*\*;21-MAR-94;Build 123  ;;Per VA Directive 6402, this routine should not be modified.  ;  Q  ; ALLIDS(IBIFN,IBXSAVE,IBSTRIP,SEG) ; Return all of the Provider IDS  I '$D(IBSTRIP) S IBSTRIP=0  I '$D(SEG) S SEG=""  N IBXIEN,ARINFO,ARID,ARQ,IBFRMTYP,ARIEN,ARINS,Z0,DAT,I,SORT1,SORT2,SORT3,COB,IBCCOB,IBCARE,IBCURR,IBXDATA,NPI,CUROTH **;JWS;IB\*2.0\*592;US131**  S IBFRMTYP=$$FT^IBCEF(IBIFN),IBFRMTYP=$S(IBFRMTYP=2:2,IBFRMTYP=3:1,**IBFRMTYP=7:4**,1:0)  S IBCARE=$S($$ISRX^IBCEF1(IBIFN):3,1:0) ;if an Rx refill bill  S:IBCARE=0 IBCARE=$$INPAT^IBCEF(IBIFN,1) S:'IBCARE IBCARE=2 ;1-inp,2-out  S IBCURR=$$COB^IBCEF(IBIFN) ;current bill payer sequence  ;don't create anything if form type not CMS-1500 or UB  I IBFRMTYP,'+$G(IBXSAVE("PROVINF",IBIFN)) D  . N IBZ,CUROTH  **. ;JWS;IB\*2.0\*592;US131**  . I IBFRMTYP=2**!(IBFRMTYP=4**) D OUTPT^IBCEF11(IBIFN,0)  . I IBFRMTYP=1 D HOS^IBCEF22(IBIFN)  . ; START IB\*2.0\*447 BI  . I IBCURR="A" D  Q  .. N IBRESARR  .. S IBLIMIT=5  .. D PROVINF(IBIFN,1,.IBRESARR,1,"C",IBFRMTYP,IBCARE,IBLIMIT,IBCURR,.IBXDATA)  .. M IBXSAVE=IBRESARR  .. S IBXSAVE("PROVINF",IBIFN)=IBIFN  . ; END IB\*2.0\*447 BI  . F CUROTH="C","O" D PROVIDER(IBIFN,CUROTH,.IBZ,IBFRMTYP,IBCARE,IBCURR,.IBXDATA) M IBXSAVE=IBZ  . S IBXSAVE("PROVINF",IBIFN)=IBIFN  . Q  ;  D LFIDS^IBCEF76(IBIFN,.IBXSAVE,IBSTRIP,SEG) ; Get the Lab/Facility IDs  S NPI=$P($$ORGNPI^IBCEF73A(IBIFN),U,1)  F CUROTH="C","O" D  . S IBXSAVE("LAB/FAC",IBIFN,CUROTH,1,0)=$S(NPI]"":"XX",1:"")\_U\_NPI  ;  S IBFRMTYP=$$FT^IBCEF(IBIFN)  S ARIEN=$S(IBFRMTYP=2:3,1:4)  S IBCCOB=$$COBN^IBCEF(IBIFN) ; Current Insurance  F COB=1:1:3 D  . S SORT1=$S(COB=IBCCOB:"C",1:"O")  . S SORT2=$S(SORT1="C":1,COB=1:1,COB=2&(IBCCOB=1):1,1:2)  . S ARINFO=$G(IBXSAVE("PROVINF",IBIFN,SORT1,SORT2,ARIEN,1))  . ;  . D BPIDS^IBCEF75(IBIFN,.IBXSAVE,SORT1,SORT2,COB,IBSTRIP,SEG)  ;  D EN^IBCEF81(.IBXSAVE)  ;  I SEG="OPR1" D  . I '$D(IBXSAVE("PROVINF",IBIFN,"C",1,3)) D  G ALLIDSQ  .. N SLC,CRED S SLC=0  .. F  S SLC=$O(IBXSAVE("L-PROV",IBIFN,SLC)) Q:'SLC  D  I $D(IBXSAVE("PROVINF",IBIFN,"C",1,3,"NAME")) Q  ... S CRED=$P($G(IBXSAVE("L-PROV",IBIFN,SLC,"C",1,3,"NAME")),U,4)  ... I CRED]"" S $P(IBXSAVE("PROVINF",IBIFN,"C",1,3,"NAME"),U,4)=CRED  ;  I SEG="LPUR"!(SEG="LPUR1"),$$SUB1OK^IBCEP8A(IBIFN),$G(IBXSAVE("SLC")) D  G ALLIDSQ  . N IBCNT,OUT  . ;IB\*2.0\*473/TAZ - Moved ID lookup into seperate function.  . D PSID(IBIFN,.IBXSAVE,.IDS)  . ;IB\*2.0\*473/TAZ - END  . S $P(OUT,U,6)=IDS(0)  . S $P(OUT,U,7,8)=IDS(1) ; secondary id (1) ^ secondary id qualifier(1)  . F IBCNT=1:1:IBXSAVE("SLC") S IBXSAVE("SLPRV",IBCNT)=OUT,IBXSAVE("SLPRV",IBCNT,"SLC")=IBCNT  ;  I SEG="LOPE" D SLPRV(IBIFN,.IBXSAVE,2) G ALLIDSQ  I SEG="LOP1" D SLPRV(IBIFN,.IBXSAVE,9) G ALLIDSQ  I SEG="LREN" D SLPRV(IBIFN,.IBXSAVE,3) G ALLIDSQ  I SEG="LSUP" D SLPRV(IBIFN,.IBXSAVE,5) G ALLIDSQ  I SEG="LREF" D SLPRV(IBIFN,.IBXSAVE,1) G ALLIDSQ  **;JWS;IB\*2.0\*592;US131**  **I SEG="LSUR" D SLPRV(IBIFN,.IBXSAVE,6) G ALLIDSQ  I SEG="LSUR1" D SLPRV(IBIFN,.IBXSAVE,6) G ALLIDSQ**  ; ALLIDSQ ;  Q  ; PSID(IBIFN,IBXPROV,IBXIDS) ; Build array of either the Fac/Lab ID or Rendering Provider IDs  ;IB\*2.0\*473/TAZ - Created a function to standardize IDs in LDAT and LPUR  ; Input:  ; IBXIEN - Internal Entry Number of claim  ; IBXPROV - Provider Array  ; IBXIDS - Array for IDs  ; Output:  ; IBXIDS(0) - Primary ID  ; IBXIDS(1) - Secondary ID  ;  N LINE,PLINE,PID,SID,SIDQ  K IBXIDS  ; Get Lab/Facility IDs  S (PID,SID,SIDQ)=""  S PID=$P($G(IBXPROV("LAB/FAC",IBIFN,"C",1,0)),U,2)  F LINE=1:1 Q:'$D(IBXPROV("LAB/FAC",IBIFN,"C",1,LINE)) D  I SID'="" Q  . S SIDQ=$P($G(IBXPROV("LAB/FAC",IBIFN,"C",1,LINE)),U) I ",0B,1G,G2,"'[(","\_SIDQ\_",") S SIDQ="" Q  . S SID=$P($G(IBXPROV("LAB/FAC",IBIFN,"C",1,LINE)),U,2)  I $L(PID)!$L(SID) S IBXIDS(0)=PID,IBXIDS(1)=SID\_U\_SIDQ G PSIDQ  ; Get Claim Level Rendering Provider IDs  S PID=$P($G(IBXPROV("PROVINF",IBIFN,"C",1,3,0)),U,4) ; Get claim level Rendering Provider NPI  F LINE=1:1 Q:'$D(IBXPROV("PROVINF",IBIFN,"C",1,3,LINE)) D  I SID'="" Q  . S SIDQ=$P($G(IBXPROV("PROVINF",IBIFN,"C",1,3,LINE)),U,3) I ",0B,1G,G2,"'[(","\_SIDQ\_",") S SIDQ="" Q  . S SID=$P($G(IBXPROV("PROVINF",IBIFN,"C",1,3,LINE)),U,4)  I $L(PID)!$L(SID) S IBXIDS(0)=PID,IBXIDS(1)=SID\_U\_SIDQ G PSIDQ  ; Get Line Level Rendering Provider IDs  F PLINE=1:1 Q:'$D(IBXPROV("L-PROV",IBIFN,PLINE)) D  I $L(PID)!$L(SID) Q  . S PID=$P($G(IBXPROV("L-PROV",IBIFN,PLINE,"C",1,3,0)),U,4)  . F LINE=1:1 Q:'$D(IBXPROV("L-PROV",IBIFN,PLINE,"C",1,3,LINE)) D  I SID'="" Q  .. S SIDQ=$P($G(IBXPROV("L-PROV",IBIFN,PLINE,"C",1,3,LINE)),U,3) I ",0B,1G,G2,"'[(","\_SIDQ\_",") S SIDQ="" Q  .. S SID=$P($G(IBXPROV("L-PROV",IBIFN,PLINE,"C",1,3,LINE)),U,4)  I $L(PID)!$L(SID) S IBXIDS(0)=PID,IBXIDS(1)=SID\_U\_SIDQ PSIDQ ;  Q  ; SLPRV(IBXIEN,IBX,PRTYPE) ;Build SLPRV nodes for the line provider type record  N SLC,DATA,IBCNT,NAME,OUT  S (SLC,IBCNT)=0  F  S SLC=$O(IBX("L-PROV",IBXIEN,SLC)) Q:'SLC  D  . I '$D(IBX("L-PROV",IBXIEN,SLC,"C",1,PRTYPE)) Q  . S NAME=$G(IBX("L-PROV",IBXIEN,SLC,"C",1,PRTYPE,"NAME"))  . S OUT=$P(NAME,U,1,3)\_U\_$P(NAME,U,5)\_U\_$G(IBX("L-PROV",IBXIEN,SLC,"C",1,PRTYPE,"TAXONOMY"))\_U\_$P($G(IBX("L-PROV",IBXIEN,SLC,"C",1,PRTYPE,0)),U,4)  . F IBN=1:1 Q:'$D(IBX("L-PROV",IBXIEN,SLC,"C",1,PRTYPE,IBN)) D  .. S DATA=$G(IBX("L-PROV",IBXIEN,SLC,"C",1,PRTYPE,IBN))  .. I ",0B,1G,G2,LU,"[(","\_$P(DATA,U,3)\_",") S OUT=OUT\_U\_$P(DATA,U,4)\_U\_$P(DATA,U,3)  . S IBCNT=IBCNT+1  . S IBX("SLPRV",IBCNT)=OUT  . S IBX("SLPRV",IBCNT,"SLC")=SLC SLPRVQ ;  Q  ;  ;PROVIDER  ;Input:  ; IB399 - ien of #399  ; IBPROV:  ; "C"- to get info for CURRENT provider  ; "O"- to get info for all others (in this case the array will contain info fot two providers  ; IBRES - array for results (by reference)  ; IBFRMTYP - Form Type  ; IBCARE - Care Type  ; IBCURR - current bill payer sequence  ;  ;Output:  ; IBRES - array to get back info (by reference)  ; IBRES(IBPROV,PRNUM,PRTYPE,SEQ#)=PROV^INSUR^IDTYPE^ID^FORMTYP^CARETYP  ; where:  ; IBPROV - see input parameter  ; PRNUM: 1=primary insurance provider, 2= secondary, 3 -tretiary  ; PRTYPE: Provider type(FUNCTION)   ; SEQ# : sequence number (1st is used for ID1, 2nd - for ID2, etc)  ; PROV : provider/VARIABLEPTR  ; INSUR: Insurance PTR #36 or NONE  ; IDTYPE: ID type  ; ID: ID   ; FORMTYP: Form type 1=UB,2=1500  ; CARETYP: Care type 0=both inp/outp,1=inpatient, 2=outpatient PROVIDER(IB399,IBPROV,IBRES,IBFRMTYP,IBCARE,IBCURR,IBXDATA) ;  N IBZ,IBRESARR,IBLIMIT  S IBRESARR=""  Q:IBCURR="A"  ;PATIENT's bill IB\*2.0\*447 BI Changes IBPROV to IBCURR  I IBPROV="C" D  . S IBLIMIT=5  . D:$$ISINSUR^IBCEF71(IBCURR,IB399) PROVINF(IB399,$S(IBCURR="T":3,IBCURR="S":2,IBCURR="P":1,1:1),.IBRESARR,1,IBPROV,IBFRMTYP,IBCARE,IBLIMIT,IBCURR,.IBXDATA)  I IBPROV="O" D  . S IBLIMIT=3  . I IBCURR="P" D  .. D:$$ISINSUR^IBCEF71("S",IB399) PROVINF(IB399,2,.IBRESARR,1,IBPROV,IBFRMTYP,IBCARE,IBLIMIT,IBCURR,.IBXDATA)  .. D:$$ISINSUR^IBCEF71("T",IB399) PROVINF(IB399,3,.IBRESARR,2,IBPROV,IBFRMTYP,IBCARE,IBLIMIT,IBCURR,.IBXDATA)  . I IBCURR="S" D  .. D:$$ISINSUR^IBCEF71("P",IB399) PROVINF(IB399,1,.IBRESARR,1,IBPROV,IBFRMTYP,IBCARE,IBLIMIT,IBCURR,.IBXDATA)  .. D:$$ISINSUR^IBCEF71("T",IB399) PROVINF(IB399,3,.IBRESARR,2,IBPROV,IBFRMTYP,IBCARE,IBLIMIT,IBCURR,.IBXDATA)  . I IBCURR="T" D  .. D:$$ISINSUR^IBCEF71("P",IB399) PROVINF(IB399,1,.IBRESARR,1,IBPROV,IBFRMTYP,IBCARE,IBLIMIT,IBCURR,.IBXDATA)  .. D:$$ISINSUR^IBCEF71("S",IB399) PROVINF(IB399,2,.IBRESARR,2,IBPROV,IBFRMTYP,IBCARE,IBLIMIT,IBCURR,.IBXDATA)  M IBRES=IBRESARR  Q  ;  ;-- PROVINF --  ;Create array with prov info  ;Input:  ; IB399 - ien #399  ; IBPRNUM - 1=prim ins, 2= sec, 3 -tert  ; IBRES - for results  ; IBSORT - to sort OTHER INSURANCE data   ; if PROVINF is called for "C" mode of PROVIDER subroutine then   ; IBSORT can be any (say 1)  ; if PROVINF is called for "O" mode then can be more than set of data  ; - need to sort array to use it (like IBXDATA(1) and IBXDATA(2))  ; for mode "O" it should be 1 or 2 (see PROVIDER section)  ;IBINSTP - "C" -current ins, "O"-other  ;IBFRMTYP - Form Type  ;IBCARE - Care Type  ;IBLIMIT - Limits on Secondary  ;IBCURR - Current Insurance  ;IBXDAYA - Revenue Code Array  ;Output:  ; IBRES(PRNUM,PRTYPE,SEQ#)=PROV^INSUR^IDTYPE^ID^FORMTYP^CARETYP  ; where:(see PROVIDER) PROVINF(IB399,IBPRNUM,IBRES,IBSORT,IBINSTP,IBFRMTYP,IBCARE,IBLIMIT,IBCURR,IBXDATA) ;  I $G(IB399)="" G PROVINFQ  I $G(IBINSTP)="" G PROVINFQ  I +$G(IBSORT)=0 S IBSORT=$G(IBPRNUM)  N IBPRTYP,IBINSCO,IBPROV,IB35591,IBN,IBEXC  S IBN=0  S IBINSCO=+$P($G(^DGCR(399,IB399,"M")),"^",IBPRNUM)  S IB35591=$$CH35591^IBCEF72(IBINSCO,IBFRMTYP,IBCARE)  S IBPRTYP=0  F  S IBPRTYP=$O(^DGCR(399,IB399,"PRV","B",IBPRTYP)) Q:'IBPRTYP  D  . N Z,IB355OV,IBPROV,IBARR  . S IBPROV=$$PROVPTR(IB399,IBPRTYP,0),IBEXC=""  . Q:+IBPROV=0  . S Z=$O(^DGCR(399,IB399,"PRV","B",IBPRTYP,0)) I Z S Z=$G(^DGCR(399,IB399,"PRV",Z,0))  . D GETPRV(IBINSCO,IBFRMTYP,IBCARE,IBPROV,.IBARR,IBPRTYP,IBINSTP,Z)  . M IBRES("PROVINF",IB399,IBINSTP)=IBARR  I $D(IBRES("PROVINF",IB399,IBINSTP,IBSORT))>1 S IBRES("PROVINF",IB399,IBINSTP,IBSORT)=$S(IBPRNUM=3:"T",IBPRNUM=2:"S",1:"P")  N SLC,CPLNK  S SLC=0  F  S SLC=$O(IBXDATA(SLC)) Q:'SLC   S IBXSAVE("SLC")=+SLC D  . S CPLNK=$G(IBXDATA(SLC,"CPLNK")) I 'CPLNK Q  . S IBPRTYP=0  . F  S IBPRTYP=$O(^DGCR(399,IB399,"CP",CPLNK,"LNPRV","B",IBPRTYP)) Q:'IBPRTYP  D  .. N Z,IBPROV,IBARR  .. S IBPROV=$$PROVPTR(IB399,IBPRTYP,CPLNK),IBEXC=""  .. Q:'+IBPROV  .. S Z=$O(^DGCR(399,IB399,"CP",CPLNK,"LNPRV","B",IBPRTYP,0)) I Z S Z=$G(^DGCR(399,IB399,"CP",CPLNK,"LNPRV",Z,0))  .. D GETPRV(IBINSCO,IBFRMTYP,IBCARE,IBPROV,.IBARR,IBPRTYP,IBINSTP,Z)  .. M IBRES("L-PROV",IB399,SLC,IBINSTP)=IBARR  . I $D(IBRES("L-PROV",IB399,SLC,IBINSTP,IBSORT))>1 S IBRES("L-PROV",IB399,SLC,IBINSTP,IBSORT)=$S(IBPRNUM=3:"T",IBPRNUM=2:"S",1:"P")  ; PROVINFQ ;Exit PROVINF  Q  ; GETPRV(IBINSCO,IBFRMTYP,IBCARE,IBPROV,IBRES,IBPRTYP,IBINSTP,IBD) ;  I "CO"'[$G(IBINSTP) G GETPRVQ  N IBRETARR,IBNPI,IBN,IBMRAND,IB355OV S IBRETARR=0,IB355OV=""  D PRACT^IBCEF71(IBINSCO,IBFRMTYP,IBCARE,IBPROV,.IBRETARR,IBPRTYP,$G(IBINSTP))  I $P(IBD,U,IBPRNUM+4)'="",$P(IBD,U,IBPRNUM+11)'="" S IB355OV=$P(IBD,U,IBPRNUM+4)\_U\_$P(IBD,U,IBPRNUM+11)  S IBN=0,IBMRAND=$$MCRONBIL^IBEFUNC(IB399)  ;Calculate MEDICARE (WNR) specific provider qualifier and ID for CMS-1500 secondary claims  I "34"[$G(IBPRTYP),$G(IBFRMTYP)=2,IBMRAND S IB355OV=$$MCR24K^IBCEU3(IB399,IBPROV)\_"^12"  I $P(IB355OV,U,2) D  . I $$CHCKSEC^IBCEF73(IBFRMTYP,IBPRTYP,$G(IBINSTP),$P($G(^IBE(355.97,+$P(IB355OV,U,2),0)),U,3)) D  .. S IBEXC=$P(IB355OV,U,2),IBN=IBN+1,IBRES(IBSORT,IBPRTYP,IBN)="OVERRIDE^"\_IBINSCO\_U\_$P($G(^IBE(355.97,+IBEXC,0)),U,3)\_U\_$P(IB355OV,U)\_"^^^^^"\_+IBEXC  I IB35591'="",IBEXC'=$P(IB35591,U,3) I $$CHCKSEC^IBCEF73(IBFRMTYP,IBPRTYP,$G(IBINSTP),$P(IB35591,"^")) D  . S IBN=IBN+1,IBRES(IBSORT,IBPRTYP,IBN)="DEFAULT^"\_IBINSCO\_"^"\_IB35591\_"^^",$P(IBRES(IBSORT,IBPRTYP,IBN),U,9)=$P(IB35591,U,3)  S IBNPI=$$NPI^IBCEFP1(IBPROV)  D SORT^IBCEF77(IBSORT,IBPRTYP,IB399,.IBRETARR,.IBRES,IBN,IBEXC,IBPRNUM,IBLIMIT)  S IBRES(IBSORT,IBPRTYP,0)="PRIMARY"\_U\_U\_$$STRIP^IBCEF76($S(IBNPI]"":"XX",1:"")\_U\_IBNPI,1,U,IBSTRIP)  F IBN=1:1 Q:'$D(IBRES(IBSORT,IBPRTYP,IBN)) S $P(IBRES(IBSORT,IBPRTYP,IBN),U,3,4)=$$STRIP^IBCEF76($P(IBRES(IBSORT,IBPRTYP,IBN),U,3,4),1,U,IBSTRIP)  S IBRES(IBSORT,IBPRTYP,"NAME")=$$NAME^IBCEFP1(IBPROV,IBIFN,$P(IBD,U,3),$P(IBD,U,8))  S IBRES(IBSORT,IBPRTYP,"ENTITY TYPE")=$S(IBPROV'["355.93,":1,$P($G(^IBA(355.93,+IBPROV,0)),U,2)=2:1,1:2)  S IBRES(IBSORT,IBPRTYP,"TAXONOMY")=$$TAXON^IBCEFP1(IBPROV,$P(IBD,U,15))  S IBRES(IBSORT,IBPRTYP,"COBID")=$$COBID^IBCEFP1(IB399,IBPRTYP,IBMRAND,IBD)  S IBRES(IBSORT,IBPRTYP)=IBPROV GETPRVQ ;  Q  ; PROVPTR(IBIEN399,IBFUNC,IBCP) ; Retrieve Provider Pointer from appropriate file  N IBN,RSLT  S IBCP=+$G(IBCP)  I 'IBCP D  . S IBN=$O(^DGCR(399,IBIEN399,"PRV","B",IBFUNC,0))  . I +IBN=0 S RSLT=0 Q  . S RSLT=$P($G(^DGCR(399,IBIEN399,"PRV",+IBN,0)),U,2)  I IBCP D  . S IBN=$O(^DGCR(399,IBIEN399,"CP",IBCP,"LNPRV","B",IBFUNC,0))  . I +IBN=0 S RSLT=0 Q  . S RSLT=$P($G(^DGCR(399,IBIEN399,"CP",IBCP,"LNPRV",+IBN,0)),U,2)  Q RSLT  ;  ;Input:  ;IBXIEN - Internal Entry Number for the current bill/claim  ;IBXSAVE - Array for returning the data  ;  ;Output:  ;IBXSAVE - Data Array AMB(IBXIEN,IBXSAVE) ; Gather Ambulance Data for AMB Record(s) - IB\*2.0\*447/TAZ  N NODE,CODE,CNT,IBXDATA  K IBXSAVE("AMB")  F NODE="U5","U6","U7" S IBXDATA=$G(^DGCR(399,IBXIEN,NODE)) I $TR(IBXDATA,U)'="" S IBXSAVE("AMB",NODE)=IBXDATA  S CODE="",CNT=0  F  S CODE=$O(^DGCR(399,IBXIEN,"U9","B",CODE)) Q:'CODE  D  . S IBXDATA=$P($G(^IBE(353.5,CODE,0)),U,1) I IBXDATA="" Q  . S CNT=CNT+1,IBXSAVE("AMB","U9",CNT)=IBXDATA  Q  ; SNDS2(IBXDATA,PIECE) ;Determine if a SUB2 record is necessary.  ; Input: IBXDATA  ; May contain data from field 232 of file 399.  ; Output: IBXDATA  ; Returns Output for piece 2 or 3 or 1 for any other piece (like 1.5)  ;Any time that ONE of the following criteria is met we should send a SUB2 record  ; 1. Incoming IBXDATA is not null SEND - Non-VA facility in field 232 of file 399  ; 2. If the service facility is a VA Institution in file 4 or a non-VA facility in file 355.93 SEND  ; 3. Not a switchback payer $$SENDSF^IBCEF79(IBXIEN)'=0 SEND  ;  ; MRD;IB\*2.0\*516 - Due to fields being marked for deletion, the  ; function $$SENDSF^IBCEF79 will always return '1'. Refer to  ; that function and INSFLGS^^IBCEF79 for more information.  ;  I IBXDATA="" D  . N Z  . S Z=$P($$B^IBCEF79(IBXIEN),U,3)  . ;S Z1=$$SENDSF^IBCEF79(IBXIEN)  . ;S IBXDATA=$S(Z="":0,'Z1:0,1:1)  . S IBXDATA=$S(Z="":0,1:1)  . Q  I 'IBXDATA S IBXDATA=""  I IBXDATA'="" S IBXDATA=$S(PIECE=2:77,PIECE=3:2,1:1)  Q IBXDATA | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCF23 | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** |  | | | | | | | | |
| **Related Options** | None | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  |  | | | | |  | | | |
| **Data Dictionary (DD) References** | BILL/CLAIMS [#399] | | | | | | | | |
| **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** | | | | | | | | | |
| IBCF23 ;ALB/ARH - HCFA 1500 19-90 DATA (block 24, procs and charges) ;12-JUN-93  ;;2.0;INTEGRATED BILLING;\*\*52,80,106,122,51,152,137,402,432,488,547\*\*;21-MAR-94;Build 119  ;;Per VA Directive 6402, this routine should not be modified.  ;  ;requires IBIFN,IB(0),IB("U"),IB("U1"), returns # of line items in IBFLD(24)  ;rev code array: IBRC("proc^division^basc flag^bedsection^rev code^unit chrg^Rx seq #")=units  ;proc array: IBCP(initial print ord)=proc date^proc^division^basc flag^dx^pos^tos^modifier^unit chrg^purch chrg amt^anesthesia mins^emerg indicator  ; IBCP(initial print order,seq #)=auxillary data  ;proc array: IBSS("proc^division^basc flag^dx^pos^tos^modifier^unit chrg^Rx seq #")=lowest inital print order  ;print order array: IBPO(final print ord,emerg indicator,initial print order)=""  ;print array: IBFLD(24,I)=begin dt^end dt^pos^tos^proc^dx^unit chrg^units^modifier pointer ien(s) separated by commas^purch chrg amt^anesthesia mins^emerg indicator  ; IBFLD(24,I,"AUX")=[auxillary data]  ; = "AUX" node of proc entry  ; IBFLD(24,I,"RX")= soft link to file 362.4 or null  ; if service is Rx, but no soft link  ;   ;charge item link: IBLINK(CPT IFN in multiple,RCIFN) = proc^division^basc flag^bedsection^rev code^unit chrg^rx seq #  ;  ; dx's used in arrays are ref #s  ; IB\*547 added backwards compatibility so that MRAs and EOBs would still roll/split procedures the same way as when the claim  ; was created. Any claim transmitted before IB\*547 was installed will roll/split the original way and any new  ; claim or claim transmitted after IB\*547 was transmitted will roll/split the new way.  ; When updating in the future care must be taken to disable/remove older code so that only new changes are  ; affected by the IBNWPTCH variable.  ; RVC ; charges array  D RVCE(,IBIFN)  Q  ; RVCE(IBXIEN,IBIFN) ;Entry for EDI formatter call (IBXIEN will be defined)  ; IBIFN required  N IBRC,IBCP,IBSS,IBSSO,IBSS1,IBPO,IBLINK,IBLINK1,IBLINKRX,IBK,IBAUXLN  N IBI,IBJ,IB11,IBLN,IBPDT,IBCHARG,IBMOD,IBPC,IBRX,IBRXF,IBPO2A,IBAUX,IBNWPTCH  ;  ; IB\*547/TAZ - Add IBNWPTCH variable.  S IBRX=0,IBNWPTCH=$$IBNWPTCH^IBCF23A(IBIFN,"IB\*2.0\*547")  S IBI=0 F  S IBI=$O(^DGCR(399,IBIFN,"RC",IBI)) Q:'IBI  S IBLN=^(IBI,0) D  . S IBSS="",IBPC=0 F IBJ=6,7,0,5,1,2,14 S IBPC=IBPC+1 S:IBJ $P(IBSS,U,IBPC,IBPC+1)=($P(IBLN,U,IBJ)\_U)  . I $P(IBSS,U,2)="" S $P(IBSS,U,2)=$P(^DGCR(399,IBIFN,0),U,22)  . I +IBSS S $P(IBSS,U)=$P(IBSS,U)\_";ICPT("  . S $P(IBSS,U,3)=$S($D(^DGCR(399,"ASC1",+$P(IBLN,U,6),IBIFN,IBI)):1,1:"")  . I +$P(IBLN,U,10)=3 D  Q  ; Rx  .. I '$P(IBLN,U,15) S IBRX=IBRX+1,$P(IBSS,U,8)=(100+IBRX)  .. I $P(IBLN,U,15) S $P(IBSS,U,8)=$P(IBLN,U,15)  .. S IBRC(IBSS,"RX")=$P(IBLN,U,11)\_U\_IBI\_U\_$P(IBLN,U,15)  .. S IBRC(IBSS)=$G(IBRC(IBSS))+1  . ;  . S IBRC(IBSS)=$G(IBRC(IBSS))+$P(IBLN,U,3) ; total units for similar RC  . I "4"[+$P(IBLN,U,10),$P(IBLN,U,11) D  ; Soft-link proc with the rev cd  .. S IBLINK(+$P(IBLN,U,11),IBI)=IBSS  .. S $P(IBLINK(+$P(IBLN,U,11),IBI),U,7)=$P(IBLN,U,14)  . I $P(IBLN,U,10) D  .. S IBLINK1(IBSS,IBI)=$P(IBLN,U,10)\_U\_+$P(IBLN,U,11)  . S IBRC(IBSS,"LNK")=IBI  ;  S IBSSO="" F  S IBSSO=$O(IBRC(IBSSO)) Q:IBSSO=""  I $D(IBRC(IBSSO,"RX")) D  . S IBSS=IBSSO,IBI=$P(IBRC(IBSSO,"RX"),U,2),IB11=$P(IBRC(IBSSO,"RX"),U,3)  . S IBRC(IBSSO)=1,IBLINKRX($S($P(IBSSO,U)>0:$P(IBSSO,U),$P($G(^DGCR(399,IBIFN,"CP",+IB11,0)),U)'="":$P(^(0),U),1:0),+IB11,+IBRC(IBSSO,"RX"))=IBSSO K IBRC(IBSSO,"RX")  ;  D PRC^IBCF23A ; Extract procedures PO ; print order array w/chrgs  ; combine multiple entries of same proc onto one line item via print order  ;if both have print orders defined then they should not be combined onto one line item  ;"proc^division^basc^dx^pos^tos^modifier(s)^unit chrg^purchased chg" must all be the same as well as the emergency indicator and all 'aux flds'  N IBP,Z,IBPO11  ;IB\*547/TAZ - set entire node into IBSS for post IB\*547 claims  ;S IBPO="" F S IBPO=$O(IBCP(IBPO)) Q:'IBPO S IBCP=IBCP(IBPO),IBSS=$P(IBCP,U,2,9),IBSS1="\*"\_$G(IBCP(IBPO,"AUX")),IBAUX=0 D  S IBPO="" F  S IBPO=$O(IBCP(IBPO)) Q:'IBPO  S IBCP=IBCP(IBPO),IBSS=$P(IBCP,U,2,$S(IBNWPTCH:$L(IBCP,U),1:9)),IBSS1="\*"\_$G(IBCP(IBPO,"AUX")),IBAUX=0 D  . I $D(IBSS(IBSS)),'$D(IBCP(IBPO,"RX")),IBPO>1000 D  Q  ; combine lines  .. I 'IBAUX S IBAUX=$$AUXOK^IBCF23A(.IBSS,IBSS1)  .. S IBPO1=$S(IBAUX:IBSS(IBSS,IBAUX),1:IBPO)  .. I 'IBAUX S Z=+$O(IBSS(IBSS,"A"),-1)+1,IBSS(IBSS,Z)=IBPO  .. I IBPO>1000!(IBPO1>1000) S IBPO(IBPO1,+$P(IBCP,U,12),IBPO)="" D  ... I $O(IBCP(IBPO,"L",0)) S Z=$O(IBCP(IBPO,"L",0)),IBPO(IBPO1,+$P(IBCP,U,12),IBPO,"L",Z)=IBCP(IBPO,"L",Z) K IBCP(IBPO,"L",Z)  . S IBAUX=+$O(IBSS(IBSS,"A"),-1)+1,IBSS(IBSS,"AUX-X",IBAUX)=IBSS1  . S IBSS(IBSS,IBAUX)=+IBPO,IBPO(+IBPO,+$P(IBCP,U,12),IBPO)=""  . S Z=0 F  S Z=$O(IBCP(IBPO,Z)) Q:'Z  S IBPO(+IBPO,+$P(IBCP,U,12),IBPO,Z)=""  . I $O(IBCP(IBPO,"L",0)) S Z=$O(IBCP(IBPO,"L",0)),IBPO(+IBPO,+$P(IBCP,U,12),IBPO,"L",Z)=IBCP(IBPO,"L",Z) K IBCP(IBPO,"L",Z)  . S IBSS(IBSS,IBAUX,"AUX")=IBSS1,IBPO(+IBPO,+$P(IBCP,U,12),IBPO,"AUX")=$E(IBSS1,2,$L(IBSS1))  . I $D(IBCP(IBPO,"RX")) S IBPO(+IBPO,+$P(IBCP,U,12),IBPO,"RX")=IBCP(IBPO,"RX"),IBSS(IBSS,IBAUX,"RX")=IBCP(IBPO)  ;  ; Find any remaining rev codes w/units that ref existing procedures  S IBP(0)=0  F IBP=3,2 Q:$G(IBP(0)) S IBRV="" F  S IBRV=$O(IBRC(IBRV)) Q:IBRV=""  I IBRV,IBRC(IBRV) D  . S IBSS1=$O(IBSS($P(IBRV,U,1,IBP))) Q:$P(IBRV,U,1,IBP)'=$P(IBSS1,U,1,IBP)  . S IBP(0)=1,Z=0  . F  S Z=$O(IBSS(IBSS1,Z)) Q:'Z  I $G(IBSS(IBSS1,Z)) D  Q  .. I $D(IBCP(IBSS(IBSS1,Z))),$P(IBCP(IBSS(IBSS1,Z)),U,9)=$P(IBSS1,U,8) D  ... N Q,Q0  ... ; S Q=$O(IBCP(""),-1)+1,Q0=$P(IBCP(IBSS(IBSS1,Z)),U,12) ; WCJ;IB\*488  ... S Q=IBSS(IBSS1,Z),Q0=$P(IBCP(IBSS(IBSS1,Z)),U,12) ; WCJ;IB\*488  ... ;M IBPO(Q,$P(IBCP(IBSS(IBSS1,Z)),U,12),Q)=IBPO(IBSS(IBSS1,Z),$P(IBCP(IBSS(IBSS1,Z)),U,12),IBSS(IBSS1,Z)),IBCP(Q)=IBCP(IBSS(IBSS1,Z)) ; WCJ;IB\*488  ... ;S $P(IBCP(Q),U,9)=$P(IBRV,U,6) ; WCJ;IB\*488  ... ;F Z0=1:1:(IBRC(IBRV)-1) S IBPO(Q,Q0,Q+(Z0\*.01))=IBPO(Q,Q0,Q) I Z0=99,(IBRC(IBRV)'=100) S IBPO(Q,Q0,Q\_".991")=(IBRC(IBRV)-1)\_"^99" Q ; Only put first 99 in array  ... F Z0=1:1:(IBRC(IBRV)) S IBPO(Q,Q0,Q+(Z0\*.001))=IBPO(Q,Q0,Q) ; changing to .001 allows us up to 999 and the units field only allows 800. ; WCJ;IB\*488  ... S IBRC(IBRV)=0  ; PRTARR ;print proc array  S IBREV="",IBPO1="",IBI=0 F  S IBPO1=$O(IBPO(IBPO1)) Q:IBPO1=""  D  . K IBRXF  . S IBEMG="" F  S IBEMG=$O(IBPO(IBPO1,IBEMG)) Q:IBEMG=""!("01"'[IBEMG) S IBPO2="" D  .. S IBDT1=99999999,IBDT2="",(IBMIN,IBUNIT)=0,(IBCHARG,IBAUX)=""  .. F  S IBPO2=$O(IBPO(IBPO1,IBEMG,IBPO2)) Q:IBPO2=""  D  ... I IBPO2#1=.991 D  Q:IBPO2#1=.991  .... N Z  .... S Z=$G(IBPO(IBPO1,IBEMG,IBPO2)) Q:'Z  .... I ($P(Z,U,2)+1)>Z Q  .... S $P(IBPO(IBPO1,IBEMG,IBPO2),U,2)=($P(Z,U,2)+1),IBPO2=(IBPO2\1)\_".99"  ... S Z=0 F  S Z=$O(IBPO(IBPO1,IBEMG,IBPO2,Z)) Q:'Z  S IBUNIT=IBUNIT+1  ... I $D(IBCP(IBPO1)) S IBPO11=IBPO1  ... S IBPO2A=$S($D(IBCP(IBPO2\1)):IBPO2\1,'$D(IBCP(IBPO2)):IBPO11,1:IBPO2)  ... S IBCHARG=$P(IBCP(IBPO2A),U,9),IBPCHG=$P(IBCP(IBPO2A),U,10)  ... ; I IBCHARG<10000,IBCHARG\*(IBUNIT+1)'<10000 D Q ;$9,999 limit per line ;WCJ IB\*488  ... I IBCHARG<10000000,IBCHARG\*(IBUNIT+1)'<10000000 D  Q  ; increased to $9,999,999 charge limit per line since that is printed form space limit ;WCJ IB\*488  .... N Z S Z=$O(IBPO(IBPO1\1+1),-1),Z=Z+$S(IBPO1+.001'=Z:.001,1:0) M IBPO(Z,IBEMG,IBPO2)=IBPO(IBPO1,IBEMG,IBPO2) K IBPO(IBPO1,IBEMG,IBPO2)  ... S IBUNIT=IBUNIT+1,IBSS=IBCP(IBPO2A),IBMIN=IBMIN+$P(IBSS,U,11)  ... S IBSS=$G(IBSS)\_U\_$G(IBCP(IBPO2A,"LNK"))  ... S Z=$O(IBPO(IBPO1,IBEMG,IBPO2,"L",0)) I Z D  .... S Z0=0  .... F Z=Z:1 Q:'$O(IBPO(IBPO1,IBEMG,IBPO2,"L",0))!(Z0=IBUNIT) I $D(IBPO(IBPO1,IBEMG,IBPO2,"L",Z)) S IBSS("L",Z)=IBPO(IBPO1,IBEMG,IBPO2,"L",Z),Z0=Z0+1 K IBPO(IBPO1,IBEMG,IBPO2,"L",Z)  ... S:IBDT1>+IBSS IBDT1=+IBSS S:IBDT2<+IBSS IBDT2=+IBSS  .. S IBAUX=$G(IBCP(IBPO1,"AUX")) S:$D(IBCP(IBPO1,"RX")) IBRXF=IBCP(IBPO1,"RX")  .. I IBUNIT D B24^IBCF23A  .. K IBRXF  ;  ;print any chrgs not associated with a proc (ie. not enough procs or proc not in "CP" level)  S IBRV="" F  S IBRV=$O(IBRC(IBRV)) Q:IBRV=""  I +IBRC(IBRV) D  D B24^IBCF23A K IBRXF  . S IBUNIT=+IBRC(IBRV),IBCHARG=$P(IBRV,U,6),IBDT1=+IB("U"),IBDT2=$P(IB("U"),U,2),IBREV=$P(IBRV,U,5),IBEMG=0,IBAUX=""  . S IBSS="^"\_$S(+IBRV:$P(IBRV,U),1:$P($G(^DGCR(399.1,+$P(IBRV,U,4),0)),U))  . S IBSS=$G(IBSS)\_U\_$$RC2CP^IBCEF22(IBIFN,+$G(IBRC(IBRV,"LNK")))  . S Z=$O(IBLINK1(IBRV,0)) I Z D  .. S Z0=0  .. F Z=Z:1 Q:'$O(IBLINK1(IBRV,0))!(Z0=IBUNIT) I $D(IBLINK1(IBRV,Z)) S IBSS("L",Z)=IBLINK1(IBRV,Z),Z0=Z0+1 K IBLINK1(IBRV,Z)  ; OFFSET ;  S IBFLD(24)=IBI ;line item count  K IBRC,IBCP,IBSS,IBPO,IBPO1,IBPO2,IBLN,IBRV,IBRV1,IBPDT,IBDT1,IBDT2,IBCHARG,IBMIN,IBUNIT,IBREV,IBLINK,IBLINK1,IBEMG,IBPCHG,Z  Q  ; DATE(X) ; Fm dt in X ==> YYYYMMDD  Q $$DT^IBCEFG1(X,,"D8")  ; B24 ; Moved to IBCF23A for space  D B24^IBCF23A  Q  ; | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCF23 ;ALB/ARH - HCFA 1500 19-90 DATA (block 24, procs and charges) ;12-JUN-93  ;;2.0;INTEGRATED BILLING;\*\*52,80,106,122,51,152,137,402,432,488,547,**592**\*\*;21-MAR-94;Build 119  ;;Per VA Directive 6402, this routine should not be modified.  ;  ;requires IBIFN,IB(0),IB("U"),IB("U1"), returns # of line items in IBFLD(24)  ;rev code array: IBRC("proc^division^basc flag^bedsection^rev code^unit chrg^Rx seq #")=units  ;proc array: IBCP(initial print ord)=proc date^proc^division^basc flag^dx^pos^tos^modifier^unit chrg^purch chrg amt^anesthesia mins^emerg indicator  ; IBCP(initial print order,seq #)=auxillary data  ;proc array: IBSS("proc^division^basc flag^dx^pos^tos^modifier^unit chrg^Rx seq #")=lowest inital print order  ;print order array: IBPO(final print ord,emerg indicator,initial print order)=""  ;print array: IBFLD(24,I)=begin dt^end dt^pos^tos^proc^dx^unit chrg^units^modifier pointer ien(s) separated by commas^purch chrg amt^anesthesia mins^emerg indicator  ; IBFLD(24,I,"AUX")=[auxillary data]  ; = "AUX" node of proc entry  ; IBFLD(24,I,"RX")= soft link to file 362.4 or null  ; if service is Rx, but no soft link  ;   ;charge item link: IBLINK(CPT IFN in multiple,RCIFN) = proc^division^basc flag^bedsection^rev code^unit chrg^rx seq #  ;  ; dx's used in arrays are ref #s  ; IB\*547 added backwards compatibility so that MRAs and EOBs would still roll/split procedures the same way as when the claim  ; was created. Any claim transmitted before IB\*547 was installed will roll/split the original way and any new  ; claim or claim transmitted after IB\*547 was transmitted will roll/split the new way.  ; When updating in the future care must be taken to disable/remove older code so that only new changes are  ; affected by the IBNWPTCH variable.  ; RVC ; charges array  D RVCE(,IBIFN)  Q  ; RVCE(IBXIEN,IBIFN) ;Entry for EDI formatter call (IBXIEN will be defined)  ; IBIFN required  N IBRC,IBCP,IBSS,IBSSO,IBSS1,IBPO,IBLINK,IBLINK1,IBLINKRX,IBK,IBAUXLN  **;JWS;IB\*2.0\*592;US131**  N IBI,IBJ,IB11,IBLN,IBPDT,IBCHARG,IBMOD,IBPC,IBRX,IBRXF,IBPO2A,IBAUX,IBNWPTCH,**IBDEN,IBDEN1**  ;  ; IB\*547/TAZ - Add IBNWPTCH variable.  S IBRX=0,IBNWPTCH=$$IBNWPTCH^IBCF23A(IBIFN,"IB\*2.0\*547")  S IBI=0 F  S IBI=$O(^DGCR(399,IBIFN,"RC",IBI)) Q:'IBI  S IBLN=^(IBI,0) D  . S IBSS="",IBPC=0 F IBJ=6,7,0,5,1,2,14 S IBPC=IBPC+1 S:IBJ $P(IBSS,U,IBPC,IBPC+1)=($P(IBLN,U,IBJ)\_U)  . I $P(IBSS,U,2)="" S $P(IBSS,U,2)=$P(^DGCR(399,IBIFN,0),U,22)  . I +IBSS S $P(IBSS,U)=$P(IBSS,U)\_";ICPT("  . S $P(IBSS,U,3)=$S($D(^DGCR(399,"ASC1",+$P(IBLN,U,6),IBIFN,IBI)):1,1:"")  . I +$P(IBLN,U,10)=3 D  Q  ; Rx  .. I '$P(IBLN,U,15) S IBRX=IBRX+1,$P(IBSS,U,8)=(100+IBRX)  .. I $P(IBLN,U,15) S $P(IBSS,U,8)=$P(IBLN,U,15)  .. S IBRC(IBSS,"RX")=$P(IBLN,U,11)\_U\_IBI\_U\_$P(IBLN,U,15)  .. S IBRC(IBSS)=$G(IBRC(IBSS))+1  . ;  . S IBRC(IBSS)=$G(IBRC(IBSS))+$P(IBLN,U,3) ; total units for similar RC  . I "4"[+$P(IBLN,U,10),$P(IBLN,U,11) D  ; Soft-link proc with the rev cd  .. S IBLINK(+$P(IBLN,U,11),IBI)=IBSS  .. S $P(IBLINK(+$P(IBLN,U,11),IBI),U,7)=$P(IBLN,U,14)  . I $P(IBLN,U,10) D  .. S IBLINK1(IBSS,IBI)=$P(IBLN,U,10)\_U\_+$P(IBLN,U,11)  . S IBRC(IBSS,"LNK")=IBI  ;  S IBSSO="" F  S IBSSO=$O(IBRC(IBSSO)) Q:IBSSO=""  I $D(IBRC(IBSSO,"RX")) D  . S IBSS=IBSSO,IBI=$P(IBRC(IBSSO,"RX"),U,2),IB11=$P(IBRC(IBSSO,"RX"),U,3)  . S IBRC(IBSSO)=1,IBLINKRX($S($P(IBSSO,U)>0:$P(IBSSO,U),$P($G(^DGCR(399,IBIFN,"CP",+IB11,0)),U)'="":$P(^(0),U),1:0),+IB11,+IBRC(IBSSO,"RX"))=IBSSO K IBRC(IBSSO,"RX")  ;  D PRC^IBCF23A ; Extract procedures PO ; print order array w/chrgs  ; combine multiple entries of same proc onto one line item via print order  ;if both have print orders defined then they should not be combined onto one line item  ;"proc^division^basc^dx^pos^tos^modifier(s)^unit chrg^purchased chg" must all be the same as well as the emergency indicator and all 'aux flds'  N IBP,Z,IBPO11  ;IB\*547/TAZ - set entire node into IBSS for post IB\*547 claims  ;S IBPO="" F S IBPO=$O(IBCP(IBPO)) Q:'IBPO S IBCP=IBCP(IBPO),IBSS=$P(IBCP,U,2,9),IBSS1="\*"\_$G(IBCP(IBPO,"AUX")),IBAUX=0 D  S IBPO="" F  S IBPO=$O(IBCP(IBPO)) Q:'IBPO  S IBCP=IBCP(IBPO),IBSS=$P(IBCP,U,2,$S(IBNWPTCH:$L(IBCP,U),1:9)),IBSS1="\*"\_$G(IBCP(IBPO,"AUX")),IBAUX=0 D  . I $D(IBSS(IBSS)),'$D(IBCP(IBPO,"RX")),IBPO>1000 D  Q  ; combine lines  .. I 'IBAUX S IBAUX=$$AUXOK^IBCF23A(.IBSS,IBSS1)  .. S IBPO1=$S(IBAUX:IBSS(IBSS,IBAUX),1:IBPO)  .. I 'IBAUX S Z=+$O(IBSS(IBSS,"A"),-1)+1,IBSS(IBSS,Z)=IBPO  .. I IBPO>1000!(IBPO1>1000) S IBPO(IBPO1,+$P(IBCP,U,12),IBPO)="" D  ... I $O(IBCP(IBPO,"L",0)) S Z=$O(IBCP(IBPO,"L",0)),IBPO(IBPO1,+$P(IBCP,U,12),IBPO,"L",Z)=IBCP(IBPO,"L",Z) K IBCP(IBPO,"L",Z)  . S IBAUX=+$O(IBSS(IBSS,"A"),-1)+1,IBSS(IBSS,"AUX-X",IBAUX)=IBSS1  . S IBSS(IBSS,IBAUX)=+IBPO,IBPO(+IBPO,+$P(IBCP,U,12),IBPO)=""  . S Z=0 F  S Z=$O(IBCP(IBPO,Z)) Q:'Z  S IBPO(+IBPO,+$P(IBCP,U,12),IBPO,Z)=""  . I $O(IBCP(IBPO,"L",0)) S Z=$O(IBCP(IBPO,"L",0)),IBPO(+IBPO,+$P(IBCP,U,12),IBPO,"L",Z)=IBCP(IBPO,"L",Z) K IBCP(IBPO,"L",Z)  . S IBSS(IBSS,IBAUX,"AUX")=IBSS1,IBPO(+IBPO,+$P(IBCP,U,12),IBPO,"AUX")=$E(IBSS1,2,$L(IBSS1))  . I $D(IBCP(IBPO,"RX")) S IBPO(+IBPO,+$P(IBCP,U,12),IBPO,"RX")=IBCP(IBPO,"RX"),IBSS(IBSS,IBAUX,"RX")=IBCP(IBPO) **. ;JWS;IB\*2.0\*592;US131  . I $D(IBCP(IBPO,"DEN")) S IBPO(+IBPO,+$P(IBCP,U,12),IBPO,"DEN")=IBCP(IBPO,"DEN")  . I $D(IBCP(IBPO,"DEN1")) M IBPO(+IBPO,+$P(IBCP,U,12),IBPO,"DEN1")=IBCP(IBPO,"DEN1")  . ;end ;JWS;IB\*2.0\*592;US131;**  ; Find any remaining rev codes w/units that ref existing procedures  S IBP(0)=0  F IBP=3,2 Q:$G(IBP(0)) S IBRV="" F  S IBRV=$O(IBRC(IBRV)) Q:IBRV=""  I IBRV,IBRC(IBRV) D  . S IBSS1=$O(IBSS($P(IBRV,U,1,IBP))) Q:$P(IBRV,U,1,IBP)'=$P(IBSS1,U,1,IBP)  . S IBP(0)=1,Z=0  . F  S Z=$O(IBSS(IBSS1,Z)) Q:'Z  I $G(IBSS(IBSS1,Z)) D  Q  .. I $D(IBCP(IBSS(IBSS1,Z))),$P(IBCP(IBSS(IBSS1,Z)),U,9)=$P(IBSS1,U,8) D  ... N Q,Q0  ... ; S Q=$O(IBCP(""),-1)+1,Q0=$P(IBCP(IBSS(IBSS1,Z)),U,12) ; WCJ;IB\*488  ... S Q=IBSS(IBSS1,Z),Q0=$P(IBCP(IBSS(IBSS1,Z)),U,12) ; WCJ;IB\*488  ... ;M IBPO(Q,$P(IBCP(IBSS(IBSS1,Z)),U,12),Q)=IBPO(IBSS(IBSS1,Z),$P(IBCP(IBSS(IBSS1,Z)),U,12),IBSS(IBSS1,Z)),IBCP(Q)=IBCP(IBSS(IBSS1,Z)) ; WCJ;IB\*488  ... ;S $P(IBCP(Q),U,9)=$P(IBRV,U,6) ; WCJ;IB\*488  ... ;F Z0=1:1:(IBRC(IBRV)-1) S IBPO(Q,Q0,Q+(Z0\*.01))=IBPO(Q,Q0,Q) I Z0=99,(IBRC(IBRV)'=100) S IBPO(Q,Q0,Q\_".991")=(IBRC(IBRV)-1)\_"^99" Q ; Only put first 99 in array  ... F Z0=1:1:(IBRC(IBRV)) S IBPO(Q,Q0,Q+(Z0\*.001))=IBPO(Q,Q0,Q) ; changing to .001 allows us up to 999 and the units field only allows 800. ; WCJ;IB\*488  ... S IBRC(IBRV)=0  ; PRTARR ;print proc array  S IBREV="",IBPO1="",IBI=0 F  S IBPO1=$O(IBPO(IBPO1)) Q:IBPO1=""  D  . K IBRXF  . S IBEMG="" F  S IBEMG=$O(IBPO(IBPO1,IBEMG)) Q:IBEMG=""!("01"'[IBEMG) S IBPO2="" D  .. S IBDT1=99999999,IBDT2="",(IBMIN,IBUNIT)=0,(IBCHARG,IBAUX)=""  .. F  S IBPO2=$O(IBPO(IBPO1,IBEMG,IBPO2)) Q:IBPO2=""  D  ... I IBPO2#1=.991 D  Q:IBPO2#1=.991  .... N Z  .... S Z=$G(IBPO(IBPO1,IBEMG,IBPO2)) Q:'Z  .... I ($P(Z,U,2)+1)>Z Q  .... S $P(IBPO(IBPO1,IBEMG,IBPO2),U,2)=($P(Z,U,2)+1),IBPO2=(IBPO2\1)\_".99"  ... S Z=0 F  S Z=$O(IBPO(IBPO1,IBEMG,IBPO2,Z)) Q:'Z  S IBUNIT=IBUNIT+1  ... I $D(IBCP(IBPO1)) S IBPO11=IBPO1  ... S IBPO2A=$S($D(IBCP(IBPO2\1)):IBPO2\1,'$D(IBCP(IBPO2)):IBPO11,1:IBPO2)  ... S IBCHARG=$P(IBCP(IBPO2A),U,9),IBPCHG=$P(IBCP(IBPO2A),U,10)  ... ; I IBCHARG<10000,IBCHARG\*(IBUNIT+1)'<10000 D Q ;$9,999 limit per line ;WCJ IB\*488  ... I IBCHARG<10000000,IBCHARG\*(IBUNIT+1)'<10000000 D  Q  ; increased to $9,999,999 charge limit per line since that is printed form space limit ;WCJ IB\*488  .... N Z S Z=$O(IBPO(IBPO1\1+1),-1),Z=Z+$S(IBPO1+.001'=Z:.001,1:0) M IBPO(Z,IBEMG,IBPO2)=IBPO(IBPO1,IBEMG,IBPO2) K IBPO(IBPO1,IBEMG,IBPO2)  ... S IBUNIT=IBUNIT+1,IBSS=IBCP(IBPO2A),IBMIN=IBMIN+$P(IBSS,U,11)  ... S IBSS=$G(IBSS)\_U\_$G(IBCP(IBPO2A,"LNK"))  ... S Z=$O(IBPO(IBPO1,IBEMG,IBPO2,"L",0)) I Z D  .... S Z0=0  .... F Z=Z:1 Q:'$O(IBPO(IBPO1,IBEMG,IBPO2,"L",0))!(Z0=IBUNIT) I $D(IBPO(IBPO1,IBEMG,IBPO2,"L",Z)) S IBSS("L",Z)=IBPO(IBPO1,IBEMG,IBPO2,"L",Z),Z0=Z0+1 K IBPO(IBPO1,IBEMG,IBPO2,"L",Z)  ... S:IBDT1>+IBSS IBDT1=+IBSS S:IBDT2<+IBSS IBDT2=+IBSS  .. S IBAUX=$G(IBCP(IBPO1,"AUX")) S:$D(IBCP(IBPO1,"RX")) IBRXF=IBCP(IBPO1,"RX")  .. **;JWS;IB\*2.0\*592;US131  .. S IBDEN=$G(IBCP(IBPO1,"DEN"))  .. I $D(IBCP(IBPO1,"DEN1")) M IBDEN1=IBCP(IBPO1,"DEN1")  .. ;end ;JWS;IB\*2.0\*592;US131**  .. I IBUNIT D B24^IBCF23A  .. K IBRXF  ;  ;print any chrgs not associated with a proc (ie. not enough procs or proc not in "CP" level)  S IBRV="" F  S IBRV=$O(IBRC(IBRV)) Q:IBRV=""  I +IBRC(IBRV) D  D B24^IBCF23A K IBRXF  . S IBUNIT=+IBRC(IBRV),IBCHARG=$P(IBRV,U,6),IBDT1=+IB("U"),IBDT2=$P(IB("U"),U,2),IBREV=$P(IBRV,U,5),IBEMG=0,IBAUX=""  . S IBSS="^"\_$S(+IBRV:$P(IBRV,U),1:$P($G(^DGCR(399.1,+$P(IBRV,U,4),0)),U))  . S IBSS=$G(IBSS)\_U\_$$RC2CP^IBCEF22(IBIFN,+$G(IBRC(IBRV,"LNK")))  . S Z=$O(IBLINK1(IBRV,0)) I Z D  .. S Z0=0  .. F Z=Z:1 Q:'$O(IBLINK1(IBRV,0))!(Z0=IBUNIT) I $D(IBLINK1(IBRV,Z)) S IBSS("L",Z)=IBLINK1(IBRV,Z),Z0=Z0+1 K IBLINK1(IBRV,Z)  ; OFFSET ;  S IBFLD(24)=IBI ;line item count  K IBRC,IBCP,IBSS,IBPO,IBPO1,IBPO2,IBLN,IBRV,IBRV1,IBPDT,IBDT1,IBDT2,IBCHARG,IBMIN,IBUNIT,IBREV,IBLINK,IBLINK1,IBEMG,IBPCHG,Z  Q  ; DATE(X) ; Fm dt in X ==> YYYYMMDD  Q $$DT^IBCEFG1(X,,"D8")  ; B24 ; Moved to IBCF23A for space  D B24^IBCF23A  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** | BILL/CLAIMS [#399]  EDI TRANSMIT BILL [#364]  EDI TRANSMISSION BATCH [#364.1] | | | | | | | | |
| **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**,592**\*\*;21-MAR-94;Build 1  ;;Per VA Directive 6402, this routine should not be modified.  ;  ; $$INSTALDT^XPDUTL(IBPATCH,.IBARY) - ICR 10141  ; B24 ; set individual entries in print array, external format  ; IBAUX = additional data for EDI output  ; IBRXF = array of RX procedures  **;JWS;IB\*2.0\*592;US131  ; IBDEN = Dental data for EDI output  ; IBDEN1 = array of Dental data for EDI output**  N IBX,Z,IBD1,IBD2,IBCPLINK  S IBI=IBI+1,IBPROC=$P(IBSS,U,2),IBD1=$$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  **;JWS;IB\*2.0\*592;US131  I $G(IBDEN)'="" S IBFLD(24,IBI,"DEN")=$G(IBDEN)  I $D(IBDEN1) M IBFLD(24,IBI,"DEN1")=IBDEN1  ;end ;JWS;IB\*2.0\*592;US131**  Q  ; AUXOK(IBSS,IBSS1) ; Check all other flds are the same to combine procs  ; IBSS = subscript of IBCP to check for dups to combine - pass by ref  ; IBSS(IBSS,"AUX-X",n) = all the previously extracted line items for the  ; same set of basic data, but having different "AUX" data  ; IBSS1 = the "AUX" data of the current IBCP entry  ;  ; Returns entry # in IBSS array if match found, or 0 if no match  ; Set the IBSS "AUX-X" node for no match  N Z,Z0  S Z=0 F  S Z=$O(IBSS(IBSS,"AUX-X",Z)) Q:'Z  I IBSS1=IBSS(IBSS,"AUX-X",Z) Q  I 'Z S Z0=+$O(IBSS(IBSS,"AUX-X",""),-1)+1,IBSS(IBSS,"AUX-X",Z0)=IBSS1  Q +Z  ; PRC ; Extract procedure data for HCFA 1500  ; IBRC(IBSS) = #rev codes with same billing criteria (IBSS)  ; IBLINK('CP' ien,'RC' ien) = IBSS including modifiers,rx seq in pc 7,8  ; IBLINK1(IBSS, 'RC' ien) = auto (1)^ 'CP' ien (soft link)  ;  ; proc array w/chrg  **;JWS;IB\*2.0\*592;US131**  N IBPR,IBP,**IBDENLN  S IBI=0 F  S IBI=$O(^DGCR(399,IBIFN,"CP",IBI)) Q:'IBI  S IBLN=^(IBI,0),IBAUXLN=$G(^("AUX")),IBDENLN=$G(^("DEN")) D  . I $O(^DGCR(399,IBIFN,"CP",IBI,"DEN1",0)) M IBDENLN("DEN1")=^DGCR(399,IBIFN,"CP",IBI,"DEN1")**  **. ;end ;JWS;IB\*2.0\*592;US131**  . N Z,Z0,Z1,Q1  . S IBPDT=$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  **. ;JWS;IB\*2.0\*592;US131  . I $G(IBDENLN)'="" S IBCP(IBPO,"DEN")=IBDENLN  . I $O(IBDENLN("DEN1",0)) M IBCP(IBPO,"DEN1")=IBDENLN("DEN1")  . ;end ;JWS;IB\*2.0\*592;US131**  . ; 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 | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBJTCA1 | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** |  | | | | | | | | |
| **Related Options** | None | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  |  | | | | |  | | | |
| **Data Dictionary (DD) References** | BILL/CLAIMS [#399] | | | | | | | | |
| **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** | | | | | | | | | |
| IBJTCA1 ;ALB/ARH - TPI CLAIMS INFO BUILD ;10/31/07 14:17  ;;2.0;INTEGRATED BILLING;\*\*39,80,106,137,223,276,363,384,432,452,473,497,521,516\*\*;21-MAR-94;Build 123  ;;Per VA Directive 6402, this routine should not be modified.  ; BLD ; build array for Third Party Joint Inquiry Claims Info screen, IBIFN must be defined  ;  N X,IBY,IBZ,IBZ0,IBI,IBT,IBD,IBLN,IBLR,IBD0,IBDI1,IBDM,IBDM1,IBDU,IBDS,IBDU2,IBID0,IBID13,IBNC,IBTC,IBTW,IBSW,IBGRPB,IBGRPE,IBWNR,IBDTX,IBBX19,IBPRVO,IBNABP,IBLVL,IBCNT,IBPRVTYP,IBVL  N IBXSAVE  ; IB\*2.0\*473 bi  S VALMCNT=0,X="",IBD0=$G(^DGCR(399,+$G(IBIFN),0)) I IBD0="" S VALMQUIT="" G BLDQ  F IBI="M","M1","U","S","U2","TX" S @("IBD"\_IBI)=$G(^DGCR(399,+IBIFN,IBI))  S IBDI1=$P(IBD0,U,21),IBDI1=$S(IBDI1="S":2,IBDI1="T":3,1:1) S IBDI1=$$POLICY^IBCEF(IBIFN,,IBDI1)  S IBID0=$G(^DIC(36,+IBDI1,0)),IBID13=$G(^DIC(36,+IBDI1,.13))  ;  S (IBLN,VALMCNT)=1  ;  ; MRD;IB\*2.0\*516 - Try to make the following more readable; also  ; added IBTC(7), IBTW(7) and IBSW(7).  ;IB\*2.0\*432/TAZ - Added IBTW(6) and IBSW(6)  ;S (IBNC(1),IBTC(1),IBTC(4),IBTC(6))=2,IBTC(5)=78,(IBNC(2),IBTC(2))=42,IBNC(3)=35,IBTW(1)=15,IBTW(2)=16,IBTW(4)=12,IBTW(5)=1,IBTW(6)=20,IBSW(1)=23,IBSW(2)=21,IBSW(4)=60,IBSW(5)=1,IBSW(6)=49  ;  S IBNC(1)=2,IBTC(1)=2,IBTW(1)=15,IBSW(1)=23  S IBNC(2)=42,IBTC(2)=42,IBTW(2)=16,IBSW(2)=21  S IBNC(3)=35  S IBTC(4)=2,IBTW(4)=12,IBSW(4)=60  S IBTC(5)=78,IBTW(5)=1,IBSW(5)=1  S IBTC(6)=2,IBTW(6)=20,IBSW(6)=49  S IBTC(7)=2,IBTW(7)=20,IBSW(7)=58  ;  S IBLR=1  ;  S IBT="Insurance Demographics" S IBLN=$$SETN(IBT,IBLN,IBLR,1)  S IBWNR=$$WNRBILL^IBEFUNC(IBIFN)  S IBNABP=$$NABP^IBNCPDPU(IBIFN)  S IBT=$S(IBWNR:" \*",1:" ")\_"Bill Payer: ",IBD=$P(IBID0,U,1) S IBLN=$$SET(IBT,IBD,IBLN,IBLR)  S IBT="Claim Address: " D  S IBD=$P(IBDM,U,5) S IBLN=$$SET(IBT,IBD,IBLN,IBLR)  . I $P(IBID0,U,1)'=$P(IBDM,U,4) S IBD=$P(IBDM,U,4) S IBLN=$$SET(IBT,IBD,IBLN,IBLR) S IBT=""  I $P(IBDM,U,6)'="" S IBT="",IBD=$P(IBDM,U,6) S IBLN=$$SET(IBT,IBD,IBLN,IBLR)  I $P(IBDM1,U,1)'="" S IBT="",IBD=$P(IBDM1,U,1) S IBLN=$$SET(IBT,IBD,IBLN,IBLR)  S IBT="",IBD=$P(IBDM,U,7),IBD=IBD\_$S(IBD'="":", ",1:"")\_$P($G(^DIC(5,+$P(IBDM,U,8),0)),U,2)\_" "\_$P(IBDM,U,9),IBLN=$$SET(IBT,IBD,IBLN,IBLR)  S IBT="Claim Phone: ",IBD=$P($$BADD^IBJTU3(+IBIFN),U,2) S IBLN=$$SET(IBT,IBD,IBLN,IBLR)  S IBLN=$$SET("","",IBLN,5)  ;  ; MRD;IB\*2.0\*516 - Use an IBLR of 7 for this section, then reset below.  S IBLR=7  S IBT="Subscriber Demographics" S IBLN=$$SETN(IBT,IBLN,1,1)  S IBT="Group Number: ",IBD=$P(IBDI1,U,3) S IBLN=$$SET(IBT,IBD,IBLN,IBLR)  S IBT="Group Name: ",IBD=$P(IBDI1,U,15) S IBLN=$$SET(IBT,IBD,IBLN,IBLR)  S IBT="Subscriber ID: ",IBD=$P(IBDI1,U,2) S IBLN=$$SET(IBT,IBD,IBLN,IBLR)  S IBT="Employer: ",IBD=$$EMPL(+DFN) S IBLN=$$SET(IBT,IBD,IBLN,IBLR)  S IBT="Insured's Name: ",IBD=$P(IBDI1,U,17) S IBLN=$$SET(IBT,IBD,IBLN,IBLR)  S IBT="Relationship: ",IBD=$$EXSET^IBJU1($P(IBDI1,U,16),2.312,16) S IBLN=$$SET(IBT,IBD,IBLN,IBLR)  S IBLR=1  ;  S (IBNC(1),IBTC(1))=2,(IBNC(2),IBTC(2))=42,IBNC(3)=29,IBTW(1)=12,IBTW(2)=16,IBSW(1)=26,IBSW(2)=22  S (IBT,IBD)="" S IBLN=$$SET(IBT,IBD,IBLN,IBLR)  ;  I $$FT^IBCEF(IBIFN)=2 D  . N IBXDATA,IBXSAVE K ^TMP("IBXSAVE",$J)  . D F^IBCEF("N-HCFA 1500 BOX 19",,,IBIFN)  . I IBXDATA'="" S IBBX19(1)=$E(IBXDATA,1,40) S:$E(IBXDATA,41,$L(IBXDATA))'="" IBBX19(2)=$E(IBXDATA,41,$L(IBXDATA))  ;  S IBGRPB=IBLN,IBLR=1  S IBT="Claim Information" S IBLN=$$SETN(IBT,IBLN,3,1)  S IBT="Bill Type: ",IBD=$$EXSET^IBJU1($P(IBD0,U,5),399,.05) S IBLN=$$SET(IBT,IBD,IBLN,IBLR)  S IBT="Time Frame: ",IBD=$$EXSET^IBJU1($P(IBD0,U,6),399,.06) S IBLN=$$SET(IBT,IBD,IBLN,IBLR)  S IBT="Rate Type: ",IBD=$P($G(^DGCR(399.3,+$P(IBD0,U,7),0)),U,1) S IBLN=$$SET(IBT,IBD,IBLN,IBLR)  S IBT="AR Status: ",IBD=$P($$ARSTATA^IBJTU4(IBIFN),U,1) S IBLN=$$SET(IBT,IBD,IBLN,IBLR)  S IBT=" Sequence: ",IBD=$P($$EXSET^IBJU1($P(IBD0,U,21),399,.21)," ",1) S IBLN=$$SET(IBT,IBD,IBLN,IBLR)  S IBT="Purch Svc: ",IBD=$S($P(IBDU2,U,11)="":"NO",1:$$EXPAND^IBTRE(399,233,$P(IBDU2,U,11))),IBLN=$$SET(IBT,IBD,IBLN,4)  I $P(IBDM1,"^",8) S IBT=" ECME No: ",IBD=$P($P(IBDM1,"^",8),";",1),IBLN=$$SET(IBT,IBD,IBLN,IBLR)  I $L($P(IBDM1,"^",9)) S IBT="ECME Ap No: ",IBD=$P(IBDM1,"^",9),IBLN=$$SET(IBT,IBD,IBLN,IBLR)  I IBNABP'="" S IBT=$S(($L($TR(IBNABP," ",""))=7):" NCPDP No: ",1:" NPI: "),IBD=IBNABP,IBLN=$$SET(IBT,IBD,IBLN,IBLR)  ; IB\*2.0\*521 add Claim HPID to display  S IBD=$S($P(IBD0,U,21)="P":$P(IBDM1,U,13),$P(IBD0,U,21)="S":$P(IBDM1,U,14),$P(IBD0,U,21)="T":$P(IBDM1,U,15),1:"")  S:IBD="" IBD=$$HPD^IBCNHUT1(+IBDI1) S IBVL=$$HOD^IBCNHUT1(IBD,+IBDI1,IBD) S IBT=$P(IBVL,U,2)\_": ",IBLN=$$SET(IBT,IBD,IBLN,IBLR)  I IBWNR S IBT="MRA Status: ",IBD=$S($P(IBDTX,U,5):$P(IBDTX,U,5),1:"NOT RECEIVED"),IBLN=$$SET(IBT,$S(IBD:$$EXPAND^IBTRE(399,24,IBD),1:IBD),IBLN,IBLR)  I $G(IBBX19(1))'="" D  . S IBT=" Box 19: ",IBD=IBBX19(1),IBLN=$$SET(IBT,IBD,IBLN,IBLR)  . I $G(IBBX19(2))'="" S IBT=$J("",11),IBD=IBBX19(2),IBLN=$$SET(IBT,IBD,IBLN,IBLR)  ;  S IBLR=6,IBPRVO=""  S IBT="Providers: ",IBD="NONE"  ;IB\*2.0\*432/TAZ - Changed how providers are displayed to take line-level providers into account.  ;D F^IBCEF("N-ALL PROVIDERS","IBZ",,IBIFN)  D F^IBCEF("N-ALL PROVIDERS 1","IBZ",,IBIFN)  S IBZ0=0  S IBLVL=0  ;F S Z=$O(IBZ(Z)) Q:'Z D  ;. I $G(IBZ(Z)),$G(IBZ(Z,1))'="" S IBLN=$$SET(IBT,"(OLD PROV DATA) "\_IBZ(Z,1),IBLN,IBLR),IBZ0=1 Q  ;. I $P($G(IBZ(Z,1)),U)'="" S IBD=$E($$EXPAND^IBTRE(399.0222,.01,Z)\_":"\_$J("",15),1,15)\_$P(IBZ(Z,1),U)\_$S($P(IBZ(Z,1),U,4)'="":" ("\_$P(IBZ(Z,1),U,4)\_")",1:"") S IBLN=$$SET(IBT,IBD,IBLN,IBLR) S IBT=$J("",11),IBZ0=1  ;I 'IBZ0 S IBLN=$$SET(IBT,IBD,IBLN,IBLR)  S IBLVL=0  F  S IBLVL=$O(IBZ(IBLVL)) Q:'IBLVL  D  . S IBT=IBT\_$S(IBLVL=1:"Claim: ",1:"Line: ")  . S IBPRVTYP="",IBCNT=0  . F  S IBCNT=$O(IBZ(IBLVL,IBCNT)) Q:'IBCNT  D  .. I IBLVL=1 S IBD=$J("",5)  .. I IBLVL=2 S IBD=$E("("\_IBCNT\_")"\_$J("",5),1,5)  .. F  S IBPRVTYP=$O(IBZ(IBLVL,IBCNT,IBPRVTYP)) Q:'IBPRVTYP  D  ... S IBD=IBD\_$E($$EXPAND^IBTRE(399.0222,.01,IBPRVTYP)\_":"\_$J("",15),1,15)  ... S IBD=IBD\_$P(IBZ(IBLVL,IBCNT,IBPRVTYP),U)  ... I $L($P(IBZ(IBLVL,IBCNT,IBPRVTYP),U,4)) S IBD=IBD\_" ("\_$P(IBZ(IBLVL,IBCNT,IBPRVTYP),U,4)\_")"  ... S IBLN=$$SET(IBT,IBD,IBLN,IBLR),IBT="",IBD=$J("",5)  ;  S IBGRPE=IBLN,IBLN=IBGRPB+1,IBLR=2  ;  S IBT="Charge Type: ",IBD=$$EXSET^IBJU1($P(IBD0,U,27),399,.27) S IBLN=$$SET(IBT,IBD,IBLN,IBLR)  S IBT="Service Dates: ",IBD=$$DATE^IBJU1($P(IBDU,U,1))\_" - "\_$$DATE^IBJU1($P(IBDU,U,2)) S IBLN=$$SET(IBT,IBD,IBLN,IBLR)  S IBT="Orig Claim: ",IBD=$$BILL^RCJIBFN2(+IBIFN) S IBLN=$$SET(IBT,$J($P(IBD,U,1),9,2),IBLN,IBLR)  S IBT="Balance Due: ",IBD=$J($P(IBD,U,3),9,2) S IBLN=$$SET(IBT,IBD,IBLN,IBLR)  I +$P(IBDM,U,2) S IBX=$S($P(IBD0,U,21)="P":2,1:1) D  S IBLN=$$SET(IBT,IBD,IBLN,IBLR)  . S IBT=$S(IBX=2:"Secondary",1:"Primary")\_": ",IBD=$P($G(^DIC(36,+$P(IBDM,U,IBX),0)),U,1)  . S IBX=$P(IBDU2,U,(IBX+3)) I +IBX S IBX="("\_$J(IBX,0,2)\_")" S IBD=$E(IBD,1,(IBSW(IBLR)-$L(IBX)-2))\_" "\_IBX  I +$P(IBDM,U,3) S IBX=$S($P(IBD0,U,21)="T":2,1:3) D  S IBLN=$$SET(IBT,IBD,IBLN,IBLR)  . S IBT=$S(IBX=2:"Secondary",1:"Tertiary")\_": ",IBD=$P($G(^DIC(36,+$P(IBDM,U,IBX),0)),U,1)  . S IBX=$P(IBDU2,U,(IBX+3)) I +IBX S IBX="("\_$J(IBX,0,2)\_")" S IBD=$E(IBD,1,(IBSW(IBLR)-$L(IBX)-2))\_" "\_IBX  S IBLN=$$SET("","",IBLN,5)  I IBWNR S IBT="MRA Rec Date: " D  S IBLN=$$SET(IBT,IBD,IBLN,2)  . N Z  . ; find last MRA for receipt date  . S (IBD,Z)="" F  S Z=$O(^IBM(361.1,"B",IBIFN,Z),-1) Q:'Z  I $P($G(^IBM(361.1,Z,0)),U,4)=1 S IBD=$$DATE^IBJU1($P($P(^IBM(361.1,Z,0),U,6),".")) Q  F Z=IBLN:1:IBGRPE S IBLN=$$SET("","",IBLN,5)  ;  S (IBLN,VALMCNT)=$S(IBLN>IBGRPE:IBLN,1:IBGRPE)  ;  S IBGRPB=IBLN,IBLR=1  D CONT^IBJTCA2  ; COPAY I $O(^IBA(362.4,"C",IBIFN,0)) D  . S (IBT,IBD)="" S IBLN=$$SET(IBT,IBD,IBLN,IBLR) ; blank line  . S IBNC(1)=21,IBT="Related Prescription Copay Information" S IBLN=$$SETN(IBT,IBLN,1,1)  . N IBZ,IBX,IBC,IBCAP  . S IBZ=0 F  S IBZ=$O(^IBA(362.4,"C",IBIFN,IBZ)) Q:'IBZ  D  .. K ^TMP("IBTPJI",$J)  .. S IBC=$G(^IBA(362.4,IBZ,0))  .. D:$P(IBC,"^",5) RX^PSO52API($P(IBD0,"^",2),"IBTPJI",$P(IBC,"^",5),"","I^")  .. ; original fill  .. I $P(IBC,"^",10)=0 D  ... S IBX=+$G(^TMP($J,"IBTPJI",$P(IBD0,"^",2),+$P(IBC,"^",5),106)),IBCAP=+$G(^(106.6))  .. ; refills  .. E  D  ... S IBX=+$G(^TMP($J,"IBTPJI",$P(IBD0,"^",2),+$P(IBC,"^",5),"IB",+$P(IBC,"^",10),9)),IBCAP=+$G(^(9.1))  .. I '$G(IBX),$G(IBCAP) S IBT=" <copay exceeded cap>",IBLN=$$SET(IBT,"",IBLN,4) Q  .. I '$G(IBX) S IBT=" <none found>",IBLN=$$SET(IBT,"",IBLN,4) Q  .. S IBX=$G(^IB(IBX,0))  .. S IBT="Rx: "\_$P(IBC,"^")\_" Chg: $"\_$FN($P(IBX,"^",7),",",2)\_" Status: "\_$$TITLE^XLFSTR($$EXTERNAL^DILFD(350,.05,"",$P(IBX,"^",5)))\_" Bill: "\_$P(IBX,"^",11)  .. S IBLN=$$SET(IBT,"",IBLN,4)  K ^TMP("IBTPJI",$J)  ;  S (IBLN,VALMCNT)=IBLN-1  ; BLDQ Q  ; EMPL(DFN) ; returns employer name  Q $P($G(^DPT(+DFN,.311)),U,1)  ; SET(TTL,DATA,LN,LR) ;  N IBY  S IBY=$J(TTL,IBTW(LR))\_DATA D SET1(IBY,LN,IBTC(LR),(IBTW(LR)+IBSW(LR)))  S LN=LN+1  Q LN  ; SETN(TTL,LN,LR,RV) ;  N IBY  S IBY=" "\_TTL\_" " D SET1(IBY,LN,IBNC(LR),$L(IBY),$G(RV))  S LN=LN+1  Q LN  ; SET1(STR,LN,COL,WD,RV) ; set up TMP array with screen data  N IBX S IBX=$G(^TMP("IBJTCA",$J,LN,0))  S IBX=$$SETSTR^VALM1(STR,IBX,COL,WD)  D SET^VALM10(LN,IBX) I $G(RV)'="" D CNTRL^VALM10(LN,COL,WD,IORVON,IORVOFF)  Q | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBJTCA1 ;ALB/ARH - TPI CLAIMS INFO BUILD ;10/31/07 14:17  ;;2.0;INTEGRATED BILLING;\*\*39,80,106,137,223,276,363,384,432,452,473,497,521,516**,592**\*\*;21-MAR-94;Build 123  ;;Per VA Directive 6402, this routine should not be modified.  ; BLD ; build array for Third Party Joint Inquiry Claims Info screen, IBIFN must be defined  ;  N X,IBY,IBZ,IBZ0,IBI,IBT,IBD,IBLN,IBLR,IBD0,IBDI1,IBDM,IBDM1,IBDU,IBDS,IBDU2,IBID0,IBID13,IBNC,IBTC,IBTW,IBSW,IBGRPB,IBGRPE,IBWNR,IBDTX,IBBX19,IBPRVO,IBNABP,IBLVL,IBCNT,IBPRVTYP,IBVL  N IBXSAVE  ; IB\*2.0\*473 bi  S VALMCNT=0,X="",IBD0=$G(^DGCR(399,+$G(IBIFN),0)) I IBD0="" S VALMQUIT="" G BLDQ  F IBI="M","M1","U","S","U2","TX" S @("IBD"\_IBI)=$G(^DGCR(399,+IBIFN,IBI))  S IBDI1=$P(IBD0,U,21),IBDI1=$S(IBDI1="S":2,IBDI1="T":3,1:1) S IBDI1=$$POLICY^IBCEF(IBIFN,,IBDI1)  S IBID0=$G(^DIC(36,+IBDI1,0)),IBID13=$G(^DIC(36,+IBDI1,.13))  ;  S (IBLN,VALMCNT)=1  ;  ; MRD;IB\*2.0\*516 - Try to make the following more readable; also  ; added IBTC(7), IBTW(7) and IBSW(7).  ;IB\*2.0\*432/TAZ - Added IBTW(6) and IBSW(6)  ;S (IBNC(1),IBTC(1),IBTC(4),IBTC(6))=2,IBTC(5)=78,(IBNC(2),IBTC(2))=42,IBNC(3)=35,IBTW(1)=15,IBTW(2)=16,IBTW(4)=12,IBTW(5)=1,IBTW(6)=20,IBSW(1)=23,IBSW(2)=21,IBSW(4)=60,IBSW(5)=1,IBSW(6)=49  ;  S IBNC(1)=2,IBTC(1)=2,IBTW(1)=15,IBSW(1)=23  S IBNC(2)=42,IBTC(2)=42,IBTW(2)=16,IBSW(2)=21  S IBNC(3)=35  S IBTC(4)=2,IBTW(4)=12,IBSW(4)=60  S IBTC(5)=78,IBTW(5)=1,IBSW(5)=1  S IBTC(6)=2,IBTW(6)=20,IBSW(6)=49  S IBTC(7)=2,IBTW(7)=20,IBSW(7)=58  ;  S IBLR=1  ;  S IBT="Insurance Demographics" S IBLN=$$SETN(IBT,IBLN,IBLR,1)  S IBWNR=$$WNRBILL^IBEFUNC(IBIFN)  S IBNABP=$$NABP^IBNCPDPU(IBIFN)  S IBT=$S(IBWNR:" \*",1:" ")\_"Bill Payer: ",IBD=$P(IBID0,U,1) S IBLN=$$SET(IBT,IBD,IBLN,IBLR)  S IBT="Claim Address: " D  S IBD=$P(IBDM,U,5) S IBLN=$$SET(IBT,IBD,IBLN,IBLR)  . I $P(IBID0,U,1)'=$P(IBDM,U,4) S IBD=$P(IBDM,U,4) S IBLN=$$SET(IBT,IBD,IBLN,IBLR) S IBT=""  I $P(IBDM,U,6)'="" S IBT="",IBD=$P(IBDM,U,6) S IBLN=$$SET(IBT,IBD,IBLN,IBLR)  I $P(IBDM1,U,1)'="" S IBT="",IBD=$P(IBDM1,U,1) S IBLN=$$SET(IBT,IBD,IBLN,IBLR)  S IBT="",IBD=$P(IBDM,U,7),IBD=IBD\_$S(IBD'="":", ",1:"")\_$P($G(^DIC(5,+$P(IBDM,U,8),0)),U,2)\_" "\_$P(IBDM,U,9),IBLN=$$SET(IBT,IBD,IBLN,IBLR)  S IBT="Claim Phone: ",IBD=$P($$BADD^IBJTU3(+IBIFN),U,2) S IBLN=$$SET(IBT,IBD,IBLN,IBLR)  S IBLN=$$SET("","",IBLN,5)  ;  ; MRD;IB\*2.0\*516 - Use an IBLR of 7 for this section, then reset below.  S IBLR=7  S IBT="Subscriber Demographics" S IBLN=$$SETN(IBT,IBLN,1,1)  S IBT="Group Number: ",IBD=$P(IBDI1,U,3) S IBLN=$$SET(IBT,IBD,IBLN,IBLR)  S IBT="Group Name: ",IBD=$P(IBDI1,U,15) S IBLN=$$SET(IBT,IBD,IBLN,IBLR)  S IBT="Subscriber ID: ",IBD=$P(IBDI1,U,2) S IBLN=$$SET(IBT,IBD,IBLN,IBLR)  S IBT="Employer: ",IBD=$$EMPL(+DFN) S IBLN=$$SET(IBT,IBD,IBLN,IBLR)  S IBT="Insured's Name: ",IBD=$P(IBDI1,U,17) S IBLN=$$SET(IBT,IBD,IBLN,IBLR)  S IBT="Relationship: ",IBD=$$EXSET^IBJU1($P(IBDI1,U,16),2.312,16) S IBLN=$$SET(IBT,IBD,IBLN,IBLR)  S IBLR=1  ;  S (IBNC(1),IBTC(1))=2,(IBNC(2),IBTC(2))=42,IBNC(3)=29,IBTW(1)=12,IBTW(2)=16,IBSW(1)=26,IBSW(2)=22  S (IBT,IBD)="" S IBLN=$$SET(IBT,IBD,IBLN,IBLR)  **;JWS:IB\*2.0\*592:US131 - added dental claim #7**  I $$FT^IBCEF(IBIFN)=2**!($$FT^IBCEF(IBIFN)=7)** D  . N IBXDATA,IBXSAVE K ^TMP("IBXSAVE",$J)  . D F^IBCEF("N-HCFA 1500 BOX 19",,,IBIFN)  . I IBXDATA'="" S IBBX19(1)=$E(IBXDATA,1,40) S:$E(IBXDATA,41,$L(IBXDATA))'="" IBBX19(2)=$E(IBXDATA,41,$L(IBXDATA))  ;  S IBGRPB=IBLN,IBLR=1  S IBT="Claim Information" S IBLN=$$SETN(IBT,IBLN,3,1)  S IBT="Bill Type: ",IBD=$$EXSET^IBJU1($P(IBD0,U,5),399,.05) S IBLN=$$SET(IBT,IBD,IBLN,IBLR)  S IBT="Time Frame: ",IBD=$$EXSET^IBJU1($P(IBD0,U,6),399,.06) S IBLN=$$SET(IBT,IBD,IBLN,IBLR)  S IBT="Rate Type: ",IBD=$P($G(^DGCR(399.3,+$P(IBD0,U,7),0)),U,1) S IBLN=$$SET(IBT,IBD,IBLN,IBLR)  S IBT="AR Status: ",IBD=$P($$ARSTATA^IBJTU4(IBIFN),U,1) S IBLN=$$SET(IBT,IBD,IBLN,IBLR)  S IBT=" Sequence: ",IBD=$P($$EXSET^IBJU1($P(IBD0,U,21),399,.21)," ",1) S IBLN=$$SET(IBT,IBD,IBLN,IBLR)  S IBT="Purch Svc: ",IBD=$S($P(IBDU2,U,11)="":"NO",1:$$EXPAND^IBTRE(399,233,$P(IBDU2,U,11))),IBLN=$$SET(IBT,IBD,IBLN,4)  I $P(IBDM1,"^",8) S IBT=" ECME No: ",IBD=$P($P(IBDM1,"^",8),";",1),IBLN=$$SET(IBT,IBD,IBLN,IBLR)  I $L($P(IBDM1,"^",9)) S IBT="ECME Ap No: ",IBD=$P(IBDM1,"^",9),IBLN=$$SET(IBT,IBD,IBLN,IBLR)  I IBNABP'="" S IBT=$S(($L($TR(IBNABP," ",""))=7):" NCPDP No: ",1:" NPI: "),IBD=IBNABP,IBLN=$$SET(IBT,IBD,IBLN,IBLR)  ; IB\*2.0\*521 add Claim HPID to display  S IBD=$S($P(IBD0,U,21)="P":$P(IBDM1,U,13),$P(IBD0,U,21)="S":$P(IBDM1,U,14),$P(IBD0,U,21)="T":$P(IBDM1,U,15),1:"")  S:IBD="" IBD=$$HPD^IBCNHUT1(+IBDI1) S IBVL=$$HOD^IBCNHUT1(IBD,+IBDI1,IBD) S IBT=$P(IBVL,U,2)\_": ",IBLN=$$SET(IBT,IBD,IBLN,IBLR)  I IBWNR S IBT="MRA Status: ",IBD=$S($P(IBDTX,U,5):$P(IBDTX,U,5),1:"NOT RECEIVED"),IBLN=$$SET(IBT,$S(IBD:$$EXPAND^IBTRE(399,24,IBD),1:IBD),IBLN,IBLR)  I $G(IBBX19(1))'="" D  . S IBT=" Box 19: ",IBD=IBBX19(1),IBLN=$$SET(IBT,IBD,IBLN,IBLR)  . I $G(IBBX19(2))'="" S IBT=$J("",11),IBD=IBBX19(2),IBLN=$$SET(IBT,IBD,IBLN,IBLR)  ;  S IBLR=6,IBPRVO=""  S IBT="Providers: ",IBD="NONE"  ;IB\*2.0\*432/TAZ - Changed how providers are displayed to take line-level providers into account.  ;D F^IBCEF("N-ALL PROVIDERS","IBZ",,IBIFN)  D F^IBCEF("N-ALL PROVIDERS 1","IBZ",,IBIFN)  S IBZ0=0  S IBLVL=0  ;F S Z=$O(IBZ(Z)) Q:'Z D  ;. I $G(IBZ(Z)),$G(IBZ(Z,1))'="" S IBLN=$$SET(IBT,"(OLD PROV DATA) "\_IBZ(Z,1),IBLN,IBLR),IBZ0=1 Q  ;. I $P($G(IBZ(Z,1)),U)'="" S IBD=$E($$EXPAND^IBTRE(399.0222,.01,Z)\_":"\_$J("",15),1,15)\_$P(IBZ(Z,1),U)\_$S($P(IBZ(Z,1),U,4)'="":" ("\_$P(IBZ(Z,1),U,4)\_")",1:"") S IBLN=$$SET(IBT,IBD,IBLN,IBLR) S IBT=$J("",11),IBZ0=1  ;I 'IBZ0 S IBLN=$$SET(IBT,IBD,IBLN,IBLR)  S IBLVL=0  F  S IBLVL=$O(IBZ(IBLVL)) Q:'IBLVL  D  . S IBT=IBT\_$S(IBLVL=1:"Claim: ",1:"Line: ")  . S IBPRVTYP="",IBCNT=0  . F  S IBCNT=$O(IBZ(IBLVL,IBCNT)) Q:'IBCNT  D  .. I IBLVL=1 S IBD=$J("",5)  .. I IBLVL=2 S IBD=$E("("\_IBCNT\_")"\_$J("",5),1,5)  .. F  S IBPRVTYP=$O(IBZ(IBLVL,IBCNT,IBPRVTYP)) Q:'IBPRVTYP  D  ... S IBD=IBD\_$E($$EXPAND^IBTRE(399.0222,.01,IBPRVTYP)\_":"\_$J("",15),1,15)  ... S IBD=IBD\_$P(IBZ(IBLVL,IBCNT,IBPRVTYP),U)  ... I $L($P(IBZ(IBLVL,IBCNT,IBPRVTYP),U,4)) S IBD=IBD\_" ("\_$P(IBZ(IBLVL,IBCNT,IBPRVTYP),U,4)\_")"  ... S IBLN=$$SET(IBT,IBD,IBLN,IBLR),IBT="",IBD=$J("",5)  ;  S IBGRPE=IBLN,IBLN=IBGRPB+1,IBLR=2  ;  S IBT="Charge Type: ",IBD=$$EXSET^IBJU1($P(IBD0,U,27),399,.27) S IBLN=$$SET(IBT,IBD,IBLN,IBLR)  S IBT="Service Dates: ",IBD=$$DATE^IBJU1($P(IBDU,U,1))\_" - "\_$$DATE^IBJU1($P(IBDU,U,2)) S IBLN=$$SET(IBT,IBD,IBLN,IBLR)  S IBT="Orig Claim: ",IBD=$$BILL^RCJIBFN2(+IBIFN) S IBLN=$$SET(IBT,$J($P(IBD,U,1),9,2),IBLN,IBLR)  S IBT="Balance Due: ",IBD=$J($P(IBD,U,3),9,2) S IBLN=$$SET(IBT,IBD,IBLN,IBLR)  I +$P(IBDM,U,2) S IBX=$S($P(IBD0,U,21)="P":2,1:1) D  S IBLN=$$SET(IBT,IBD,IBLN,IBLR)  . S IBT=$S(IBX=2:"Secondary",1:"Primary")\_": ",IBD=$P($G(^DIC(36,+$P(IBDM,U,IBX),0)),U,1)  . S IBX=$P(IBDU2,U,(IBX+3)) I +IBX S IBX="("\_$J(IBX,0,2)\_")" S IBD=$E(IBD,1,(IBSW(IBLR)-$L(IBX)-2))\_" "\_IBX  I +$P(IBDM,U,3) S IBX=$S($P(IBD0,U,21)="T":2,1:3) D  S IBLN=$$SET(IBT,IBD,IBLN,IBLR)  . S IBT=$S(IBX=2:"Secondary",1:"Tertiary")\_": ",IBD=$P($G(^DIC(36,+$P(IBDM,U,IBX),0)),U,1)  . S IBX=$P(IBDU2,U,(IBX+3)) I +IBX S IBX="("\_$J(IBX,0,2)\_")" S IBD=$E(IBD,1,(IBSW(IBLR)-$L(IBX)-2))\_" "\_IBX  S IBLN=$$SET("","",IBLN,5)  I IBWNR S IBT="MRA Rec Date: " D  S IBLN=$$SET(IBT,IBD,IBLN,2)  . N Z  . ; find last MRA for receipt date  . S (IBD,Z)="" F  S Z=$O(^IBM(361.1,"B",IBIFN,Z),-1) Q:'Z  I $P($G(^IBM(361.1,Z,0)),U,4)=1 S IBD=$$DATE^IBJU1($P($P(^IBM(361.1,Z,0),U,6),".")) Q  F Z=IBLN:1:IBGRPE S IBLN=$$SET("","",IBLN,5)  ;  S (IBLN,VALMCNT)=$S(IBLN>IBGRPE:IBLN,1:IBGRPE)  ;  S IBGRPB=IBLN,IBLR=1  D CONT^IBJTCA2  ; COPAY I $O(^IBA(362.4,"C",IBIFN,0)) D  . S (IBT,IBD)="" S IBLN=$$SET(IBT,IBD,IBLN,IBLR) ; blank line  . S IBNC(1)=21,IBT="Related Prescription Copay Information" S IBLN=$$SETN(IBT,IBLN,1,1)  . N IBZ,IBX,IBC,IBCAP  . S IBZ=0 F  S IBZ=$O(^IBA(362.4,"C",IBIFN,IBZ)) Q:'IBZ  D  .. K ^TMP("IBTPJI",$J)  .. S IBC=$G(^IBA(362.4,IBZ,0))  .. D:$P(IBC,"^",5) RX^PSO52API($P(IBD0,"^",2),"IBTPJI",$P(IBC,"^",5),"","I^")  .. ; original fill  .. I $P(IBC,"^",10)=0 D  ... S IBX=+$G(^TMP($J,"IBTPJI",$P(IBD0,"^",2),+$P(IBC,"^",5),106)),IBCAP=+$G(^(106.6))  .. ; refills  .. E  D  ... S IBX=+$G(^TMP($J,"IBTPJI",$P(IBD0,"^",2),+$P(IBC,"^",5),"IB",+$P(IBC,"^",10),9)),IBCAP=+$G(^(9.1))  .. I '$G(IBX),$G(IBCAP) S IBT=" <copay exceeded cap>",IBLN=$$SET(IBT,"",IBLN,4) Q  .. I '$G(IBX) S IBT=" <none found>",IBLN=$$SET(IBT,"",IBLN,4) Q  .. S IBX=$G(^IB(IBX,0))  .. S IBT="Rx: "\_$P(IBC,"^")\_" Chg: $"\_$FN($P(IBX,"^",7),",",2)\_" Status: "\_$$TITLE^XLFSTR($$EXTERNAL^DILFD(350,.05,"",$P(IBX,"^",5)))\_" Bill: "\_$P(IBX,"^",11)  .. S IBLN=$$SET(IBT,"",IBLN,4)  K ^TMP("IBTPJI",$J)  ;  S (IBLN,VALMCNT)=IBLN-1  ; BLDQ Q  ; EMPL(DFN) ; returns employer name  Q $P($G(^DPT(+DFN,.311)),U,1)  ; SET(TTL,DATA,LN,LR) ;  N IBY  S IBY=$J(TTL,IBTW(LR))\_DATA D SET1(IBY,LN,IBTC(LR),(IBTW(LR)+IBSW(LR)))  S LN=LN+1  Q LN  ; SETN(TTL,LN,LR,RV) ;  N IBY  S IBY=" "\_TTL\_" " D SET1(IBY,LN,IBNC(LR),$L(IBY),$G(RV))  S LN=LN+1  Q LN  ; SET1(STR,LN,COL,WD,RV) ; set up TMP array with screen data  N IBX S IBX=$G(^TMP("IBJTCA",$J,LN,0))  S IBX=$$SETSTR^VALM1(STR,IBX,COL,WD)  D SET^VALM10(LN,IBX) I $G(RV)'="" D CNTRL^VALM10(LN,COL,WD,IORVON,IORVOFF)  Q | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCNSC0 | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** |  | | | | | | | | |
| **Related Options** | None | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  |  | | | | |  | | | |
| **Data Dictionary (DD) References** |  | | | | | | | | |
| **Related Protocols** | None | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Current Logic** | | | | | | | | | |
| IBCNSC0 ;ALB/NLR - INSURANCE COMPANY EDIT - ;12-MAR-1993  ;;2.0;INTEGRATED BILLING;\*\*371,547\*\*;21-MAR-94;Build 119  ;;Per VA Directive 6402, this routine should not be modified.  ; CLAIMS1 ; display Inpatient Claims information  N OFFSET,START,IBCNS12,IBADD  ;WCJ;IB\*2.0\*547  ;S START=27,OFFSET=2  S START=28+(2\*$G(IBACMAX)),OFFSET=2  D SET^IBCNSP(START,OFFSET+20," Inpatient Claims Office Information ",IORVON,IORVOFF)  ;  ;WCJ;IB\*2.0\*547;Call New API  ;S IBCNS12=$$ADDRESS(IBCNS,.12,5)  S IBCNS12=$$ADD2(IBCNS,.12,5)  ;  D SET^IBCNSP(START+1,OFFSET," Company Name: "\_$P($G(^DIC(36,+$P(IBCNS12,"^",7),0)),"^",1))  D SET^IBCNSP(START+2,OFFSET," Street: "\_$P(IBCNS12,"^",1))  D SET^IBCNSP(START+3,OFFSET," Street 2: "\_$P(IBCNS12,"^",2))  N OFFSET S OFFSET=45  D SET^IBCNSP(START+1,OFFSET," Street 3: "\_$P(IBCNS12,"^",3)) S IBADD=1  D SET^IBCNSP(START+1+IBADD,OFFSET," City/State: "\_$E($P(IBCNS12,"^",4),1,15)\_$S($P(IBCNS12,"^",4)="":"",1:", ")\_$P($G(^DIC(5,+$P(IBCNS12,"^",5),0)),"^",2)\_" "\_$E($P(IBCNS12,"^",6),1,5))  D SET^IBCNSP(START+2+IBADD,OFFSET," Phone: "\_$P(IBCNS12,"^",8))  D SET^IBCNSP(START+3+IBADD,OFFSET," Fax: "\_$P(IBCNS12,"^",9))  Q  ; R1Q Q CLAIMS2 ; display Outpatient Claims information  ;  N OFFSET,START,IBCNS16,IBADD  ;WCJ;IB\*2.0\*547  ;S START=34,OFFSET=2  S START=35+(2\*$G(IBACMAX)),OFFSET=2  D SET^IBCNSP(START,OFFSET+20," Outpatient Claims Office Information ",IORVON,IORVOFF)  ;  ;WCJ;IB\*2.0\*547;Call New API  ;S IBCNS16=$$ADDRESS(IBCNS,.16,6)  S IBCNS16=$$ADD2(IBCNS,.16,6)  ;  D SET^IBCNSP(START+1,OFFSET," Company Name: "\_$P($G(^DIC(36,+$P(IBCNS16,"^",7),0)),"^",1))  D SET^IBCNSP(START+2,OFFSET," Street: "\_$P(IBCNS16,"^",1))  D SET^IBCNSP(START+3,OFFSET," Street 2: "\_$P(IBCNS16,"^",2))  N OFFSET S OFFSET=45  D SET^IBCNSP(START+1,OFFSET," Street 3: "\_$P(IBCNS16,"^",3)) S IBADD=1  D SET^IBCNSP(START+1+IBADD,OFFSET," City/State: "\_$E($P(IBCNS16,"^",4),1,15)\_$S($P(IBCNS16,"^",4)="":"",1:", ")\_$P($G(^DIC(5,+$P(IBCNS16,"^",5),0)),"^",2)\_" "\_$E($P(IBCNS16,"^",6),1,5))  D SET^IBCNSP(START+2+IBADD,OFFSET," Phone: "\_$P(IBCNS16,"^",8))  D SET^IBCNSP(START+3+IBADD,OFFSET," Fax: "\_$P(IBCNS16,"^",9))  Q  ;  ; Only adding comments on patch 547. Changes are on the ADD2 tag below.  ; This tag is called from the Output formatter.  ; It returns a "complete" address  ; It judges an address complete if it has a state (don't ask why, I am just adding the comments)  ; If the address it wants is not complete, it returns the main address.  ; These addresses go out on claims and claims (X12 837) don't like partial addresses. ADDRESS(INS,NODE,PH) ; -- generic find address  ;  N IBX,INSSAVE,IBPH,IBFX,IBCNT,IBA  S IBX="" ;S IBPH="",IBFX="",IBA=""  ; REDO ; gather insurance carrier's main address information   S IBX=$G(^DIC(36,+INS,.11)),IBPH=$P($G(^DIC(36,+INS,.13)),"^",1),IBFX=$P(IBX,"^",9)  ;S IBCNT=$G(IBCNT)+1  ;  ; -- if process the same co. more than once you are in an infinite loop  ;I $D(IBCNT(IBCNS)) G ADDREQ  ;S IBCNT(IBCNS)=""  ;  ; -- gather address information from specific office (Claims, Appeals, Inquiry)  ;  I $P($G(^DIC(36,+INS,+NODE)),"^",5) S IBX=$G(^DIC(36,+INS,+NODE)),IBPH=$P($G(^DIC(36,+INS,.13)),"^",PH),IBFX=$P($G(IBX),"^",9)  I $P($G(^DIC(36,+INS,+NODE)),"^",7) S INSSAVE=INS,INS=$P($G(^DIC(36,+INS,+NODE)),"^",7) I INSSAVE'=INS G REDO  ; ADDRESQ ; concatenate company name, address, phone and fax   S $P(IBA,"^",1,6)=$P($G(IBX),"^",1,6)  S $P(IBA,"^",7)=INS  S $P(IBA,"^",8)=IBPH  S $P(IBA,"^",9)=IBFX ADDREQ Q IBA  ;  ; WCJ;IB\*2.0\*547;  ; This is a new tag which is just called from the insurance company editor screens.  ; The billers/insurance verifiers want to see what data is actually in the insurance company file.  ; They don't care if it's complete. Heck, a phone number may be enough.  ; This will just return what is in the file for the ins company that handles that type of claims.  ; Input: INS - IREN to file 36  ; NODE - Node in File 36 (corresponds to Claims, Appeals, Inquiry...)  ; PH - Location of Phone number in node .13 ADD2(INS,NODE,PH) ;  N IBX,INSSAVE,IBFX,IBPH,IBA  F  S IBX=$G(^DIC(36,+INS,+NODE)) Q:'$P(IBX,U,7) S INSSAVE=INS,INS=$P(IBX,U,7) Q:INSSAVE=INS  ; concatenate company name, address, phone and fax   S IBPH=$P($G(^DIC(36,+INS,.13)),U,PH),IBFX=$P(IBX,U,9)  S $P(IBA,U,1,6)=$P(IBX,U,1,6),$P(IBA,U,7)=INS,$P(IBA,U,8)=IBPH,$P(IBA,U,9)=IBFX  Q IBA | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCNSC0 ;ALB/NLR - INSURANCE COMPANY EDIT - ;12-MAR-1993  ;;2.0;INTEGRATED BILLING;\*\*371,547,**592**\*\*;21-MAR-94;Build 119  ;;Per VA Directive 6402, this routine should not be modified.  ; CLAIMS1 ; display Inpatient Claims information  N OFFSET,START,IBCNS12,IBADD  ;WCJ;IB\*2.0\*547  ;S START=27,OFFSET=2  S START=28+(2\*$G(IBACMAX)),OFFSET=2  D SET^IBCNSP(START,OFFSET+20," Inpatient Claims Office Information ",IORVON,IORVOFF)  ;  ;WCJ;IB\*2.0\*547;Call New API  ;S IBCNS12=$$ADDRESS(IBCNS,.12,5)  S IBCNS12=$$ADD2(IBCNS,.12,5)  ;  D SET^IBCNSP(START+1,OFFSET," Company Name: "\_$P($G(^DIC(36,+$P(IBCNS12,"^",7),0)),"^",1))  D SET^IBCNSP(START+2,OFFSET," Street: "\_$P(IBCNS12,"^",1))  D SET^IBCNSP(START+3,OFFSET," Street 2: "\_$P(IBCNS12,"^",2))  N OFFSET S OFFSET=45  D SET^IBCNSP(START+1,OFFSET," Street 3: "\_$P(IBCNS12,"^",3)) S IBADD=1  D SET^IBCNSP(START+1+IBADD,OFFSET," City/State: "\_$E($P(IBCNS12,"^",4),1,15)\_$S($P(IBCNS12,"^",4)="":"",1:", ")\_$P($G(^DIC(5,+$P(IBCNS12,"^",5),0)),"^",2)\_" "\_$E($P(IBCNS12,"^",6),1,5))  D SET^IBCNSP(START+2+IBADD,OFFSET," Phone: "\_$P(IBCNS12,"^",8))  D SET^IBCNSP(START+3+IBADD,OFFSET," Fax: "\_$P(IBCNS12,"^",9))  Q  ; R1Q Q CLAIMS2 ; display Outpatient Claims information  ;  N OFFSET,START,IBCNS16,IBADD  ;WCJ;IB\*2.0\*547  ;S START=34,OFFSET=2  S START=35+(2\*$G(IBACMAX)),OFFSET=2  D SET^IBCNSP(START,OFFSET+20," Outpatient Claims Office Information ",IORVON,IORVOFF)  ;  ;WCJ;IB\*2.0\*547;Call New API  ;S IBCNS16=$$ADDRESS(IBCNS,.16,6)  S IBCNS16=$$ADD2(IBCNS,.16,6)  ;  D SET^IBCNSP(START+1,OFFSET," Company Name: "\_$P($G(^DIC(36,+$P(IBCNS16,"^",7),0)),"^",1))  D SET^IBCNSP(START+2,OFFSET," Street: "\_$P(IBCNS16,"^",1))  D SET^IBCNSP(START+3,OFFSET," Street 2: "\_$P(IBCNS16,"^",2))  N OFFSET S OFFSET=45  D SET^IBCNSP(START+1,OFFSET," Street 3: "\_$P(IBCNS16,"^",3)) S IBADD=1  D SET^IBCNSP(START+1+IBADD,OFFSET," City/State: "\_$E($P(IBCNS16,"^",4),1,15)\_$S($P(IBCNS16,"^",4)="":"",1:", ")\_$P($G(^DIC(5,+$P(IBCNS16,"^",5),0)),"^",2)\_" "\_$E($P(IBCNS16,"^",6),1,5))  D SET^IBCNSP(START+2+IBADD,OFFSET," Phone: "\_$P(IBCNS16,"^",8))  D SET^IBCNSP(START+3+IBADD,OFFSET," Fax: "\_$P(IBCNS16,"^",9))  Q  ;  ; Only adding comments on patch 547. Changes are on the ADD2 tag below.  ; This tag is called from the Output formatter.  ; It returns a "complete" address  ; It judges an address complete if it has a state (don't ask why, I am just adding the comments)  ; If the address it wants is not complete, it returns the main address.  ; These addresses go out on claims and claims (X12 837) don't like partial addresses. ADDRESS(INS,NODE,PH) ; -- generic find address  ;  N IBX,INSSAVE,IBPH,IBFX,IBCNT,IBA  S IBX="" ;S IBPH="",IBFX="",IBA=""  ; REDO ; gather insurance carrier's main address information   S IBX=$G(^DIC(36,+INS,.11)),IBPH=$P($G(^DIC(36,+INS,.13)),"^",1),IBFX=$P(IBX,"^",9)  ;S IBCNT=$G(IBCNT)+1  ;  ; -- if process the same co. more than once you are in an infinite loop  ;I $D(IBCNT(IBCNS)) G ADDREQ  ;S IBCNT(IBCNS)=""  ;  ; -- gather address information from specific office (Claims, Appeals, Inquiry**, Dental)** **;JWS;IB\*2.0\*592;Changed below for DENTAL insurance mailing address  I $P($G(^DIC(36,+INS,+NODE)),"^",5) D  . S IBX=$G(^DIC(36,+INS,+NODE))  . I +NODE=.19 S IBPH=$P(IBX,"^",PH)  . E  S IBPH=$P($G(^DIC(36,+INS,.13)),"^",PH)  . S IBFX=$P($G(IBX),"^",9)  I $P($G(^DIC(36,+INS,+NODE)),"^",7) S INSSAVE=INS,INS=$P($G(^DIC(36,+INS,+NODE)),"^",7) I INSSAVE'=INS G REDO**  ; ADDRESQ ; concatenate company name, address, phone and fax   S $P(IBA,"^",1,6)=$P($G(IBX),"^",1,6)  S $P(IBA,"^",7)=INS  S $P(IBA,"^",8)=IBPH  S $P(IBA,"^",9)=IBFX ADDREQ Q IBA  ;  ; WCJ;IB\*2.0\*547;  ; This is a new tag which is just called from the insurance company editor screens.  ; The billers/insurance verifiers want to see what data is actually in the insurance company file.  ; They don't care if it's complete. Heck, a phone number may be enough.  ; This will just return what is in the file for the ins company that handles that type of claims.  ; Input: INS - IREN to file 36  ; NODE - Node in File 36 (corresponds to Claims, Appeals, Inquiry...)  ; PH - Location of Phone number in node .13 ADD2(INS,NODE,PH) ;  N IBX,INSSAVE,IBFX,IBPH,IBA  F  S IBX=$G(^DIC(36,+INS,+NODE)) Q:'$P(IBX,U,7) S INSSAVE=INS,INS=$P(IBX,U,7) Q:INSSAVE=INS  ; concatenate company name, address, phone and fax   S IBPH=$P($G(^DIC(36,+INS,.13)),U,PH),IBFX=$P(IBX,U,9)  **;JWS;IB\*2.0\*592;Dental mailing address  I +NODE=.19 S IBPH=$P($G(^DIC(36,+INS,.19)),U,11)**  S $P(IBA,U,1,6)=$P(IBX,U,1,6),$P(IBA,U,7)=INS,$P(IBA,U,8)=IBPH,$P(IBA,U,9)=IBFX  Q IBA | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCNSC1 | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** |  | | | | | | | | |
| **Related Options** | None | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  |  | | | | |  | | | |
| **Data Dictionary (DD) References** |  | | | | | | | | |
| **Related Protocols** | None | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Current Logic** | | | | | | | | | |
| IBCNSC1 ;ALB/NLR - IBCNS INSURANCE COMPANY ;23-MAR-93  ;;2.0;INTEGRATED BILLING;\*\*62,137,232,291,320,348,349,371,400,519,516,547\*\*;21-MAR-94;Build 119  ;;Per VA Directive 6402, this routine should not be modified.  ; % G EN^IBCNSC  ; AI ; -- (In)Activate Company  D FULL^VALM1 W !!  I '$D(^XUSEC("IB INSURANCE SUPERVISOR",DUZ)) D SORRY G EXIT  D ^IBCNSC2  G EXIT CC ; -- Change Insurance Company  D FULL^VALM1 W !!  S IBCNS1=IBCNS K IBCNS D INSCO^IBCNSC  I '$D(IBCNS) S IBCNS=IBCNS1  K IBCNS1,VALMQUIT  G EXIT EA ; -- Billing,Claims,Appeals,Inquiry,Telephone,Main,Remarks,Synonyms  D FULL^VALM1  ;  ; IB\*2\*320 - check key for associate company action  I $G(IBY)=",13,",'$$KCHK^XUSRB("IB EDI INSURANCE EDIT") D  G EXIT  . W !!?5,"You must hold the IB EDI INSURANCE EDIT key to access this option."  . D PAUSE^VALM1  . Q  ;  W !!  D MAIN  ;  ; -- was company deleted  I '$D(^DIC(36,IBCNS)) W !!,"<DELETED>",!! S VALMQUIT="" Q  ; EXIT ;  D HDR^IBCNSC,BLD^IBCNSC  S VALMBCK="R"  Q MAIN ; -- Call edit template  N IBEDIKEY,Z  L +^DIC(36,+IBCNS):5 I '$T D LOCKED^IBTRCD1 G MAINQ  I $G(IBY)=",12," D FACID  F Z=1,2,4,9,13,14 S IBEDIKEY(Z)=$P($G(^DIC(36,+IBCNS,3)),U,Z) ; save EDI data fields  F Z=1:1:8 S IBEDIKEY(Z,6)=$P($G(^DIC(36,+IBCNS,6)),U,Z) ; save EDI data fields  I $G(IBY)'=",12," N DIE,DA,DR S DIE="^DIC(36,",(DA,Y)=IBCNS,DR="[IBEDIT INS CO1]" D ^DIE K DIE S:$D(Y) IB("^")=1 D:$TR($P($G(^DIC(36,IBCNS,6)),U,1,8),U)]"" CUIDS(IBCNS)  I $G(IBY)=",12," D EDITID^IBCEP(+IBCNS)  I $F(",6,1,",$G(IBY)) D CLEANIDS^IBCNSC(+IBCNS) ;clean up any errant nodes on alternate payert IDS  I $F(",6,13,",$G(IBY)) D PARENT^IBCNSC02(+IBCNS) ; parent/child management  L -^DIC(36,+IBCNS)  ; IB\*2.0\*519: If field 3.02 or 3.04 has changed, trigger HL7 to update the NIF  I (IBEDIKEY(2)'=$P($G(^DIC(36,+IBCNS,3)),U,2))!(IBEDIKEY(4)'=$P($G(^DIC(36,+IBCNS,3)),U,4)) D EXR^IBCNHUT1(IBCNS),SEND^IBCNHHLO(IBCNS) MAINQ Q  ; FACID ; -- Edit facility ids  D FACID^IBCEP2B(+IBCNS,"E")  Q  ; SORRY ; -- can't inactivate, don't have key  W !!,"You do not have access to Inactivate entries. See your application coordinator.",! D PAUSE^VALM1  Q PRESCR ;  N OFFSET,START,IBCNS18,IBADD  ;  ;WCJ;IB\*2.0\*547;Call New API  ;S IBCNS18=$$ADDRESS^IBCNSC0(IBCNS,.18,11)  S IBCNS18=$$ADD2^IBCNSC0(IBCNS,.18,11)  ;  ;WCJ;IB\*2.0\*547  ;S START=41,OFFSET=2  S START=42+(2\*$G(IBACMAX)),OFFSET=2  D SET^IBCNSP(START,OFFSET+19," Prescription Claims Office Information ",IORVON,IORVOFF)  D SET^IBCNSP(START+1,OFFSET," Company Name: "\_$P($G(^DIC(36,+$P(IBCNS18,"^",7),0)),"^",1))  D SET^IBCNSP(START+2,OFFSET," Street: "\_$P(IBCNS18,"^",1))  D SET^IBCNSP(START+3,OFFSET," Street 2: "\_$P(IBCNS18,"^",2))  ; D SET^IBCNSP(START+4,OFFSET,"Claim Off. ID: "\_$P(IBCNS18,"^",11))  N OFFSET S OFFSET=45  D SET^IBCNSP(START+1,OFFSET," Street 3: "\_$P(IBCNS18,"^",3)) S IBADD=1  D SET^IBCNSP(START+1+IBADD,OFFSET," City/State: "\_$E($P(IBCNS18,"^",4),1,15)\_$S($P(IBCNS18,"^",4)="":"",1:", ")\_$P($G(^DIC(5,+$P(IBCNS18,"^",5),0)),"^",2)\_" "\_$E($P(IBCNS18,"^",6),1,5))  D SET^IBCNSP(START+2+IBADD,OFFSET," Phone: "\_$P(IBCNS18,"^",8))  D SET^IBCNSP(START+3+IBADD,OFFSET," Fax: "\_$P(IBCNS18,"^",9))  Q  ; PROVID N OFFSET,START,IBCNS4,IBCNS3,IBDISP,Z,LINE  S START=$O(^TMP("IBCNSC",$J,""),-1)+1  S (IB1ST("PROVID"),LINE)=START  S OFFSET=2,IBCNS4=$G(^DIC(36,IBCNS,4)),IBCNS3=$G(^(3))  ;   D SET^IBCNSP(LINE,OFFSET+25,"Provider IDs",IORVON,IORVOFF)  N OFFSET  S LINE=LINE+1,OFFSET=1  D SET^IBCNSP(LINE,OFFSET,"Billing Provider Secondary ID")  ;  N Z,Z0,Z1,IBS,I,DIV,FT,CU,CUF,DIVISION,FORMTYPE,PIDT  S Z=0 F  S Z=$O(^IBA(355.92,"B",+IBCNS,Z)) Q:'Z  D  . S Z0=$G(^IBA(355.92,Z,0))  . Q:'$P(Z0,U,6)!($P(Z0,U,7)="") ; Quit if no provider id or id type  . Q:'($P(Z0,U,8)="E")  . S IBS(+$P(Z0,U,5),+$P(Z0,U,3),+$P(Z0,U,4))=$P(Z0,U,6)\_U\_$P(Z0,U,7)  ;  S DIV="" F  S DIV=$O(IBS(DIV)) Q:DIV=""  D  . S DIVISION=$$DIV^IBCEP7(DIV)  . S CU="",CUF=0 F  S CU=$O(IBS(DIV,CU)) Q:CU=""  D  .. S FT="" F  S FT=$O(IBS(DIV,CU,FT)) Q:FT=""  D  ... S FORMTYPE=$S(FT=1:"UB-04",FT=2:"1500",1:"UNKNOWN")  ... S LINE=LINE+1  ... I 'CUF,+CU S CUF=1 S TEXT=$P(DIVISION,"/")\_" Care Units :",OFFSET=5 D SET^IBCNSP(LINE,OFFSET,TEXT) S LINE=LINE+1  ... I CU=0 S TEXT=DIVISION\_"/"\_FORMTYPE\_": "\_$$GET1^DIQ(355.97,$P(IBS(DIV,CU,FT),U),.03,"E")\_" "\_$P(IBS(DIV,CU,FT),U,2),OFFSET=2  ... I +CU S TEXT=$$EXPAND^IBTRE(355.92,.03,CU)\_"/"\_FORMTYPE\_": "\_$$GET1^DIQ(355.97,$P(IBS(DIV,CU,FT),U),.03,"E")\_" "\_$P(IBS(DIV,CU,FT),U,2),OFFSET=5  ... D SET^IBCNSP(LINE,OFFSET,TEXT)  ;  S LINE=LINE+1 D SET^IBCNSP(LINE,2," ")  ;  K IBS  S OFFSET=1,LINE=LINE+1  D SET^IBCNSP(LINE,OFFSET,"Additional Billing Provider Secondary IDs")  S Z=0 F  S Z=$O(^IBA(355.92,"B",+IBCNS,Z)) Q:'Z  D  . S Z0=$G(^IBA(355.92,Z,0))  . Q:'$P(Z0,U,6)!($P(Z0,U,7)="") ; Quit if no provider id or id type  . Q:'($P(Z0,U,8)="A")  . ; IBS(DIVISION,FORMTYPE,IDTYPE)=ID  . S IBS(+$P(Z0,U,5),+$P(Z0,U,4),+$P(Z0,U,6))=$P(Z0,U,7)  ;  S DIVISION=$$DIV^IBCEP7(0)  S DIV="" F  S DIV=$O(IBS(DIV)) Q:DIV=""  D  . S FT="" F  S FT=$O(IBS(DIV,FT)) Q:FT=""  D  .. S FORMTYPE=$S(FT=1:"UB-04",FT=2:"1500",1:"UNKNOWN")  .. S TEXT=DIVISION\_"/"\_FORMTYPE\_": "  .. S LINE=LINE+1,OFFSET=2  .. D SET^IBCNSP(LINE,OFFSET,TEXT)  .. S PIDT="" F  S PIDT=$O(IBS(DIV,FT,PIDT)) Q:PIDT=""  D  ... S LINE=LINE+1  ... S TEXT=$$GET1^DIQ(355.97,PIDT,.03,"E")\_" "\_IBS(DIV,FT,PIDT),OFFSET=5  ... D SET^IBCNSP(LINE,OFFSET,TEXT)  ;  S LINE=LINE+1 D SET^IBCNSP(LINE,2," ")  ;  K IBS  S OFFSET=1,LINE=LINE+1  D SET^IBCNSP(LINE,OFFSET,"VA-Laboratory or Facility Secondary IDs")  S Z=0 F  S Z=$O(^IBA(355.92,"B",+IBCNS,Z)) Q:'Z  D  . S Z0=$G(^IBA(355.92,Z,0))  . Q:'$P(Z0,U,6)!($P(Z0,U,7)="") ; Quit if no provider id or id type  . Q:'($P(Z0,U,8)="LF")  . ; IBS(DIVISION,FORMTYPE,IDTYPE)=ID  . S IBS(+$P(Z0,U,5),+$P(Z0,U,4),+$P(Z0,U,6))=$P(Z0,U,7)  ;  S DIVISION=$$DIV^IBCEP7(0)  S DIV="" F  S DIV=$O(IBS(DIV)) Q:DIV=""  D  . S FT="" F  S FT=$O(IBS(DIV,FT)) Q:FT=""  D  .. S FORMTYPE=$S(FT=1:"UB-04",FT=2:"1500",1:"UNKNOWN")  .. S TEXT=DIVISION\_"/"\_FORMTYPE\_": "  .. S LINE=LINE+1,OFFSET=2  .. D SET^IBCNSP(LINE,OFFSET,TEXT)  .. S PIDT="" F  S PIDT=$O(IBS(DIV,FT,PIDT)) Q:PIDT=""  D  ... S LINE=LINE+1  ... ;S TEXT=$$EXPAND^IBTRE(355.92,.06,PIDT)\_" "\_IBS(DIV,FT,PIDT),OFFSET=5  ... S TEXT=$$GET1^DIQ(355.97,PIDT,.03,"E")\_" "\_IBS(DIV,FT,PIDT),OFFSET=5  ... D SET^IBCNSP(LINE,OFFSET,TEXT)  ;  ;  S LINE=LINE+1 D SET^IBCNSP(LINE,2," ")  S LINE=LINE+1 D SET^IBCNSP(LINE,2," ")  S OFFSET=2  S LINE=LINE+1 D SET^IBCNSP(LINE,OFFSET+25,"ID Parameters",IORVON,IORVOFF)  ;  S IBCNS4=$G(^DIC(36,IBCNS,4)),IBCNS3=$G(^(3)),OFFSET=1  S TEXT="Attending/Rendering Provider Secondary ID Qualifier (1500): "\_$$EXPAND^IBTRE(36,4.01,+$P(IBCNS4,U))  S LINE=LINE+1  D SET^IBCNSP(LINE,OFFSET,TEXT)  ;  S TEXT="Attending/Rendering Provider Secondary ID Qualifier (UB-04): "\_$$EXPAND^IBTRE(36,4.02,+$P(IBCNS4,U,2))  S LINE=LINE+1  D SET^IBCNSP(LINE,OFFSET,TEXT)  ;  S TEXT="Attending/Rendering Secondary ID Requirement: "\_$$EXPAND^IBTRE(36,4.03,+$P(IBCNS4,U,3))  S LINE=LINE+1  D SET^IBCNSP(LINE,OFFSET,TEXT)  ;  S TEXT="Referring Provider Secondary ID Qualifier (1500): "\_$$EXPAND^IBTRE(36,4.04,+$P(IBCNS4,U,4))  S LINE=LINE+1  D SET^IBCNSP(LINE,OFFSET,TEXT)  ;  S TEXT="Referring Provider Secondary ID Requirement: "\_$$EXPAND^IBTRE(36,4.05,+$P(IBCNS4,U,5))  S LINE=LINE+1  D SET^IBCNSP(LINE,OFFSET,TEXT)  ;  S TEXT="Use Att/Rend ID as Billing Provider Sec. ID (1500): "\_$$EXPAND^IBTRE(36,4.06,+$P(IBCNS4,U,6))  S LINE=LINE+1  D SET^IBCNSP(LINE,OFFSET,TEXT)  ;  S TEXT="Use Att/Rend ID as Billing Provider Sec. ID (UB-04): "\_$$EXPAND^IBTRE(36,4.08,+$P(IBCNS4,U,8))  S LINE=LINE+1  D SET^IBCNSP(LINE,OFFSET,TEXT)  ;  ; MRD;IB\*2.0\*516 - Marked fields 4.07, 4.11, 4.12 and 4.13 for  ; deletion and removed all references to them.  ;S TEXT="Always use main VAMC as Billing Provider (1500)?: "\_$$EXPAND^IBTRE(36,4.11,+$P(IBCNS4,U,11))  ;S LINE=LINE+1  ;D SET^IBCNSP(LINE,OFFSET,TEXT)  ;  ;S TEXT="Always use main VAMC as Billing Provider (UB-04)?: "\_$$EXPAND^IBTRE(36,4.12,+$P(IBCNS4,U,12))  ;S LINE=LINE+1  ;D SET^IBCNSP(LINE,OFFSET,TEXT)  ;  ;I $P(IBCNS4,U,11)!($P(IBCNS4,U,12)) D  ;.S TEXT="Send VA Lab/Facility IDs or Facility Data for VAMC?: "\_$$EXPAND^IBTRE(36,4.07,+$P(IBCNS4,U,7))  ;.S LINE=LINE+1  ;.D SET^IBCNSP(LINE,OFFSET,TEXT)  ;.;  ;.S TEXT="Use the Billing Provider (VAMC) Name and Street Address?: "\_$$EXPAND^IBTRE(36,4.13,+$P(IBCNS4,U,13))  ;.S LINE=LINE+1  ;.D SET^IBCNSP(LINE,OFFSET,TEXT)  ;.Q  ;  S TEXT="Transmit no Billing Provider Sec. ID for the Electronic Plan Types: "  S LINE=LINE+1  D SET^IBCNSP(LINE,OFFSET,TEXT)  ;  N TAR,ERR,IBCT  D LIST^DIC(36.013,","\_IBCNS\_",",".01",,10,,,,,,"TAR","ERR")  F IBCT=1:1:+$G(TAR("DILIST",0)) D  . S TEXT=TAR("DILIST",1,IBCT)  . S LINE=LINE+1  . D SET^IBCNSP(LINE,OFFSET,TEXT)  ;  S LINE=LINE+1 D SET^IBCNSP(LINE,2," ")  S LINE=LINE+1 D SET^IBCNSP(LINE,2," ")  Q  ;  INSDEF(IBINS,IBPTYP) ; Returns the default id # for an ins co, if possible  N X  S X=""  I IBINS,IBPTYP S X=$P($G(^IBA(355.91,+$O(^IBA(355.91,"AC",IBINS,IBPTYP,"\*N/A\*","")),0)),U,7)  Q X  ; CUIDS(IBCNS) ;  N DIE,DA,DR,PIECE,DAT6,Y  S DAT6=$P(^DIC(36,IBCNS,6),U,1,8) ; get the Payer IDs  ;  ; Make sure each qualifier has an ID and vice versa  F PIECE=1,3,5,7 D  . I $TR($P(DAT6,U,PIECE,PIECE+1),U)="" Q  ; both blank  . I $P(DAT6,U,PIECE)]"",$P(DAT6,U,PIECE+1)]"" Q  ; both have data  . S DIE="^DIC(36,",(DA,Y)=IBCNS,DR="6.0"\_$S($P(DAT6,U,PIECE)]"":PIECE,1:PIECE+1)\_"////@"  . D ^DIE K DIE  ;  S DAT6=$P($G(^DIC(36,IBCNS,6)),U,1,8) ; get the Payer IDs again since they may have changed above.  ;  ; Make sure the first pair of ID/Qual are populated if the 2nd pair is. If not, move em over.  ; This is done for institutional then professional  F PIECE=1,5 D  . I $P(DAT6,U,PIECE)]"" Q  ; already has set one  . I $P(DAT6,U,PIECE+2)="" Q  ; has no second set  . S DIE="^DIC(36,",(DA,Y)=IBCNS  . ; deleting the qualifier triggers deletion of the ID  . S DR="6.0"\_PIECE\_"////"\_$P(DAT6,U,PIECE+2)\_";6.0"\_(PIECE+1)\_"////"\_$P(DAT6,U,PIECE+3)\_";6.0"\_(PIECE+2)\_"////@"  . D ^DIE K DIE  Q | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCNSC1 ;ALB/NLR - IBCNS INSURANCE COMPANY ;23-MAR-93  ;;2.0;INTEGRATED BILLING;\*\*62,137,232,291,320,348,349,371,400,519,516,547,**592**\*\*;21-MAR-94;Build 119  ;;Per VA Directive 6402, this routine should not be modified.  ; % G EN^IBCNSC  ; AI ; -- (In)Activate Company  D FULL^VALM1 W !!  I '$D(^XUSEC("IB INSURANCE SUPERVISOR",DUZ)) D SORRY G EXIT  D ^IBCNSC2  G EXIT CC ; -- Change Insurance Company  D FULL^VALM1 W !!  S IBCNS1=IBCNS K IBCNS D INSCO^IBCNSC  I '$D(IBCNS) S IBCNS=IBCNS1  K IBCNS1,VALMQUIT  G EXIT EA ; -- Billing,Claims,Appeals,Inquiry,Telephone,Main,Remarks,Synonyms  D FULL^VALM1  ;  ; IB\*2\*320 - check key for associate company action  I $G(IBY)=",13,",'$$KCHK^XUSRB("IB EDI INSURANCE EDIT") D  G EXIT  . W !!?5,"You must hold the IB EDI INSURANCE EDIT key to access this option."  . D PAUSE^VALM1  . Q  ;  W !!  D MAIN  ;  ; -- was company deleted  I '$D(^DIC(36,IBCNS)) W !!,"<DELETED>",!! S VALMQUIT="" Q  ; EXIT ;  D HDR^IBCNSC,BLD^IBCNSC  S VALMBCK="R"  Q MAIN ; -- Call edit template  N IBEDIKEY,Z  L +^DIC(36,+IBCNS):5 I '$T D LOCKED^IBTRCD1 G MAINQ  I $G(IBY)=",12," D FACID  **;JWS;IB\*2.0\*592;add field .15 (piece 15) Dental EDI Payer ID**  F Z=1,2,4,9,13,14,**15** S IBEDIKEY(Z)=$P($G(^DIC(36,+IBCNS,3)),U,Z) ; save EDI data fields  F Z=1:1:8 S IBEDIKEY(Z,6)=$P($G(^DIC(36,+IBCNS,6)),U,Z) ; save EDI data fields  I $G(IBY)'=",12," N DIE,DA,DR S DIE="^DIC(36,",(DA,Y)=IBCNS,DR="[IBEDIT INS CO1]" D ^DIE K DIE S:$D(Y) IB("^")=1 D:$TR($P($G(^DIC(36,IBCNS,6)),U,1,8),U)]"" CUIDS(IBCNS)  I $G(IBY)=",12," D EDITID^IBCEP(+IBCNS)  I $F(",6,1,",$G(IBY)) D CLEANIDS^IBCNSC(+IBCNS) ;clean up any errant nodes on alternate payert IDS  I $F(",6,13,",$G(IBY)) D PARENT^IBCNSC02(+IBCNS) ; parent/child management  L -^DIC(36,+IBCNS)  ; IB\*2.0\*519: If field 3.02 or 3.04 has changed, trigger HL7 to update the NIF  I (IBEDIKEY(2)'=$P($G(^DIC(36,+IBCNS,3)),U,2))!(IBEDIKEY(4)'=$P($G(^DIC(36,+IBCNS,3)),U,4)) D EXR^IBCNHUT1(IBCNS),SEND^IBCNHHLO(IBCNS) MAINQ Q  ; FACID ; -- Edit facility ids  D FACID^IBCEP2B(+IBCNS,"E")  Q  ; SORRY ; -- can't inactivate, don't have key  W !!,"You do not have access to Inactivate entries. See your application coordinator.",! D PAUSE^VALM1  Q PRESCR ;  N OFFSET,START,IBCNS18,IBADD  ;  ;WCJ;IB\*2.0\*547;Call New API  ;S IBCNS18=$$ADDRESS^IBCNSC0(IBCNS,.18,11)  S IBCNS18=$$ADD2^IBCNSC0(IBCNS,.18,11)  ;  ;WCJ;IB\*2.0\*547  ;S START=41,OFFSET=2  S START=42+(2\*$G(IBACMAX)),OFFSET=2  D SET^IBCNSP(START,OFFSET+19," Prescription Claims Office Information ",IORVON,IORVOFF)  D SET^IBCNSP(START+1,OFFSET," Company Name: "\_$P($G(^DIC(36,+$P(IBCNS18,"^",7),0)),"^",1))  D SET^IBCNSP(START+2,OFFSET," Street: "\_$P(IBCNS18,"^",1))  D SET^IBCNSP(START+3,OFFSET," Street 2: "\_$P(IBCNS18,"^",2))  ; D SET^IBCNSP(START+4,OFFSET,"Claim Off. ID: "\_$P(IBCNS18,"^",11))  N OFFSET S OFFSET=45  D SET^IBCNSP(START+1,OFFSET," Street 3: "\_$P(IBCNS18,"^",3)) S IBADD=1  D SET^IBCNSP(START+1+IBADD,OFFSET," City/State: "\_$E($P(IBCNS18,"^",4),1,15)\_$S($P(IBCNS18,"^",4)="":"",1:", ")\_$P($G(^DIC(5,+$P(IBCNS18,"^",5),0)),"^",2)\_" "\_$E($P(IBCNS18,"^",6),1,5))  D SET^IBCNSP(START+2+IBADD,OFFSET," Phone: "\_$P(IBCNS18,"^",8))  D SET^IBCNSP(START+3+IBADD,OFFSET," Fax: "\_$P(IBCNS18,"^",9))  Q  ; PROVID N OFFSET,START,IBCNS4,IBCNS3,IBDISP,Z,LINE  S START=$O(^TMP("IBCNSC",$J,""),-1)+1  S (IB1ST("PROVID"),LINE)=START  S OFFSET=2,IBCNS4=$G(^DIC(36,IBCNS,4)),IBCNS3=$G(^(3))  ;   D SET^IBCNSP(LINE,OFFSET+25,"Provider IDs",IORVON,IORVOFF)  N OFFSET  S LINE=LINE+1,OFFSET=1  D SET^IBCNSP(LINE,OFFSET,"Billing Provider Secondary ID")  ;  N Z,Z0,Z1,IBS,I,DIV,FT,CU,CUF,DIVISION,FORMTYPE,PIDT  S Z=0 F  S Z=$O(^IBA(355.92,"B",+IBCNS,Z)) Q:'Z  D  . S Z0=$G(^IBA(355.92,Z,0))  . Q:'$P(Z0,U,6)!($P(Z0,U,7)="") ; Quit if no provider id or id type  . Q:'($P(Z0,U,8)="E")  . S IBS(+$P(Z0,U,5),+$P(Z0,U,3),+$P(Z0,U,4))=$P(Z0,U,6)\_U\_$P(Z0,U,7)  ;  S DIV="" F  S DIV=$O(IBS(DIV)) Q:DIV=""  D  . S DIVISION=$$DIV^IBCEP7(DIV)  . S CU="",CUF=0 F  S CU=$O(IBS(DIV,CU)) Q:CU=""  D  .. S FT="" F  S FT=$O(IBS(DIV,CU,FT)) Q:FT=""  D  ... **;JWS;IB\*2.0\*592 - Dental form 7 (J430D)**  ... S FORMTYPE=$S(FT=1:"UB-04",FT=2:"1500",**FT=4:"J430D**",1:"UNKNOWN")  ... S LINE=LINE+1  ... I 'CUF,+CU S CUF=1 S TEXT=$P(DIVISION,"/")\_" Care Units :",OFFSET=5 D SET^IBCNSP(LINE,OFFSET,TEXT) S LINE=LINE+1  ... I CU=0 S TEXT=DIVISION\_"/"\_FORMTYPE\_": "\_$$GET1^DIQ(355.97,$P(IBS(DIV,CU,FT),U),.03,"E")\_" "\_$P(IBS(DIV,CU,FT),U,2),OFFSET=2  ... I +CU S TEXT=$$EXPAND^IBTRE(355.92,.03,CU)\_"/"\_FORMTYPE\_": "\_$$GET1^DIQ(355.97,$P(IBS(DIV,CU,FT),U),.03,"E")\_" "\_$P(IBS(DIV,CU,FT),U,2),OFFSET=5  ... D SET^IBCNSP(LINE,OFFSET,TEXT)  ;  S LINE=LINE+1 D SET^IBCNSP(LINE,2," ")  ;  K IBS  S OFFSET=1,LINE=LINE+1  D SET^IBCNSP(LINE,OFFSET,"Additional Billing Provider Secondary IDs")  S Z=0 F  S Z=$O(^IBA(355.92,"B",+IBCNS,Z)) Q:'Z  D  . S Z0=$G(^IBA(355.92,Z,0))  . Q:'$P(Z0,U,6)!($P(Z0,U,7)="") ; Quit if no provider id or id type  . Q:'($P(Z0,U,8)="A")  . ; IBS(DIVISION,FORMTYPE,IDTYPE)=ID  . S IBS(+$P(Z0,U,5),+$P(Z0,U,4),+$P(Z0,U,6))=$P(Z0,U,7)  ;  S DIVISION=$$DIV^IBCEP7(0)  S DIV="" F  S DIV=$O(IBS(DIV)) Q:DIV=""  D  . S FT="" F  S FT=$O(IBS(DIV,FT)) Q:FT=""  D  .. **;JWS;IB\*2.0\*592 - Dental form 7 (J430D)**  .. S FORMTYPE=$S(FT=1:"UB-04",FT=2:"1500",**FT=4:"J430D",**1:"UNKNOWN")  .. S TEXT=DIVISION\_"/"\_FORMTYPE\_": "  .. S LINE=LINE+1,OFFSET=2  .. D SET^IBCNSP(LINE,OFFSET,TEXT)  .. S PIDT="" F  S PIDT=$O(IBS(DIV,FT,PIDT)) Q:PIDT=""  D  ... S LINE=LINE+1  ... S TEXT=$$GET1^DIQ(355.97,PIDT,.03,"E")\_" "\_IBS(DIV,FT,PIDT),OFFSET=5  ... D SET^IBCNSP(LINE,OFFSET,TEXT)  ;  S LINE=LINE+1 D SET^IBCNSP(LINE,2," ")  ;  K IBS  S OFFSET=1,LINE=LINE+1  D SET^IBCNSP(LINE,OFFSET,"VA-Laboratory or Facility Secondary IDs")  S Z=0 F  S Z=$O(^IBA(355.92,"B",+IBCNS,Z)) Q:'Z  D  . S Z0=$G(^IBA(355.92,Z,0))  . Q:'$P(Z0,U,6)!($P(Z0,U,7)="") ; Quit if no provider id or id type  . Q:'($P(Z0,U,8)="LF")  . ; IBS(DIVISION,FORMTYPE,IDTYPE)=ID  . S IBS(+$P(Z0,U,5),+$P(Z0,U,4),+$P(Z0,U,6))=$P(Z0,U,7)  ;  S DIVISION=$$DIV^IBCEP7(0)  S DIV="" F  S DIV=$O(IBS(DIV)) Q:DIV=""  D  . S FT="" F  S FT=$O(IBS(DIV,FT)) Q:FT=""  D  **.. ;JWS;IB\*2.0\*592 - Dental form 7 (J430D)**  .. S FORMTYPE=$S(FT=1:"UB-04",FT=2:"1500",**FT=4:"J430D",**1:"UNKNOWN")  .. S TEXT=DIVISION\_"/"\_FORMTYPE\_": "  .. S LINE=LINE+1,OFFSET=2  .. D SET^IBCNSP(LINE,OFFSET,TEXT)  .. S PIDT="" F  S PIDT=$O(IBS(DIV,FT,PIDT)) Q:PIDT=""  D  ... S LINE=LINE+1  ... ;S TEXT=$$EXPAND^IBTRE(355.92,.06,PIDT)\_" "\_IBS(DIV,FT,PIDT),OFFSET=5  ... S TEXT=$$GET1^DIQ(355.97,PIDT,.03,"E")\_" "\_IBS(DIV,FT,PIDT),OFFSET=5  ... D SET^IBCNSP(LINE,OFFSET,TEXT)  ;  S LINE=LINE+1 D SET^IBCNSP(LINE,2," ")  S LINE=LINE+1 D SET^IBCNSP(LINE,2," ")  S OFFSET=2  S LINE=LINE+1 D SET^IBCNSP(LINE,OFFSET+25,"ID Parameters",IORVON,IORVOFF)  ;  S IBCNS4=$G(^DIC(36,IBCNS,4)),IBCNS3=$G(^(3)),OFFSET=1  S TEXT="Attending/Rendering Provider Secondary ID Qualifier (1500): "\_$$EXPAND^IBTRE(36,4.01,+$P(IBCNS4,U))  S LINE=LINE+1  D SET^IBCNSP(LINE,OFFSET,TEXT)  ;  **;JWS;IB\*2.8\*592;add J430D displayed values**  **S TEXT="Attending/Rendering Provider Secondary ID Qualifier (J430D): "\_$$EXPAND^IBTRE(36,4.14,+$P(IBCNS4,U,14))  S LINE=LINE+1  D SET^IBCNSP(LINE,OFFSET,TEXT)**  ;  S TEXT="Attending/Rendering Provider Secondary ID Qualifier (UB-04): "\_$$EXPAND^IBTRE(36,4.02,+$P(IBCNS4,U,2))  S LINE=LINE+1  D SET^IBCNSP(LINE,OFFSET,TEXT)  ;  S TEXT="Attending/Rendering Secondary ID Requirement: "\_$$EXPAND^IBTRE(36,4.03,+$P(IBCNS4,U,3))  S LINE=LINE+1  D SET^IBCNSP(LINE,OFFSET,TEXT)  ;  S TEXT="Referring Provider Secondary ID Qualifier (1500): "\_$$EXPAND^IBTRE(36,4.04,+$P(IBCNS4,U,4))  S LINE=LINE+1  D SET^IBCNSP(LINE,OFFSET,TEXT)  ; **;JWS;IB\*2.0\*592;add J430D form information  S TEST="Use Att/Rend ID as Billing Provider Sec. ID (J430D): "\_$$EXPAND^IBTRE(36,4.15,+$P(IBCNS4,U,15))  S LINE=LINE+1  D SET^IBCNSP(LINE,OFFSET,TEXT)**  ;  S TEXT="Referring Provider Secondary ID Requirement: "\_$$EXPAND^IBTRE(36,4.05,+$P(IBCNS4,U,5))  S LINE=LINE+1  D SET^IBCNSP(LINE,OFFSET,TEXT)  ;  S TEXT="Use Att/Rend ID as Billing Provider Sec. ID (1500): "\_$$EXPAND^IBTRE(36,4.06,+$P(IBCNS4,U,6))  S LINE=LINE+1  D SET^IBCNSP(LINE,OFFSET,TEXT)  ;  **;JWS;IB\*2.0\*592; added J430D values  S TEXT="Use Att/Rend ID as Billing Provider Sec. ID (J430D): "\_$$EXPAND^IBTRE(36,4.16,+$P(IBCNS4,U,16))  S LINE=LINE+1  D SET^IBCNSP(LINE,OFFSET,TEXT)**  ;  S TEXT="Use Att/Rend ID as Billing Provider Sec. ID (UB-04): "\_$$EXPAND^IBTRE(36,4.08,+$P(IBCNS4,U,8))  S LINE=LINE+1  D SET^IBCNSP(LINE,OFFSET,TEXT)  ;  ; MRD;IB\*2.0\*516 - Marked fields 4.07, 4.11, 4.12 and 4.13 for  ; deletion and removed all references to them.  ;S TEXT="Always use main VAMC as Billing Provider (1500)?: "\_$$EXPAND^IBTRE(36,4.11,+$P(IBCNS4,U,11))  ;S LINE=LINE+1  ;D SET^IBCNSP(LINE,OFFSET,TEXT)  ;  ;S TEXT="Always use main VAMC as Billing Provider (UB-04)?: "\_$$EXPAND^IBTRE(36,4.12,+$P(IBCNS4,U,12))  ;S LINE=LINE+1  ;D SET^IBCNSP(LINE,OFFSET,TEXT)  ;  ;I $P(IBCNS4,U,11)!($P(IBCNS4,U,12)) D  ;.S TEXT="Send VA Lab/Facility IDs or Facility Data for VAMC?: "\_$$EXPAND^IBTRE(36,4.07,+$P(IBCNS4,U,7))  ;.S LINE=LINE+1  ;.D SET^IBCNSP(LINE,OFFSET,TEXT)  ;.;  ;.S TEXT="Use the Billing Provider (VAMC) Name and Street Address?: "\_$$EXPAND^IBTRE(36,4.13,+$P(IBCNS4,U,13))  ;.S LINE=LINE+1  ;.D SET^IBCNSP(LINE,OFFSET,TEXT)  ;.Q  ;  S TEXT="Transmit no Billing Provider Sec. ID for the Electronic Plan Types: "  S LINE=LINE+1  D SET^IBCNSP(LINE,OFFSET,TEXT)  ;  N TAR,ERR,IBCT  D LIST^DIC(36.013,","\_IBCNS\_",",".01",,10,,,,,,"TAR","ERR")  F IBCT=1:1:+$G(TAR("DILIST",0)) D  . S TEXT=TAR("DILIST",1,IBCT)  . S LINE=LINE+1  . D SET^IBCNSP(LINE,OFFSET,TEXT)  ;  S LINE=LINE+1 D SET^IBCNSP(LINE,2," ")  S LINE=LINE+1 D SET^IBCNSP(LINE,2," ")  Q  ;  INSDEF(IBINS,IBPTYP) ; Returns the default id # for an ins co, if possible  N X  S X=""  I IBINS,IBPTYP S X=$P($G(^IBA(355.91,+$O(^IBA(355.91,"AC",IBINS,IBPTYP,"\*N/A\*","")),0)),U,7)  Q X  ; CUIDS(IBCNS) ;  N DIE,DA,DR,PIECE,DAT6,Y  S DAT6=$P(^DIC(36,IBCNS,6),U,1,8) ; get the Payer IDs  ;  ; Make sure each qualifier has an ID and vice versa  F PIECE=1,3,5,7 D  . I $TR($P(DAT6,U,PIECE,PIECE+1),U)="" Q  ; both blank  . I $P(DAT6,U,PIECE)]"",$P(DAT6,U,PIECE+1)]"" Q  ; both have data  . S DIE="^DIC(36,",(DA,Y)=IBCNS,DR="6.0"\_$S($P(DAT6,U,PIECE)]"":PIECE,1:PIECE+1)\_"////@"  . D ^DIE K DIE  ;  S DAT6=$P($G(^DIC(36,IBCNS,6)),U,1,8) ; get the Payer IDs again since they may have changed above.  ;  ; Make sure the first pair of ID/Qual are populated if the 2nd pair is. If not, move em over.  ; This is done for institutional then professional  F PIECE=1,5 D  . I $P(DAT6,U,PIECE)]"" Q  ; already has set one  . I $P(DAT6,U,PIECE+2)="" Q  ; has no second set  . S DIE="^DIC(36,",(DA,Y)=IBCNS  . ; deleting the qualifier triggers deletion of the ID  . S DR="6.0"\_PIECE\_"////"\_$P(DAT6,U,PIECE+2)\_";6.0"\_(PIECE+1)\_"////"\_$P(DAT6,U,PIECE+3)\_";6.0"\_(PIECE+2)\_"////@"  . D ^DIE K DIE  Q | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCEPB | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** |  | | | | | | | | |
| **Related Options** | None | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  |  | | | | |  | | | |
| **Data Dictionary (DD) References** |  | | | | | | | | |
| **Related Protocols** | None | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Current Logic** | | | | | | | | | |
| IBCEPB ;ALB/WCJ - Insurance company ID parameters ;22-DEC-2005  ;;2.0;INTEGRATED BILLING;\*\*320,348,349,400,516\*\*;21-MAR-94;Build 123  ;;Per VA Directive 6402, this routine should not be modified. EN ; -- main entry point for IBCE INSCO ID MAINT  D EN^VALM("IBCE INSCO ID MAINT")  Q  ; HDR ; -- header code  N PCF,PCDISP  I '$D(IBCNS) N IBCNS S IBCNS=IBINS  S PCF=$P($G(^DIC(36,+IBCNS,3)),U,13),PCDISP=$S(PCF="P":"(Parent)",1:"")  S VALMHDR(1)="Insurance Co: "\_$P($G(^DIC(36,+IBCNS,0)),U)\_PCDISP  Q  ; INIT ; Initialize  D CLEAN^VALM10  I '$D(IBCNS) N IBCNS S IBCNS=IBINS  N IBLCT  S IBLCT=0  ; Display the list  D SET1(.IBLCT,"Attending/Rendering Provider Secondary ID")  D SET1(.IBLCT,"Default ID (1500) : "\_$$GET1^DIQ(36,IBCNS,4.01))  D SET1(.IBLCT,"Default ID (UB-04): "\_$$GET1^DIQ(36,IBCNS,4.02))  D SET1(.IBLCT,"Require ID on Claim: "\_$$GET1^DIQ(36,IBCNS,4.03))  D SET1(.IBLCT," ")  D SET1(.IBLCT,"Referring Provider Secondary ID")  D SET1(.IBLCT,"Default ID (1500): "\_$$GET1^DIQ(36,IBCNS,4.04))  D SET1(.IBLCT,"Require ID on Claim: "\_$$GET1^DIQ(36,IBCNS,4.05))  D SET1(.IBLCT," ")  D SET1(.IBLCT,"Billing Provider Secondary IDs")  D SET1(.IBLCT,"Use Att/Rend ID as Billing Provider Sec. ID (1500)? : "\_$$GET1^DIQ(36,IBCNS,4.06))  D SET1(.IBLCT,"Use Att/Rend ID as Billing Provider Sec. ID (UB-04)?: "\_$$GET1^DIQ(36,IBCNS,4.08))  D SET1(.IBLCT,"Transmit no Billing Provider Sec ID for the following Electronic Plan Types:")  D LIST^DIC(36.013,","\_IBCNS\_",",".01",,10,,,,,,"TAR","ERR")  F I=1:1:+$G(TAR("DILIST",0)) D  . D SET1(.IBLCT,TAR("DILIST",1,I))  D SET1(.IBLCT," ")  D SET1(.IBLCT,"Billing Provider/Service Facility")  ;  S IBCNS4=$G(^DIC(36,+IBCNS,4))  ; MRD;IB\*2.0\*516 - Marked fields 4.07, 4.11, 4.12 and 4.13 for  ; deletion and removed all references to them.  ;D SET1(.IBLCT,"Always use main VAMC as Billing Provider (1500)?: "\_$$EXPAND^IBTRE(36,4.11,+$P(IBCNS4,U,11)))  ;D SET1(.IBLCT,"Always use main VAMC as Billing Provider (UB-04)?: "\_$$EXPAND^IBTRE(36,4.12,+$P(IBCNS4,U,12)))  ;I $P(IBCNS4,U,11)!($P(IBCNS4,U,12)) D  ;.D SET1(.IBLCT,"Send VA Lab/Facility IDs or Facility Data for VAMC?: "\_$$EXPAND^IBTRE(36,4.07,+$P(IBCNS4,U,7)))  ;.D SET1(.IBLCT,"Use the Billing Prov (VAMC) Name and Street Address?: "\_$$EXPAND^IBTRE(36,4.13,+$P(IBCNS4,U,13)))  ;.Q  ;  S VALMBG=1,VALMCNT=IBLCT  Q  ; SET1(IBLCT,TEXT,IBCT) ;  S IBLCT=IBLCT+1 D SET^VALM10(IBLCT,TEXT)  Q  ; EXPND ;  Q HELP ;  Q EXIT ;  D CLEAN^VALM10  Q  ; IDPARAM ;  D FULL^VALM1  N DIE,DA,DR  I '$D(IBCNS) N IBCNS S IBCNS=IBINS  S DIE="^DIC(36,",(DA,Y)=IBCNS,DR="[IBEDIT INS CO1]"  I '$D(IBY) N IBY S IBY=",12,"  D ^DIE K DIE  K ^TMP("IBCE\_PRVFAC\_MAINT",$J)  D INIT  S VALMBCK="R"  Q  ; BILLPRVP ;  D FULL^VALM1  D EN^IBCEPC  D INIT  K ^TMP("IBCE\_PRVFAC\_MAINT",$J)  S VALMBCK="R"  Q | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCEPB ;ALB/WCJ - Insurance company ID parameters ;22-DEC-2005  ;;2.0;INTEGRATED BILLING;\*\*320,348,349,400,516,**592**\*\*;21-MAR-94;Build 123  ;;Per VA Directive 6402, this routine should not be modified. EN ; -- main entry point for IBCE INSCO ID MAINT  D EN^VALM("IBCE INSCO ID MAINT")  Q  ; HDR ; -- header code  N PCF,PCDISP  I '$D(IBCNS) N IBCNS S IBCNS=IBINS  S PCF=$P($G(^DIC(36,+IBCNS,3)),U,13),PCDISP=$S(PCF="P":"(Parent)",1:"")  S VALMHDR(1)="Insurance Co: "\_$P($G(^DIC(36,+IBCNS,0)),U)\_PCDISP  Q  ; INIT ; Initialize  D CLEAN^VALM10  I '$D(IBCNS) N IBCNS S IBCNS=IBINS  N IBLCT  S IBLCT=0  ; Display the list  D SET1(.IBLCT,"Attending/Rendering Provider Secondary ID")  D SET1(.IBLCT,"Default ID (1500): "\_$$GET1^DIQ(36,IBCNS,4.01))  **;JWS;IB\*2.0\*592; add field for Att/Rend Sec ID for J430D**  **D SET1(.IBLCT,"Default ID (J430D): "\_$$GET1^DIQ(36,IBCNS,4.14))**  D SET1(.IBLCT,"Default ID (UB-04): "\_$$GET1^DIQ(36,IBCNS,4.02))  D SET1(.IBLCT,"Require ID on Claim: "\_$$GET1^DIQ(36,IBCNS,4.03))  D SET1(.IBLCT," ")  D SET1(.IBLCT,"Referring Provider Secondary ID")  D SET1(.IBLCT,"Default ID (1500): "\_$$GET1^DIQ(36,IBCNS,4.04))  **;JWS;IB\*2.0\*592;add field for Refer Pro Sec ID for J430D  D SET1(.IBLCT,"Default ID (J430D): "\_$$GET1^DIQ(36,IBCNS,4.15))**  D SET1(.IBLCT,"Require ID on Claim: "\_$$GET1^DIQ(36,IBCNS,4.05))  D SET1(.IBLCT," ") **;JWS;IB\*2.0\*592  D SET1(.IBLCT,"Assistant Surgeon Secondary ID")**  **D SET1(.IBLCT,"Default ID (J430D): "\_$$GET1^DIQ(36,IBCNS,4.17))**  D SET1(.IBLCT," ")  D SET1(.IBLCT,"Billing Provider Secondary IDs")  D SET1(.IBLCT,"Use Att/Rend ID as Billing Provider Sec. ID (1500)?: "\_$$GET1^DIQ(36,IBCNS,4.06))  **;JWS;IB\*2.0\*592; add field for Att/Rend ID as Billing Prov sec id for J430D  D SET1(.IBLCT,"Use Att/Rend ID as Billing Provider Sec. ID (J430D)?: "\_$$GET1^DIQ(36,IBCNS,4.16))**  D SET1(.IBLCT,"Use Att/Rend ID as Billing Provider Sec. ID (UB-04)?: "\_$$GET1^DIQ(36,IBCNS,4.08))  D SET1(.IBLCT,"Transmit no Billing Provider Sec ID for the following Electronic Plan Types:")  D LIST^DIC(36.013,","\_IBCNS\_",",".01",,10,,,,,,"TAR","ERR")  F I=1:1:+$G(TAR("DILIST",0)) D  . D SET1(.IBLCT,TAR("DILIST",1,I))  D SET1(.IBLCT," ")  D SET1(.IBLCT,"Billing Provider/Service Facility")  ;  S IBCNS4=$G(^DIC(36,+IBCNS,4))  ; MRD;IB\*2.0\*516 - Marked fields 4.07, 4.11, 4.12 and 4.13 for  ; deletion and removed all references to them.  ;D SET1(.IBLCT,"Always use main VAMC as Billing Provider (1500)?: "\_$$EXPAND^IBTRE(36,4.11,+$P(IBCNS4,U,11)))  ;D SET1(.IBLCT,"Always use main VAMC as Billing Provider (UB-04)?: "\_$$EXPAND^IBTRE(36,4.12,+$P(IBCNS4,U,12)))  ;I $P(IBCNS4,U,11)!($P(IBCNS4,U,12)) D  ;.D SET1(.IBLCT,"Send VA Lab/Facility IDs or Facility Data for VAMC?: "\_$$EXPAND^IBTRE(36,4.07,+$P(IBCNS4,U,7)))  ;.D SET1(.IBLCT,"Use the Billing Prov (VAMC) Name and Street Address?: "\_$$EXPAND^IBTRE(36,4.13,+$P(IBCNS4,U,13)))  ;.Q  ;  S VALMBG=1,VALMCNT=IBLCT  Q  ; SET1(IBLCT,TEXT,IBCT) ;  S IBLCT=IBLCT+1 D SET^VALM10(IBLCT,TEXT)  Q  ; EXPND ;  Q HELP ;  Q EXIT ;  D CLEAN^VALM10  Q  ; IDPARAM ;  D FULL^VALM1  N DIE,DA,DR  I '$D(IBCNS) N IBCNS S IBCNS=IBINS  S DIE="^DIC(36,",(DA,Y)=IBCNS,DR="[IBEDIT INS CO1]"  I '$D(IBY) N IBY S IBY=",12,"  D ^DIE K DIE  K ^TMP("IBCE\_PRVFAC\_MAINT",$J)  D INIT  S VALMBCK="R"  Q  ; BILLPRVP ;  D FULL^VALM1  D EN^IBCEPC  D INIT  K ^TMP("IBCE\_PRVFAC\_MAINT",$J)  S VALMBCK="R"  Q | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCEF2 | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** |  | | | | | | | | |
| **Related Options** | None | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  |  | | | | |  | | | |
| **Data Dictionary (DD) References** |  | | | | | | | | |
| **Related Protocols** | None | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Current Logic** | | | | | | | | | |
| IBCEF2 ;ALB/TMP - FORMATTER SPECIFIC BILL FUNCTIONS ;8/6/03 10:54am  ;;2.0;INTEGRATED BILLING;\*\*52,85,51,137,232,155,296,349,403,400,432,488,461,547\*\*;21-MAR-94;Build 119  ;;Per VA Directive 6402, this routine should not be modified.  ; HOS(IBIFN) ; Extract rev codes for inst. episode into IBXDATA  ; Moved for space  D HOS^IBCEF22(IBIFN)  Q  ; OTHINS(IBIFN) ;Determine 'other insurance' node (I1,I2)  ; If primary bill, other ins is secondary  ; If sec or tert bill, other ins is primary  ;IBIFN = bill ien  N Z  S Z=$$COBN^IBCEF(IBIFN)  Q "I"\_$S(Z=1:2,1:1)  ; OTHINS1(IBIFN) ; Returns the COB #'s of all 'other insurance' as a string  ;IBIFN = bill ien  N IBC,Z  S Z=$$COBN^IBCEF(IBIFN)  I Z=1 S IBC=$S($D(^DGCR(399,IBIFN,"I2")):$S($D(^DGCR(399,IBIFN,"I3")):23,1:2),1:"") ;Primary=>2 or 23  I Z=2 S IBC="1"\_$S($D(^DGCR(399,IBIFN,"I3")):3,1:"") ;Secondary=>1 or 13  I Z=3 S IBC="12" ;Tertiary =>12 OTHQ Q IBC  ; RECVR(IBIFN) ; Returns the V.A. internal routing id of the current ins  ; co for 837  ;IBIFN = bill ien  N MCR,NUM,IBPH  S IBPH=$P("P^H",U,$$FT^IBCEF(IBIFN)-1)  S NUM="ENVOY"\_IBPH  ; If rate type is CHAMPVA, send 'CHAMVA'  I $P($G(^DGCR(399.3,+$P($G(^DGCR(399,IBIFN,0)),U,7),0)),U)="CHAMPVA" S NUM="CHAMV"\_IBPH  I NUM["ENVOY",$$MCRWNR^IBEFUNC(+$$CURR(IBIFN)) D  . S MCR=$P("B^A",U,$$FT^IBCEF(IBIFN)-1) ; PART A/B for MEDICARE  . S NUM="PART"\_MCR  Q NUM  ; ALLPAYID(IBIFN,IBXDATA,SEQ) ; Returns clearinghouse id for all (SEQ="")  ; or a specific (SEQ=1,2,3) ins co's for 837 in IBXDATA(n) for bill ien  ; IBIFN  ; EJK \*296\* Add IBMRA - MRA Claim type.   ; EJK \*296\* Add IBEBI - Electronic Billing ID  ;  ;WCJ;IB\*2.0\*547 - added IBM2  ;N Z,Z0,Z1,A,IBM,IBINST,IBMCR,IBX,IBMRA,IBEBI  N Z,Z0,Z1,A,IBM,IBM2,IBINST,IBMCR,IBX,IBMRA,IBEBI  ;S IBXDATA="",IBM=$G(^DGCR(399,IBIFN,"M"))  S IBXDATA="",IBM=$G(^DGCR(399,IBIFN,"M")),IBM2=$G(^DGCR(399,IBIFN,"M2"))  F Z=1:1:3 I $S('$G(SEQ):1,1:Z=SEQ) S Z0=$P(IBM,U,Z) I Z0 D  S:A'="" IBXDATA(Z)=A  . S A=""  . ;WCJ;IB\*2.0\*547  . I $P(IBM2,U,Z\*2)]"" S A=$P(IBM2,U,Z\*2) Q  ; grab new alternate payer IDs from bill if they exist  . ;  . S IBINST=($$FT^IBCEF(IBIFN)=3) ;Is bill UB-04?  . ; EJK \*296\* Get IBEBI based on Prof. or Inst. claim  . I IBINST S IBEBI=$P($G(^DIC(36,Z0,3)),U,4)  . I 'IBINST S IBEBI=$P($G(^DIC(36,Z0,3)),U,2)  . S IBEBI=$$UP^XLFSTR(IBEBI)  . ; EJK \*296\* If this is a Medicare claim, it may be printed or transmitted.   . S IBMRA=$$MRASEC^IBCEF4(IBIFN) ;Is claim 2ndary to an MRA?   . S IBMCR=$$MCRONBIL^IBEFUNC(IBIFN),Z1=$G(^DGCR(399,IBIFN,"TX"))  . Q:$P(Z1,U,8)=1!$S('$P(Z1,U,9):0,1:$$MRASEC^IBCEF4(IBIFN)) ;Force local prnt  . S A=$S($P(Z1,U,8)'=2:$P($G(^DIC(36,Z0,3)),U,$S(IBINST:4,1:2)),1:"")  . S A=$$UP^XLFSTR(A)  . ;  . ; RPRNT = CMS-1500 Rx bills  . ; IPRNT = Inst MRA secondary claims  . ; PPRNT = Prof MRA secondary claims  . ; HPRNT = inst printed bills (non-MRA, force print at clearinghouse)  . ; SPRNT = prof printed bills (non-MRA, force print at clearinghouse)  . ;  . ; Default to appropriate 'xPRNT' if Rx bill or COB bill or forced to  . ; print - claims must print at clearinghouse  . ;  . ; Rx bills on CMS-1500  . ;IB\*2.0\*432/TAZ Claims no longer print at clearinghouse  . ;I 'IBINST,$$ISRX^IBCEF1(IBIFN) S A="RPRNT" Q  . ;  . ; Claim forced to print at clearinghouse (Field #27)  . I $P(Z1,U,8)=2 S A=$S(IBINST:"H",1:"S")\_"PRNT" Q  . ;  . ; EJK \*296\* Send IBEBI for MRA secondary claims if it exists  . I Z>1,IBMRA,IBEBI'="" S A=IBEBI Q  . ;  . ; MRA secondary claim  . I Z>1,IBMCR=1,$P(Z1,U,5)="C" S A=$S(IBINST:"I",1:"P")\_"PRNT" Q  . ;  . ; Medicare is current payer (MRA request claim)  . I $$WNRBILL^IBEFUNC(IBIFN,Z) S A=$S(IBINST:"12M61",1:"SMTX1") Q  . ;  . ; IB\*296 - Do not modify the payer ID for CHAMPVA (HAC)  . I A=84146 Q  . I A=84147 Q  . ;  . ; If not a primary bill force to print  . ;IB\*2.0\*432/TAZ secondary bills will now be processed  . ;I Z>1,Z=$$COBN^IBCEF(IBIFN) S A=$S(IBINST:"H",1:"S")\_"PRNT" Q  . Q  ;  Q  ; PAYERID(IBIFN) ; Returns clearinghouse id for current ins co  ; IBIFN = bill ien  N NUM,IBSEQ  ; Determine the current ins co's # to identify at WEBMD  ; Envoy changed to WEBMD in patch 232  S IBSEQ=+$$COBN^IBCEF(IBIFN)  D ALLPAYID(IBIFN,.NUM,IBSEQ) S NUM=$G(NUM(IBSEQ))  Q $G(NUM)  ; CURR(IBIFN) ; Returns ien of the current insurance  ; company for bill ien IBIFN  Q $$FINDINS^IBCEF1(IBIFN)  ; ADMDT(IBIFN,NOOUTCK) ; Calculate admission/start of care date/time  D ADMDT^IBCEF21(IBIFN,$G(NOOUTCK)) ; Moved for space  Q  ; DISDT(IBIFN) ; Calculate discharge date  D DISDT^IBCEF21(IBIFN) ; Moved for space  Q  ; INDTS(IBIFN) ; Function returns the admit ^ discharge date/time of admission if patient is an inpatient on bill's event date  N Z,Z0,DFN,VAINDT,VAIN S Z0=""  S Z=$G(^DGCR(399,+$G(IBIFN),0)),DFN=$P(Z,U,2),VAINDT=$P(Z,U,3)  I +DFN,+VAINDT D INP^VADPT I +VAIN(1) S Z0=+VAIN(7)\_U\_+$G(^DGPM(+$P($G(^DGPM(+VAIN(1),0)),U,17),0))  Q Z0  ; TXMT(IBIFN) ; Function moved - use new call in IBCEF4  Q $$TXMT^IBCEF4(IBIFN)  ;  ; ID(LN,VAL) ; Set EXTRACT GLOBAL for multi-valued record  ; ids for Austin  ; LN = the line # being extracted  ; VAL = the value of the element being extracted  ;  ; Assumes IBXPG exists  ;  Q:LN<2  D SETGBL^IBCEFG(IBXPG,LN,1,VAL,.IBXSIZE)  Q  ; ID1(LN,DX,CT,DCT,ECT) ;Special entry point for diagnoses to 'save' the fact  ; a dx code is an e-code.  ; LN is last entry # output, returned as the entry # (IBXLINE) to assign to this entry  ; DX = the actual Dx code array(RECORD ID). Pass by reference, DX returned null if  ; dx was not output  ; CT = the ct on the 'DC' entry. pass by reference, returned null if  ; the end of the valid dx codes has been reached  ; DCT= Count of regular DX codes. UB-04 can have 25 non External Cause codes.  ; ECT= Count of External Cause codes. UB-04 can have 12 External Cause codes.  ; External Cause of Injury codes and qualifier changed with ICD-10: E-codes in ICD-9, V,X,W,Y-codes in ICD-10  N IBINS,VAL,CNT,DXIEN,DXQ,EDX,I,POA,ICDV  S IBINS=($$FT^IBCEF(IBXIEN)=3)  S VAL="DC"\_CT  S VAL=$E(VAL\_" ",1,4)  S DCT=+$G(DCT),ECT=+$G(ECT) ;Make sure variables are initialized.  ;  S EDX=0,DX=$G(DX)  S ICDV=$$ICD9VER^IBACSV(+$G(DX(CT)))  I ICDV=1,$E(DX)="E" S EDX=1 ; TRUE if ECI ICD-9 Dx (e-code)  I ICDV=30,"VWXY"[$E(DX) S EDX=1 ; TRUE if ECI ICD-10 Dx  ;  S I=$S(EDX:3,1:2)  ;  S:'EDX DXQ=$S(+$G(^TMP("DCX",$J,2))>0:"BF",1:"BK") ; first non e-code DX is principal (qualifier "BK"), the rest have qualifier "BF"  ;  I IBINS D  I DX="" G IDX1  .;I CT>28 S CT="" Q ; Max of 28 codes for institutional/UB  .I EDX S ECT=ECT+1 I ECT>12 S DX="" Q  ;Only 12 E-codes allowed  .I 'EDX S DCT=DCT+1 I DCT>25 S DX="" Q  ;Only 25 DX codes allowed  .S DXIEN=$P(DX(CT),U,2) Q:DXIEN=""  .; IB\*2.0\*547 - no longer stuff a 1 for POA, send a blank if null  .S POA=$P($G(^IBA(362.3,DXIEN,0)),U,4) ; I POA="",$$INPAT^IBCEF(IBXIEN) S POA=1 ; POA indicator defaults to "1", if not present on inpatient claim  .S:EDX DXQ="BN" ; e-code DX qualifier  .Q  ;  I 'IBINS S:EDX DXQ="BF" S POA="" ; on CMS-1500 e-code DX qualifiers are "BF" and there's no POA  ;  I ICDV=30 S DXQ="A"\_DXQ ; adjust Qualifier for ICD-10 codes  ;  ;Changed 8 to 12 so we can transmit 12 codes. BAA \*488\*  I 'IBINS,CT>12 S ^TMP("IBXSAVE",$J,"DX",IBXIEN)=$G(^TMP("IBXSAVE",$J,"DX",IBXIEN))+1,^TMP("IBXSAVE",$J,"DX",IBXIEN,$P(DX(+^TMP("IBXSAVE",$J,"DX",IBXIEN)),U,2))=$G(^TMP("IBXSAVE",$J,"DX",IBXIEN)) S DX="" Q  ;  I CT'="",DX'="" D  .; populate ^TMP("DCX") scratch global  .S ^TMP("DCX",$J,1)=CT,CNT=$G(^TMP("DCX",$J,I))+1,^TMP("DCX",$J,I)=CNT  .S (^TMP("DCX",$J,I,CNT),^TMP("DCX",$J,1,CT))=DX\_U\_DXQ\_U\_POA  .S LN=LN+1 D ID(LN,VAL) S ^TMP("IBXSAVE",$J,"DX",IBXIEN,$P(DX(LN),U,2))=LN,^TMP("IBXSAVE",$J,"DX",IBXIEN)=CT,CT=CT+1  .Q  ; IDX1 ;  Q  ; M(CT) ; Calculate multi-valued field for 837 extract  ; CT = passed by reference/the record ID counter  S CT=CT+1  ;IB\*2.0\*547/TAZ Increase counter to 25  ;Q $E(CT#12+$S(CT#12:0,1:12)\_" ",1,2)  Q $E(CT#25+$S(CT#25:0,1:25)\_" ",1,2)  ; SVITM(IBA,LINE) ; Saves the linked items from the bill data extract into  ; an array the formatter will use to link Rxs and prosthetics  ; to an SV1 or SV2 line item, if possible. Kills off IBA array entries  ; after they are 'moved'  ; IBA = array that contains the data to be saved  ; subscripts are (line #,item type,item pointer)=ct  N Z0,Z1  S Z0="" F  S Z0=$O(IBA("OUTPT",LINE,Z0)) Q:Z0=""  I Z0?1N.N  S Z1="" F  S Z1=$O(IBA("OUTPT",LINE,Z0,Z1)) Q:Z1=""  S ^TMP($J,"IBITEM",Z0,Z1,LINE)=IBA("OUTPT",LINE,Z0,Z1) K IBA("OUTPT",LINE,Z0,Z1)  Q  ; LINK(IBTYP,IBDATA) ; Link the item with a service line, if possible  ; IBTYP = the code for the type of item  ; returned incremented if no link is made  ; IBDATA = the extracted data string that identifies the item.   ; Returns the line to link to or null if no link  N IBLN,IBKEY,Z  S IBLN=""  S IBKEY=$S(IBTYP=3:$P(IBDATA,U,9),IBTYP=5:$P(IBDATA,U,4),1:"") Q:IBKEY=""  I $D(^TMP($J,"IBITEM",IBTYP,IBKEY)) D  G:IBLN LINKQ  .S Z=0 F  S Z=$O(^TMP($J,"IBITEM",IBTYP,IBKEY,Z)) Q:'Z  I ^TMP($J,"IBITEM",IBTYP,IBKEY,Z) S IBLN=Z,^TMP($J,"IBITEM",IBTYP,IBKEY,Z)=^TMP($J,"IBITEM",IBTYP,IBKEY,Z)-1 Q  I $D(^TMP($J,"IBITEM",IBTYP,0)) S IBKEY=0 D  .S Z=0 F  S Z=$O(^TMP($J,"IBITEM",IBTYP,IBKEY,Z)) Q:'Z  I ^TMP($J,"IBITEM",IBTYP,IBKEY,Z) S IBLN=Z,^TMP($J,"IBITEM",IBTYP,IBKEY,Z)=^TMP($J,"IBITEM",IBTYP,IBKEY,Z)-1 Q LINKQ Q IBLN  ; COID(IBIFN) ; Claim office ID - moved for space  Q $$COID^IBCEF21(IBIFN)  ; PPOL(IBIFN,COB) ; return IFN of patient policy on a bill defined by COB (fields 399,112-114)  N X,Y,PPOL S PPOL=""  I +$G(IBIFN) S X=$G(^DGCR(399,+IBIFN,"M")) I +$G(COB),COB<4 S Y=COB+11,PPOL=$P(X,U,Y)  Q PPOL  ; LADJ(SUB,LINE,SEQ1,GRP,IBXSAVE,PIECE) ; Extract line level adjustments  ; SUB = 1st subscript in IBXSAVE array to use  ; LINE = 2nd subscript  ; SEQ1 = 4th subscript  ; GRP = 5th subscript  ; IBXSAVE = array that has the data for COB line level adjustments  ; PIECE = # of the piece on the 0-node of the line level  ; adjustment reason to be extracted  ;  N A,B  S (A,B)=0  F  S A=$O(IBXSAVE(SUB,LINE,"COB",SEQ1,GRP,A)) Q:'A  D  . S B=B+1,IBXDATA(B)=$P(IBXSAVE(SUB,LINE,"COB",SEQ1,GRP,A),U,PIECE)  Q  ; ESGHPST(IBIFN,COB) ; return insureds employ status if bill policy defined by COB is an Employer Sponsored Group Health Plan  Q $$ESGHPST^IBCEF21(IBIFN,COB) ;Tag moved  ; ESGHPNL(IBIFN,COB) ; return employer name and location if bill policy defined by COB is an Employer Sponsored Group Health Plan  Q $$ESGHPNL^IBCEF21(IBIFN,COB) ;Tag moved  ; AMTOUT(A,B,C,IBXSAVE) ; format output amount  ;  N Z,K,IBZ,IBARR K IBXDATA S (IBZ,K)=0,IBARR="IBXSAVE("""\_A\_""")" F  S IBZ=$O(@IBARR@(IBZ)) Q:'IBZ  S K=K+1,Z=0 F  S Z=$O(@IBARR@(IBZ,Z)) Q:'Z  I $P($G(@IBARR@(IBZ,Z,B)),U,C) S IBXDATA(K)=$$DOLLAR^IBCEFG1($G(IBXDATA(K))+$P(@IBARR@(IBZ,Z,B),U,C))  Q | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCEF2 ;ALB/TMP - FORMATTER SPECIFIC BILL FUNCTIONS ;8/6/03 10:54am  ;;2.0;INTEGRATED BILLING;\*\*52,85,51,137,232,155,296,349,403,400,432,488,461,547,**592**\*\*;21-MAR-94;Build 119  ;;Per VA Directive 6402, this routine should not be modified.  ; HOS(IBIFN) ; Extract rev codes for inst. episode into IBXDATA  ; Moved for space  D HOS^IBCEF22(IBIFN)  Q  ; OTHINS(IBIFN) ;Determine 'other insurance' node (I1,I2)  ; If primary bill, other ins is secondary  ; If sec or tert bill, other ins is primary  ;IBIFN = bill ien  N Z  S Z=$$COBN^IBCEF(IBIFN)  Q "I"\_$S(Z=1:2,1:1)  ; OTHINS1(IBIFN) ; Returns the COB #'s of all 'other insurance' as a string  ;IBIFN = bill ien  N IBC,Z  S Z=$$COBN^IBCEF(IBIFN)  I Z=1 S IBC=$S($D(^DGCR(399,IBIFN,"I2")):$S($D(^DGCR(399,IBIFN,"I3")):23,1:2),1:"") ;Primary=>2 or 23  I Z=2 S IBC="1"\_$S($D(^DGCR(399,IBIFN,"I3")):3,1:"") ;Secondary=>1 or 13  I Z=3 S IBC="12" ;Tertiary =>12 OTHQ Q IBC  ; RECVR(IBIFN) ; Returns the V.A. internal routing id of the current ins  ; co for 837  ;IBIFN = bill ien  N MCR,NUM,IBPH **;JWS;IB\*2.0\*592:Dental form #7**  S IBPH=$P("P^H^^^^**DENTAL**",U,$$FT^IBCEF(IBIFN)-1)  S NUM=$S**($$FT^IBCEF(IBIFN)=7:IBPH**,1:"ENVOY"\_IBPH)  ; If rate type is CHAMPVA, send 'CHAMVA'  I $P($G(^DGCR(399.3,+$P($G(^DGCR(399,IBIFN,0)),U,7),0)),U)="CHAMPVA" S NUM="CHAMV"\_IBPH  I NUM["ENVOY",$$MCRWNR^IBEFUNC(+$$CURR(IBIFN)) D  . **;JWS;IB\*2.0\*592:Dental form #7**  . S MCR=$P("B^A^^^^**B**",U,$$FT^IBCEF(IBIFN)-1) ; PART A/B for MEDICARE  . S NUM="PART"\_MCR  Q NUM  ; ALLPAYID(IBIFN,IBXDATA,SEQ) ; Returns clearinghouse id for all (SEQ="")  ; or a specific (SEQ=1,2,3) ins co's for 837 in IBXDATA(n) for bill ien  ; IBIFN  ; EJK \*296\* Add IBMRA - MRA Claim type.   ; EJK \*296\* Add IBEBI - Electronic Billing ID  ;  ;WCJ;IB\*2.0\*547 - added IBM2  ;N Z,Z0,Z1,A,IBM,IBINST,IBMCR,IBX,IBMRA,IBEBI  N Z,Z0,Z1,A,IBM,IBM2,IBINST,IBMCR,IBX,IBMRA,IBEBI  ;S IBXDATA="",IBM=$G(^DGCR(399,IBIFN,"M"))  S IBXDATA="",IBM=$G(^DGCR(399,IBIFN,"M")),IBM2=$G(^DGCR(399,IBIFN,"M2"))  F Z=1:1:3 I $S('$G(SEQ):1,1:Z=SEQ) S Z0=$P(IBM,U,Z) I Z0 D  S:A'="" IBXDATA(Z)=A  . S A=""  . ;WCJ;IB\*2.0\*547  . I $P(IBM2,U,Z\*2)]"" S A=$P(IBM2,U,Z\*2) Q  ; grab new alternate payer IDs from bill if they exist  . ;  . S IBINST=($$FT^IBCEF(IBIFN)=3) ;Is bill UB-04?  . ; EJK \*296\* Get IBEBI based on Prof. or Inst. claim  . I IBINST S IBEBI=$P($G(^DIC(36,Z0,3)),U,4)  . I 'IBINST S IBEBI=$P($G(^DIC(36,Z0,3)),U,2)  **. ;JWS;IB\*2.0\*592;Dental payer id  . I $$FT^IBCEF(IBIFN)=7 S IBEBI=$P($G(^DIC(36,Z0,3)),U,15)**  . S IBEBI=$$UP^XLFSTR(IBEBI)  . ; EJK \*296\* If this is a Medicare claim, it may be printed or transmitted.   . S IBMRA=$$MRASEC^IBCEF4(IBIFN) ;Is claim 2ndary to an MRA?   . S IBMCR=$$MCRONBIL^IBEFUNC(IBIFN),Z1=$G(^DGCR(399,IBIFN,"TX"))  . Q:$P(Z1,U,8)=1!$S('$P(Z1,U,9):0,1:$$MRASEC^IBCEF4(IBIFN)) ;Force local prnt  . S A=$S($P(Z1,U,8)'=2:$P($G(^DIC(36,Z0,3)),U,$S(IBINST:4,1:2)),1:"")  . S A=$$UP^XLFSTR(A)  . ;  . ; RPRNT = CMS-1500 Rx bills  . ; IPRNT = Inst MRA secondary claims  . ; PPRNT = Prof MRA secondary claims  . ; HPRNT = inst printed bills (non-MRA, force print at clearinghouse)  . ; SPRNT = prof printed bills (non-MRA, force print at clearinghouse)  . ;  . ; Default to appropriate 'xPRNT' if Rx bill or COB bill or forced to  . ; print - claims must print at clearinghouse  . ;  . ; Rx bills on CMS-1500  . ;IB\*2.0\*432/TAZ Claims no longer print at clearinghouse  . ;I 'IBINST,$$ISRX^IBCEF1(IBIFN) S A="RPRNT" Q  . ;  . ; Claim forced to print at clearinghouse (Field #27)  . I $P(Z1,U,8)=2 S A=$S(IBINST:"H",1:"S")\_"PRNT" Q  . ;  . ; EJK \*296\* Send IBEBI for MRA secondary claims if it exists  . I Z>1,IBMRA,IBEBI'="" S A=IBEBI Q  . ;  . ; MRA secondary claim  . I Z>1,IBMCR=1,$P(Z1,U,5)="C" S A=$S(IBINST:"I",1:"P")\_"PRNT" Q  . ;  . ; Medicare is current payer (MRA request claim)  . I $$WNRBILL^IBEFUNC(IBIFN,Z) S A=$S(IBINST:"12M61",1:"SMTX1") Q  . ;  . ; IB\*296 - Do not modify the payer ID for CHAMPVA (HAC)  . I A=84146 Q  . I A=84147 Q  . ;  . ; If not a primary bill force to print  . ;IB\*2.0\*432/TAZ secondary bills will now be processed  . ;I Z>1,Z=$$COBN^IBCEF(IBIFN) S A=$S(IBINST:"H",1:"S")\_"PRNT" Q  . Q  ;  Q  ; PAYERID(IBIFN) ; Returns clearinghouse id for current ins co  ; IBIFN = bill ien  N NUM,IBSEQ  ; Determine the current ins co's # to identify at WEBMD  ; Envoy changed to WEBMD in patch 232  S IBSEQ=+$$COBN^IBCEF(IBIFN)  D ALLPAYID(IBIFN,.NUM,IBSEQ) S NUM=$G(NUM(IBSEQ))  Q $G(NUM)  ; CURR(IBIFN) ; Returns ien of the current insurance  ; company for bill ien IBIFN  Q $$FINDINS^IBCEF1(IBIFN)  ; ADMDT(IBIFN,NOOUTCK) ; Calculate admission/start of care date/time  D ADMDT^IBCEF21(IBIFN,$G(NOOUTCK)) ; Moved for space  Q  ; DISDT(IBIFN) ; Calculate discharge date  D DISDT^IBCEF21(IBIFN) ; Moved for space  Q  ; INDTS(IBIFN) ; Function returns the admit ^ discharge date/time of admission if patient is an inpatient on bill's event date  N Z,Z0,DFN,VAINDT,VAIN S Z0=""  S Z=$G(^DGCR(399,+$G(IBIFN),0)),DFN=$P(Z,U,2),VAINDT=$P(Z,U,3)  I +DFN,+VAINDT D INP^VADPT I +VAIN(1) S Z0=+VAIN(7)\_U\_+$G(^DGPM(+$P($G(^DGPM(+VAIN(1),0)),U,17),0))  Q Z0  ; TXMT(IBIFN) ; Function moved - use new call in IBCEF4  Q $$TXMT^IBCEF4(IBIFN)  ;  ; ID(LN,VAL) ; Set EXTRACT GLOBAL for multi-valued record  ; ids for Austin  ; LN = the line # being extracted  ; VAL = the value of the element being extracted  ;  ; Assumes IBXPG exists  ;  Q:LN<2  D SETGBL^IBCEFG(IBXPG,LN,1,VAL,.IBXSIZE)  Q  ; ID1(LN,DX,CT,DCT,ECT) ;Special entry point for diagnoses to 'save' the fact  ; a dx code is an e-code.  ; LN is last entry # output, returned as the entry # (IBXLINE) to assign to this entry  ; DX = the actual Dx code array(RECORD ID). Pass by reference, DX returned null if  ; dx was not output  ; CT = the ct on the 'DC' entry. pass by reference, returned null if  ; the end of the valid dx codes has been reached  ; DCT= Count of regular DX codes. UB-04 can have 25 non External Cause codes.  ; ECT= Count of External Cause codes. UB-04 can have 12 External Cause codes.  ; External Cause of Injury codes and qualifier changed with ICD-10: E-codes in ICD-9, V,X,W,Y-codes in ICD-10  N IBINS,VAL,CNT,DXIEN,DXQ,EDX,I,POA,ICDV  S IBINS=($$FT^IBCEF(IBXIEN)=3)  S VAL="DC"\_CT  S VAL=$E(VAL\_" ",1,4)  S DCT=+$G(DCT),ECT=+$G(ECT) ;Make sure variables are initialized.  ;  S EDX=0,DX=$G(DX)  S ICDV=$$ICD9VER^IBACSV(+$G(DX(CT)))  I ICDV=1,$E(DX)="E" S EDX=1 ; TRUE if ECI ICD-9 Dx (e-code)  I ICDV=30,"VWXY"[$E(DX) S EDX=1 ; TRUE if ECI ICD-10 Dx  ;  S I=$S(EDX:3,1:2)  ;  S:'EDX DXQ=$S(+$G(^TMP("DCX",$J,2))>0:"BF",1:"BK") ; first non e-code DX is principal (qualifier "BK"), the rest have qualifier "BF"  ;  I IBINS D  I DX="" G IDX1  .;I CT>28 S CT="" Q ; Max of 28 codes for institutional/UB  .I EDX S ECT=ECT+1 I ECT>12 S DX="" Q  ;Only 12 E-codes allowed  .I 'EDX S DCT=DCT+1 I DCT>25 S DX="" Q  ;Only 25 DX codes allowed  .S DXIEN=$P(DX(CT),U,2) Q:DXIEN=""  .; IB\*2.0\*547 - no longer stuff a 1 for POA, send a blank if null  .S POA=$P($G(^IBA(362.3,DXIEN,0)),U,4) ; I POA="",$$INPAT^IBCEF(IBXIEN) S POA=1 ; POA indicator defaults to "1", if not present on inpatient claim  .S:EDX DXQ="BN" ; e-code DX qualifier  .Q  ;  I 'IBINS S:EDX DXQ="BF" S POA="" ; on CMS-1500 e-code DX qualifiers are "BF" and there's no POA  ;  I ICDV=30 S DXQ="A"\_DXQ ; adjust Qualifier for ICD-10 codes  ;  ;Changed 8 to 12 so we can transmit 12 codes. BAA \*488\*  I 'IBINS,CT>12 S ^TMP("IBXSAVE",$J,"DX",IBXIEN)=$G(^TMP("IBXSAVE",$J,"DX",IBXIEN))+1,^TMP("IBXSAVE",$J,"DX",IBXIEN,$P(DX(+^TMP("IBXSAVE",$J,"DX",IBXIEN)),U,2))=$G(^TMP("IBXSAVE",$J,"DX",IBXIEN)) S DX="" Q  ;  I CT'="",DX'="" D  .; populate ^TMP("DCX") scratch global  .S ^TMP("DCX",$J,1)=CT,CNT=$G(^TMP("DCX",$J,I))+1,^TMP("DCX",$J,I)=CNT  .S (^TMP("DCX",$J,I,CNT),^TMP("DCX",$J,1,CT))=DX\_U\_DXQ\_U\_POA  .S LN=LN+1 D ID(LN,VAL) S ^TMP("IBXSAVE",$J,"DX",IBXIEN,$P(DX(LN),U,2))=LN,^TMP("IBXSAVE",$J,"DX",IBXIEN)=CT,CT=CT+1  .Q  ; IDX1 ;  Q  ; M(CT) ; Calculate multi-valued field for 837 extract  ; CT = passed by reference/the record ID counter  S CT=CT+1  ;IB\*2.0\*547/TAZ Increase counter to 25  ;Q $E(CT#12+$S(CT#12:0,1:12)\_" ",1,2)  Q $E(CT#25+$S(CT#25:0,1:25)\_" ",1,2)  ; SVITM(IBA,LINE) ; Saves the linked items from the bill data extract into  ; an array the formatter will use to link Rxs and prosthetics  ; to an SV1 or SV2 line item, if possible. Kills off IBA array entries  ; after they are 'moved'  ; IBA = array that contains the data to be saved  ; subscripts are (line #,item type,item pointer)=ct  N Z0,Z1  S Z0="" F  S Z0=$O(IBA("OUTPT",LINE,Z0)) Q:Z0=""  I Z0?1N.N  S Z1="" F  S Z1=$O(IBA("OUTPT",LINE,Z0,Z1)) Q:Z1=""  S ^TMP($J,"IBITEM",Z0,Z1,LINE)=IBA("OUTPT",LINE,Z0,Z1) K IBA("OUTPT",LINE,Z0,Z1)  Q  ; LINK(IBTYP,IBDATA) ; Link the item with a service line, if possible  ; IBTYP = the code for the type of item  ; returned incremented if no link is made  ; IBDATA = the extracted data string that identifies the item.   ; Returns the line to link to or null if no link  N IBLN,IBKEY,Z  S IBLN=""  S IBKEY=$S(IBTYP=3:$P(IBDATA,U,9),IBTYP=5:$P(IBDATA,U,4),1:"") Q:IBKEY=""  I $D(^TMP($J,"IBITEM",IBTYP,IBKEY)) D  G:IBLN LINKQ  .S Z=0 F  S Z=$O(^TMP($J,"IBITEM",IBTYP,IBKEY,Z)) Q:'Z  I ^TMP($J,"IBITEM",IBTYP,IBKEY,Z) S IBLN=Z,^TMP($J,"IBITEM",IBTYP,IBKEY,Z)=^TMP($J,"IBITEM",IBTYP,IBKEY,Z)-1 Q  I $D(^TMP($J,"IBITEM",IBTYP,0)) S IBKEY=0 D  .S Z=0 F  S Z=$O(^TMP($J,"IBITEM",IBTYP,IBKEY,Z)) Q:'Z  I ^TMP($J,"IBITEM",IBTYP,IBKEY,Z) S IBLN=Z,^TMP($J,"IBITEM",IBTYP,IBKEY,Z)=^TMP($J,"IBITEM",IBTYP,IBKEY,Z)-1 Q LINKQ Q IBLN  ; COID(IBIFN) ; Claim office ID - moved for space  Q $$COID^IBCEF21(IBIFN)  ; PPOL(IBIFN,COB) ; return IFN of patient policy on a bill defined by COB (fields 399,112-114)  N X,Y,PPOL S PPOL=""  I +$G(IBIFN) S X=$G(^DGCR(399,+IBIFN,"M")) I +$G(COB),COB<4 S Y=COB+11,PPOL=$P(X,U,Y)  Q PPOL  ; LADJ(SUB,LINE,SEQ1,GRP,IBXSAVE,PIECE) ; Extract line level adjustments  ; SUB = 1st subscript in IBXSAVE array to use  ; LINE = 2nd subscript  ; SEQ1 = 4th subscript  ; GRP = 5th subscript  ; IBXSAVE = array that has the data for COB line level adjustments  ; PIECE = # of the piece on the 0-node of the line level  ; adjustment reason to be extracted  ;  N A,B  S (A,B)=0  F  S A=$O(IBXSAVE(SUB,LINE,"COB",SEQ1,GRP,A)) Q:'A  D  . S B=B+1,IBXDATA(B)=$P(IBXSAVE(SUB,LINE,"COB",SEQ1,GRP,A),U,PIECE)  Q  ; ESGHPST(IBIFN,COB) ; return insureds employ status if bill policy defined by COB is an Employer Sponsored Group Health Plan  Q $$ESGHPST^IBCEF21(IBIFN,COB) ;Tag moved  ; ESGHPNL(IBIFN,COB) ; return employer name and location if bill policy defined by COB is an Employer Sponsored Group Health Plan  Q $$ESGHPNL^IBCEF21(IBIFN,COB) ;Tag moved  ; AMTOUT(A,B,C,IBXSAVE) ; format output amount  ;  N Z,K,IBZ,IBARR K IBXDATA S (IBZ,K)=0,IBARR="IBXSAVE("""\_A\_""")" F  S IBZ=$O(@IBARR@(IBZ)) Q:'IBZ  S K=K+1,Z=0 F  S Z=$O(@IBARR@(IBZ,Z)) Q:'Z  I $P($G(@IBARR@(IBZ,Z,B)),U,C) S IBXDATA(K)=$$DOLLAR^IBCEFG1($G(IBXDATA(K))+$P(@IBARR@(IBZ,Z,B),U,C))  Q | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCEF72 | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** |  | | | | | | | | |
| **Related Options** | None | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  |  | | | | |  | | | |
| **Data Dictionary (DD) References** |  | | | | | | | | |
| **Related Protocols** | None | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Current Logic** | | | | | | | | | |
| IBCEF72 ;WOIFO/SS - FORMATTER AND EXTRACTOR SPECIFIC BILL FUNCTIONS ;8/6/03 10:56am  ;;2.0;INTEGRATED BILLING;\*\*232,320,349,432,516\*\*;21-MAR-94;Build 123  ;;Per VA Directive 6402, this routine should not be modified.  ;  ;  ;Input:  ;IBINSCO - ptr to #36  ;IBFRMTYP 0=unknwn/both,1=UB,2=1500  ;IBCARE - 0=unknwn or both inp/outp,1=inpatient, 2=outpatient, 3 -RX  ;Output: X12 IDtype^ID^ID TYPE ptr to file 355.97 CH35591(IBINSCO,IBFRMTYP,IBCARE) ;  N IB35591,IBRET,IB1  S IB35591=0,IBRET=""  F  S IB35591=$O(^IBA(355.91,"B",IBINSCO,IB35591)) Q:+IB35591=0 Q:IBRET'=""  D  . S IB1=$G(^IBA(355.91,IB35591,0))  . I '($P(IB1,"^",4)=0!(IBFRMTYP=0)) Q:$P(IB1,"^",4)'=IBFRMTYP  ;if wrong form type  . I ($P(IB1,"^",5)=3)!(IBCARE=3) Q:IBCARE'=$P(IB1,"^",5) ;if not RX  . I ($P(IB1,"^",5)=1)!($P(IB1,"^",5)=2) I (IBCARE=1)!(IBCARE=2) Q:$P(IB1,"^",5)'=IBCARE  ;if wrong care type  . S IBRET=$P($G(^IBE(355.97,+$P(IB1,"^",6),0)),"^",3)\_"^"\_$P(IB1,"^",7)\_U\_+$P(IB1,U,6)  Q IBRET  ; FINDEIN(IBXIEN,IBPROV,IBFAC,IBS) ; find EIN for facility/ SSN for person  ; IBXIEN = ien of bill entry file 399  ; IBFAC = 1 if facility, 0 if individual provider  ; IBPROV = ien of provider (vp format)  ; IBS = 1 if person's EIN should be returned if there, otherwise SSN  ; FUNCTION RETURNS   ; EIN or SSN ^ 24 for EIN, 34 for SSN or null if none found  N Z,Z0,IBARR,IBEIN,IBSSN  S (IBEIN,IBSSN)=""  D ALLID^IBCEP8(IBPROV,"",.IBARR)  S Z=0 F  S Z=$O(IBARR(Z)) Q:'Z  D  Q:IBEIN'=""  . I $G(IBFAC) Q:$P(IBARR(Z),U,7)'="EI"  S IBEIN=$P(IBARR(Z),U,2)\_U\_24 Q  . I $P(IBARR(Z),U,7)="SY" D  Q  .. I $G(IBS) S IBSSN=$P(IBARR(Z),U,2)\_U\_34 Q  . S IBEIN=$P(IBARR(Z),U,2)\_U\_24  . I $G(IBS),$P(IBARR(Z),U,7)="EI" S IBEIN=$P(IBARR(Z),U,2)\_U\_24  I $G(IBS),IBEIN="" S IBEIN=IBSSN  Q IBEIN  ;  ; NONVAID(IBXIEN,IBX,IBFAC,IBS) ; Find the non-VA provider default id  ; IBXIEN = the ien of the bill (file 399)  ; IBX = id data returned if passed by reference  ; IBFAC = 1 if getting the id for the facility or 0 for rendering prov  ; IBS = 1 if getting id for person, but need the EIN if there  ; Function returns the id^type of id^person/facility flag:  ; Type of id: 1 = SSN 2 = EIN 0 = not found  ; person/facility: 1 = person 2 = facility  N Z,IBXSAVE,IBU2,IBTYPE,IBZ,IBF,IBPROV,Q,Q0  S IBTYPE=2,IBU2=$G(^DGCR(399,IBXIEN,"U2")),IBF=2,IBPROV=""  ;  S Z=$P(IBU2,U,10)  I 'Z S IBX="",IBTYPE=0 G NONVAQ ; Not a non-VA facility  S IBPROV=Z\_";IBA(355.93,"  ;  ; Get EIN  I $G(IBFAC) D  G NONVAQ  . S IBX=$P($$FINDEIN(IBXIEN,IBPROV,IBFAC),U),IBTYPE=2  ;  ; Get EIN/SSN  I '$G(IBFAC) D  G NONVAQ  . S IBX="",IBF=1  . S Q0=($$FT^IBCEF(IBXIEN)=3)+3 ; 3 for rendering/4 for attending  . S Q=+$O(^DGCR(399,IBXIEN,"PRV","B",Q0,0))  . S IBPROV=$P($G(^DGCR(399,IBXIEN,"PRV",Q,0)),U,2)  . I IBPROV S IBX=$$FINDEIN(IBXIEN,IBPROV,IBFAC,$G(IBS)),IBTYPE=$S($P(IBX,U,2)=24:2,$P(IBX,U,2)=34:1,1:0),IBX=$P(IBX,U)  ; NONVAQ I IBTYPE,IBX="",$P(IBU2,U,12)'="" S IBX=$P(IBU2,U,12) ; pull from 399  S IBX=$G(IBX)  Q IBX\_U\_IBTYPE\_U\_IBF  ;----  ;checks if there is data for OP\* segments and   ;then populates PROV COB SEQ  ;Input:  ;IBXIEN - ien in #399  ;IBSAVE - "in" array (i.e. IBXSAVE)  ;IBDATA - "out" array (i.e. IBXDATA)  ;IBFUNC - FUNCTION from #399 (1-refering, 2 -operating, etc)  ;IBSEGM - segment record ID, optional  ;Output:  ; IBDATA with formatted output PROVSEQ(IBXIEN,IBSAVE,IBDATA,IBFUNC,IBSEGM) ;  N IB1,IBINS,IBFL  ;S IBFL=$S(IBFUNC=3!(IBFUNC=4):1,1:0)  F IB1=1,2 D  . I '$$ISINSUR^IBCEF71($G(IBSAVE("PROVINF",IBXIEN,"O",IB1)),IBXIEN) Q  ;don't create anything if there is no such insurance  . ;\*432/TAZ - Removed. Attending and Rendering can be on same bill now.  . ;I IBFL S IBFUNC=$S($O(IBSAVE("PROVINF",IBXIEN,"O",IB1,3,0)):3,1:4)  . I '$O(IBSAVE("PROVINF",IBXIEN,"O",IB1,IBFUNC,0)) Q  . S IBDATA(IB1)=$G(IBSAVE("PROVINF",IBXIEN,"O",IB1))  . I $G(IBSEGM)'="" D ID^IBCEF2(IB1,IBSEGM)  Q  ; OUTPRVID(IBXIEN,IBXSAVE) ; Extract the outside provider or facility ids  ; into IBXSAVE array  ; Function returns 1 if person or 2 if facility ids or "" if neither  N Z,IBXDATA,IBPERSON,TAG  ;WCJ;11/1/2005 Extract the first 3 chars of Z instead.  S Z=$E($$PSPRV^IBCEF7(IBXIEN),1,3),IBPERSON=""  ;EJK 8/23/05 IB\*320 - CHANGED Z=101 TO Z=1010. Z WILL ALWAYS BE A 4 DIGIT #.   ; WCJ 11/1/2005 ; Removed EJK's change and added above change  I Z=111!(Z=101) S TAG=$S(Z=101:"OUTSIDE FAC PROVIDER INF",1:"CUR/OTH PROVIDER INFO") D F^IBCEF("N-ALL "\_TAG) S IBPERSON=$S('$E(Z,2):2,1:1)  Q IBPERSON  ; OUTPRV(IBREC,IBXIEN,IBXSAVE) ; Extract the outside provider or facility ids  ; into IBXSAVE array  ; Function returns 1 if person or 2 if facility ids or "" if neither  ; IBREC = the record whose ids should be returned  N IBPERSON,IBFRM,IBTYPE,IBFAC  I IBREC="SUB1"!(IBREC="OP6") D  . K IBXSAVE("PROVINF",IBXIEN),IBXSAVE("PROVINF\_FAC",IBXIEN)  . S IBPERSON=$$OUTPRVID(IBXIEN,.IBXSAVE),IBFAC=$S(IBPERSON=1:0,1:1)  E  D  . K IBXSAVE("PROVINF\_FAC",IBXIEN)  . D F^IBCEF("N-ALL OUTSIDE FAC PROVIDER INF")  . S IBPERSON=2,IBFAC=1  S IBFRM=$$FT^IBCEF(IBXIEN),IBFRM=$S(IBFRM=2:2,1:1)  S IBTYPE=$S(IBREC["SUB":"C",1:"O")  D CHCKSUB^IBCEF73(IBFRM,IBREC,IBFAC,IBTYPE,.IBXSAVE)  Q IBPERSON  ;  ;get IENs in file #36 for other insurances OTHINS(IB399,IBRES) ;  N IBFRMTYP,Z,Z1,Z2,Z4  S Z=$$COBN^IBCEF(IB399),Z0=0  F Z1=1:1:3 I Z1'=Z,$D(^DGCR(399,IB399,"I"\_Z1)) D  . S Z0=Z0+1  . ; MRD;IB\*2.0\*516 - Added HPID as second piece.  . S IBRES(Z0)=+$G(^DGCR(399,IB399,"I"\_Z1))\_U\_$P(^DGCR(399,IB399,"M1"),U,12+Z1)  . Q  Q  ;  ;get other insurance EDI ID NUMBERs OTHINSID(IB399,IBRES) ;insurance EDI   N IBFRMTYP,IBZ,Z0,Z1,Z4  S IBFRMTYP=$$FT^IBCEF(IB399),IBFRMTYP=$S(IBFRMTYP=2:2,IBFRMTYP=3:1,1:0)  S Z4=$S(IBFRMTYP=1:4,1:2) ;UB - piece4,1500 or BOTH -piece 2  D OTHINS(IB399,.IBZ)  S Z1=0  F Z0=1,2 I $G(IBZ(Z0)) D  . S IBRES(Z0)=$S($$MCRWNR^IBEFUNC(+IBZ(Z0)):$S(IBFRMTYP=1:"12M61",1:"SMTX1"),1:$P($G(^DIC(36,+IBZ(Z0),3)),U,Z4))  . ; MRD;IB\*2.0\*516 - Added HPID as second piece.  . S $P(IBRES(Z0),U,2)=$P(IBZ(Z0),U,2)  . Q  Q  ;  ;get other insurance addresses OTHINADR(IB399,IBRES,IBADDFLD) ;insurance EDI  N IBZ,Z0,Z1,Z4  D OTHINS(IB399,.IBZ)  S Z1=0  I IBADDFLD=18 D  Q  . F Z0=1:1:2 I $G(IBZ(Z0)) D  . . S IBRES(Z0)=$P($G(^DIC(36,+IBZ(Z0),.11)),U,1)  . . S IBRES(Z0)=$E(IBRES(Z0),1,55)  I IBADDFLD=18.9 D  Q  . F Z0=1:1:2 I $G(IBZ(Z0)) D  . . S IBRES(Z0)=$P($G(^DIC(36,+IBZ(Z0),.11)),U,1)  . . S Z4=$P($G(^DIC(36,+IBZ(Z0),.11)),U,2) S:Z4'="" IBRES(Z0)=IBRES(Z0)\_", "\_Z4  . . S Z4=$P($G(^DIC(36,+IBZ(Z0),.11)),U,3) S:Z4'="" IBRES(Z0)=IBRES(Z0)\_", "\_Z4  . . S Z4=$P($G(^DIC(36,+IBZ(Z0),.11)),U,4) S:Z4'="" IBRES(Z0)=IBRES(Z0)\_", "\_Z4  . . S Z4=$P($G(^DIC(5,+$P($G(^DIC(36,+IBZ(Z0),.11)),U,5),0)),U,2) S:Z4'="" IBRES(Z0)=IBRES(Z0)\_", "\_Z4  . . S Z4=$P($G(^DIC(36,+IBZ(Z0),.11)),U,6) S:Z4'="" IBRES(Z0)=IBRES(Z0)\_", "\_Z4  . . S IBRES(Z0)=$E(IBRES(Z0),1,157)  I IBADDFLD=19 D  Q  . F Z0=1:1:2 I $G(IBZ(Z0)) D  . . S IBRES(Z0)=$P($G(^DIC(36,+IBZ(Z0),.11)),U,2)  . . S IBRES(Z0)=IBRES(Z0)\_" "\_$P($G(^DIC(36,+IBZ(Z0),.11)),U,3)  . . S IBRES(Z0)=$E(IBRES(Z0),1,55)  I IBADDFLD=20 D  Q  . F Z0=1:1:2 I $G(IBZ(Z0)) D  . . S IBRES(Z0)=$P($G(^DIC(36,+IBZ(Z0),.11)),U,4)  . . S IBRES(Z0)=$E(IBRES(Z0),1,30)  I IBADDFLD=21 D  Q  . F Z0=1:1:2 I $G(IBZ(Z0)) D  . . S IBRES(Z0)=$P($G(^DIC(5,+$P($G(^DIC(36,+IBZ(Z0),.11)),U,5),0)),U,2)  . . S IBRES(Z0)=$E(IBRES(Z0),1,2)  I IBADDFLD=22 D  Q  . F Z0=1:1:2 I $G(IBZ(Z0)) D  . . S IBRES(Z0)=$P($G(^DIC(36,+IBZ(Z0),.11)),U,6)  . . S IBRES(Z0)=$E(IBRES(Z0),1,15)  Q  ; SFIDQ(IBXIEN,IBXSAVE,IBXDATA) ; Find the service facility id qualifier for  ; 837 record SUB2-5  ;IBXIEN = ien of 399  ;Pass by reference: IBXSAVE (input/output) IBXDATA (output)  N B,Z  K IBXSAVE("NVID")  D  ; protect IBXDATA  . N IBXDATA  . D F^IBCEF("N-RENDERING INSTITUTION")  . S:IBXDATA'="" IBXSAVE("IBFAC")=IBXDATA  I $P($G(IBXSAVE("IBFAC")),U,2)'=1 K IBXDATA Q  S Z=$$PSPRV^IBCEF7(IBXIEN)  ;WCJ 11/04/2005 If a Non-VA facility   I $E(Z) D  . S IBXSAVE("NVID")=$$NONVAID^IBCEF72(IBXIEN,.B,$E(Z),1)  .; S IBXSAVE("NVID")=$$NONVAID^IBCEF72(IBXIEN,.B,'$E(Z,2),1)  . S IBXDATA=$P("^34^24",U,$P(IBXSAVE("NVID"),U,2)+1)  ;S Z=$$PSPRV^IBCEF7(IBXIEN),IBXSAVE("NVID")=$$NONVAID^IBCEF72(IBXIEN,.B,'$E(Z,2),1),IBXDATA=24  Q  ; OTHP36(IBXIEN,IBZOUT) ;  N Z,Z0,Z1,IBZ  D F^IBCEF("N-ALL INSURANCE CO 837 ID","IBZ")  F Z=1,2,3 S IBZOUT(Z)=+$$POLICY^IBCEF(IBXIEN,1,$E("PST",Z))  Q  ;  ;---------SORT-----------  ;IBPRNUM - seq #  ;IBPRTYP - type of provider (use FUNCTION value from file 399, fld 222)  ;IB399 = ien file 399  ;IBSRC,IBDST - source,destination arrays  ;IBN - starting #  ;Output:  ; IBDST(1-primary/2-secondary provider,Provider type(FUNCTION),N)=  ; =provider/VARIABLEPTR^Insurance PTR #36 or NONE^ID type^ID^Form type^Care type^state ptr #5 for state license #  ; where N is numeration (1 for ID1, 2 for ID2, etc) GETSSN(IBPTR) ;look for SSN in #200 first and if not found then look at #355.9  ;if in file #200  I $P(IBPTR,";",2)="VA(200," Q $$SSN200^IBCEF73(IBPTR)  ;if in 355.93 then use 355.9  Q $$SSN3559^IBCEF73(IBPTR)  ;--  ;SSN3559  ;Find SSN from 355.9  ;Input:  ; Variable pointer to ^VA(200 or ^IBA(355.93  ;Output:  ; SSN or null  ; PADNDC(Z) ;PAD LEADING ZERO'S INTO A NON 5-4-2 FORMAT NDC NUMBER  ;Z IS ITERATION, ONLY PAD CURRENT NDC NUMBER  N NDC  S NDC=$P(IBXSAVE("OUTPT",Z,"RX"),"^",3)  Q:$L(NDC)=13  I $L(NDC)=14 D  Q  . S $P(NDC,"-",1)=$E($P(NDC,"-",1),2,$L($P(NDC,"-",1)))  . S $P(IBXSAVE("OUTPT",Z,"RX"),"^",3)=NDC  I $L($P(NDC,"-",1))'=5 S $P(NDC,"-",1)="0"\_$P(NDC,"-",1)  I $L($P(NDC,"-",2))'=4 S $P(NDC,"-",2)="0"\_$P(NDC,"-",2)  I $L($P(NDC,"-",3))'=2 S $P(NDC,"-",3)="0"\_$P(NDC,"-",3)  S $P(IBXSAVE("OUTPT",Z,"RX"),"^",3)=NDC  Q  ; | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCEF72 ;WOIFO/SS - FORMATTER AND EXTRACTOR SPECIFIC BILL FUNCTIONS ;8/6/03 10:56am  ;;2.0;INTEGRATED BILLING;\*\*232,320,349,432,516,**592**\*\*;21-MAR-94;Build 123  ;;Per VA Directive 6402, this routine should not be modified.  ;  ;  ;Input:  ;IBINSCO - ptr to #36  ;IBFRMTYP 0=unknwn/both,1=UB,2=1500,4=J430D  ;IBCARE - 0=unknwn or both inp/outp,1=inpatient, 2=outpatient, 3 -RX  ;Output: X12 IDtype^ID^ID TYPE ptr to file 355.97 CH35591(IBINSCO,IBFRMTYP,IBCARE) ;  N IB35591,IBRET,IB1  S IB35591=0,IBRET=""  F  S IB35591=$O(^IBA(355.91,"B",IBINSCO,IB35591)) Q:+IB35591=0 Q:IBRET'=""  D  . S IB1=$G(^IBA(355.91,IB35591,0))  . I '($P(IB1,"^",4)=0!(IBFRMTYP=0)) Q:$P(IB1,"^",4)'=IBFRMTYP  ;if wrong form type  . I ($P(IB1,"^",5)=3)!(IBCARE=3) Q:IBCARE'=$P(IB1,"^",5) ;if not RX  . I ($P(IB1,"^",5)=1)!($P(IB1,"^",5)=2) I (IBCARE=1)!(IBCARE=2) Q:$P(IB1,"^",5)'=IBCARE  ;if wrong care type  . S IBRET=$P($G(^IBE(355.97,+$P(IB1,"^",6),0)),"^",3)\_"^"\_$P(IB1,"^",7)\_U\_+$P(IB1,U,6)  Q IBRET  ; FINDEIN(IBXIEN,IBPROV,IBFAC,IBS) ; find EIN for facility/ SSN for person  ; IBXIEN = ien of bill entry file 399  ; IBFAC = 1 if facility, 0 if individual provider  ; IBPROV = ien of provider (vp format)  ; IBS = 1 if person's EIN should be returned if there, otherwise SSN  ; FUNCTION RETURNS   ; EIN or SSN ^ 24 for EIN, 34 for SSN or null if none found  N Z,Z0,IBARR,IBEIN,IBSSN  S (IBEIN,IBSSN)=""  D ALLID^IBCEP8(IBPROV,"",.IBARR)  S Z=0 F  S Z=$O(IBARR(Z)) Q:'Z  D  Q:IBEIN'=""  . I $G(IBFAC) Q:$P(IBARR(Z),U,7)'="EI"  S IBEIN=$P(IBARR(Z),U,2)\_U\_24 Q  . I $P(IBARR(Z),U,7)="SY" D  Q  .. I $G(IBS) S IBSSN=$P(IBARR(Z),U,2)\_U\_34 Q  . S IBEIN=$P(IBARR(Z),U,2)\_U\_24  . I $G(IBS),$P(IBARR(Z),U,7)="EI" S IBEIN=$P(IBARR(Z),U,2)\_U\_24  I $G(IBS),IBEIN="" S IBEIN=IBSSN  Q IBEIN  ;  ; NONVAID(IBXIEN,IBX,IBFAC,IBS) ; Find the non-VA provider default id  ; IBXIEN = the ien of the bill (file 399)  ; IBX = id data returned if passed by reference  ; IBFAC = 1 if getting the id for the facility or 0 for rendering prov  ; IBS = 1 if getting id for person, but need the EIN if there  ; Function returns the id^type of id^person/facility flag:  ; Type of id: 1 = SSN 2 = EIN 0 = not found  ; person/facility: 1 = person 2 = facility  N Z,IBXSAVE,IBU2,IBTYPE,IBZ,IBF,IBPROV,Q,Q0  S IBTYPE=2,IBU2=$G(^DGCR(399,IBXIEN,"U2")),IBF=2,IBPROV=""  ;  S Z=$P(IBU2,U,10)  I 'Z S IBX="",IBTYPE=0 G NONVAQ ; Not a non-VA facility  S IBPROV=Z\_";IBA(355.93,"  ;  ; Get EIN  I $G(IBFAC) D  G NONVAQ  . S IBX=$P($$FINDEIN(IBXIEN,IBPROV,IBFAC),U),IBTYPE=2  ;  ; Get EIN/SSN  I '$G(IBFAC) D  G NONVAQ  . S IBX="",IBF=1  . S Q0=($$FT^IBCEF(IBXIEN)=3)+3 ; 3 for rendering/4 for attending  . S Q=+$O(^DGCR(399,IBXIEN,"PRV","B",Q0,0))  . S IBPROV=$P($G(^DGCR(399,IBXIEN,"PRV",Q,0)),U,2)  . I IBPROV S IBX=$$FINDEIN(IBXIEN,IBPROV,IBFAC,$G(IBS)),IBTYPE=$S($P(IBX,U,2)=24:2,$P(IBX,U,2)=34:1,1:0),IBX=$P(IBX,U)  ; NONVAQ I IBTYPE,IBX="",$P(IBU2,U,12)'="" S IBX=$P(IBU2,U,12) ; pull from 399  S IBX=$G(IBX)  Q IBX\_U\_IBTYPE\_U\_IBF  ;----  ;checks if there is data for OP\* segments and   ;then populates PROV COB SEQ  ;Input:  ;IBXIEN - ien in #399  ;IBSAVE - "in" array (i.e. IBXSAVE)  ;IBDATA - "out" array (i.e. IBXDATA)  ;IBFUNC - FUNCTION from #399 (1-refering, 2 -operating, etc)  ;IBSEGM - segment record ID, optional  ;Output:  ; IBDATA with formatted output PROVSEQ(IBXIEN,IBSAVE,IBDATA,IBFUNC,IBSEGM) ;  N IB1,IBINS,IBFL  ;S IBFL=$S(IBFUNC=3!(IBFUNC=4):1,1:0)  F IB1=1,2 D  . I '$$ISINSUR^IBCEF71($G(IBSAVE("PROVINF",IBXIEN,"O",IB1)),IBXIEN) Q  ;don't create anything if there is no such insurance  . ;\*432/TAZ - Removed. Attending and Rendering can be on same bill now.  . ;I IBFL S IBFUNC=$S($O(IBSAVE("PROVINF",IBXIEN,"O",IB1,3,0)):3,1:4)  . I '$O(IBSAVE("PROVINF",IBXIEN,"O",IB1,IBFUNC,0)) Q  . S IBDATA(IB1)=$G(IBSAVE("PROVINF",IBXIEN,"O",IB1))  . I $G(IBSEGM)'="" D ID^IBCEF2(IB1,IBSEGM)  Q  ; OUTPRVID(IBXIEN,IBXSAVE) ; Extract the outside provider or facility ids  ; into IBXSAVE array  ; Function returns 1 if person or 2 if facility ids or "" if neither  N Z,IBXDATA,IBPERSON,TAG  ;WCJ;11/1/2005 Extract the first 3 chars of Z instead.  S Z=$E($$PSPRV^IBCEF7(IBXIEN),1,3),IBPERSON=""  ;EJK 8/23/05 IB\*320 - CHANGED Z=101 TO Z=1010. Z WILL ALWAYS BE A 4 DIGIT #.   ; WCJ 11/1/2005 ; Removed EJK's change and added above change  I Z=111!(Z=101) S TAG=$S(Z=101:"OUTSIDE FAC PROVIDER INF",1:"CUR/OTH PROVIDER INFO") D F^IBCEF("N-ALL "\_TAG) S IBPERSON=$S('$E(Z,2):2,1:1)  Q IBPERSON  ; OUTPRV(IBREC,IBXIEN,IBXSAVE) ; Extract the outside provider or facility ids  ; into IBXSAVE array  ; Function returns 1 if person or 2 if facility ids or "" if neither  ; IBREC = the record whose ids should be returned  N IBPERSON,IBFRM,IBTYPE,IBFAC  I IBREC="SUB1"!(IBREC="OP6") D  . K IBXSAVE("PROVINF",IBXIEN),IBXSAVE("PROVINF\_FAC",IBXIEN)  . S IBPERSON=$$OUTPRVID(IBXIEN,.IBXSAVE),IBFAC=$S(IBPERSON=1:0,1:1)  E  D  . K IBXSAVE("PROVINF\_FAC",IBXIEN)  . D F^IBCEF("N-ALL OUTSIDE FAC PROVIDER INF")  . S IBPERSON=2,IBFAC=1  S IBFRM=$$FT^IBCEF(IBXIEN),IBFRM=$S(IBFRM=2:2,1:1)  S IBTYPE=$S(IBREC["SUB":"C",1:"O")  D CHCKSUB^IBCEF73(IBFRM,IBREC,IBFAC,IBTYPE,.IBXSAVE)  Q IBPERSON  ;  ;get IENs in file #36 for other insurances OTHINS(IB399,IBRES) ;  N IBFRMTYP,Z,Z1,Z2,Z4  S Z=$$COBN^IBCEF(IB399),Z0=0  F Z1=1:1:3 I Z1'=Z,$D(^DGCR(399,IB399,"I"\_Z1)) D  . S Z0=Z0+1  . ; MRD;IB\*2.0\*516 - Added HPID as second piece.  . S IBRES(Z0)=+$G(^DGCR(399,IB399,"I"\_Z1))\_U\_$P(^DGCR(399,IB399,"M1"),U,12+Z1)  . Q  Q  ;  ;get other insurance EDI ID NUMBERs OTHINSID(IB399,IBRES) ;insurance EDI   N IBFRMTYP,IBZ,Z0,Z1,Z4  **;JWS;IB\*2.0\*592;Dental form**  S IBFRMTYP=$$FT^IBCEF(IB399),IBFRMTYP=$S(IBFRMTYP=2:2,IBFRMTYP=3:1**,IBFRMTYP=7:4**,1:0)  S Z4=$S(IBFRMTYP=1:4,**IBFRMTYP=4:15**,1:2) ;UB - piece4,**DENTAL - piece 15**, 1500 or BOTH -piece 2,   D OTHINS(IB399,.IBZ)  S Z1=0  F Z0=1,2 I $G(IBZ(Z0)) D  . S IBRES(Z0)=$S($$MCRWNR^IBEFUNC(+IBZ(Z0)):$S(IBFRMTYP=1:"12M61",1:"SMTX1"),1:$P($G(^DIC(36,+IBZ(Z0),3)),U,Z4))  . ; MRD;IB\*2.0\*516 - Added HPID as second piece.  . S $P(IBRES(Z0),U,2)=$P(IBZ(Z0),U,2)  . Q  Q  ;  ;get other insurance addresses OTHINADR(IB399,IBRES,IBADDFLD) ;insurance EDI  N IBZ,Z0,Z1,Z4  D OTHINS(IB399,.IBZ)  S Z1=0  I IBADDFLD=18 D  Q  . F Z0=1:1:2 I $G(IBZ(Z0)) D  . . S IBRES(Z0)=$P($G(^DIC(36,+IBZ(Z0),.11)),U,1)  . . S IBRES(Z0)=$E(IBRES(Z0),1,55)  I IBADDFLD=18.9 D  Q  . F Z0=1:1:2 I $G(IBZ(Z0)) D  . . S IBRES(Z0)=$P($G(^DIC(36,+IBZ(Z0),.11)),U,1)  . . S Z4=$P($G(^DIC(36,+IBZ(Z0),.11)),U,2) S:Z4'="" IBRES(Z0)=IBRES(Z0)\_", "\_Z4  . . S Z4=$P($G(^DIC(36,+IBZ(Z0),.11)),U,3) S:Z4'="" IBRES(Z0)=IBRES(Z0)\_", "\_Z4  . . S Z4=$P($G(^DIC(36,+IBZ(Z0),.11)),U,4) S:Z4'="" IBRES(Z0)=IBRES(Z0)\_", "\_Z4  . . S Z4=$P($G(^DIC(5,+$P($G(^DIC(36,+IBZ(Z0),.11)),U,5),0)),U,2) S:Z4'="" IBRES(Z0)=IBRES(Z0)\_", "\_Z4  . . S Z4=$P($G(^DIC(36,+IBZ(Z0),.11)),U,6) S:Z4'="" IBRES(Z0)=IBRES(Z0)\_", "\_Z4  . . S IBRES(Z0)=$E(IBRES(Z0),1,157)  I IBADDFLD=19 D  Q  . F Z0=1:1:2 I $G(IBZ(Z0)) D  . . S IBRES(Z0)=$P($G(^DIC(36,+IBZ(Z0),.11)),U,2)  . . S IBRES(Z0)=IBRES(Z0)\_" "\_$P($G(^DIC(36,+IBZ(Z0),.11)),U,3)  . . S IBRES(Z0)=$E(IBRES(Z0),1,55)  I IBADDFLD=20 D  Q  . F Z0=1:1:2 I $G(IBZ(Z0)) D  . . S IBRES(Z0)=$P($G(^DIC(36,+IBZ(Z0),.11)),U,4)  . . S IBRES(Z0)=$E(IBRES(Z0),1,30)  I IBADDFLD=21 D  Q  . F Z0=1:1:2 I $G(IBZ(Z0)) D  . . S IBRES(Z0)=$P($G(^DIC(5,+$P($G(^DIC(36,+IBZ(Z0),.11)),U,5),0)),U,2)  . . S IBRES(Z0)=$E(IBRES(Z0),1,2)  I IBADDFLD=22 D  Q  . F Z0=1:1:2 I $G(IBZ(Z0)) D  . . S IBRES(Z0)=$P($G(^DIC(36,+IBZ(Z0),.11)),U,6)  . . S IBRES(Z0)=$E(IBRES(Z0),1,15)  Q  ; SFIDQ(IBXIEN,IBXSAVE,IBXDATA) ; Find the service facility id qualifier for  ; 837 record SUB2-5  ;IBXIEN = ien of 399  ;Pass by reference: IBXSAVE (input/output) IBXDATA (output)  N B,Z  K IBXSAVE("NVID")  D  ; protect IBXDATA  . N IBXDATA  . D F^IBCEF("N-RENDERING INSTITUTION")  . S:IBXDATA'="" IBXSAVE("IBFAC")=IBXDATA  I $P($G(IBXSAVE("IBFAC")),U,2)'=1 K IBXDATA Q  S Z=$$PSPRV^IBCEF7(IBXIEN)  ;WCJ 11/04/2005 If a Non-VA facility   I $E(Z) D  . S IBXSAVE("NVID")=$$NONVAID^IBCEF72(IBXIEN,.B,$E(Z),1)  .; S IBXSAVE("NVID")=$$NONVAID^IBCEF72(IBXIEN,.B,'$E(Z,2),1)  . S IBXDATA=$P("^34^24",U,$P(IBXSAVE("NVID"),U,2)+1)  ;S Z=$$PSPRV^IBCEF7(IBXIEN),IBXSAVE("NVID")=$$NONVAID^IBCEF72(IBXIEN,.B,'$E(Z,2),1),IBXDATA=24  Q  ; OTHP36(IBXIEN,IBZOUT) ;  N Z,Z0,Z1,IBZ  D F^IBCEF("N-ALL INSURANCE CO 837 ID","IBZ")  F Z=1,2,3 S IBZOUT(Z)=+$$POLICY^IBCEF(IBXIEN,1,$E("PST",Z))  Q  ;  ;---------SORT-----------  ;IBPRNUM - seq #  ;IBPRTYP - type of provider (use FUNCTION value from file 399, fld 222)  ;IB399 = ien file 399  ;IBSRC,IBDST - source,destination arrays  ;IBN - starting #  ;Output:  ; IBDST(1-primary/2-secondary provider,Provider type(FUNCTION),N)=  ; =provider/VARIABLEPTR^Insurance PTR #36 or NONE^ID type^ID^Form type^Care type^state ptr #5 for state license #  ; where N is numeration (1 for ID1, 2 for ID2, etc) GETSSN(IBPTR) ;look for SSN in #200 first and if not found then look at #355.9  ;if in file #200  I $P(IBPTR,";",2)="VA(200," Q $$SSN200^IBCEF73(IBPTR)  ;if in 355.93 then use 355.9  Q $$SSN3559^IBCEF73(IBPTR)  ;--  ;SSN3559  ;Find SSN from 355.9  ;Input:  ; Variable pointer to ^VA(200 or ^IBA(355.93  ;Output:  ; SSN or null  ; PADNDC(Z) ;PAD LEADING ZERO'S INTO A NON 5-4-2 FORMAT NDC NUMBER  ;Z IS ITERATION, ONLY PAD CURRENT NDC NUMBER  N NDC  S NDC=$P(IBXSAVE("OUTPT",Z,"RX"),"^",3)  Q:$L(NDC)=13  I $L(NDC)=14 D  Q  . S $P(NDC,"-",1)=$E($P(NDC,"-",1),2,$L($P(NDC,"-",1)))  . S $P(IBXSAVE("OUTPT",Z,"RX"),"^",3)=NDC  I $L($P(NDC,"-",1))'=5 S $P(NDC,"-",1)="0"\_$P(NDC,"-",1)  I $L($P(NDC,"-",2))'=4 S $P(NDC,"-",2)="0"\_$P(NDC,"-",2)  I $L($P(NDC,"-",3))'=2 S $P(NDC,"-",3)="0"\_$P(NDC,"-",3)  S $P(IBXSAVE("OUTPT",Z,"RX"),"^",3)=NDC  Q  ; | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCEP2A | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** |  | | | | | | | | |
| **Related Options** | None | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  |  | | | | |  | | | |
| **Data Dictionary (DD) References** |  | | | | | | | | |
| **Related Protocols** | None | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Current Logic** | | | | | | | | | |
| IBCEP2A ;ALB/TMP - EDI UTILITIES for provider ID ;25-APR-01  ;;2.0;INTEGRATED BILLING;\*\*137,232,320,348,349,400\*\*;21-MAR-94;Build 52  ;;Per VHA Directive 2004-038, this routine should not be modified.  ; ALT(IBPERF,IBSRC,IBALT,IBINS4,IBPTYP) ; set source level to next higher level   ; or set the alternate type and source if performing provider id  ; alternate type and source exist  ; IBPERF = 1 if performing provider id is requested  ; IBINS4 = '4' node of insurance co (file 36)  ; Pass IBPTYP by reference to get alternate provider id type  ; Pass IBALT by reference. Set to 1 if alternate id is to be used next  ;  I '$G(IBPERF)!($P(IBINS4,U,3)=1) S IBSRC=IBSRC-1 G ALTQ  S IBSRC=""  I '$G(IBALT),$P(IBINS4,U,3)=2,$P(IBINS4,U,10),$P(IBINS4,U,11) S IBALT=1,IBSRC=$P(IBINS4,U,11),IBPTYP=$P(IBINS4,U,10) S:IBPTYP="" IBPTYP=$P(IBINS4,U)  ; ALTQ Q IBSRC  ; IDSET(IBPTYP,IBINS4,IBPERF,IBSPEC,IBSRC,IBUP) ; set variables for provider id type search  N Z  S IBSPEC=$G(^IBE(355.97,+IBPTYP,1))  S Z=$S($G(IBPERF):2,$P(IBSPEC,U,5):6,$P(IBSPEC,U,6):4,1:2)  S IBSRC=$P(IBINS4,U,Z),IBUP=$P(IBINS4,U,$S(IBSRC:Z+1,1:0))  Q  ; CAREST(IBIFN) ; Return state file ien of state where care was performed  ; IBIFN = ien of bill in file 399  N STATE,IBU2,NVAFAC,IB0,EVDT,IBDIV,INST  S STATE=""  ;  ; non-VA care  S IBU2=$G(^DGCR(399,IBIFN,"U2"))  S NVAFAC=+$P(IBU2,U,10) ; non-VA facility  I NVAFAC S STATE=+$P($G(^IBA(355.93,NVAFAC,0)),U,7) G CARESTX  ;  ; VA care  S IB0=$G(^DGCR(399,IBIFN,0))  S EVDT=$P(IB0,U,3) ; claim event date  I 'EVDT S EVDT=DT                          ; - default today if undefined  S IBDIV=+$P(IB0,U,22) ; division ptr file 40.8  I 'IBDIV S IBDIV=$$PRIM^VASITE(EVDT) ; - default primary division as of event date  I IBDIV'>0 S IBDIV=$$PRIM^VASITE() ; - default main division as of today's date  S INST=+$$SITE^VASITE(EVDT,IBDIV) ; division institution ptr file 4  I INST'>0 S INST=+$$SITE^VASITE(DT,IBDIV) ; - default div as of today's date  I INST'>0 S INST=+$$SITE^VASITE ; - default main institution  S STATE=+$P($G(^DIC(4,INST,0)),U,2) ; state file ien from Institution file  ; CARESTX ;  Q STATE  ; RECALCA(IBIFN) ; Recalculate all performing provider id's on bill IBIFN  ; IBIFN = ien of bill entry (file 399)  N IBZ,IBZ0,IBX,IBP,IBSEQ,DA,DIE,DR,DIR,X,Y  ;  D EN^DDIOL("THIS FUNCTION HAS BEEN DISABLED",,"!") Q  ;  S DA(1)=IBIFN  I '$D(^XUSEC("IB SUPERVISOR",DUZ)) D EN^DDIOL("YOU ARE NOT AUTHORIZED TO PERFORM THIS FUNCTION",,"!")  S IBZ=0 F  S IBZ=$O(^DGCR(399,IBIFN,"PRV",IBZ)) Q:'IBZ  S IBP=$G(^(IBZ,0)) I $P(IBP,U,2)'="" D  . S DA=IBZ  . F IBZ0=5:1:7 Q:'$G(^DGCR(399,IBIFN,"I"\_(IBZ0-4))) D  .. S IBSEQ=$$EXPAND^IBTRE(399.0222,.01,+IBP)\_" "\_$P("PRIMARY^SECONDARY^TERTIARY",U,IBZ0-4)\_" PROVIDER ID "  .. S IBX=$$RECALC(.DA,IBZ0-4,$P(IBP,U,IBZ0),1)  .. I IBX'="",IBX=$P(IBP,U,IBZ0) D EN^DDIOL(IBSEQ\_"NO CHANGE NEEDED",,"!") Q  .. I IBX'="",IBX'=$P(IBP,U,IBZ0) D  Q  ... S DR=(IBZ0/100)\_"////"\_IBX,DIE="^DGCR(399,"\_DA(1)\_",""PRV""," D ^DIE  ... D EN^DDIOL(IBSEQ\_"CHANGED TO "\_IBX,,"!")  .. D EN^DDIOL(IBSEQ\_"NOT FOUND",,"!")  Q  ; RECALC(IBDA,IBSEQ,IBX,IBD) ; Recalculate id #, if possible - called  ; from input transforms in subfile 399.0222, fields .05-.07  ; IBDA = DA array of the provider entry (file 399.0222)  ; IBSEQ = the numeric COB sequence of the provider id (1-3)  ; IBX = the current value of the id in the subfile  ; IBD = flag that if set to 1 will suppress the display text  ;  N IBPN,IBZ  S IBPN=$P($G(^DGCR(399,IBDA(1),"PRV",IBDA,0)),U,2)  I IBPN="" D:'$G(IBD) EN^DDIOL(" CAN'T CALCULATE WITHOUT A PROVIDER NAME","","?0") G RECALCQ  S IBZ=$$GETID^IBCEP2(IBDA(1),2,IBPN,IBSEQ)  I IBZ="" D:'$G(IBD) EN^DDIOL(" ID COULD NOT BE DETERMINED","","?0") G RECALCQ  D:'$G(IBD) EN^DDIOL(" "\_IBZ\_$S(IBZ'=IBX:"",1:" (no change)"),"","?0")  S IBX=IBZ  ; RECALCQ Q IBX  ; PERFPRV(IBIFN) ; Returns the variable pointer of the 'performing provider'  ; (attending or rendering) for a bill IBIFN  N IBP,IBPT,IBQ,Z  S Z=$$FT^IBCEF(IBIFN),IBPT=$S(Z=2:3,Z=3:4,1:0)  D GETPRV^IBCEU(IBIFN,IBPT,.IBP)  Q $P($G(IBP(IBPT,1)),U,3)  ; INSPAR(IBIFN,SEQ) ;  N Z,Z4,Z0  Q:$G(X)'="??"  S:'$G(SEQ) SEQ=$$COBN^IBCEF(IBIFN)  S Z=+$G(^DGCR(399,IBIFN,"I"\_SEQ)),Z4=$G(^DIC(36,Z,4))  I Z D  . D EN^DDIOL(">"\_$J("",20)\_"-- PERFORMING PROVIDER ID PARAMETERS --",,"!")  . S Z0=$P(" PRIMARY^SECONDARY^ TERTIARY",U,SEQ)\_" INSURANCE: "\_$P($G(^DIC(36,Z,0)),U)  . D EN^DDIOL(">"\_$J("",(80-$L(Z0))\2)\_Z0,,"!")  . D EN^DDIOL("> Secondary Perf Prov ID Type (1500): "\_$$EXPAND^IBTRE(36,4.01,+Z4),,"!")  . D EN^DDIOL("> Secondary Perf Prov ID Type (UB04): "\_$$EXPAND^IBTRE(36,4.02,$P(Z4,U,2)),,"!")  . D EN^DDIOL("> Secondary Perf Prov IDs Required: "\_$$EXPAND^IBTRE(36,4.03,$P(Z4,U,3)),,"!")  . D EN^DDIOL(" ",,"!")  Q  ; GETTYP(IBXIEN,IBCOBN,IBFUNC) ; Function returns provider id type for insurance co  ; with COB of IBCOBN on claim ien IBXIEN in first ^ pc and 1 in second  ; ^ piece if the id is required  ;   ; IBFUNC=1:REFERRING;2:OPERATING;3:RENDERING;4:ATTENDING;5:SUPERVISING;9:OTHER  ;   N A,R,Z,Z0  S A="",R=0  S:'$G(IBCOBN)!(IBCOBN>3) IBCOBN=$$COBN^IBCEF(IBXIEN)  S Z=+$G(^DGCR(399,IBXIEN,"I"\_+IBCOBN))  I Z D  . S Z0=$$FT^IBCEF(IBXIEN)  . S A=+$P($G(^DIC(36,Z,4)),U,$S(Z0=2&($G(IBFUNC)=1):4,Z0=2:1,1:2))  . I A,$G(IBFUNC)'=1 S R=$P($G(^DIC(36,Z,4)),U,3),R=$S('R:0,R=3:1,R=1:Z0=2,R=2:Z0=3,1:0)  . I A,$G(IBFUNC)=1 S R=+$P($G(^DIC(36,Z,4)),U,5),R=$S('R:0,Z0'=2:0,1:1)  Q A\_U\_R  ; UNIQ1(IBIFN,IBINS,IBPTYP,IBPROV,IBUNIT,IBCU,IBT) ; Match most-least specific  ; \*\*\* SEE PARAMETER DEFINITIONS IN IBCEP3 \*\*\*  ;  ; Start in file 355.9 (Specific Provider)  ; IBPROV = (variable pointer syntax) provider on bill IBIFN  ;  N Q,Z0,Z1,Z2,IBID,IBX  S IBID=""  S IBX=$P($G(^IBA(355.9,+IBCU,0)),U,3) S:"0"[IBX IBX="\*N/A\*"  S Z0=$$FT^IBCEF(IBIFN),Z0=$S(Z0=2:2,Z0=3:1,1:0),Z1=$$INPAT^IBCEF(IBIFN) S:'Z1 Z1=2 S Z2=$$ISRX^IBCEF1(IBIFN)  ;  ; Match all elements  F Q=$S(Z2:3,1:Z1),$S(Z2:Z1,1:"") I Q'="",$D(^IBA(355.9,"AUNIQ",IBPROV,IBINS,IBX,Z0,Q,IBPTYP,IBCU)) S IBID=$P($G(^IBA(355.9,IBCU,0)),U,7),$P(IBT,U,2,3)=(IBCU\_U\_355.9) Q  G:IBID'="" UNIQ1Q  ;  ; Match both form types,specific I/O element  F Q=$S(Z2:3,1:Z1),$S(Z2:Z1,1:"") I Q'="",$D(^IBA(355.9,"AUNIQ",IBPROV,IBINS,IBX,0,Q,IBPTYP,IBCU)) S IBID=$P($G(^IBA(355.9,IBCU,0)),U,7),$P(IBT,U,2,3)=(IBCU\_U\_355.9) Q  G:IBID'="" UNIQ1Q  ;  ; Match specific form type, both I/O element or Rx  F Q=$S(Z2:3,1:0),$S(Z2:0,1:"") I Q'="",$D(^IBA(355.9,"AUNIQ",IBPROV,IBINS,IBX,Z0,Q,IBPTYP,IBCU)) S IBID=$P($G(^IBA(355.9,IBCU,0)),U,7),$P(IBT,U,2,3)=(IBCU\_U\_355.9) Q  G:IBID'="" UNIQ1Q  ;  ; Match both form types, both I/O element or Rx  F Q=$S(Z2:3,1:0),$S(Z2:0,1:"") I Q'="",$D(^IBA(355.9,"AUNIQ",IBPROV,IBINS,IBX,0,Q,IBPTYP,IBCU)) S IBID=$P($G(^IBA(355.9,IBCU,0)),U,7),$P(IBT,U,2,3)=(IBCU\_U\_355.9) Q  ; UNIQ1Q Q IBID  ; UNIQ2(IBIFN,IBINS,IBPTYP,IBUNIT,IBCU,IBT) ; Match on most-least specific  ; \*\*\* SEE PARAMETER DEFINITIONS IN IBCEP3 \*\*\*  ;  ; Start in file 355.91 (Specific Insurance)  ;  N Q,Z0,Z1,Z2,IBID,IBX  S IBID="" S:"0"[$G(IBUNIT) IBUNIT="\*N/A\*"  S Z0=$$FT^IBCEF(IBIFN),Z0=$S(Z0=2:2,Z0=3:1,1:0),Z1=$$INPAT^IBCEF(IBIFN) S:'Z1 Z1=2 S Z2=$$ISRX^IBCEF1(IBIFN)  ;  ; Match all elements  F Q=$S(Z2:3,1:Z1),$S(Z2:Z1,1:"") I Q'="",$D(^IBA(355.91,"AUNIQ",IBINS,IBUNIT,Z0,Q,IBPTYP,IBCU)) S IBID=$P($G(^IBA(355.91,IBCU,0)),U,7),$P(IBT,U,2,3)=(IBCU\_U\_355.91) Q  G:IBID'="" UNIQ2Q  ;  ; Match both form types,specific I/O element  F Q=$S(Z2:3,1:Z1),$S(Z2:Z1,1:"") I Q'="",$D(^IBA(355.91,"AUNIQ",IBINS,IBUNIT,0,Q,IBPTYP,IBCU)) S IBID=$P($G(^IBA(355.91,IBCU,0)),U,7),$P(IBT,U,2,3)=(IBCU\_U\_355.91) Q  G:IBID'="" UNIQ2Q  ;  ; Match specific form type, both I/O element or Rx  F Q=$S(Z2:3,1:0),$S(Z2:0,1:"") I Q'="",$D(^IBA(355.91,"AUNIQ",IBINS,IBUNIT,Z0,Q,IBPTYP,IBCU)) S IBID=$P($G(^IBA(355.91,IBCU,0)),U,7),$P(IBT,U,2,3)=(IBCU\_U\_355.91) Q  G:IBID'="" UNIQ2Q  ;  ; Match both form types, both I/O elements or Rx  F Q=$S(Z2:3,1:0),$S(Z2:0,1:"") I Q'="",$D(^IBA(355.91,"AUNIQ",IBINS,IBUNIT,0,Q,IBPTYP,IBCU)) S IBID=$P($G(^IBA(355.91,IBCU,0)),U,7),$P(IBT,U,2,3)=(IBCU\_U\_355.91) Q  ; UNIQ2Q Q IBID  ; | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCEP2A ;ALB/TMP - EDI UTILITIES for provider ID ;25-APR-01  ;;2.0;INTEGRATED BILLING;\*\*137,232,320,348,349,400,**592**\*\*;21-MAR-94;Build 52  ;;Per VHA Directive 2004-038, this routine should not be modified.  ; ALT(IBPERF,IBSRC,IBALT,IBINS4,IBPTYP) ; set source level to next higher level   ; or set the alternate type and source if performing provider id  ; alternate type and source exist  ; IBPERF = 1 if performing provider id is requested  ; IBINS4 = '4' node of insurance co (file 36)  ; Pass IBPTYP by reference to get alternate provider id type  ; Pass IBALT by reference. Set to 1 if alternate id is to be used next  ;  I '$G(IBPERF)!($P(IBINS4,U,3)=1) S IBSRC=IBSRC-1 G ALTQ  S IBSRC=""  I '$G(IBALT),$P(IBINS4,U,3)=2,$P(IBINS4,U,10),$P(IBINS4,U,11) S IBALT=1,IBSRC=$P(IBINS4,U,11),IBPTYP=$P(IBINS4,U,10) S:IBPTYP="" IBPTYP=$P(IBINS4,U)  ; ALTQ Q IBSRC  ; IDSET(IBPTYP,IBINS4,IBPERF,IBSPEC,IBSRC,IBUP) ; set variables for provider id type search  N Z  S IBSPEC=$G(^IBE(355.97,+IBPTYP,1))  S Z=$S($G(IBPERF):2,$P(IBSPEC,U,5):6,$P(IBSPEC,U,6):4,1:2)  S IBSRC=$P(IBINS4,U,Z),IBUP=$P(IBINS4,U,$S(IBSRC:Z+1,1:0))  Q  ; CAREST(IBIFN) ; Return state file ien of state where care was performed  ; IBIFN = ien of bill in file 399  N STATE,IBU2,NVAFAC,IB0,EVDT,IBDIV,INST  S STATE=""  ;  ; non-VA care  S IBU2=$G(^DGCR(399,IBIFN,"U2"))  S NVAFAC=+$P(IBU2,U,10) ; non-VA facility  I NVAFAC S STATE=+$P($G(^IBA(355.93,NVAFAC,0)),U,7) G CARESTX  ;  ; VA care  S IB0=$G(^DGCR(399,IBIFN,0))  S EVDT=$P(IB0,U,3) ; claim event date  I 'EVDT S EVDT=DT                          ; - default today if undefined  S IBDIV=+$P(IB0,U,22) ; division ptr file 40.8  I 'IBDIV S IBDIV=$$PRIM^VASITE(EVDT) ; - default primary division as of event date  I IBDIV'>0 S IBDIV=$$PRIM^VASITE() ; - default main division as of today's date  S INST=+$$SITE^VASITE(EVDT,IBDIV) ; division institution ptr file 4  I INST'>0 S INST=+$$SITE^VASITE(DT,IBDIV) ; - default div as of today's date  I INST'>0 S INST=+$$SITE^VASITE ; - default main institution  S STATE=+$P($G(^DIC(4,INST,0)),U,2) ; state file ien from Institution file  ; CARESTX ;  Q STATE  ; RECALCA(IBIFN) ; Recalculate all performing provider id's on bill IBIFN  ; IBIFN = ien of bill entry (file 399)  N IBZ,IBZ0,IBX,IBP,IBSEQ,DA,DIE,DR,DIR,X,Y  ;  D EN^DDIOL("THIS FUNCTION HAS BEEN DISABLED",,"!") Q  ;  S DA(1)=IBIFN  I '$D(^XUSEC("IB SUPERVISOR",DUZ)) D EN^DDIOL("YOU ARE NOT AUTHORIZED TO PERFORM THIS FUNCTION",,"!")  S IBZ=0 F  S IBZ=$O(^DGCR(399,IBIFN,"PRV",IBZ)) Q:'IBZ  S IBP=$G(^(IBZ,0)) I $P(IBP,U,2)'="" D  . S DA=IBZ  . F IBZ0=5:1:7 Q:'$G(^DGCR(399,IBIFN,"I"\_(IBZ0-4))) D  .. S IBSEQ=$$EXPAND^IBTRE(399.0222,.01,+IBP)\_" "\_$P("PRIMARY^SECONDARY^TERTIARY",U,IBZ0-4)\_" PROVIDER ID "  .. S IBX=$$RECALC(.DA,IBZ0-4,$P(IBP,U,IBZ0),1)  .. I IBX'="",IBX=$P(IBP,U,IBZ0) D EN^DDIOL(IBSEQ\_"NO CHANGE NEEDED",,"!") Q  .. I IBX'="",IBX'=$P(IBP,U,IBZ0) D  Q  ... S DR=(IBZ0/100)\_"////"\_IBX,DIE="^DGCR(399,"\_DA(1)\_",""PRV""," D ^DIE  ... D EN^DDIOL(IBSEQ\_"CHANGED TO "\_IBX,,"!")  .. D EN^DDIOL(IBSEQ\_"NOT FOUND",,"!")  Q  ; RECALC(IBDA,IBSEQ,IBX,IBD) ; Recalculate id #, if possible - called  ; from input transforms in subfile 399.0222, fields .05-.07  ; IBDA = DA array of the provider entry (file 399.0222)  ; IBSEQ = the numeric COB sequence of the provider id (1-3)  ; IBX = the current value of the id in the subfile  ; IBD = flag that if set to 1 will suppress the display text  ;  N IBPN,IBZ  S IBPN=$P($G(^DGCR(399,IBDA(1),"PRV",IBDA,0)),U,2)  I IBPN="" D:'$G(IBD) EN^DDIOL(" CAN'T CALCULATE WITHOUT A PROVIDER NAME","","?0") G RECALCQ  S IBZ=$$GETID^IBCEP2(IBDA(1),2,IBPN,IBSEQ)  I IBZ="" D:'$G(IBD) EN^DDIOL(" ID COULD NOT BE DETERMINED","","?0") G RECALCQ  D:'$G(IBD) EN^DDIOL(" "\_IBZ\_$S(IBZ'=IBX:"",1:" (no change)"),"","?0")  S IBX=IBZ  ; RECALCQ Q IBX  ; PERFPRV(IBIFN) ; Returns the variable pointer of the 'performing provider'  ; (attending or rendering) for a bill IBIFN  N IBP,IBPT,IBQ,Z  S Z=$$FT^IBCEF(IBIFN),IBPT=$S(Z=2:3,Z=3:4,1:0)  D GETPRV^IBCEU(IBIFN,IBPT,.IBP)  Q $P($G(IBP(IBPT,1)),U,3)  ; INSPAR(IBIFN,SEQ) ;  N Z,Z4,Z0  Q:$G(X)'="??"  S:'$G(SEQ) SEQ=$$COBN^IBCEF(IBIFN)  S Z=+$G(^DGCR(399,IBIFN,"I"\_SEQ)),Z4=$G(^DIC(36,Z,4))  I Z D  . D EN^DDIOL(">"\_$J("",20)\_"-- PERFORMING PROVIDER ID PARAMETERS --",,"!")  . S Z0=$P(" PRIMARY^SECONDARY^ TERTIARY",U,SEQ)\_" INSURANCE: "\_$P($G(^DIC(36,Z,0)),U)  . D EN^DDIOL(">"\_$J("",(80-$L(Z0))\2)\_Z0,,"!")  . D EN^DDIOL("> Secondary Perf Prov ID Type (1500): "\_$$EXPAND^IBTRE(36,4.01,+Z4),,"!")  **. ;JWS;IB\*2.0\*592;form J430D**  **. D EN^DDIOL("> Secondary Perf Prov ID Type (J430D): "\_$$EXPAND^IBTRE(36,4.14,$P(Z4,U,14)),,"!")**  . D EN^DDIOL("> Secondary Perf Prov ID Type (UB04): "\_$$EXPAND^IBTRE(36,4.02,$P(Z4,U,2)),,"!")  . D EN^DDIOL("> Secondary Perf Prov IDs Required: "\_$$EXPAND^IBTRE(36,4.03,$P(Z4,U,3)),,"!")  . D EN^DDIOL(" ",,"!")  Q  ; GETTYP(IBXIEN,IBCOBN,IBFUNC) ; Function returns provider id type for insurance co  ; with COB of IBCOBN on claim ien IBXIEN in first ^ pc and 1 in second  ; ^ piece if the id is required  ;   ; IBFUNC=1:REFERRING;2:OPERATING;3:RENDERING;4:ATTENDING;5:SUPERVISING;6:ASSISTANT SURGEON;9:OTHER  ;   N A,R,Z,Z0  S A="",R=0  S:'$G(IBCOBN)!(IBCOBN>3) IBCOBN=$$COBN^IBCEF(IBXIEN)  S Z=+$G(^DGCR(399,IBXIEN,"I"\_+IBCOBN))  I Z D  . S Z0=$$FT^IBCEF(IBXIEN)  **. ;JRA IB\*2.0\*592 Treat Dental Form 7 same as CMS-1500**  . **;**S A=+$P($G(^DIC(36,Z,4)),U,$S(Z0=2&($G(IBFUNC)=1):4,Z0=2:1,1:2)) **;JRA IB\*2.0\*592 ';'**  . **;I** A,$G(IBFUNC)'=1 S R=$P($G(^DIC(36,Z,4)),U,3),R=$S('R:0,R=3:1,R=1:Z0=2,R=2:Z0=3,1:0) **;JRA IB\*2.0\*592 ';'**  . **;I** A,$G(IBFUNC)=1 S R=+$P($G(^DIC(36,Z,4)),U,5),R=$S('R:0,Z0'=2:0,1:1) **;JRA IB\*2.0\*592 ';'**  . S A=+$P($G(^DIC(36,Z,4)),U,$S((Z0=2)&($G(IBFUNC)=1):4**,(Z0=7)&($G(IBFUNC)=1):15**,Z0=2:1,**Z0=7:14**,1:2)) **;JRA IB\*2.0\*592;JWS**  **. ;JWS;IB\*2.0\*592;Assistant Surgeon default  . I A,$G(IBFUNC)=6 S A=17**  . I A,$G(IBFUNC)'=1 S R=$P($G(^DIC(36,Z,4)),U,3),R=$S('R:0,R=3:**Z0'=7,**R=1:Z0=2,R=2:Z0=3,**R=4:Z0=7**,**R=5:Z0'=3,R=6:1**,1:0) **;JRA IB\*2.0\*592**  . I A,$G(IBFUNC)=1 S R=+$P($G(^DIC(36,Z,4)),U,5),R=$S('R:0**,Z0=3:0**,1:1) **;JRA IB\*2.0\*592**  Q A\_U\_R  ; UNIQ1(IBIFN,IBINS,IBPTYP,IBPROV,IBUNIT,IBCU,IBT) ; Match most-least specific  ; \*\*\* SEE PARAMETER DEFINITIONS IN IBCEP3 \*\*\*  ;  ; Start in file 355.9 (Specific Provider)  ; IBPROV = (variable pointer syntax) provider on bill IBIFN  ;  N Q,Z0,Z1,Z2,IBID,IBX  S IBID=""  S IBX=$P($G(^IBA(355.9,+IBCU,0)),U,3) S:"0"[IBX IBX="\*N/A\*"  **;JRA IB\*2.0\*592 new Dental Form 7**  **;**S Z0=$$FT^IBCEF(IBIFN),Z0=$S(Z0=2:2,Z0=3:1,1:0),Z1=$$INPAT^IBCEF(IBIFN) S:'Z1 Z1=2 S Z2=$$ISRX^IBCEF1(IBIFN) **;JRA IB\*2.0\*592 ';'**  S Z0=$$FT^IBCEF(IBIFN),Z0=$S(Z0=2:2,Z0=3:1,**Z0=7:4**,1:0),Z1=$$INPAT^IBCEF(IBIFN) S:'Z1 Z1=2 S Z2=$$ISRX^IBCEF1(IBIFN) **;JWS;JRA IB\*2.0\*592**  ;  ; Match all elements  F Q=$S(Z2:3,1:Z1),$S(Z2:Z1,1:"") I Q'="",$D(^IBA(355.9,"AUNIQ",IBPROV,IBINS,IBX,Z0,Q,IBPTYP,IBCU)) S IBID=$P($G(^IBA(355.9,IBCU,0)),U,7),$P(IBT,U,2,3)=(IBCU\_U\_355.9) Q  G:IBID'="" UNIQ1Q  ;  ; Match both form types,specific I/O element  F Q=$S(Z2:3,1:Z1),$S(Z2:Z1,1:"") I Q'="",$D(^IBA(355.9,"AUNIQ",IBPROV,IBINS,IBX,0,Q,IBPTYP,IBCU)) S IBID=$P($G(^IBA(355.9,IBCU,0)),U,7),$P(IBT,U,2,3)=(IBCU\_U\_355.9) Q  G:IBID'="" UNIQ1Q  ;  ; Match specific form type, both I/O element or Rx  F Q=$S(Z2:3,1:0),$S(Z2:0,1:"") I Q'="",$D(^IBA(355.9,"AUNIQ",IBPROV,IBINS,IBX,Z0,Q,IBPTYP,IBCU)) S IBID=$P($G(^IBA(355.9,IBCU,0)),U,7),$P(IBT,U,2,3)=(IBCU\_U\_355.9) Q  G:IBID'="" UNIQ1Q  ;  ; Match both form types, both I/O element or Rx  F Q=$S(Z2:3,1:0),$S(Z2:0,1:"") I Q'="",$D(^IBA(355.9,"AUNIQ",IBPROV,IBINS,IBX,0,Q,IBPTYP,IBCU)) S IBID=$P($G(^IBA(355.9,IBCU,0)),U,7),$P(IBT,U,2,3)=(IBCU\_U\_355.9) Q  ; UNIQ1Q Q IBID  ; UNIQ2(IBIFN,IBINS,IBPTYP,IBUNIT,IBCU,IBT) ; Match on most-least specific  ; \*\*\* SEE PARAMETER DEFINITIONS IN IBCEP3 \*\*\*  ;  ; Start in file 355.91 (Specific Insurance)  ;  N Q,Z0,Z1,Z2,IBID,IBX  S IBID="" S:"0"[$G(IBUNIT) IBUNIT="\*N/A\*"  **;JRA IB\*2.0\*592 Dental Form 7**  **;**S Z0=$$FT^IBCEF(IBIFN),Z0=$S(Z0=2:2,Z0=3:1,1:0),Z1=$$INPAT^IBCEF(IBIFN) S:'Z1 Z1=2 S Z2=$$ISRX^IBCEF1(IBIFN) **;JRA IB\*2.0\*592 ';'**  S Z0=$$FT^IBCEF(IBIFN),Z0=$S(Z0=2:2,Z0=3:1,**Z0=7:4**,1:0),Z1=$$INPAT^IBCEF(IBIFN) S:'Z1 Z1=2 S Z2=$$ISRX^IBCEF1(IBIFN) **;JWS;JRA IB\*2.0\*592**  ;  ; Match all elements  F Q=$S(Z2:3,1:Z1),$S(Z2:Z1,1:"") I Q'="",$D(^IBA(355.91,"AUNIQ",IBINS,IBUNIT,Z0,Q,IBPTYP,IBCU)) S IBID=$P($G(^IBA(355.91,IBCU,0)),U,7),$P(IBT,U,2,3)=(IBCU\_U\_355.91) Q  G:IBID'="" UNIQ2Q  ;  ; Match both form types,specific I/O element  F Q=$S(Z2:3,1:Z1),$S(Z2:Z1,1:"") I Q'="",$D(^IBA(355.91,"AUNIQ",IBINS,IBUNIT,0,Q,IBPTYP,IBCU)) S IBID=$P($G(^IBA(355.91,IBCU,0)),U,7),$P(IBT,U,2,3)=(IBCU\_U\_355.91) Q  G:IBID'="" UNIQ2Q  ;  ; Match specific form type, both I/O element or Rx  F Q=$S(Z2:3,1:0),$S(Z2:0,1:"") I Q'="",$D(^IBA(355.91,"AUNIQ",IBINS,IBUNIT,Z0,Q,IBPTYP,IBCU)) S IBID=$P($G(^IBA(355.91,IBCU,0)),U,7),$P(IBT,U,2,3)=(IBCU\_U\_355.91) Q  G:IBID'="" UNIQ2Q  ;  ; Match both form types, both I/O elements or Rx  F Q=$S(Z2:3,1:0),$S(Z2:0,1:"") I Q'="",$D(^IBA(355.91,"AUNIQ",IBINS,IBUNIT,0,Q,IBPTYP,IBCU)) S IBID=$P($G(^IBA(355.91,IBCU,0)),U,7),$P(IBT,U,2,3)=(IBCU\_U\_355.91) Q  ; UNIQ2Q Q IBID  ; | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCEF21 | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** |  | | | | | | | | |
| **Related Options** | None | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  |  | | | | |  | | | |
| **Data Dictionary (DD) References** |  | | | | | | | | |
| **Related Protocols** | None | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Current Logic** | | | | | | | | | |
| IBCEF21 ;ALB/TMP - FORMATTER SPECIFIC BILL FUNCTIONS CONTINUED ; 3/9/11 1:12pm  ;;2.0;INTEGRATED BILLING;\*\*51,296,371,389,448,516\*\*;21-MAR-94;Build 123  ;;Per VA Directive 6402, this routine should not be modified.  ; COID(IBIFN) ; Claim office ID  N IBCOID,IBCOID1,IBIN  S IBIN=$$CURR^IBCEF2(IBIFN),IBCOID1="",IBCOID=$P($$ADDRESS^IBCNSC0(IBIN,.11,5),U,11)  ;  I IBIN D  . I $D(^IBA(364.2,"C",IBIFN)) S IBCOID1=$P($$ADDRESS^IBCNSC0(IBIN,.18,5),U,11) Q  ;Rx  . I $P($G(^DGCR(399,IBIFN,0)),U,5)<3 S IBCOID1=$P($$ADDRESS^IBCNSC0(IBIN,.12,5),U,11) Q  ;Inpt  . I $P($G(^DGCR(399,IBIFN,0)),U,5)'<3 S IBCOID1=$P($$ADDRESS^IBCNSC0(IBIN,.16,5),U,11) Q  ;Outpt  ;  Q $S(IBCOID1'="":IBCOID1,1:IBCOID)  ; ESGHPST(IBIFN,COB) ; return insureds employment status if the bill policy defined by COB is an Employer Sponsored Group Health Plan  ; ESGHP FLAG (2.312,2.1) ^ the employment status (2.312,2.11)  ;  N PPOL,DFN,X,Y S Y=""  S PPOL=$$PPOL^IBCEF2($G(IBIFN),$G(COB)),DFN=$P($G(^DGCR(399,+$G(IBIFN),0)),U,2)  I +PPOL,+DFN S X=$G(^DPT(DFN,.312,+PPOL,2)) S Y=+$P(X,U,10)\_U\_$P(X,U,11)  Q Y  ; ESGHPNL(IBIFN,COB) ; return employer name and location if the bill policy defined by COB is an Employer Sponsored Group Health Plan  ; ESGHP FLAG (2.312,2.1) ^ employer name (2.312,2.015) ^ employer city (2.312,2.05)   ; ^ employer state abbr (2.312,2.06) ^ employer state ifn (2.312,2.06)  ;  N PPOL,DFN,X,Y S Y=""  S PPOL=$$PPOL^IBCEF2($G(IBIFN),$G(COB)),DFN=$P($G(^DGCR(399,+$G(IBIFN),0)),U,2)  I +PPOL,+DFN S X=$G(^DPT(DFN,.312,+PPOL,2)) S Y=+$P(X,U,10)\_U\_$P(X,U,9)\_U\_$P(X,U,5)\_U\_$P($G(^DIC(5,+$P(X,U,6),0)),U,2)\_U\_$P(X,U,6)  Q Y  ; REMARKS(IBIFN) ; Compile array of bill remarks  ;IBIFN = bill ien  N Z,Z0,Z1,IBARRAY,IBSM  S Z=0  ;S:$P($G(^DGCR(399,IBIFN,"U1")),U,2) Z=Z+1,Z0=$P(^("U1"),U,2),IBXDATA(Z)="OFFSET AMOUNT: "\_"$"\_+$P(Z0,".")\_"."\_$E($P(Z0,".",2)\_"00",1,2)  S:$P($G(^DGCR(399,IBIFN,"U1")),U,8)'="" Z=Z+1,IBXDATA(Z)=$P(^("U1"),U,8) ;Bill comment on bill  S Z0=$G(^DGCR(399,IBIFN,0)),Z1=$G(^DGCR(399.3,+$P(Z0,U,7),0))  D SET^IBCSC5B(IBIFN,.IBARRAY)  I $P($G(IBARRAY),U,2) D  ;Prosthetics  . S Z0=0 F  S Z0=$O(IBARRAY(Z0)) Q:Z0=""  S Z1=0 F  S Z1=$O(IBARRAY(Z0,Z1)) Q:'Z1  S Z=Z+1,IBXDATA(Z)="Prosthetic: "\_$E($$PINB^IBCSC5B(+IBARRAY(Z0,Z1)),1,39)\_" "\_$E(Z0,4,5)\_"/"\_$E(Z0,6,7)\_"/"\_$E(Z0,1,2)  Q  ; CREM(IBIFN) ; Compile array of bill remarks common to every bill  ;IBIFN = bill ien  N Z  S Z=0  S:$P($G(^IBE(350.9,1,1)),U,4)'="" Z=Z+1,IBXDATA(Z)=$P(^(1),U,4) ;Site specific 'every bill' comment  Q  ; ADMDT(IBIFN,NOOUTCK) ; Calculate admission/start of care date/time  ; IBIFN = bill ien  ; NOOUTCK = flag that will:  ; (1) no check for inpt episode overlap for outpt  ; (0 or null) performs check for inpt episode overlap for outpt  ;   ; Returns IBXDATA = fileman date format  N Z,Z0,Z1  S Z=$G(^DGCR(399,IBIFN,0)),Z1=$P($G(^("U")),U,20),Z0=$$INPAT^IBCEF(IBIFN,1)  S IBXDATA=$S(Z0&$P(Z,U,8):$P($G(^DGPT(+$P(Z,U,8),0)),U,2),1:"")  S:'IBXDATA IBXDATA=$P(Z,U,3)\_$S(Z0&(Z1<25):"."\_$E("0",$L(Z1))\_Z1,1:"")  ; Check to see if outpt episode (date in event date) overlaps inpt  ; episode - use admit date if it does  I 'Z0,IBXDATA,'$G(NOOUTCK) D  . N VAINDT,VAIN,DFN  . S VAINDT=IBXDATA,DFN=$P($G(^DGCR(399,IBIFN,0)),U,2)  . D INP^VADPT S IBXDATA=+VAIN(7) S:'IBXDATA IBXDATA=""  I 'IBXDATA,'Z0 S IBXDATA=$$SERVDT^IBCEF(IBIFN,,2)  Q  ; DISDT(IBIFN) ; Calculate discharge date  ; IBIFN = bill ien  N Z,Z0  S Z=$$INPAT^IBCEF(IBIFN,1),Z0=$G(^DGCR(399,IBIFN,0))  I Z S IBXDATA=+$G(^DGPT(+$P(Z0,U,8),70)) S:'IBXDATA IBXDATA=$P(Z0,U,16)  I 'Z N VAINDT,VAIN,DFN S DFN=$P($G(^DGCR(399,IBIFN,0)),U,2) D INP^VADPT I VAIN(1) S IBXDATA=+$G(^DGPM(+$P($G(^DGPM(+VAIN(1),0)),U,17),0))  Q  ; INSSECID(IBIFN,TYPE,SEQ) ; Extract subscriber and patient prim/sec ID's  ; IBIFN required  ; TYPE is either "PAT" or "SUB" to indicate we need to extract either  ; patient or subscriber ID information. Default="SUB".  ; SEQ is the insurance sequence# (1,2,3). Default is current ins seq#.  ;  ; Output:  ; Function returns an 8-piece string as follows.  ; [1] primary qualifier  ; [2] primary ID  ; [3] secondary qual(1)  ; [4] secondary ID(1)  ; [5] secondary qual(2)  ; [6] secondary ID(2)  ; [7] secondary qual(3)  ; [8] secondary ID(3)  ;  NEW DATA,DFN,POL,IB0,IB5,REL  S DATA=""  S IBIFN=+$G(IBIFN) I 'IBIFN G INSSX  I $G(TYPE)="" S TYPE="SUB"               ; default type of ID's to get  I '$F(".PAT.SUB.","."\_TYPE\_".") G INSSX  I '$G(SEQ) S SEQ=$$COBN^IBCEF(IBIFN) ; default current ins seq#  I '$F(".1.2.3.","."\_SEQ\_".") G INSSX  S DFN=+$P($G(^DGCR(399,IBIFN,0)),U,2) I 'DFN G INSSX  S POL=+$P($G(^DGCR(399,IBIFN,"M")),U,SEQ+11) I 'POL G INSSX  ;IB\*2.0\*516/baa - Use HIPAA compliant fields  ;S IB0=$G(^DPT(DFN,.312,POL,0)) I IB0="" G INSSX ;516 - baa  S IB0=$$ZND^IBCNS1(DFN,POL) I IB0="" G INSSX  ;516 - baa  S IB5=$G(^DPT(DFN,.312,POL,5))  S REL=+$P(IB0,U,16) ; pat rel to insured  S $P(DATA,U,1)="MI"  S $P(DATA,U,2)=$P(IB0,U,2) ; subscriber primary ID   S $P(DATA,U,3,8)=$P(IB5,U,2,7) ; subscriber secondary data  I TYPE="PAT",REL'=1 D  . S $P(DATA,U,2)=$P(IB5,U,1) ; patient primary ID  . S $P(DATA,U,3,8)=$P(IB5,U,8,13) ; patient secondary data  . Q  ;  S DATA=$$SCRUB(DATA) ; scrub the data INSSX ;  Q DATA  ; SCRUB(DATA) ; Scrub the 8-piece string gathered above  NEW PCE  ;  ; make sure you can't have an ID without a qualifier or a qualifier  ; without an ID. Check all 4 pairs.  F PCE=1,3,5,7 D  . I $P(DATA,U,PCE)'="",$P(DATA,U,PCE+1)'="" Q  . S ($P(DATA,U,PCE),$P(DATA,U,PCE+1))=""  . Q  ;  ; fill in secondary gaps. If Set1 and Set2 are blank, but Set3 exists  ; then move Set3 to Set1 and delete Set3.  I $P(DATA,U,3)="",$P(DATA,U,5)="",$P(DATA,U,7)'="" D  . S $P(DATA,U,3)=$P(DATA,U,7),$P(DATA,U,4)=$P(DATA,U,8)  . S ($P(DATA,U,7),$P(DATA,U,8))=""  . Q  ;  ; fill in secondary gaps more generically.  ; If Set(n) is blank, but Set(n+1) exists, then move it up.  F PCE=3,5 D  . I $P(DATA,U,PCE)="",$P(DATA,U,PCE+2)'="" D  .. S $P(DATA,U,PCE)=$P(DATA,U,PCE+2)  .. S $P(DATA,U,PCE+1)=$P(DATA,U,PCE+3)  .. S ($P(DATA,U,PCE+2),$P(DATA,U,PCE+3))=""  .. Q  . Q  ;  Q DATA  ; | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCEF21 ;ALB/TMP - FORMATTER SPECIFIC BILL FUNCTIONS CONTINUED ; 3/9/11 1:12pm  ;;2.0;INTEGRATED BILLING;\*\*51,296,371,389,448,516,**592**\*\*;21-MAR-94;Build 123  ;;Per VA Directive 6402, this routine should not be modified.  ; COID(IBIFN) ; Claim office ID  N IBCOID,IBCOID1,IBIN  S IBIN=$$CURR^IBCEF2(IBIFN),IBCOID1="",IBCOID=$P($$ADDRESS^IBCNSC0(IBIN,.11,5),U,11)  ;  I IBIN D  . I $D(^IBA(364.2,"C",IBIFN)) S IBCOID1=$P($$ADDRESS^IBCNSC0(IBIN,.18,5),U,11) Q  ;Rx  . I $P($G(^DGCR(399,IBIFN,0)),U,5)<3 S IBCOID1=$P($$ADDRESS^IBCNSC0(IBIN,.12,5),U,11) Q  ;Inpt  . I $P($G(^DGCR(399,IBIFN,0)),U,5)'<3 S IBCOID1=$P($$ADDRESS^IBCNSC0(IBIN,.16,5),U,11) Q  ;Outpt **. ;JWS;IB\*2.0\*592;Dental insurance mailing address info  . I $$FT^IBCEF(IBIFN)=7 S IBCOID1=$P($$ADDRESS^IBCNSC0(IBIN,.19,11),U,11) Q  ;Dental**  ;  Q $S(IBCOID1'="":IBCOID1,1:IBCOID)  ; ESGHPST(IBIFN,COB) ; return insureds employment status if the bill policy defined by COB is an Employer Sponsored Group Health Plan  ; ESGHP FLAG (2.312,2.1) ^ the employment status (2.312,2.11)  ;  N PPOL,DFN,X,Y S Y=""  S PPOL=$$PPOL^IBCEF2($G(IBIFN),$G(COB)),DFN=$P($G(^DGCR(399,+$G(IBIFN),0)),U,2)  I +PPOL,+DFN S X=$G(^DPT(DFN,.312,+PPOL,2)) S Y=+$P(X,U,10)\_U\_$P(X,U,11)  Q Y  ; ESGHPNL(IBIFN,COB) ; return employer name and location if the bill policy defined by COB is an Employer Sponsored Group Health Plan  ; ESGHP FLAG (2.312,2.1) ^ employer name (2.312,2.015) ^ employer city (2.312,2.05)   ; ^ employer state abbr (2.312,2.06) ^ employer state ifn (2.312,2.06)  ;  N PPOL,DFN,X,Y S Y=""  S PPOL=$$PPOL^IBCEF2($G(IBIFN),$G(COB)),DFN=$P($G(^DGCR(399,+$G(IBIFN),0)),U,2)  I +PPOL,+DFN S X=$G(^DPT(DFN,.312,+PPOL,2)) S Y=+$P(X,U,10)\_U\_$P(X,U,9)\_U\_$P(X,U,5)\_U\_$P($G(^DIC(5,+$P(X,U,6),0)),U,2)\_U\_$P(X,U,6)  Q Y  ; REMARKS(IBIFN) ; Compile array of bill remarks  ;IBIFN = bill ien  N Z,Z0,Z1,IBARRAY,IBSM  S Z=0  ;S:$P($G(^DGCR(399,IBIFN,"U1")),U,2) Z=Z+1,Z0=$P(^("U1"),U,2),IBXDATA(Z)="OFFSET AMOUNT: "\_"$"\_+$P(Z0,".")\_"."\_$E($P(Z0,".",2)\_"00",1,2)  S:$P($G(^DGCR(399,IBIFN,"U1")),U,8)'="" Z=Z+1,IBXDATA(Z)=$P(^("U1"),U,8) ;Bill comment on bill  S Z0=$G(^DGCR(399,IBIFN,0)),Z1=$G(^DGCR(399.3,+$P(Z0,U,7),0))  D SET^IBCSC5B(IBIFN,.IBARRAY)  I $P($G(IBARRAY),U,2) D  ;Prosthetics  . S Z0=0 F  S Z0=$O(IBARRAY(Z0)) Q:Z0=""  S Z1=0 F  S Z1=$O(IBARRAY(Z0,Z1)) Q:'Z1  S Z=Z+1,IBXDATA(Z)="Prosthetic: "\_$E($$PINB^IBCSC5B(+IBARRAY(Z0,Z1)),1,39)\_" "\_$E(Z0,4,5)\_"/"\_$E(Z0,6,7)\_"/"\_$E(Z0,1,2)  Q  ; CREM(IBIFN) ; Compile array of bill remarks common to every bill  ;IBIFN = bill ien  N Z  S Z=0  S:$P($G(^IBE(350.9,1,1)),U,4)'="" Z=Z+1,IBXDATA(Z)=$P(^(1),U,4) ;Site specific 'every bill' comment  Q  ; ADMDT(IBIFN,NOOUTCK) ; Calculate admission/start of care date/time  ; IBIFN = bill ien  ; NOOUTCK = flag that will:  ; (1) no check for inpt episode overlap for outpt  ; (0 or null) performs check for inpt episode overlap for outpt  ;   ; Returns IBXDATA = fileman date format  N Z,Z0,Z1  S Z=$G(^DGCR(399,IBIFN,0)),Z1=$P($G(^("U")),U,20),Z0=$$INPAT^IBCEF(IBIFN,1)  S IBXDATA=$S(Z0&$P(Z,U,8):$P($G(^DGPT(+$P(Z,U,8),0)),U,2),1:"")  S:'IBXDATA IBXDATA=$P(Z,U,3)\_$S(Z0&(Z1<25):"."\_$E("0",$L(Z1))\_Z1,1:"")  ; Check to see if outpt episode (date in event date) overlaps inpt  ; episode - use admit date if it does  I 'Z0,IBXDATA,'$G(NOOUTCK) D  . N VAINDT,VAIN,DFN  . S VAINDT=IBXDATA,DFN=$P($G(^DGCR(399,IBIFN,0)),U,2)  . D INP^VADPT S IBXDATA=+VAIN(7) S:'IBXDATA IBXDATA=""  I 'IBXDATA,'Z0 S IBXDATA=$$SERVDT^IBCEF(IBIFN,,2)  Q  ; DISDT(IBIFN) ; Calculate discharge date  ; IBIFN = bill ien  N Z,Z0  S Z=$$INPAT^IBCEF(IBIFN,1),Z0=$G(^DGCR(399,IBIFN,0))  I Z S IBXDATA=+$G(^DGPT(+$P(Z0,U,8),70)) S:'IBXDATA IBXDATA=$P(Z0,U,16)  I 'Z N VAINDT,VAIN,DFN S DFN=$P($G(^DGCR(399,IBIFN,0)),U,2) D INP^VADPT I VAIN(1) S IBXDATA=+$G(^DGPM(+$P($G(^DGPM(+VAIN(1),0)),U,17),0))  Q  ; INSSECID(IBIFN,TYPE,SEQ) ; Extract subscriber and patient prim/sec ID's  ; IBIFN required  ; TYPE is either "PAT" or "SUB" to indicate we need to extract either  ; patient or subscriber ID information. Default="SUB".  ; SEQ is the insurance sequence# (1,2,3). Default is current ins seq#.  ;  ; Output:  ; Function returns an 8-piece string as follows.  ; [1] primary qualifier  ; [2] primary ID  ; [3] secondary qual(1)  ; [4] secondary ID(1)  ; [5] secondary qual(2)  ; [6] secondary ID(2)  ; [7] secondary qual(3)  ; [8] secondary ID(3)  ;  NEW DATA,DFN,POL,IB0,IB5,REL  S DATA=""  S IBIFN=+$G(IBIFN) I 'IBIFN G INSSX  I $G(TYPE)="" S TYPE="SUB"               ; default type of ID's to get  I '$F(".PAT.SUB.","."\_TYPE\_".") G INSSX  I '$G(SEQ) S SEQ=$$COBN^IBCEF(IBIFN) ; default current ins seq#  I '$F(".1.2.3.","."\_SEQ\_".") G INSSX  S DFN=+$P($G(^DGCR(399,IBIFN,0)),U,2) I 'DFN G INSSX  S POL=+$P($G(^DGCR(399,IBIFN,"M")),U,SEQ+11) I 'POL G INSSX  ;IB\*2.0\*516/baa - Use HIPAA compliant fields  ;S IB0=$G(^DPT(DFN,.312,POL,0)) I IB0="" G INSSX ;516 - baa  S IB0=$$ZND^IBCNS1(DFN,POL) I IB0="" G INSSX  ;516 - baa  S IB5=$G(^DPT(DFN,.312,POL,5))  S REL=+$P(IB0,U,16) ; pat rel to insured  S $P(DATA,U,1)="MI"  S $P(DATA,U,2)=$P(IB0,U,2) ; subscriber primary ID   S $P(DATA,U,3,8)=$P(IB5,U,2,7) ; subscriber secondary data  I TYPE="PAT",REL'=1 D  . S $P(DATA,U,2)=$P(IB5,U,1) ; patient primary ID  . S $P(DATA,U,3,8)=$P(IB5,U,8,13) ; patient secondary data  . Q  ;  S DATA=$$SCRUB(DATA) ; scrub the data INSSX ;  Q DATA  ; SCRUB(DATA) ; Scrub the 8-piece string gathered above  NEW PCE  ;  ; make sure you can't have an ID without a qualifier or a qualifier  ; without an ID. Check all 4 pairs.  F PCE=1,3,5,7 D  . I $P(DATA,U,PCE)'="",$P(DATA,U,PCE+1)'="" Q  . S ($P(DATA,U,PCE),$P(DATA,U,PCE+1))=""  . Q  ;  ; fill in secondary gaps. If Set1 and Set2 are blank, but Set3 exists  ; then move Set3 to Set1 and delete Set3.  I $P(DATA,U,3)="",$P(DATA,U,5)="",$P(DATA,U,7)'="" D  . S $P(DATA,U,3)=$P(DATA,U,7),$P(DATA,U,4)=$P(DATA,U,8)  . S ($P(DATA,U,7),$P(DATA,U,8))=""  . Q  ;  ; fill in secondary gaps more generically.  ; If Set(n) is blank, but Set(n+1) exists, then move it up.  F PCE=3,5 D  . I $P(DATA,U,PCE)="",$P(DATA,U,PCE+2)'="" D  .. S $P(DATA,U,PCE)=$P(DATA,U,PCE+2)  .. S $P(DATA,U,PCE+1)=$P(DATA,U,PCE+3)  .. S ($P(DATA,U,PCE+2),$P(DATA,U,PCE+3))=""  .. Q  . Q  ;  Q DATA  ; | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCNSC0 | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** |  | | | | | | | | |
| **Related Options** | None | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  |  | | | | |  | | | |
| **Data Dictionary (DD) References** |  | | | | | | | | |
| **Related Protocols** | None | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Current Logic** | | | | | | | | | |
| IBCNSC0 ;ALB/NLR - INSURANCE COMPANY EDIT - ;12-MAR-1993  ;;2.0;INTEGRATED BILLING;\*\*371,547\*\*;21-MAR-94;Build 119  ;;Per VA Directive 6402, this routine should not be modified.  ; CLAIMS1 ; display Inpatient Claims information  N OFFSET,START,IBCNS12,IBADD  ;WCJ;IB\*2.0\*547  ;S START=27,OFFSET=2  S START=28+(2\*$G(IBACMAX)),OFFSET=2  D SET^IBCNSP(START,OFFSET+20," Inpatient Claims Office Information ",IORVON,IORVOFF)  ;  ;WCJ;IB\*2.0\*547;Call New API  ;S IBCNS12=$$ADDRESS(IBCNS,.12,5)  S IBCNS12=$$ADD2(IBCNS,.12,5)  ;  D SET^IBCNSP(START+1,OFFSET," Company Name: "\_$P($G(^DIC(36,+$P(IBCNS12,"^",7),0)),"^",1))  D SET^IBCNSP(START+2,OFFSET," Street: "\_$P(IBCNS12,"^",1))  D SET^IBCNSP(START+3,OFFSET," Street 2: "\_$P(IBCNS12,"^",2))  N OFFSET S OFFSET=45  D SET^IBCNSP(START+1,OFFSET," Street 3: "\_$P(IBCNS12,"^",3)) S IBADD=1  D SET^IBCNSP(START+1+IBADD,OFFSET," City/State: "\_$E($P(IBCNS12,"^",4),1,15)\_$S($P(IBCNS12,"^",4)="":"",1:", ")\_$P($G(^DIC(5,+$P(IBCNS12,"^",5),0)),"^",2)\_" "\_$E($P(IBCNS12,"^",6),1,5))  D SET^IBCNSP(START+2+IBADD,OFFSET," Phone: "\_$P(IBCNS12,"^",8))  D SET^IBCNSP(START+3+IBADD,OFFSET," Fax: "\_$P(IBCNS12,"^",9))  Q  ; R1Q Q CLAIMS2 ; display Outpatient Claims information  ;  N OFFSET,START,IBCNS16,IBADD  ;WCJ;IB\*2.0\*547  ;S START=34,OFFSET=2  S START=35+(2\*$G(IBACMAX)),OFFSET=2  D SET^IBCNSP(START,OFFSET+20," Outpatient Claims Office Information ",IORVON,IORVOFF)  ;  ;WCJ;IB\*2.0\*547;Call New API  ;S IBCNS16=$$ADDRESS(IBCNS,.16,6)  S IBCNS16=$$ADD2(IBCNS,.16,6)  ;  D SET^IBCNSP(START+1,OFFSET," Company Name: "\_$P($G(^DIC(36,+$P(IBCNS16,"^",7),0)),"^",1))  D SET^IBCNSP(START+2,OFFSET," Street: "\_$P(IBCNS16,"^",1))  D SET^IBCNSP(START+3,OFFSET," Street 2: "\_$P(IBCNS16,"^",2))  N OFFSET S OFFSET=45  D SET^IBCNSP(START+1,OFFSET," Street 3: "\_$P(IBCNS16,"^",3)) S IBADD=1  D SET^IBCNSP(START+1+IBADD,OFFSET," City/State: "\_$E($P(IBCNS16,"^",4),1,15)\_$S($P(IBCNS16,"^",4)="":"",1:", ")\_$P($G(^DIC(5,+$P(IBCNS16,"^",5),0)),"^",2)\_" "\_$E($P(IBCNS16,"^",6),1,5))  D SET^IBCNSP(START+2+IBADD,OFFSET," Phone: "\_$P(IBCNS16,"^",8))  D SET^IBCNSP(START+3+IBADD,OFFSET," Fax: "\_$P(IBCNS16,"^",9))  Q  ;  ; Only adding comments on patch 547. Changes are on the ADD2 tag below.  ; This tag is called from the Output formatter.  ; It returns a "complete" address  ; It judges an address complete if it has a state (don't ask why, I am just adding the comments)  ; If the address it wants is not complete, it returns the main address.  ; These addresses go out on claims and claims (X12 837) don't like partial addresses. ADDRESS(INS,NODE,PH) ; -- generic find address  ;  N IBX,INSSAVE,IBPH,IBFX,IBCNT,IBA  S IBX="" ;S IBPH="",IBFX="",IBA=""  ; REDO ; gather insurance carrier's main address information   S IBX=$G(^DIC(36,+INS,.11)),IBPH=$P($G(^DIC(36,+INS,.13)),"^",1),IBFX=$P(IBX,"^",9)  ;S IBCNT=$G(IBCNT)+1  ;  ; -- if process the same co. more than once you are in an infinite loop  ;I $D(IBCNT(IBCNS)) G ADDREQ  ;S IBCNT(IBCNS)=""  ;  ; -- gather address information from specific office (Claims, Appeals, Inquiry)  ;  I $P($G(^DIC(36,+INS,+NODE)),"^",5) S IBX=$G(^DIC(36,+INS,+NODE)),IBPH=$P($G(^DIC(36,+INS,.13)),"^",PH),IBFX=$P($G(IBX),"^",9)  I $P($G(^DIC(36,+INS,+NODE)),"^",7) S INSSAVE=INS,INS=$P($G(^DIC(36,+INS,+NODE)),"^",7) I INSSAVE'=INS G REDO  ; ADDRESQ ; concatenate company name, address, phone and fax   S $P(IBA,"^",1,6)=$P($G(IBX),"^",1,6)  S $P(IBA,"^",7)=INS  S $P(IBA,"^",8)=IBPH  S $P(IBA,"^",9)=IBFX ADDREQ Q IBA  ;  ; WCJ;IB\*2.0\*547;  ; This is a new tag which is just called from the insurance company editor screens.  ; The billers/insurance verifiers want to see what data is actually in the insurance company file.  ; They don't care if it's complete. Heck, a phone number may be enough.  ; This will just return what is in the file for the ins company that handles that type of claims.  ; Input: INS - IREN to file 36  ; NODE - Node in File 36 (corresponds to Claims, Appeals, Inquiry...)  ; PH - Location of Phone number in node .13 ADD2(INS,NODE,PH) ;  N IBX,INSSAVE,IBFX,IBPH,IBA  F  S IBX=$G(^DIC(36,+INS,+NODE)) Q:'$P(IBX,U,7) S INSSAVE=INS,INS=$P(IBX,U,7) Q:INSSAVE=INS  ; concatenate company name, address, phone and fax   S IBPH=$P($G(^DIC(36,+INS,.13)),U,PH),IBFX=$P(IBX,U,9)  S $P(IBA,U,1,6)=$P(IBX,U,1,6),$P(IBA,U,7)=INS,$P(IBA,U,8)=IBPH,$P(IBA,U,9)=IBFX  Q IBA | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCNSC0 ;ALB/NLR - INSURANCE COMPANY EDIT - ;12-MAR-1993  ;;2.0;INTEGRATED BILLING;\*\*371,547,**592**\*\*;21-MAR-94;Build 119  ;;Per VA Directive 6402, this routine should not be modified.  ; CLAIMS1 ; display Inpatient Claims information  N OFFSET,START,IBCNS12,IBADD  ;WCJ;IB\*2.0\*547  ;S START=27,OFFSET=2  S START=28+(2\*$G(IBACMAX)),OFFSET=2  D SET^IBCNSP(START,OFFSET+20," Inpatient Claims Office Information ",IORVON,IORVOFF)  ;  ;WCJ;IB\*2.0\*547;Call New API  ;S IBCNS12=$$ADDRESS(IBCNS,.12,5)  S IBCNS12=$$ADD2(IBCNS,.12,5)  ;  D SET^IBCNSP(START+1,OFFSET," Company Name: "\_$P($G(^DIC(36,+$P(IBCNS12,"^",7),0)),"^",1))  D SET^IBCNSP(START+2,OFFSET," Street: "\_$P(IBCNS12,"^",1))  D SET^IBCNSP(START+3,OFFSET," Street 2: "\_$P(IBCNS12,"^",2))  N OFFSET S OFFSET=45  D SET^IBCNSP(START+1,OFFSET," Street 3: "\_$P(IBCNS12,"^",3)) S IBADD=1  D SET^IBCNSP(START+1+IBADD,OFFSET," City/State: "\_$E($P(IBCNS12,"^",4),1,15)\_$S($P(IBCNS12,"^",4)="":"",1:", ")\_$P($G(^DIC(5,+$P(IBCNS12,"^",5),0)),"^",2)\_" "\_$E($P(IBCNS12,"^",6),1,5))  D SET^IBCNSP(START+2+IBADD,OFFSET," Phone: "\_$P(IBCNS12,"^",8))  D SET^IBCNSP(START+3+IBADD,OFFSET," Fax: "\_$P(IBCNS12,"^",9))  Q  ; R1Q Q CLAIMS2 ; display Outpatient Claims information  ;  N OFFSET,START,IBCNS16,IBADD  ;WCJ;IB\*2.0\*547  ;S START=34,OFFSET=2  S START=35+(2\*$G(IBACMAX)),OFFSET=2  D SET^IBCNSP(START,OFFSET+20," Outpatient Claims Office Information ",IORVON,IORVOFF)  ;  ;WCJ;IB\*2.0\*547;Call New API  ;S IBCNS16=$$ADDRESS(IBCNS,.16,6)  S IBCNS16=$$ADD2(IBCNS,.16,6)  ;  D SET^IBCNSP(START+1,OFFSET," Company Name: "\_$P($G(^DIC(36,+$P(IBCNS16,"^",7),0)),"^",1))  D SET^IBCNSP(START+2,OFFSET," Street: "\_$P(IBCNS16,"^",1))  D SET^IBCNSP(START+3,OFFSET," Street 2: "\_$P(IBCNS16,"^",2))  N OFFSET S OFFSET=45  D SET^IBCNSP(START+1,OFFSET," Street 3: "\_$P(IBCNS16,"^",3)) S IBADD=1  D SET^IBCNSP(START+1+IBADD,OFFSET," City/State: "\_$E($P(IBCNS16,"^",4),1,15)\_$S($P(IBCNS16,"^",4)="":"",1:", ")\_$P($G(^DIC(5,+$P(IBCNS16,"^",5),0)),"^",2)\_" "\_$E($P(IBCNS16,"^",6),1,5))  D SET^IBCNSP(START+2+IBADD,OFFSET," Phone: "\_$P(IBCNS16,"^",8))  D SET^IBCNSP(START+3+IBADD,OFFSET," Fax: "\_$P(IBCNS16,"^",9))  Q  ;  ; Only adding comments on patch 547. Changes are on the ADD2 tag below.  ; This tag is called from the Output formatter.  ; It returns a "complete" address  ; It judges an address complete if it has a state (don't ask why, I am just adding the comments)  ; If the address it wants is not complete, it returns the main address.  ; These addresses go out on claims and claims (X12 837) don't like partial addresses. ADDRESS(INS,NODE,PH) ; -- generic find address  ;  N IBX,INSSAVE,IBPH,IBFX,IBCNT,IBA  S IBX="" ;S IBPH="",IBFX="",IBA=""  ; REDO ; gather insurance carrier's main address information   S IBX=$G(^DIC(36,+INS,.11)),IBPH=$P($G(^DIC(36,+INS,.13)),"^",1),IBFX=$P(IBX,"^",9)  ;S IBCNT=$G(IBCNT)+1  ;  ; -- if process the same co. more than once you are in an infinite loop  ;I $D(IBCNT(IBCNS)) G ADDREQ  ;S IBCNT(IBCNS)=""  ;  ; -- gather address information from specific office (Claims, Appeals, Inquiry**, Dental)**  **;JWS;IB\*2.0\*592;Changed below for DENTAL insurance mailing address  I $P($G(^DIC(36,+INS,+NODE)),"^",5) D  . S IBX=$G(^DIC(36,+INS,+NODE))  . I +NODE=.19 S IBPH=$P(IBX,"^",PH)  . E**S IBPH=$P($G(^DIC(36,+INS,.13)),"^",PH)  . S IBFX=$P($G(IBX),"^",9)  I $P($G(^DIC(36,+INS,+NODE)),"^",7) S INSSAVE=INS,INS=$P($G(^DIC(36,+INS,+NODE)),"^",7) I INSSAVE'=INS G REDO  ; ADDRESQ ; concatenate company name, address, phone and fax   S $P(IBA,"^",1,6)=$P($G(IBX),"^",1,6)  S $P(IBA,"^",7)=INS  S $P(IBA,"^",8)=IBPH  S $P(IBA,"^",9)=IBFX ADDREQ Q IBA  ;  ; WCJ;IB\*2.0\*547;  ; This is a new tag which is just called from the insurance company editor screens.  ; The billers/insurance verifiers want to see what data is actually in the insurance company file.  ; They don't care if it's complete. Heck, a phone number may be enough.  ; This will just return what is in the file for the ins company that handles that type of claims.  ; Input: INS - IREN to file 36  ; NODE - Node in File 36 (corresponds to Claims, Appeals, Inquiry...)  ; PH - Location of Phone number in node .13 ADD2(INS,NODE,PH) ;  N IBX,INSSAVE,IBFX,IBPH,IBA  F  S IBX=$G(^DIC(36,+INS,+NODE)) Q:'$P(IBX,U,7) S INSSAVE=INS,INS=$P(IBX,U,7) Q:INSSAVE=INS  ; concatenate company name, address, phone and fax   S IBPH=$P($G(^DIC(36,+INS,.13)),U,PH),IBFX=$P(IBX,U,9)  **;JWS;IB\*2.0\*592;Dental mailing address  I +NODE=.19 S IBPH=$P($G(^DIC(36,+INS,.19)),U,11)**  S $P(IBA,U,1,6)=$P(IBX,U,1,6),$P(IBA,U,7)=INS,$P(IBA,U,8)=IBPH,$P(IBA,U,9)=IBFX  Q IBA | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCNSC01 | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** |  | | | | | | | | |
| **Related Options** | None | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  |  | | | | |  | | | |
| **Data Dictionary (DD) References** |  | | | | | | | | |
| **Related Protocols** | None | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Current Logic** | | | | | | | | | |
| IBCNSC01 ;ALB/NLR - INSURANCE COMPANY EDIT ;6/1/05 10:06am  ;;2.0;INTEGRATED BILLING;\*\*52,137,191,184,232,320,349,371,399,416,432,494,519,547\*\*;21-MAR-94;Build 119  ;;Per VA Directive 6402, this routine should not be modified.  ; PARAM ; -- Insurance company parameters region  N OFFSET,START,IBCNS0,IBCNS03,IBCNS06,IBCNS08,IBCNS13,IBCNS3,IBHPD  S IBCNS0=$G(^DIC(36,+IBCNS,0)),IBCNS3=$G(^(3))  S IBCNS03=$P(IBCNS0,"^",3),IBCNS06=$P(IBCNS0,"^",6),IBCNS08=$P(IBCNS0,"^",8)  S IBCNS13=$G(^DIC(36,+IBCNS,.13))  S START=1,OFFSET=2  D SET^IBCNSP(START,OFFSET+25," Billing Parameters ",IORVON,IORVOFF)  ;  D SET^IBCNSP(START+1,OFFSET+1,"Signature Required?: "\_$S(+IBCNS03:"YES",1:"NO"))  D SET^IBCNSP(START+2,OFFSET+10,"Reimburse?: "\_$E($$EXPAND^IBTRE(36,1,$P(IBCNS0,"^",2)),1,21))  D SET^IBCNSP(START+3,OFFSET+3,"Mult. Bedsections: "\_$S(+IBCNS06:"YES",IBCNS06=0:"NO",1:""))  D SET^IBCNSP(START+4,OFFSET+6,"One Opt. Visit: "\_$S(+IBCNS08:"YES",1:"NO"))  D SET^IBCNSP(START+5,OFFSET+4,"Diff. Rev. Codes: "\_$P(IBCNS0,"^",7))  D SET^IBCNSP(START+6,OFFSET+1,"Amb. Sur. Rev. Code: "\_$P(IBCNS0,"^",9))  D SET^IBCNSP(START+7,OFFSET+1,"Rx Refill Rev. Code: "\_$P(IBCNS0,"^",15))  D SET^IBCNSP(START+8,OFFSET+3,"Filing Time Frame: "\_$P(IBCNS0,"^",12)\_$S(+$P(IBCNS0,"^",18):" ("\_$$FTFN^IBCNSU31(,+IBCNS)\_")",1:""))  ;  S OFFSET=45  D SET^IBCNSP(START+1,OFFSET+4,"Type Of Coverage: "\_$$EXPAND^IBTRE(36,.13,+$P(IBCNS0,U,13)))  D SET^IBCNSP(START+2,OFFSET+7,"Billing Phone: "\_$P(IBCNS13,"^",2))  D SET^IBCNSP(START+3,OFFSET+2,"Verification Phone: "\_$P(IBCNS13,"^",4))  D SET^IBCNSP(START+4,OFFSET+2,"Precert Comp. Name: "\_$P($G(^DIC(36,+$P(IBCNS13,"^",9),0)),"^",1))  D SET^IBCNSP(START+5,OFFSET+7,"Precert Phone: "\_$$PHONE(IBCNS13))  I +IBCNS3=2 D SET^IBCNSP(START+6,OFFSET,"Max # Test Bills/Day: "\_$P(IBCNS3,U,6))  ;  S START=11,OFFSET=2  D SET^IBCNSP(START,OFFSET+28," EDI Parameters ",IORVON,IORVOFF)  D SET^IBCNSP(START+1,OFFSET+13,"Transmit?: "\_$S(+IBCNS3=1:"YES-LIVE",+IBCNS3=2:"TEST ONLY",1:"NO"))  D SET^IBCNSP(START+2,OFFSET+1,"Inst Payer Primary ID: "\_$P(IBCNS3,U,4))  ;  ;WCJ;IB\*2.0\*547; Lots o Changes below to include new Alternate Primary ID  N IBAC,IBACND,LOOP  S IBACMAX=0  F IBACND=15,16 D  .S LOOP=0 F  S LOOP=$O(^DIC(36,+IBCNS,IBACND,LOOP)) Q:'+LOOP  D  ..S IBAC(IBACND,"CT")=$G(IBAC(IBACND,"CT"))+1 I IBAC(IBACND,"CT")>IBACMAX S IBACMAX=IBAC(IBACND,"CT")  ..S IBAC(IBACND,IBAC(IBACND,"CT"))=$P($G(^DIC(36,+IBCNS,IBACND,LOOP,0)),U,1,2)  ;  S LOOP=0 F  S LOOP=$O(IBAC(15,LOOP)) Q:'LOOP  D  .D SET^IBCNSP(START+2+(LOOP\*2-1),OFFSET,"Alt-I Payer Prim ID Type: "\_$$GET1^DIQ(355.98,+$P($G(IBAC(15,LOOP)),U),.01))  .D SET^IBCNSP(START+2+(LOOP\*2),OFFSET,"Alt-Inst Payer Prim ID: "\_$P($G(IBAC(15,LOOP)),U,2))  ;  D SET^IBCNSP(START+3+(2\*IBACMAX),OFFSET,"Inst Payer Sec ID Qual: "\_$$GET1^DIQ(36,+IBCNS,6.01))  D SET^IBCNSP(START+4+(2\*IBACMAX),OFFSET+5,"Inst Payer Sec ID: "\_$$GET1^DIQ(36,+IBCNS,6.02))  D SET^IBCNSP(START+5+(2\*IBACMAX),OFFSET,"Inst Payer Sec ID Qual: "\_$$GET1^DIQ(36,+IBCNS,6.03))  D SET^IBCNSP(START+6+(2\*IBACMAX),OFFSET+5,"Inst Payer Sec ID: "\_$$GET1^DIQ(36,+IBCNS,6.04))  D SET^IBCNSP(START+7+(2\*IBACMAX),OFFSET+12,"Bin Number: "\_$P($G(^DIC(36,+IBCNS,3)),"^",3)) ;  ;  ;IB\*2.0\*547;WCJ Added and bumped HPID down  D SET^IBCNSP(START+8+(2\*IBACMAX),OFFSET+10,"UMO (278) ID: "\_$P($G(^DIC(36,+IBCNS,7)),U))  ;ib\*2.0\*519  S IBHPD=$$HPD^IBCNHUT1(+IBCNS)  D SET^IBCNSP(START+9+(2\*IBACMAX),OFFSET+13,$P($$HOD^IBCNHUT1(IBHPD),U,2)\_": "\_IBHPD)  ;  S OFFSET=41  D SET^IBCNSP(START+1,OFFSET+8," Insurance Type: "\_$$EXPAND^IBTRE(36,3.09,+$P(IBCNS3,U,9)))  D SET^IBCNSP(START+2,OFFSET+1," Prof Payer Primary ID: "\_$P(IBCNS3,U,2))  ;  S LOOP=0 F  S LOOP=$O(IBAC(16,LOOP)) Q:'LOOP  D  .D SET^IBCNSP(START+2+(LOOP\*2-1),OFFSET+1,"Alt-P Payer Prim ID Type: "\_$$GET1^DIQ(355.98,+$P($G(IBAC(16,LOOP)),U),.01))  .D SET^IBCNSP(START+2+(LOOP\*2),OFFSET+1,"Alt-Prof Payer Prim ID: "\_$P($G(IBAC(16,LOOP)),U,2))  ;  D SET^IBCNSP(START+3+(2\*IBACMAX),OFFSET," Prof Payer Sec ID Qual: "\_$$GET1^DIQ(36,+IBCNS,6.05))  D SET^IBCNSP(START+4+(2\*IBACMAX),OFFSET+5," Prof Payer Sec ID: "\_$$GET1^DIQ(36,+IBCNS,6.06))  D SET^IBCNSP(START+5+(2\*IBACMAX),OFFSET," Prof Payer Sec ID Qual: "\_$$GET1^DIQ(36,+IBCNS,6.07))  D SET^IBCNSP(START+6+(2\*IBACMAX),OFFSET+5," Prof Payer Sec ID: "\_$$GET1^DIQ(36,+IBCNS,6.08))  ;IB\*2.0\*432/TAZ Added fields 6.09 and 6.1  D SET^IBCNSP(START+7+(2\*IBACMAX),OFFSET-3," Prnt Sec/Tert Auto Claims: "\_$$GET1^DIQ(36,+IBCNS,6.09))  D SET^IBCNSP(START+8+(2\*IBACMAX),OFFSET-5," Prnt Med Sec Claims w/o MRA: "\_$$GET1^DIQ(36,+IBCNS,6.1))  Q  ; PHONE(IBCNS13) ; -- Compute precert company phone  N IBX,IBSAVE,IBCNT S IBX=""  I '$P(IBCNS13,"^",9) S IBX=$P(IBCNS13,"^",3) G PHONEQ REDOX S IBSAVE=+$P(IBCNS13,"^",9)  S IBCNT=$G(IBCNT)+1  ; -- if you process the same co. more than once you are in an infinite loop  I $D(IBCNT(IBCNS)) G PHONEQ  S IBCNT(IBCNS)=""  S IBCNS13=$G(^DIC(36,+$P(IBCNS13,"^",9),.13))  S IBX=$P(IBCNS13,"^") S:$L($P(IBCNS13,"^",3)) IBX=$P(IBCNS13,"^",3)  ; -- if process the same co. more than once you are in an infinite loop  I $P(IBCNS13,"^",9),$P(IBCNS13,"^",9)'=IBSAVE G REDOX PHONEQ Q IBX  ; MAIN ; -- Insurance company main address  N OFFSET,START,IBCNS11,IBCNS13,IBADD  S IBCNS11=$G(^DIC(36,+IBCNS,.11))  S IBCNS13=$G(^DIC(36,+IBCNS,.13))  ;  ;S START=21,OFFSET=25  S START=22+(2\*IBACMAX),OFFSET=25  D SET^IBCNSP(START,OFFSET," Main Mailing Address ",IORVON,IORVOFF)  N OFFSET S OFFSET=2  D SET^IBCNSP(START+1,OFFSET," Street: "\_$P(IBCNS11,"^",1)) S IBADD=1  D SET^IBCNSP(START+2,OFFSET," Street 2: "\_$P(IBCNS11,"^",2)) S IBADD=2  D SET^IBCNSP(START+3,OFFSET," Street 3: "\_$P(IBCNS11,"^",3)) S IBADD=3  ; D SET^IBCNSP(START+4,OFFSET,"Claim Off. ID: "\_$P(IBCNS11,U,11))  N OFFSET S OFFSET=45  D SET^IBCNSP(START+1,OFFSET," City/State: "\_$E($P(IBCNS11,"^",4),1,15)\_$S($P(IBCNS11,"^",4)="":"",1:", ")\_$P($G(^DIC(5,+$P(IBCNS11,"^",5),0)),"^",2)\_" "\_$E($P(IBCNS11,"^",6),1,5))  D SET^IBCNSP(START+2,OFFSET," Phone: "\_$P(IBCNS13,"^",1))  D SET^IBCNSP(START+3,OFFSET," Fax: "\_$P(IBCNS11,"^",9))  Q  ;  ; PAYER ; This procedure builds the display for the payer associated with  ; this insurance company.  ; ESG - 7/29/02 - IIV project  ; - 9/9/09 - eIV updated  ; - 2/4/13 - remove ePharmacy references (IB\*2\*494)  ;  NEW PAYERIEN,PAYR,APPDATA,APP,DATA,APPNAME,A1,A2,A3,A4,A5,A6,A7,A8  NEW START,TITLE,OFFSET,IBLINE  S PAYERIEN=$P($G(^DIC(36,+IBCNS,3)),U,10),PAYR="",APPDATA=0  I PAYERIEN D  . S PAYR=$G(^IBE(365.12,PAYERIEN,0))  . S APP=0  . F  S APP=$O(^IBE(365.12,PAYERIEN,1,APP)) Q:'APP  D  .. S DATA=$G(^IBE(365.12,PAYERIEN,1,APP,0))  .. S APPNAME=$$EXTERNAL^DILFD(365.121,.01,"",$P(DATA,U,1))  .. I APPNAME="" Q  .. I APPNAME="IIV" S APPNAME="eIV"   ; IB\*2\*416 - change external display to be eIV  .. I APPNAME="E-PHARM" Q             ; IB\*2\*494 - don't display ePharmacy application data  .. I $D(APPDATA(APPNAME)) Q  .. S (A1,A2,A3,A4,A5,A6,A7)="NO",A8=""  .. I $P(DATA,U,2) S A1="YES"      ; national active  .. I $P(DATA,U,3) S A2="YES"      ; local active  .. I $P(DATA,U,7) S A3="YES"      ; auto-accept  .. I $P(DATA,U,8) S A4="YES"      ; ident inquiries require subscr ID (\*416 field not used)  .. I $P(DATA,U,9) S A5="YES"      ; use SSN for subscriber ID (\*416 field not used)  .. I $P(DATA,U,10) S A6="YES"     ; transmit SSN (\*416 field not used)  .. I $P(DATA,U,11) S A7="YES"     ; deactivated?  .. ; A8 = deactivation date  .. I $P(DATA,U,12) S A8=$P($$FMTE^XLFDT($P(DATA,U,12),"5Z"),"@",1)  .. S APPDATA(APPNAME)=A1\_U\_A2\_U\_A3\_U\_A4\_U\_A5\_U\_A6\_U\_A7\_U\_A8  .. S APPDATA=APPDATA+1  .. Q  . Q  ;  S START=$O(^TMP("IBCNSC",$J,""),-1)+1  S IB1ST("PAYER")=START  S TITLE=" Payer Information: e-IV "     ; esg - IB\*2\*494 - remove ePharmacy reference  S OFFSET=(40-($L(TITLE)/2))\1+1  D SET^IBCNSP(START,OFFSET,TITLE,IORVON,IORVOFF)  D SET^IBCNSP(START+1,9,"Payer Name: "\_$P(PAYR,U,1))  D SET^IBCNSP(START+2,5,"VA National ID: "\_$P(PAYR,U,2))  D SET^IBCNSP(START+2,51,"CMS National ID: "\_$P(PAYR,U,3))  S IBLINE=START+2  ;  ; Handle the case where no application data is defined  I 'APPDATA D  G PAYERX  . S IBLINE=IBLINE+1  . D SET^IBCNSP(IBLINE,2," ") ; blank line  . S IBLINE=IBLINE+1  . D SET^IBCNSP(IBLINE,16,"Payer Application data is not defined!")  . Q  ;  ; Display all the applications  S APPNAME=""  F  S APPNAME=$O(APPDATA(APPNAME)) Q:APPNAME=""  D  . S IBLINE=IBLINE+1  . D SET^IBCNSP(IBLINE,2," ") ; blank line  . ;  . S IBLINE=IBLINE+1  . D SET^IBCNSP(IBLINE,2,"Payer Application: "\_APPNAME)  . D SET^IBCNSP(IBLINE,51,"FSC Auto-Update: "\_$P(APPDATA(APPNAME),U,3))  . ;  . S IBLINE=IBLINE+1  . D SET^IBCNSP(IBLINE,4,"National Active: "\_$P(APPDATA(APPNAME),U,1))  . D SET^IBCNSP(IBLINE,55,"Deactivated: "\_$P(APPDATA(APPNAME),U,7))  . ;  . S IBLINE=IBLINE+1  . D SET^IBCNSP(IBLINE,7,"Local Active: "\_$P(APPDATA(APPNAME),U,2))  . ;  . ; If no deactivated date, then exit  . I $P(APPDATA(APPNAME),U,8)="" Q  . ;  . D SET^IBCNSP(IBLINE,50,"Date Deactivated: "\_$P(APPDATA(APPNAME),U,8))  . ;  . Q PAYERX ;  ; Two trailing blank lines after payer information display  S IBLINE=IBLINE+1  D SET^IBCNSP(IBLINE,2," ") ; blank line  S IBLINE=IBLINE+1  D SET^IBCNSP(IBLINE,2," ") ; blank line  Q  ;  ; REMARKS ;  ;  N OFFSET,START,IBLCNT,IBI  S START=$O(^TMP("IBCNSC",$J,""),-1)+1,OFFSET=2  S IB1ST("REM")=START  ;  D SET^IBCNSP(START,OFFSET," Remarks ",IORVON,IORVOFF)  S (IBLCNT,IBI)=0 F  S IBI=$O(^DIC(36,+IBCNS,11,IBI)) Q:IBI<1 D  . S IBLCNT=IBLCNT+1  . D SET^IBCNSP(START+IBLCNT,OFFSET," "\_$E($G(^DIC(36,+IBCNS,11,IBI,0)),1,80))  . Q  D SET^IBCNSP(START+IBLCNT+1,OFFSET," ") ; blank line after remarks  Q  ; SYN ;  N OFFSET,START,SYN,SYNOI  S START=$O(^TMP("IBCNSC",$J,""),-1)+1,OFFSET=2  S IB1ST("SYN")=START  D SET^IBCNSP(START,OFFSET," Synonyms ",IORVON,IORVOFF)  S SYN="" F SYNOI=1:1:8 S SYN=$O(^DIC(36,+IBCNS,10,"B",SYN)) Q:SYN=""  D SET^IBCNSP(START+SYNOI,OFFSET,$S(SYNOI>7:" ...edit to see more...",1:" "\_SYN))  Q  ; | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCNSC01 ;ALB/NLR - INSURANCE COMPANY EDIT ;6/1/05 10:06am  ;;2.0;INTEGRATED BILLING;\*\*52,137,191,184,232,320,349,371,399,416,432,494,519,547,**592**\*\*;21-MAR-94;Build 119  ;;Per VA Directive 6402, this routine should not be modified.  ; PARAM ; -- Insurance company parameters region  N OFFSET,START,IBCNS0,IBCNS03,IBCNS06,IBCNS08,IBCNS13,IBCNS3,IBHPD  S IBCNS0=$G(^DIC(36,+IBCNS,0)),IBCNS3=$G(^(3))  S IBCNS03=$P(IBCNS0,"^",3),IBCNS06=$P(IBCNS0,"^",6),IBCNS08=$P(IBCNS0,"^",8)  S IBCNS13=$G(^DIC(36,+IBCNS,.13))  S START=1,OFFSET=2  D SET^IBCNSP(START,OFFSET+25," Billing Parameters ",IORVON,IORVOFF)  ;  D SET^IBCNSP(START+1,OFFSET+1,"Signature Required?: "\_$S(+IBCNS03:"YES",1:"NO"))  D SET^IBCNSP(START+2,OFFSET+10,"Reimburse?: "\_$E($$EXPAND^IBTRE(36,1,$P(IBCNS0,"^",2)),1,21))  D SET^IBCNSP(START+3,OFFSET+3,"Mult. Bedsections: "\_$S(+IBCNS06:"YES",IBCNS06=0:"NO",1:""))  D SET^IBCNSP(START+4,OFFSET+6,"One Opt. Visit: "\_$S(+IBCNS08:"YES",1:"NO"))  D SET^IBCNSP(START+5,OFFSET+4,"Diff. Rev. Codes: "\_$P(IBCNS0,"^",7))  D SET^IBCNSP(START+6,OFFSET+1,"Amb. Sur. Rev. Code: "\_$P(IBCNS0,"^",9))  D SET^IBCNSP(START+7,OFFSET+1,"Rx Refill Rev. Code: "\_$P(IBCNS0,"^",15))  D SET^IBCNSP(START+8,OFFSET+3,"Filing Time Frame: "\_$P(IBCNS0,"^",12)\_$S(+$P(IBCNS0,"^",18):" ("\_$$FTFN^IBCNSU31(,+IBCNS)\_")",1:""))  ;  S OFFSET=45  D SET^IBCNSP(START+1,OFFSET+4,"Type Of Coverage: "\_$$EXPAND^IBTRE(36,.13,+$P(IBCNS0,U,13)))  D SET^IBCNSP(START+2,OFFSET+7,"Billing Phone: "\_$P(IBCNS13,"^",2))  D SET^IBCNSP(START+3,OFFSET+2,"Verification Phone: "\_$P(IBCNS13,"^",4))  D SET^IBCNSP(START+4,OFFSET+2,"Precert Comp. Name: "\_$P($G(^DIC(36,+$P(IBCNS13,"^",9),0)),"^",1))  D SET^IBCNSP(START+5,OFFSET+7,"Precert Phone: "\_$$PHONE(IBCNS13))  I +IBCNS3=2 D SET^IBCNSP(START+6,OFFSET,"Max # Test Bills/Day: "\_$P(IBCNS3,U,6))  ;  S START=11,OFFSET=2  D SET^IBCNSP(START,OFFSET+28," EDI Parameters ",IORVON,IORVOFF)  D SET^IBCNSP(START+1,OFFSET+13,"Transmit?: "\_$S(+IBCNS3=1:"YES-LIVE",+IBCNS3=2:"TEST ONLY",1:"NO"))  D SET^IBCNSP(START+2,OFFSET+1,"Inst Payer Primary ID: "\_$P(IBCNS3,U,4))  **;JWS;IB\*2.0\*592;Dental Payer ID  D SET^IBCNSP(START+4,OFFSET+7,"Dental Payer ID: "\_$P(IBCNS3,U,15))**  ;  ;WCJ;IB\*2.0\*547; Lots o Changes below to include new Alternate Primary ID  N IBAC,IBACND,LOOP  S IBACMAX=0  F IBACND=15,16 D  .S LOOP=0 F  S LOOP=$O(^DIC(36,+IBCNS,IBACND,LOOP)) Q:'+LOOP  D  ..S IBAC(IBACND,"CT")=$G(IBAC(IBACND,"CT"))+1 I IBAC(IBACND,"CT")>IBACMAX S IBACMAX=IBAC(IBACND,"CT")  ..S IBAC(IBACND,IBAC(IBACND,"CT"))=$P($G(^DIC(36,+IBCNS,IBACND,LOOP,0)),U,1,2)  ;  S LOOP=0 F  S LOOP=$O(IBAC(15,LOOP)) Q:'LOOP  D  .D SET^IBCNSP(START+2+(LOOP\*2-1),OFFSET,"Alt-I Payer Prim ID Type: "\_$$GET1^DIQ(355.98,+$P($G(IBAC(15,LOOP)),U),.01))  .D SET^IBCNSP(START+2+(LOOP\*2),OFFSET,"Alt-Inst Payer Prim ID: "\_$P($G(IBAC(15,LOOP)),U,2))  ;  D SET^IBCNSP(START+3+(2\*IBACMAX),OFFSET,"Inst Payer Sec ID Qual: "\_$$GET1^DIQ(36,+IBCNS,6.01))  D SET^IBCNSP(START+4+(2\*IBACMAX),OFFSET+5,"Inst Payer Sec ID: "\_$$GET1^DIQ(36,+IBCNS,6.02))  D SET^IBCNSP(START+5+(2\*IBACMAX),OFFSET,"Inst Payer Sec ID Qual: "\_$$GET1^DIQ(36,+IBCNS,6.03))  D SET^IBCNSP(START+6+(2\*IBACMAX),OFFSET+5,"Inst Payer Sec ID: "\_$$GET1^DIQ(36,+IBCNS,6.04))  D SET^IBCNSP(START+7+(2\*IBACMAX),OFFSET+12,"Bin Number: "\_$P($G(^DIC(36,+IBCNS,3)),"^",3)) ;  ;  ;IB\*2.0\*547;WCJ Added and bumped HPID down  D SET^IBCNSP(START+8+(2\*IBACMAX),OFFSET+10,"UMO (278) ID: "\_$P($G(^DIC(36,+IBCNS,7)),U))  ;ib\*2.0\*519  S IBHPD=$$HPD^IBCNHUT1(+IBCNS)  D SET^IBCNSP(START+9+(2\*IBACMAX),OFFSET+13,$P($$HOD^IBCNHUT1(IBHPD),U,2)\_": "\_IBHPD)  ;  S OFFSET=41  D SET^IBCNSP(START+1,OFFSET+8," Insurance Type: "\_$$EXPAND^IBTRE(36,3.09,+$P(IBCNS3,U,9)))  D SET^IBCNSP(START+2,OFFSET+1," Prof Payer Primary ID: "\_$P(IBCNS3,U,2))  ;  S LOOP=0 F  S LOOP=$O(IBAC(16,LOOP)) Q:'LOOP  D  .D SET^IBCNSP(START+2+(LOOP\*2-1),OFFSET+1,"Alt-P Payer Prim ID Type: "\_$$GET1^DIQ(355.98,+$P($G(IBAC(16,LOOP)),U),.01))  .D SET^IBCNSP(START+2+(LOOP\*2),OFFSET+1,"Alt-Prof Payer Prim ID: "\_$P($G(IBAC(16,LOOP)),U,2))  ;  D SET^IBCNSP(START+3+(2\*IBACMAX),OFFSET," Prof Payer Sec ID Qual: "\_$$GET1^DIQ(36,+IBCNS,6.05))  D SET^IBCNSP(START+4+(2\*IBACMAX),OFFSET+5," Prof Payer Sec ID: "\_$$GET1^DIQ(36,+IBCNS,6.06))  D SET^IBCNSP(START+5+(2\*IBACMAX),OFFSET," Prof Payer Sec ID Qual: "\_$$GET1^DIQ(36,+IBCNS,6.07))  D SET^IBCNSP(START+6+(2\*IBACMAX),OFFSET+5," Prof Payer Sec ID: "\_$$GET1^DIQ(36,+IBCNS,6.08))  ;IB\*2.0\*432/TAZ Added fields 6.09 and 6.1  D SET^IBCNSP(START+7+(2\*IBACMAX),OFFSET-3," Prnt Sec/Tert Auto Claims: "\_$$GET1^DIQ(36,+IBCNS,6.09))  D SET^IBCNSP(START+8+(2\*IBACMAX),OFFSET-5," Prnt Med Sec Claims w/o MRA: "\_$$GET1^DIQ(36,+IBCNS,6.1))  Q  ; PHONE(IBCNS13) ; -- Compute precert company phone  N IBX,IBSAVE,IBCNT S IBX=""  I '$P(IBCNS13,"^",9) S IBX=$P(IBCNS13,"^",3) G PHONEQ REDOX S IBSAVE=+$P(IBCNS13,"^",9)  S IBCNT=$G(IBCNT)+1  ; -- if you process the same co. more than once you are in an infinite loop  I $D(IBCNT(IBCNS)) G PHONEQ  S IBCNT(IBCNS)=""  S IBCNS13=$G(^DIC(36,+$P(IBCNS13,"^",9),.13))  S IBX=$P(IBCNS13,"^") S:$L($P(IBCNS13,"^",3)) IBX=$P(IBCNS13,"^",3)  ; -- if process the same co. more than once you are in an infinite loop  I $P(IBCNS13,"^",9),$P(IBCNS13,"^",9)'=IBSAVE G REDOX PHONEQ Q IBX  ; MAIN ; -- Insurance company main address  N OFFSET,START,IBCNS11,IBCNS13,IBADD  S IBCNS11=$G(^DIC(36,+IBCNS,.11))  S IBCNS13=$G(^DIC(36,+IBCNS,.13))  ;  ;S START=21,OFFSET=25  S START=22+(2\*IBACMAX),OFFSET=25  D SET^IBCNSP(START,OFFSET," Main Mailing Address ",IORVON,IORVOFF)  N OFFSET S OFFSET=2  D SET^IBCNSP(START+1,OFFSET," Street: "\_$P(IBCNS11,"^",1)) S IBADD=1  D SET^IBCNSP(START+2,OFFSET," Street 2: "\_$P(IBCNS11,"^",2)) S IBADD=2  D SET^IBCNSP(START+3,OFFSET," Street 3: "\_$P(IBCNS11,"^",3)) S IBADD=3  ; D SET^IBCNSP(START+4,OFFSET,"Claim Off. ID: "\_$P(IBCNS11,U,11))  N OFFSET S OFFSET=45  D SET^IBCNSP(START+1,OFFSET," City/State: "\_$E($P(IBCNS11,"^",4),1,15)\_$S($P(IBCNS11,"^",4)="":"",1:", ")\_$P($G(^DIC(5,+$P(IBCNS11,"^",5),0)),"^",2)\_" "\_$E($P(IBCNS11,"^",6),1,5))  D SET^IBCNSP(START+2,OFFSET," Phone: "\_$P(IBCNS13,"^",1))  D SET^IBCNSP(START+3,OFFSET," Fax: "\_$P(IBCNS11,"^",9))  Q  ;  ; PAYER ; This procedure builds the display for the payer associated with  ; this insurance company.  ; ESG - 7/29/02 - IIV project  ; - 9/9/09 - eIV updated  ; - 2/4/13 - remove ePharmacy references (IB\*2\*494)  ;  NEW PAYERIEN,PAYR,APPDATA,APP,DATA,APPNAME,A1,A2,A3,A4,A5,A6,A7,A8  NEW START,TITLE,OFFSET,IBLINE  S PAYERIEN=$P($G(^DIC(36,+IBCNS,3)),U,10),PAYR="",APPDATA=0  I PAYERIEN D  . S PAYR=$G(^IBE(365.12,PAYERIEN,0))  . S APP=0  . F  S APP=$O(^IBE(365.12,PAYERIEN,1,APP)) Q:'APP  D  .. S DATA=$G(^IBE(365.12,PAYERIEN,1,APP,0))  .. S APPNAME=$$EXTERNAL^DILFD(365.121,.01,"",$P(DATA,U,1))  .. I APPNAME="" Q  .. I APPNAME="IIV" S APPNAME="eIV"   ; IB\*2\*416 - change external display to be eIV  .. I APPNAME="E-PHARM" Q             ; IB\*2\*494 - don't display ePharmacy application data  .. I $D(APPDATA(APPNAME)) Q  .. S (A1,A2,A3,A4,A5,A6,A7)="NO",A8=""  .. I $P(DATA,U,2) S A1="YES"      ; national active  .. I $P(DATA,U,3) S A2="YES"      ; local active  .. I $P(DATA,U,7) S A3="YES"      ; auto-accept  .. I $P(DATA,U,8) S A4="YES"      ; ident inquiries require subscr ID (\*416 field not used)  .. I $P(DATA,U,9) S A5="YES"      ; use SSN for subscriber ID (\*416 field not used)  .. I $P(DATA,U,10) S A6="YES"     ; transmit SSN (\*416 field not used)  .. I $P(DATA,U,11) S A7="YES"     ; deactivated?  .. ; A8 = deactivation date  .. I $P(DATA,U,12) S A8=$P($$FMTE^XLFDT($P(DATA,U,12),"5Z"),"@",1)  .. S APPDATA(APPNAME)=A1\_U\_A2\_U\_A3\_U\_A4\_U\_A5\_U\_A6\_U\_A7\_U\_A8  .. S APPDATA=APPDATA+1  .. Q  . Q  ;  S START=$O(^TMP("IBCNSC",$J,""),-1)+1  S IB1ST("PAYER")=START  S TITLE=" Payer Information: e-IV "     ; esg - IB\*2\*494 - remove ePharmacy reference  S OFFSET=(40-($L(TITLE)/2))\1+1  D SET^IBCNSP(START,OFFSET,TITLE,IORVON,IORVOFF)  D SET^IBCNSP(START+1,9,"Payer Name: "\_$P(PAYR,U,1))  D SET^IBCNSP(START+2,5,"VA National ID: "\_$P(PAYR,U,2))  D SET^IBCNSP(START+2,51,"CMS National ID: "\_$P(PAYR,U,3))  S IBLINE=START+2  ;  ; Handle the case where no application data is defined  I 'APPDATA D  G PAYERX  . S IBLINE=IBLINE+1  . D SET^IBCNSP(IBLINE,2," ") ; blank line  . S IBLINE=IBLINE+1  . D SET^IBCNSP(IBLINE,16,"Payer Application data is not defined!")  . Q  ;  ; Display all the applications  S APPNAME=""  F  S APPNAME=$O(APPDATA(APPNAME)) Q:APPNAME=""  D  . S IBLINE=IBLINE+1  . D SET^IBCNSP(IBLINE,2," ") ; blank line  . ;  . S IBLINE=IBLINE+1  . D SET^IBCNSP(IBLINE,2,"Payer Application: "\_APPNAME)  . D SET^IBCNSP(IBLINE,51,"FSC Auto-Update: "\_$P(APPDATA(APPNAME),U,3))  . ;  . S IBLINE=IBLINE+1  . D SET^IBCNSP(IBLINE,4,"National Active: "\_$P(APPDATA(APPNAME),U,1))  . D SET^IBCNSP(IBLINE,55,"Deactivated: "\_$P(APPDATA(APPNAME),U,7))  . ;  . S IBLINE=IBLINE+1  . D SET^IBCNSP(IBLINE,7,"Local Active: "\_$P(APPDATA(APPNAME),U,2))  . ;  . ; If no deactivated date, then exit  . I $P(APPDATA(APPNAME),U,8)="" Q  . ;  . D SET^IBCNSP(IBLINE,50,"Date Deactivated: "\_$P(APPDATA(APPNAME),U,8))  . ;  . Q PAYERX ;  ; Two trailing blank lines after payer information display  S IBLINE=IBLINE+1  D SET^IBCNSP(IBLINE,2," ") ; blank line  S IBLINE=IBLINE+1  D SET^IBCNSP(IBLINE,2," ") ; blank line  Q  ;  ; REMARKS ;  ;  N OFFSET,START,IBLCNT,IBI  S START=$O(^TMP("IBCNSC",$J,""),-1)+1,OFFSET=2  S IB1ST("REM")=START  ;  D SET^IBCNSP(START,OFFSET," Remarks ",IORVON,IORVOFF)  S (IBLCNT,IBI)=0 F  S IBI=$O(^DIC(36,+IBCNS,11,IBI)) Q:IBI<1 D  . S IBLCNT=IBLCNT+1  . D SET^IBCNSP(START+IBLCNT,OFFSET," "\_$E($G(^DIC(36,+IBCNS,11,IBI,0)),1,80))  . Q  D SET^IBCNSP(START+IBLCNT+1,OFFSET," ") ; blank line after remarks  Q  ; SYN ;  N OFFSET,START,SYN,SYNOI  S START=$O(^TMP("IBCNSC",$J,""),-1)+1,OFFSET=2  S IB1ST("SYN")=START  D SET^IBCNSP(START,OFFSET," Synonyms ",IORVON,IORVOFF)  S SYN="" F SYNOI=1:1:8 S SYN=$O(^DIC(36,+IBCNS,10,"B",SYN)) Q:SYN=""  D SET^IBCNSP(START+SYNOI,OFFSET,$S(SYNOI>7:" ...edit to see more...",1:" "\_SYN))  Q  ; | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCNADD | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** |  | | | | | | | | |
| **Related Options** | None | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  |  | | | | |  | | | |
| **Data Dictionary (DD) References** |  | | | | | | | | |
| **Related Protocols** | None | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Current Logic** | | | | | | | | | |
| IBCNADD ;ALB/AAS - ADDRESS RETRIEVAL ENGINE FOR FILE 399 ; 29-AUG-93  ;;2.0;INTEGRATED BILLING;\*\*52,80,377\*\*;21-MAR-94;Build 23  ;;Per VHA Directive 2004-038, this routine should not be modified.  ; ADD(DA,IBCOB) ; -- Retrieve correct billing address for a bill, mailing address of Bill Payer  ; assumes that new policy field points to valid ins. policy  ; DA = ien to file 399  ; IBCOB = payer sequence PST or 123 (optional)  ;  N X,Y,I,J,IB01,IB02,IBTYP,DFN,IBCNS,IBCDFN,IBCNT,IBAGAIN,IBFND,IBBILLTY,IBCHRGTY  S IB02=""  S DFN=$P($G(^DGCR(399,DA,0)),"^",2)  S IBBILLTY=$P($G(^DGCR(399,DA,0)),"^",5),IBCHRGTY=$P($$CHGTYPE^IBCU(DA),"^;",1)  ;  S IBCNS=+$P($G(^DGCR(399,DA,"MP")),U,1)  S IBCDFN=$P($G(^DGCR(399,DA,"MP")),U,2)  ;  ; If a specific payer sequence was passed in, get the ins. company and the policy ptr  ; No address returned for Medicare  I $G(IBCOB)'="" D  I $$MCRWNR^IBEFUNC(IBCNS) G MAINQ  . S IBCOB=$TR(IBCOB,"PST","123")  . S IBCNS=+$P($G(^DGCR(399,DA,"I"\_IBCOB)),U,1)  . S IBCDFN=+$P($G(^DGCR(399,DA,"M")),U,IBCOB+11)  . Q  ;  I 'IBCNS G MAINQ  I IBCDFN S IBCNS=+$G(^DPT(+DFN,.312,+IBCDFN,0))  I '$D(^DIC(36,+IBCNS,0)) G MAINQ  ;  ; -- if send bill to employer and state is filled in use this  I +$G(^DPT(DFN,.312,+IBCDFN,2)),+$P(^(2),"^",6) S IB02=$P(^(2),"^",2,99) G MAINQ  ; MAIN ; -- determine address for company for type bill  ;  ; -- get main address  S IB02=$S($D(^DIC(36,+IBCNS,.11)):^(.11),1:"")  S IBCNT=$G(IBCNT)+1  ;  ; -- if process the same co. more than once you are in an infinite loop  I $D(IBCNT(IBCNS)) G MAINQ ;already processed this company use main add  S IBCNT(IBCNS)=""  ;  ; -- type of charges: Rx charges - if ins company has an rx address use it, otherwise use opt address  I IBCHRGTY=3 S IBTYP="R" D @IBTYP G:$D(IBFND) MAINQ I $D(IBAGAIN) K IBAGAIN G MAIN  ;  ; -- type of bill: inpatient<3, outpatient>2  S IBTYP=$S(IBBILLTY<3:"I",1:"O")  D @IBTYP I $D(IBAGAIN) K IBAGAIN G MAIN  ;  ; -- return address MAINQ Q IB02  ; I ; -- see if there is an inpatient address  ; -- use if state is there  I $P($G(^DIC(36,+IBCNS,.12)),"^",5) S IB02=$P($G(^(.12)),"^",1,6)  ;  ; -- if other company processes claims start again  I $P($G(^DIC(36,+IBCNS,.12)),"^",7) S IBCNS=$P($G(^DIC(36,+IBCNS,.12)),"^",7) S IBAGAIN=1  Q  ; O ; -- see if there is an outpatient address  ; -- use if state is there  I $P($G(^DIC(36,+IBCNS,.16)),"^",5) S IB02=$P($G(^(.16)),"^",1,6)  ;  ; -- if other company processes claims start again  I $P($G(^DIC(36,+IBCNS,.16)),"^",7) S IBCNS=$P($G(^DIC(36,+IBCNS,.16)),"^",7) S IBAGAIN=1  Q  ; R ; -- see if there is an Rx address  ; -- use if state is there  I $P($G(^DIC(36,+IBCNS,.18)),"^",5) S IB02=$P($G(^(.18)),"^",1,6) S IBFND=1  ;  ; -- if other company processes claims start again  I $P($G(^DIC(36,+IBCNS,.18)),"^",7) S IBCNS=$P($G(^DIC(36,+IBCNS,.18)),"^",7) S IBAGAIN=1 K IBFND  Q | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCNADD ;ALB/AAS - ADDRESS RETRIEVAL ENGINE FOR FILE 399 ; 29-AUG-93  ;;2.0;INTEGRATED BILLING;\*\*52,80,377,**592**\*\*;21-MAR-94;Build 23  ;;Per VHA Directive 2004-038, this routine should not be modified.  ; ADD(DA,IBCOB) ; -- Retrieve correct billing address for a bill, mailing address of Bill Payer  ; assumes that new policy field points to valid ins. policy  ; DA = ien to file 399  ; IBCOB = payer sequence PST or 123 (optional)  ;  N X,Y,I,J,IB01,IB02,IBTYP,DFN,IBCNS,IBCDFN,IBCNT,IBAGAIN,IBFND,IBBILLTY,IBCHRGTY  S IB02=""  S DFN=$P($G(^DGCR(399,DA,0)),"^",2)  S IBBILLTY=$P($G(^DGCR(399,DA,0)),"^",5),IBCHRGTY=$P($$CHGTYPE^IBCU(DA),"^;",1)  ;  S IBCNS=+$P($G(^DGCR(399,DA,"MP")),U,1)  S IBCDFN=$P($G(^DGCR(399,DA,"MP")),U,2)  ;  ; If a specific payer sequence was passed in, get the ins. company and the policy ptr  ; No address returned for Medicare  I $G(IBCOB)'="" D  I $$MCRWNR^IBEFUNC(IBCNS) G MAINQ  . S IBCOB=$TR(IBCOB,"PST","123")  . S IBCNS=+$P($G(^DGCR(399,DA,"I"\_IBCOB)),U,1)  . S IBCDFN=+$P($G(^DGCR(399,DA,"M")),U,IBCOB+11)  . Q  ;  I 'IBCNS G MAINQ  I IBCDFN S IBCNS=+$G(^DPT(+DFN,.312,+IBCDFN,0))  I '$D(^DIC(36,+IBCNS,0)) G MAINQ  ;  ; -- if send bill to employer and state is filled in use this  I +$G(^DPT(DFN,.312,+IBCDFN,2)),+$P(^(2),"^",6) S IB02=$P(^(2),"^",2,99) G MAINQ  ; MAIN ; -- determine address for company for type bill  ;  ; -- get main address  S IB02=$S($D(^DIC(36,+IBCNS,.11)):^(.11),1:"")  S IBCNT=$G(IBCNT)+1  ;  ; -- if process the same co. more than once you are in an infinite loop  I $D(IBCNT(IBCNS)) G MAINQ ;already processed this company use main add  S IBCNT(IBCNS)=""  ;  ; -- type of charges: Rx charges - if ins company has an rx address use it, otherwise use opt address  I IBCHRGTY=3 S IBTYP="R" D @IBTYP G:$D(IBFND) MAINQ I $D(IBAGAIN) K IBAGAIN G MAIN  ;  ; -- type of bill: inpatient<3, outpatient>2  S IBTYP=$S(IBBILLTY<3:"I",1:"O")  D @IBTYP I $D(IBAGAIN) K IBAGAIN G MAIN  ;  ; -- return address MAINQ Q IB02  ; I ; -- see if there is an inpatient address  ; -- use if state is there  I $P($G(^DIC(36,+IBCNS,.12)),"^",5) S IB02=$P($G(^(.12)),"^",1,6)  ;  ; -- if other company processes claims start again  I $P($G(^DIC(36,+IBCNS,.12)),"^",7) S IBCNS=$P($G(^DIC(36,+IBCNS,.12)),"^",7) S IBAGAIN=1  Q  ; O ; -- see if there is an outpatient address  ; -- use if state is there **;JWS;IB\*2.0\*592;Dental Insurance mailing address  I $$FT^IBCEF(DA)=7 D  Q  . I $P($G(^DIC(36,+IBCNS,.19)),"^",5) S IB02=$P(^(.19),"^",1,6)  . I $P($G(^DIC(36,+IBCNS,.19)),"^",7) S IBCNS=$P(^(.19),"^",7) S IBAGAIN=1**  ;  I $P($G(^DIC(36,+IBCNS,.16)),"^",5) S IB02=$P($G(^(.16)),"^",1,6)  ;  ; -- if other company processes claims start again  I $P($G(^DIC(36,+IBCNS,.16)),"^",7) S IBCNS=$P($G(^DIC(36,+IBCNS,.16)),"^",7) S IBAGAIN=1  Q  ; R ; -- see if there is an Rx address  ; -- use if state is there  I $P($G(^DIC(36,+IBCNS,.18)),"^",5) S IB02=$P($G(^(.18)),"^",1,6) S IBFND=1  ;  ; -- if other company processes claims start again  I $P($G(^DIC(36,+IBCNS,.18)),"^",7) S IBCNS=$P($G(^DIC(36,+IBCNS,.18)),"^",7) S IBAGAIN=1 K IBFND  Q | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCNSCD1 | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** |  | | | | | | | | |
| **Related Options** | None | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  |  | | | | |  | | | |
| **Data Dictionary (DD) References** |  | | | | | | | | |
| **Related Protocols** | None | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Current Logic** | | | | | | | | | |
| IBCNSCD1 ;ALB/CPM - DELETE INSURANCE COMPANY (CON'T) ; 02-FEB-95  ;;2.0;INTEGRATED BILLING;\*\*28,46,80\*\*;21-MAR-94  ;;Per VHA Directive 10-93-142, this routine should not be modified.  ;  DQ ; Queued entry point for the final clean-up job.  ;  K ^TMP($J,"IBCNSCD")  L +^IB("IBCNSCD"):5 E  G DDQ ; another clean-up job got started  S IBC=0 F  S IBC=$O(^DIC(36,"ADEL",IBC)) Q:'IBC  S ^TMP($J,"IBCNSCD",IBC)=$P($G(^DIC(36,IBC,5)),"^",2)  I '$D(^TMP($J,"IBCNSCD")) G DDQ ; no companies to be deleted  ;  D NOW^%DTC S IBBDT=%  ;  ; - dispositions  S DFN=0 F  S DFN=$O(^DPT(DFN)) Q:'DFN  S IBC=0 F  S IBC=$O(^DPT(DFN,"DIS",IBC)) Q:'IBC  S IBCO=$P($G(^(IBC,2)),"^",6) I IBCO,$D(^TMP($J,"IBCNSCD",IBCO)) D  .S $P(^DPT(DFN,"DIS",IBC,2),"^",6)=$G(^TMP($J,"IBCNSCD",IBCO))  .S IBCT("DIS")=$G(IBCT("DIS"))+1  .I $G(^TMP($J,"IBCNSCD",IBCO))="" S IBCT("DIS",DFN,IBC)=""  ;  ; - insurance companies  S IBC=0 F  S IBC=$O(^DIC(36,IBC)) Q:'IBC  D  .S IB0=$G(^DIC(36,IBC,0)),IB12=$G(^(.12)),IB13=$G(^(.13)),IB14=$G(^(.14)),IB16=$G(^(.16)),IB18=$G(^(.18))  .K IBV  .I $P(IB0,"^",16),$D(^TMP($J,"IBCNSCD",$P(IB0,"^",16))) S IBV(0)="16^"\_^($P(IB0,"^",16))  .I $P(IB12,"^",7),$D(^TMP($J,"IBCNSCD",$P(IB12,"^",7))) S IBV(.12)="7^"\_^($P(IB12,"^",7))  .I $P(IB13,"^",9),$D(^TMP($J,"IBCNSCD",$P(IB13,"^",9))) S IBV(.13)="9^"\_^($P(IB13,"^",9))  .I $P(IB14,"^",7),$D(^TMP($J,"IBCNSCD",$P(IB14,"^",7))) S IBV(.14)="7^"\_^($P(IB14,"^",7))  .I $P(IB16,"^",7),$D(^TMP($J,"IBCNSCD",$P(IB16,"^",7))) S IBV(.16)="7^"\_^($P(IB16,"^",7))  .I $P(IB18,"^",7),$D(^TMP($J,"IBCNSCD",$P(IB18,"^",7))) S IBV(.18)="7^"\_^($P(IB18,"^",7))  .Q:'$D(IBV)  .;  .; - delete or repoint  .S IBX="" F  S IBX=$O(IBV(IBX)) Q:IBX=""  D  ..S $P(^DIC(36,IBC,IBX),"^",+IBV(IBX))=$P(IBV(IBX),"^",2)  ..S IBCT("INS",IBX)=$G(IBCT("INS",IBX))+1  ..I $P(IBV(IBX),"^",2)="" S IBCT("INS",IBX,IBC)=""  ;  ; - insurance reviews  S IBC=0 F  S IBC=$O(^IBT(356.2,IBC)) Q:'IBC  S IBCO=$P($G(^(IBC,0)),"^",8) I IBCO,$D(^TMP($J,"IBCNSCD",IBCO)) S IBCD=$G(^IBT(356.2,IBC,0)) D  .S IBVAL=$G(^TMP($J,"IBCNSCD",IBCO)) I 'IBVAL S IBVAL="@"  .S DA=IBC,DR=".08////"\_IBVAL,DIE="^IBT(356.2," D ^DIE K DA,DIE,DR  .S IBCT("IR")=$G(IBCT("IR"))+1  .I IBVAL="@" S IBCT("IR",+$P(IBCD,"^",5),+IBCD)=""  ;  ; - bills  S IBC=0 F  S IBC=$O(^DGCR(399,IBC)) Q:'IBC  S IBCNS=0 F  S IBCNS=$O(^DGCR(399,IBC,"AIC",IBCNS)) Q:'IBCNS  I $D(^TMP($J,"IBCNSCD",IBCNS)) S (IBREP,IBVAL)=$G(^(IBCNS)) D FIND  ;  ; - call AR to handle receivables  S IBCTAR=0 D INS2^RCAMINS("^TMP($J,""IBCNSCD"")",.IBCTAR)  ;  D NOW^%DTC S IBEDT=%  ;  ; - mail results  D MAIL^IBCNSCD2  ;  ; - finally, delete the companies  S IBC=0 F  S IBC=$O(^TMP($J,"IBCNSCD",IBC)) Q:'IBC  S DA=IBC,DIK="^DIC(36,",DIDEL=36 D ^DIK  ;  ; - delete task number from #350.9  S $P(^IBE(350.9,1,4),"^",8)=""  ; DDQ K IBC,IBCT,^TMP($J,"IBCNSCD")  L -^IB("IBCNSCD")  S ZTREQ="@"  Q  ;  ; FIND ; Find the carrier somewhere in the bill.  ; Required local variables are those described in CARR.  S IB0=$G(^DGCR(399,IBC,0)),IBM=$G(^("M"))  ;  ; - look for the carrier  I +IBM=IBCNS D CARR(1,"I1") ; primary  I $P(IBM,"^",2)=IBCNS D CARR(2,"I2") ; secondary  I $P(IBM,"^",3)=IBCNS D CARR(3,"I3") ; tertiary  ;  ; - kill off the x-ref  K ^DGCR(399,IBC,"AIC",IBCNS)  Q  ; CARR(IBP,IBSUB) ; Update each carrier.  ; Input: IBP -- carrier [1:primary 2:secondary 3:tertiary]  ; IBSUB -- updated subscript ["I1":prim "I2":sec "I3":tert]  ;  ; The following local variables are also required to be defined:  ; IBCNS, IB0, IBM, IBC, IBREP, IBVAL  ;  S IBCNS1=+IBREP  S $P(^DGCR(399,IBC,"M"),"^",IBP)=IBVAL  I $G(^DGCR(399,IBC,IBSUB))]"" S $P(^(IBSUB),"^",1)=IBVAL  I IBVAL="" D  .S IBS=0  .I $P(IB0,"^",2) S IBCNS1=+$G(^DPT($P(IB0,"^",2),.312,+$P(IBM,"^",IBP+11),0)) I IBCNS1 S IBS=1,$P(^DGCR(399,IBC,"M"),"^",IBP)=IBCNS1 S:$G(^(IBSUB))]"" $P(^(IBSUB),"^",1)=IBCNS1  .I 'IBS S IBCT("BL",IBP,IBC)=""  ;  I IBCNS1 S ^DGCR(399,IBC,"AIC",IBCNS1)=""  ;  I IBCNS=+$G(^DGCR(399,IBC,"MP")) D  .I $P(IB0,"^",2),+IBCNS K ^DGCR(399,"AE",$P(IB0,"^",2),IBCNS,IBC)  .S $P(^DGCR(399,IBC,"MP"),U,1)=IBCNS1  .I $P(IB0,"^",2),+IBCNS1 S ^DGCR(399,"AE",$P(IB0,"^",2),+IBCNS1,IBC)=""  ;  S IBCT("BL",IBP)=$G(IBCT("BL",IBP))+1  Q  ;  ; BILL(IBBILLN,IBCNS,IBREP) ; Callable Entry Point for Accounts Receivable  ; Input: IBBILLN -- Bill Number for bill to be repointed  ; IBCNS -- Pointer to the insurance company in file #36  ; that is being merged  ; IBREP -- Pointer to the insurance company in file #36  ; into which information is being merged  ;  N IBC,IBCT,IBVAL,IBCNS1,IB0,IBM  I $G(IBBILLN)=""!'$G(IBCNS)!($G(IBREP)="") G BILLQ  S IBC=$O(^DGCR(399,"B",IBBILLN,0)) I 'IBC G BILLQ  S IBVAL=$S(IBREP:IBREP,1:"")  D FIND BILLQ Q | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCNSCD1 ;ALB/CPM - DELETE INSURANCE COMPANY (CON'T) ; 02-FEB-95  ;;2.0;INTEGRATED BILLING;\*\*28,46,80,**592**\*\*;21-MAR-94  ;;Per VHA Directive 10-93-142, this routine should not be modified.  ;  DQ ; Queued entry point for the final clean-up job.  ;  K ^TMP($J,"IBCNSCD")  L +^IB("IBCNSCD"):5 E  G DDQ ; another clean-up job got started  S IBC=0 F  S IBC=$O(^DIC(36,"ADEL",IBC)) Q:'IBC  S ^TMP($J,"IBCNSCD",IBC)=$P($G(^DIC(36,IBC,5)),"^",2)  I '$D(^TMP($J,"IBCNSCD")) G DDQ ; no companies to be deleted  ;  D NOW^%DTC S IBBDT=%  ;  ; - dispositions  S DFN=0 F  S DFN=$O(^DPT(DFN)) Q:'DFN  S IBC=0 F  S IBC=$O(^DPT(DFN,"DIS",IBC)) Q:'IBC  S IBCO=$P($G(^(IBC,2)),"^",6) I IBCO,$D(^TMP($J,"IBCNSCD",IBCO)) D  .S $P(^DPT(DFN,"DIS",IBC,2),"^",6)=$G(^TMP($J,"IBCNSCD",IBCO))  .S IBCT("DIS")=$G(IBCT("DIS"))+1  .I $G(^TMP($J,"IBCNSCD",IBCO))="" S IBCT("DIS",DFN,IBC)=""  ;  ; - insurance companies  S IBC=0 F  S IBC=$O(^DIC(36,IBC)) Q:'IBC  D  **.;JWS;IB\*2.0\*592;add Dental Ins address**  .S IB0=$G(^DIC(36,IBC,0)),IB12=$G(^(.12)),IB13=$G(^(.13)),IB14=$G(^(.14)),IB16=$G(^(.16)),IB18=$G(^(.18)),**IB19=$G(^(.19))**  .K IBV  .I $P(IB0,"^",16),$D(^TMP($J,"IBCNSCD",$P(IB0,"^",16))) S IBV(0)="16^"\_^($P(IB0,"^",16))  .I $P(IB12,"^",7),$D(^TMP($J,"IBCNSCD",$P(IB12,"^",7))) S IBV(.12)="7^"\_^($P(IB12,"^",7))  .I $P(IB13,"^",9),$D(^TMP($J,"IBCNSCD",$P(IB13,"^",9))) S IBV(.13)="9^"\_^($P(IB13,"^",9))  .I $P(IB14,"^",7),$D(^TMP($J,"IBCNSCD",$P(IB14,"^",7))) S IBV(.14)="7^"\_^($P(IB14,"^",7))  .I $P(IB16,"^",7),$D(^TMP($J,"IBCNSCD",$P(IB16,"^",7))) S IBV(.16)="7^"\_^($P(IB16,"^",7))  .I $P(IB18,"^",7),$D(^TMP($J,"IBCNSCD",$P(IB18,"^",7))) S IBV(.18)="7^"\_^($P(IB18,"^",7)) **.;JWS;IB\*2.0\*592;add Dental Ins address  .I $P(IB19,"^",7),$D(^TMP($J,"IBCNSCD",$P(IB19,"^",7))) S IBV(.19)="7^"\_^($P(IB19,"^",7))**  .Q:'$D(IBV)  .;  .; - delete or repoint  .S IBX="" F  S IBX=$O(IBV(IBX)) Q:IBX=""  D  ..S $P(^DIC(36,IBC,IBX),"^",+IBV(IBX))=$P(IBV(IBX),"^",2)  ..S IBCT("INS",IBX)=$G(IBCT("INS",IBX))+1  ..I $P(IBV(IBX),"^",2)="" S IBCT("INS",IBX,IBC)=""  ;  ; - insurance reviews  S IBC=0 F  S IBC=$O(^IBT(356.2,IBC)) Q:'IBC  S IBCO=$P($G(^(IBC,0)),"^",8) I IBCO,$D(^TMP($J,"IBCNSCD",IBCO)) S IBCD=$G(^IBT(356.2,IBC,0)) D  .S IBVAL=$G(^TMP($J,"IBCNSCD",IBCO)) I 'IBVAL S IBVAL="@"  .S DA=IBC,DR=".08////"\_IBVAL,DIE="^IBT(356.2," D ^DIE K DA,DIE,DR  .S IBCT("IR")=$G(IBCT("IR"))+1  .I IBVAL="@" S IBCT("IR",+$P(IBCD,"^",5),+IBCD)=""  ;  ; - bills  S IBC=0 F  S IBC=$O(^DGCR(399,IBC)) Q:'IBC  S IBCNS=0 F  S IBCNS=$O(^DGCR(399,IBC,"AIC",IBCNS)) Q:'IBCNS  I $D(^TMP($J,"IBCNSCD",IBCNS)) S (IBREP,IBVAL)=$G(^(IBCNS)) D FIND  ;  ; - call AR to handle receivables  S IBCTAR=0 D INS2^RCAMINS("^TMP($J,""IBCNSCD"")",.IBCTAR)  ;  D NOW^%DTC S IBEDT=%  ;  ; - mail results  D MAIL^IBCNSCD2  ;  ; - finally, delete the companies  S IBC=0 F  S IBC=$O(^TMP($J,"IBCNSCD",IBC)) Q:'IBC  S DA=IBC,DIK="^DIC(36,",DIDEL=36 D ^DIK  ;  ; - delete task number from #350.9  S $P(^IBE(350.9,1,4),"^",8)=""  ; DDQ K IBC,IBCT,^TMP($J,"IBCNSCD")  L -^IB("IBCNSCD")  S ZTREQ="@"  Q  ;  ; FIND ; Find the carrier somewhere in the bill.  ; Required local variables are those described in CARR.  S IB0=$G(^DGCR(399,IBC,0)),IBM=$G(^("M"))  ;  ; - look for the carrier  I +IBM=IBCNS D CARR(1,"I1") ; primary  I $P(IBM,"^",2)=IBCNS D CARR(2,"I2") ; secondary  I $P(IBM,"^",3)=IBCNS D CARR(3,"I3") ; tertiary  ;  ; - kill off the x-ref  K ^DGCR(399,IBC,"AIC",IBCNS)  Q  ; CARR(IBP,IBSUB) ; Update each carrier.  ; Input: IBP -- carrier [1:primary 2:secondary 3:tertiary]  ; IBSUB -- updated subscript ["I1":prim "I2":sec "I3":tert]  ;  ; The following local variables are also required to be defined:  ; IBCNS, IB0, IBM, IBC, IBREP, IBVAL  ;  S IBCNS1=+IBREP  S $P(^DGCR(399,IBC,"M"),"^",IBP)=IBVAL  I $G(^DGCR(399,IBC,IBSUB))]"" S $P(^(IBSUB),"^",1)=IBVAL  I IBVAL="" D  .S IBS=0  .I $P(IB0,"^",2) S IBCNS1=+$G(^DPT($P(IB0,"^",2),.312,+$P(IBM,"^",IBP+11),0)) I IBCNS1 S IBS=1,$P(^DGCR(399,IBC,"M"),"^",IBP)=IBCNS1 S:$G(^(IBSUB))]"" $P(^(IBSUB),"^",1)=IBCNS1  .I 'IBS S IBCT("BL",IBP,IBC)=""  ;  I IBCNS1 S ^DGCR(399,IBC,"AIC",IBCNS1)=""  ;  I IBCNS=+$G(^DGCR(399,IBC,"MP")) D  .I $P(IB0,"^",2),+IBCNS K ^DGCR(399,"AE",$P(IB0,"^",2),IBCNS,IBC)  .S $P(^DGCR(399,IBC,"MP"),U,1)=IBCNS1  .I $P(IB0,"^",2),+IBCNS1 S ^DGCR(399,"AE",$P(IB0,"^",2),+IBCNS1,IBC)=""  ;  S IBCT("BL",IBP)=$G(IBCT("BL",IBP))+1  Q  ;  ; BILL(IBBILLN,IBCNS,IBREP) ; Callable Entry Point for Accounts Receivable  ; Input: IBBILLN -- Bill Number for bill to be repointed  ; IBCNS -- Pointer to the insurance company in file #36  ; that is being merged  ; IBREP -- Pointer to the insurance company in file #36  ; into which information is being merged  ;  N IBC,IBCT,IBVAL,IBCNS1,IB0,IBM  I $G(IBBILLN)=""!'$G(IBCNS)!($G(IBREP)="") G BILLQ  S IBC=$O(^DGCR(399,"B",IBBILLN,0)) I 'IBC G BILLQ  S IBVAL=$S(IBREP:IBREP,1:"")  D FIND BILLQ Q | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCNSCD2 | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** |  | | | | | | | | |
| **Related Options** | None | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  |  | | | | |  | | | |
| **Data Dictionary (DD) References** |  | | | | | | | | |
| **Related Protocols** | None | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Current Logic** | | | | | | | | | |
| IBCNSCD2 ;ALB/CPM - DELETE INSURANCE COMPANY (CON'T) ; 03-FEB-95  ;;Version 2.0 ; INTEGRATED BILLING ;\*\*28,46\*\*; 21-MAR-94  ;;Per VHA Directive 10-93-142, this routine should not be modified.  ;  MAIL ; Send results out.  S XMSUB="Insurance Company Deletion Clean-up Completion"  S XMDUZ="INTEGRATED BILLING PACKAGE",XMTEXT="^TMP($J,""IBT"",",XMY(DUZ)=""  ;  K ^TMP($J,"IBT") S IBC=0  D SET("The final clean-up for deleted Insurance Company(s) has completed.")  D SET(" ")  S Y=IBBDT D D^DIQ D SET("Job Start Time: "\_Y)  S Y=IBEDT D D^DIQ D SET(" Job End Time: "\_Y)  ;  D SET(" ")  D SET("DELETED COMPANY"\_$J("",24)\_"REPOINTED TO")  D SET($TR($J("",79)," ","="))  S IBX=0 F  S IBX=$O(^TMP($J,"IBCNSCD",IBX)) Q:'IBX  S IBX1=+$G(^(IBX)) D  .S X=$E($P($G(^DIC(36,IBX,0)),"^")\_" (#"\_IBX\_")"\_$J("",39),1,39)  .S X=X\_$S(IBX1:$P($G(^DIC(36,IBX1,0)),"^")\_" (#"\_IBX1\_")",1:"not repointed")  .D SET(X)  ;  D SET(" ")  D SET(" ")  D SET("1. Correction of the Disposition (sub-file #2.101) field")  D SET(" 'INJURING PARTIES INSURANCE' (#25)")  D SET(" Number of Disposition records updated: "\_+$G(IBCT("DIS")))  I $O(IBCT("DIS",0)) D  .D SET($J("",8)\_"The following dispositions had this field deleted and not merged:")  .S DFN=0 F  S DFN=$O(IBCT("DIS",DFN)) Q:'DFN  D  ..S IBNAM=$$PT^IBEFUNC(DFN),IBH=0  ..S IBX=$J("",10)\_$E($P(IBNAM,"^"),1,25)\_" ("\_$P(IBNAM,"^",3)\_")"  ..S IBDAT="" F  S IBDAT=$O(IBCT("DIS",DFN,IBDAT)) Q:IBDAT=""  D  ...S IBDAT1="Date/Time: "\_$$DAT2^IBOUTL(9999999-IBDAT)  ...I 'IBH D SET($E(IBX\_$J("",45),1,45)\_IBDAT1)  ...E  D SET($J("",45)\_IBDAT1)  ...S IBH=1  ;  ; - insurance companies  S IBINS(0)="REPOINT PATIENTS TO^.16"  S IBINS(.12)="CLAIMS (INPT) COMPANY NAME^.127"  S IBINS(.13)="PRECERT COMPANY NAME^.139"  S IBINS(.14)="APPEALS COMPANY NAME^.147"  S IBINS(.16)="CLAIMS (OPT) COMPANY NAME^.167"  S IBINS(.18)="CLAIMS (RX) COMPANY NAME^.187"  D SET(" ")  D SET("2. Correction of other Insurance Company (file #36) records:")  S IBX="" F  S IBX=$O(IBINS(IBX)) Q:IBX=""  S IBS=IBINS(IBX) D  .D SET(" Number of records with '"\_$P(IBS,"^")\_"' (#"\_$P(IBS,"^",2)\_") updated: "\_+$G(IBCT("INS",IBX)))  .I $O(IBCT("INS",IBX,0)) D  ..D SET($J("",8)\_"The following companies had this field deleted and not merged:")  ..S IBCO=0 F  S IBCO=$O(IBCT("INS",IBX,IBCO)) Q:'IBCO  D  ...D SET($J("",10)\_$P($G(^DIC(36,IBCO,0)),"^")\_" (ien "\_IBCO\_")")  ;  ; - insurance reviews  D SET(" ")  D SET("3. Correction of the Insurance Review (file #356.2) field")  D SET(" 'INSURANCE COMPANY CONTACTED' (#.08)")  D SET(" Number of Insurance Review records updated: "\_+$G(IBCT("IR")))  I $O(IBCT("IR",0)) D  .D SET($J("",8)\_"The following Insurance reviews had this field deleted and not merged:")  .S DFN=0 F  S DFN=$O(IBCT("IR",DFN)) Q:'DFN  D  ..S IBNAM=$$PT^IBEFUNC(DFN),IBH=0  ..S IBX=$J("",10)\_$E($P(IBNAM,"^"),1,25)\_" ("\_$P(IBNAM,"^",3)\_")"  ..S IBDAT="" F  S IBDAT=$O(IBCT("IR",DFN,IBDAT)) Q:IBDAT=""  D  ...S IBDAT1="Review Date/Time: "\_$$DAT2^IBOUTL(IBDAT)  ...I 'IBH D SET($E(IBX\_$J("",45),1,45)\_IBDAT1)  ...E  D SET($J("",45)\_IBDAT1)  ...S IBH=1  ;  ; - bills  K IBINS  S IBINS(1)="PRIMARY INSURANCE CARRIER^101"  S IBINS(2)="SECONDARY INSURANCE CARRIER^102"  S IBINS(3)="TERTIARY INSURANCE CARRIER^103"  D SET(" ")  D SET("4. Correction of Bill/Claims (file #399) records:")  S IBX="" F  S IBX=$O(IBINS(IBX)) Q:IBX=""  S IBS=IBINS(IBX) D  .D SET(" Number of records with '"\_$P(IBS,"^")\_"' (#"\_$P(IBS,"^",2)\_") updated: "\_+$G(IBCT("BL",IBX)))  .I $O(IBCT("BL",IBX,0)) D  ..D SET($J("",8)\_"The following bills had this field deleted and not merged:")  ..S IBCO=0 F  S IBCO=$O(IBCT("BL",IBX,IBCO)) Q:'IBCO  D  ...S IBS=$G(^DGCR(399,IBCO,0))  ...S IBNAM=$$PT^IBEFUNC(+$P(IBS,"^",2))  ...D SET($J("",10)\_$E($E($P(IBNAM,"^"),1,25)\_" ("\_$P(IBNAM,"^",3)\_")"\_$J("",35),1,35)\_"Bill #: "\_$P(IBS,"^"))  ;  ; - receivables in AR  D SET(" ")  D SET("5. Number of updated secondary and tertiary carriers of AR receivables: "\_+$G(IBCTAR))  ;  D ^XMD  K ^TMP($J,"IBT")  Q  ; SET(X) ; Set Message Text Array  S IBC=IBC+1,^TMP($J,"IBT",IBC)=X  Q | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCNSCD2 ;ALB/CPM - DELETE INSURANCE COMPANY (CON'T) ; 03-FEB-95  ;;Version 2.0 ; INTEGRATED BILLING ;\*\*28,46,**592**\*\*; 21-MAR-94  ;;Per VHA Directive 10-93-142, this routine should not be modified.  ;  MAIL ; Send results out.  S XMSUB="Insurance Company Deletion Clean-up Completion"  S XMDUZ="INTEGRATED BILLING PACKAGE",XMTEXT="^TMP($J,""IBT"",",XMY(DUZ)=""  ;  K ^TMP($J,"IBT") S IBC=0  D SET("The final clean-up for deleted Insurance Company(s) has completed.")  D SET(" ")  S Y=IBBDT D D^DIQ D SET("Job Start Time: "\_Y)  S Y=IBEDT D D^DIQ D SET(" Job End Time: "\_Y)  ;  D SET(" ")  D SET("DELETED COMPANY"\_$J("",24)\_"REPOINTED TO")  D SET($TR($J("",79)," ","="))  S IBX=0 F  S IBX=$O(^TMP($J,"IBCNSCD",IBX)) Q:'IBX  S IBX1=+$G(^(IBX)) D  .S X=$E($P($G(^DIC(36,IBX,0)),"^")\_" (#"\_IBX\_")"\_$J("",39),1,39)  .S X=X\_$S(IBX1:$P($G(^DIC(36,IBX1,0)),"^")\_" (#"\_IBX1\_")",1:"not repointed")  .D SET(X)  ;  D SET(" ")  D SET(" ")  D SET("1. Correction of the Disposition (sub-file #2.101) field")  D SET(" 'INJURING PARTIES INSURANCE' (#25)")  D SET(" Number of Disposition records updated: "\_+$G(IBCT("DIS")))  I $O(IBCT("DIS",0)) D  .D SET($J("",8)\_"The following dispositions had this field deleted and not merged:")  .S DFN=0 F  S DFN=$O(IBCT("DIS",DFN)) Q:'DFN  D  ..S IBNAM=$$PT^IBEFUNC(DFN),IBH=0  ..S IBX=$J("",10)\_$E($P(IBNAM,"^"),1,25)\_" ("\_$P(IBNAM,"^",3)\_")"  ..S IBDAT="" F  S IBDAT=$O(IBCT("DIS",DFN,IBDAT)) Q:IBDAT=""  D  ...S IBDAT1="Date/Time: "\_$$DAT2^IBOUTL(9999999-IBDAT)  ...I 'IBH D SET($E(IBX\_$J("",45),1,45)\_IBDAT1)  ...E  D SET($J("",45)\_IBDAT1)  ...S IBH=1  ;  ; - insurance companies  S IBINS(0)="REPOINT PATIENTS TO^.16"  S IBINS(.12)="CLAIMS (INPT) COMPANY NAME^.127"  S IBINS(.13)="PRECERT COMPANY NAME^.139"  S IBINS(.14)="APPEALS COMPANY NAME^.147"  S IBINS(.16)="CLAIMS (OPT) COMPANY NAME^.167"  S IBINS(.18)="CLAIMS (RX) COMPANY NAME^.187"  **;JWS;IB\*2.0\*592;add Dental Ins address  S IBINS(.19)="CLAIMS (DENTAL) COMPANY NAME^.197"**  D SET(" ")  D SET("2. Correction of other Insurance Company (file #36) records:")  S IBX="" F  S IBX=$O(IBINS(IBX)) Q:IBX=""  S IBS=IBINS(IBX) D  .D SET(" Number of records with '"\_$P(IBS,"^")\_"' (#"\_$P(IBS,"^",2)\_") updated: "\_+$G(IBCT("INS",IBX)))  .I $O(IBCT("INS",IBX,0)) D  ..D SET($J("",8)\_"The following companies had this field deleted and not merged:")  ..S IBCO=0 F  S IBCO=$O(IBCT("INS",IBX,IBCO)) Q:'IBCO  D  ...D SET($J("",10)\_$P($G(^DIC(36,IBCO,0)),"^")\_" (ien "\_IBCO\_")")  ;  ; - insurance reviews  D SET(" ")  D SET("3. Correction of the Insurance Review (file #356.2) field")  D SET(" 'INSURANCE COMPANY CONTACTED' (#.08)")  D SET(" Number of Insurance Review records updated: "\_+$G(IBCT("IR")))  I $O(IBCT("IR",0)) D  .D SET($J("",8)\_"The following Insurance reviews had this field deleted and not merged:")  .S DFN=0 F  S DFN=$O(IBCT("IR",DFN)) Q:'DFN  D  ..S IBNAM=$$PT^IBEFUNC(DFN),IBH=0  ..S IBX=$J("",10)\_$E($P(IBNAM,"^"),1,25)\_" ("\_$P(IBNAM,"^",3)\_")"  ..S IBDAT="" F  S IBDAT=$O(IBCT("IR",DFN,IBDAT)) Q:IBDAT=""  D  ...S IBDAT1="Review Date/Time: "\_$$DAT2^IBOUTL(IBDAT)  ...I 'IBH D SET($E(IBX\_$J("",45),1,45)\_IBDAT1)  ...E  D SET($J("",45)\_IBDAT1)  ...S IBH=1  ;  ; - bills  K IBINS  S IBINS(1)="PRIMARY INSURANCE CARRIER^101"  S IBINS(2)="SECONDARY INSURANCE CARRIER^102"  S IBINS(3)="TERTIARY INSURANCE CARRIER^103"  D SET(" ")  D SET("4. Correction of Bill/Claims (file #399) records:")  S IBX="" F  S IBX=$O(IBINS(IBX)) Q:IBX=""  S IBS=IBINS(IBX) D  .D SET(" Number of records with '"\_$P(IBS,"^")\_"' (#"\_$P(IBS,"^",2)\_") updated: "\_+$G(IBCT("BL",IBX)))  .I $O(IBCT("BL",IBX,0)) D  ..D SET($J("",8)\_"The following bills had this field deleted and not merged:")  ..S IBCO=0 F  S IBCO=$O(IBCT("BL",IBX,IBCO)) Q:'IBCO  D  ...S IBS=$G(^DGCR(399,IBCO,0))  ...S IBNAM=$$PT^IBEFUNC(+$P(IBS,"^",2))  ...D SET($J("",10)\_$E($E($P(IBNAM,"^"),1,25)\_" ("\_$P(IBNAM,"^",3)\_")"\_$J("",35),1,35)\_"Bill #: "\_$P(IBS,"^"))  ;  ; - receivables in AR  D SET(" ")  D SET("5. Number of updated secondary and tertiary carriers of AR receivables: "\_+$G(IBCTAR))  ;  D ^XMD  K ^TMP($J,"IBT")  Q  ; SET(X) ; Set Message Text Array  S IBC=IBC+1,^TMP($J,"IBT",IBC)=X  Q | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBJTU3 | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** |  | | | | | | | | |
| **Related Options** | None | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  |  | | | | |  | | | |
| **Data Dictionary (DD) References** |  | | | | | | | | |
| **Related Protocols** | None | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Current Logic** | | | | | | | | | |
| IBJTU3 ;ALB/ARH - TPI UTILITIES - INS ADDRESS ; 2/14/95  ;;2.0;INTEGRATED BILLING;\*\*39,80\*\*;21-MAR-94  ;;Per VHA Directive 10-93-142, this routine should not be modified.  ; BADD(IBIFN) ; returns mailing address for bill  ; returns: COMPANY NAME ^ PHONE NUMBER ^ STR 1 ^ STR 2 ^ STR 3 ^ CITY ^ STATE ^ ZIP ^ ^ FAX #  N DFN,IBX,IBCNS,IBCDFN,IBTYP  ;  S IBX="",DFN=$G(^DGCR(399,+$G(IBIFN),0))  S IBTYP=$P(DFN,U,5),DFN=+$P(DFN,U,2) I 'DFN G BADDQ  S IBCNS=$G(^DGCR(399,+IBIFN,"MP")) I 'IBCNS G BADDQ  S IBCDFN=$P(IBCNS,U,2) I +IBCDFN S IBCNS=+$G(^DPT(DFN,.312,+IBCDFN,0))  ;  ; -- if send to employer and state defined, return employer address  I +IBCDFN S IBCDFN=$G(^DPT(DFN,.312,+IBCDFN,2)) I +IBCDFN,+$P(IBCDFN,U,6) D  G BADDQ  . S IBX=$P(IBCDFN,U,9)\_U\_$P(IBCDFN,U,8)\_U\_$P(IBCDFN,U,2,7)  ;  S IBTYP=$S(IBTYP<3:"INP",1:"OPT")  S IBX=$$INSADD(+IBCNS,IBTYP)  ; BADDQ Q IBX  ;  ; INSADD(IBCNS,IBATYP) ; returns specific type of address/phone # for an insurance company, follows ptrs to company responsible  ; returns: COMPANY NAME ^ PHONE NUMBER ^ STR 1 ^ STR 2 ^ STR 3 ^ CITY ^ STATE ^ ZIP ^ ^ FAX #  ; if type does not have an address or phone number then main mailing addr/ph # is returned  ;  N IBD0,IBD13,IBADD,IBNM,IBPH,IBDN,IBCNT,IBAGAIN  S (IBADD,IBNM,IBPH)=""  ; MAIN ; -- determine address for company for type bill  ;  S IBD0=$G(^DIC(36,+$G(IBCNS),0)) I IBD0="" G MAINQ  S IBD13=$G(^DIC(36,IBCNS,.13))  ;  ; -- get name, main address, phone number  S IBNM=$P(IBD0,U,1),IBPH=$P(IBD13,U,1),IBADD=$G(^DIC(36,+IBCNS,.11))  ;  ; -- if process the same co. more than once you are in an infinate loop  I $D(IBCNT(IBCNS)) G MAINQ ;already processed this company use main add  S IBCNT(IBCNS)=""  ;  ; -- type of bill  I $G(IBATYP)'="",$T(@IBATYP)'="" D @IBATYP I $D(IBAGAIN) K IBAGAIN G MAIN  ;  ; -- return address MAINQ S IBNM=IBNM\_U\_IBPH\_U\_IBADD  Q IBNM  ; VER ; -- verification phone number  I $P(IBD13,U,4)'="" S IBPH=$P(IBD13,U,4)  Q  ; BILL ; -- billing phone number  I $P(IBD13,U,2)'="" S IBPH=$P(IBD13,U,2)  Q  ; PCERT ; -- precertification phone number  I $P(IBD13,U,3)'="" S IBPH=$P(IBD13,U,3)  ;  ; -- if other company processes precerts start again  I $P(IBD13,"^",9) S IBCNS=$P(IBD13,"^",9) S IBAGAIN=1  Q  ; INP ; -- inpatient phone number  I $P(IBD13,U,5)'="" S IBPH=$P(IBD13,U,5)  ;  ; -- see if there is an inpatient address, use if state is there  S IBDN=$G(^DIC(36,+IBCNS,.12)) I $P(IBDN,"^",5) S IBADD=IBDN  ;  ; -- if other company processes claims start again  I $P(IBDN,"^",7) S IBCNS=$P(IBDN,"^",7) S IBAGAIN=1  Q  ; OPT ; -- outpatient phone number  I $P(IBD13,U,6)'="" S IBPH=$P(IBD13,U,6)  ;  ; -- see if there is an outpatient address, use if state is there  S IBDN=$G(^DIC(36,+IBCNS,.16)) I $P(IBDN,"^",5) S IBADD=IBDN  ;  ; -- if other company processes claims start again  I $P(IBDN,"^",7) S IBCNS=$P(IBDN,"^",7) S IBAGAIN=1  Q  ; RX ; -- prescription phone number  I $P(IBD13,U,11)'="" S IBPH=$P(IBD13,U,11)  ;  ; -- see if there is an prescription address, use if state is there  S IBDN=$G(^DIC(36,+IBCNS,.18)) I $P(IBDN,"^",5) S IBADD=IBDN  ;  ; -- if other company processes claims start again  I $P(IBDN,"^",7) S IBCNS=$P(IBDN,"^",7) S IBAGAIN=1  Q  ; APL ; -- appeals phone number  I $P(IBD13,U,7)'="" S IBPH=$P(IBD13,U,7)  ;  ; -- see if there is an appeals address, use if state is there  S IBDN=$G(^DIC(36,+IBCNS,.14)) I $P(IBDN,"^",5) S IBADD=IBDN  ;  ; -- if other company processes claims start again  I $P(IBDN,"^",7) S IBCNS=$P(IBDN,"^",7) S IBAGAIN=1  Q  ; INQ ; -- inquiry phone number  I $P(IBD13,U,8)'="" S IBPH=$P(IBD13,U,8)  ;  ; -- see if there is an outpatient address, use if state is there  S IBDN=$G(^DIC(36,+IBCNS,.15)) I $P(IBDN,"^",5) S IBADD=IBDN  ;  ; -- if other company processes claims start again  I $P(IBDN,"^",7) S IBCNS=$P(IBDN,"^",7) S IBAGAIN=1  Q | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBJTU3 ;ALB/ARH - TPI UTILITIES - INS ADDRESS ; 2/14/95  ;;2.0;INTEGRATED BILLING;\*\*39,80,**592**\*\*;21-MAR-94  ;;Per VHA Directive 10-93-142, this routine should not be modified.  ; BADD(IBIFN) ; returns mailing address for bill  ; returns: COMPANY NAME ^ PHONE NUMBER ^ STR 1 ^ STR 2 ^ STR 3 ^ CITY ^ STATE ^ ZIP ^ ^ FAX #  N DFN,IBX,IBCNS,IBCDFN,IBTYP  ;  S IBX="",DFN=$G(^DGCR(399,+$G(IBIFN),0))  S IBTYP=$P(DFN,U,5),DFN=+$P(DFN,U,2) I 'DFN G BADDQ  S IBCNS=$G(^DGCR(399,+IBIFN,"MP")) I 'IBCNS G BADDQ  S IBCDFN=$P(IBCNS,U,2) I +IBCDFN S IBCNS=+$G(^DPT(DFN,.312,+IBCDFN,0))  ;  ; -- if send to employer and state defined, return employer address  I +IBCDFN S IBCDFN=$G(^DPT(DFN,.312,+IBCDFN,2)) I +IBCDFN,+$P(IBCDFN,U,6) D  G BADDQ  . S IBX=$P(IBCDFN,U,9)\_U\_$P(IBCDFN,U,8)\_U\_$P(IBCDFN,U,2,7)  ;  S IBTYP=$S(IBTYP<3:"INP",1:"OPT")  S IBX=$$INSADD(+IBCNS,IBTYP)  ; BADDQ Q IBX  ;  ; INSADD(IBCNS,IBATYP) ; returns specific type of address/phone # for an insurance company, follows ptrs to company responsible  ; returns: COMPANY NAME ^ PHONE NUMBER ^ STR 1 ^ STR 2 ^ STR 3 ^ CITY ^ STATE ^ ZIP ^ ^ FAX #  ; if type does not have an address or phone number then main mailing addr/ph # is returned  ;  N IBD0,IBD13,IBADD,IBNM,IBPH,IBDN,IBCNT,IBAGAIN  S (IBADD,IBNM,IBPH)=""  ; MAIN ; -- determine address for company for type bill  ;  S IBD0=$G(^DIC(36,+$G(IBCNS),0)) I IBD0="" G MAINQ  S IBD13=$G(^DIC(36,IBCNS,.13))  ;  ; -- get name, main address, phone number  S IBNM=$P(IBD0,U,1),IBPH=$P(IBD13,U,1),IBADD=$G(^DIC(36,+IBCNS,.11))  ;  ; -- if process the same co. more than once you are in an infinate loop  I $D(IBCNT(IBCNS)) G MAINQ ;already processed this company use main add  S IBCNT(IBCNS)=""  ;  ; -- type of bill  I $G(IBATYP)'="",$T(@IBATYP)'="" D @IBATYP I $D(IBAGAIN) K IBAGAIN G MAIN  ;  ; -- return address MAINQ S IBNM=IBNM\_U\_IBPH\_U\_IBADD  Q IBNM  ; VER ; -- verification phone number  I $P(IBD13,U,4)'="" S IBPH=$P(IBD13,U,4)  Q  ; BILL ; -- billing phone number  I $P(IBD13,U,2)'="" S IBPH=$P(IBD13,U,2)  Q  ; PCERT ; -- precertification phone number  I $P(IBD13,U,3)'="" S IBPH=$P(IBD13,U,3)  ;  ; -- if other company processes precerts start again  I $P(IBD13,"^",9) S IBCNS=$P(IBD13,"^",9) S IBAGAIN=1  Q  ; INP ; -- inpatient phone number  I $P(IBD13,U,5)'="" S IBPH=$P(IBD13,U,5)  ;  ; -- see if there is an inpatient address, use if state is there  S IBDN=$G(^DIC(36,+IBCNS,.12)) I $P(IBDN,"^",5) S IBADD=IBDN  ;  ; -- if other company processes claims start again  I $P(IBDN,"^",7) S IBCNS=$P(IBDN,"^",7) S IBAGAIN=1  Q  ; OPT ; -- outpatient phone number  **;JWS;IB\*2.0\*592;Dental Insurance mailing address  I $$FT^IBCEF(IBIFN)=7 D  Q  . I $P($G(^DIC(36,+IBCNS,.19)),"^",11)'="" S IBPH=$P(^(.19),"^",11)  . I $P($G(^DIC(36,+IBCNS,.19)),"^",5) S IB02=$P(^(.19),"^",1,6)  . I $P($G(^DIC(36,+IBCNS,.19)),"^",7) S IBCNS=$P(^(.19),"^",7) S IBAGAIN=1**  I $P(IBD13,U,6)'="" S IBPH=$P(IBD13,U,6)  ;  ; -- see if there is an outpatient address, use if state is there  S IBDN=$G(^DIC(36,+IBCNS,.16)) I $P(IBDN,"^",5) S IBADD=IBDN  ;  ; -- if other company processes claims start again  I $P(IBDN,"^",7) S IBCNS=$P(IBDN,"^",7) S IBAGAIN=1  Q  ; RX ; -- prescription phone number  I $P(IBD13,U,11)'="" S IBPH=$P(IBD13,U,11)  ;  ; -- see if there is an prescription address, use if state is there  S IBDN=$G(^DIC(36,+IBCNS,.18)) I $P(IBDN,"^",5) S IBADD=IBDN  ;  ; -- if other company processes claims start again  I $P(IBDN,"^",7) S IBCNS=$P(IBDN,"^",7) S IBAGAIN=1  Q  ; APL ; -- appeals phone number  I $P(IBD13,U,7)'="" S IBPH=$P(IBD13,U,7)  ;  ; -- see if there is an appeals address, use if state is there  S IBDN=$G(^DIC(36,+IBCNS,.14)) I $P(IBDN,"^",5) S IBADD=IBDN  ;  ; -- if other company processes claims start again  I $P(IBDN,"^",7) S IBCNS=$P(IBDN,"^",7) S IBAGAIN=1  Q  ; INQ ; -- inquiry phone number  I $P(IBD13,U,8)'="" S IBPH=$P(IBD13,U,8)  ;  ; -- see if there is an outpatient address, use if state is there  S IBDN=$G(^DIC(36,+IBCNS,.15)) I $P(IBDN,"^",5) S IBADD=IBDN  ;  ; -- if other company processes claims start again  I $P(IBDN,"^",7) S IBCNS=$P(IBDN,"^",7) S IBAGAIN=1  Q | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBY592PR | | | | | | | | |
| **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 [#364.5]  IB FORM SKELETON DEFINITION [#364.6]  IB FORM FIELD CONTENT [#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)** | | | | | | | | | |
| IBY592PR ;EDE/JWS - Pre-Installation for IB patch 592 ; 2/28/17 4:33pm  ;;2.0;INTEGRATED BILLING;\*\*592\*\*;21-MAR-94;Build 119  ;;Per VA Directive 6402, this routine should not be modified.  ;  ; delete all output formatter (O.F.) data elements included in build  D DELOF  Q  ; INCLUDE(FILE,Y) ; function to determine if O.F. entry should be included in the build  ; FILE=5,6,7 indicating file 364.x  ; Y=ien to file  NEW OK,LN,TAG,DATA  S OK=0  F LN=2:1 S TAG="ENT"\_FILE\_"+"\_LN,DATA=$P($T(@TAG),";;",2) Q:DATA=""  I $F(DATA,U\_Y\_U) S OK=1 Q  Q OK  ;  ;Delete edited entries to insure clean install of new entries  ;Delete obsolete entries. DELOF   ; Delete included OF entries  NEW FILE,DIK,LN,TAG,TAGLN,DATA,PCE,DA,Y  F FILE=5,6,7 S DIK="^IBA(364."\_FILE\_"," D  . F TAG="ENT"\_FILE,"DEL"\_FILE D  .. F LN=2:1 S TAGLN=TAG\_"+"\_LN,DATA=$P($T(@TAGLN),";;",2) Q:DATA=""  D  ... F PCE=2:1 S DA=$P(DATA,U,PCE) Q:'DA  I $D(^IBA("364."\_FILE,DA,0)) D ^DIK  Q  ;  ; Example for ENT5, ENT6, ENT7, DEL5, DEL6, and DEL7:  ;;^195^254^259^269^324^325^  ; Note: Must have beginning and ending up-carat  ;  ;-----------------------------------------------------------------------  ; 364.5 entries modified:  ;  ; 83 - N-ATT/REND PHYSICIAN NAME  ; 97 - N-NON-INSTITUTIONAL CLAIM TYPE  ; 249 - N-ATT/REND PHYSICIAN SPEC  ; 261 - N-ATT/REND PROVIDER ID  ; 370 - N-ATT/REND PHYSICIAN NAME BR  ; 378 - N-ORTHO BANDING QUALIFIER  ; 379 - N-ORTHO BANDIN DATE  ; 380 - N-ORTHO TX MTHS COUNT  ; 381 - N-ORTHO TX MTHS COUNT TRANSFER  ; 382 - N-ORTHO TX INDICATOR  ; 383 - N-TOOTH NUMBER  ; 384 - N-HCFA J430D SERV LINE (EDI)  ; ENT5 ; OF entries in file 364.5 to be included  ;  ;;^83^97^249^261^370^378^379^380^381^382^383^384^  ;  ;-----------------------------------------------------------------------  ; 364.6 entries modified:  ;  ; 2258 - seg 62, 1-RECORD ID 'DN1 '  ; 2259 - seq 62, 2-ORTHO BANDING QUALIFIER  ; 2260 - seq 62, 3-ORTHO BANDING DATE  ; 2261 - seq 62, 4-ORTHO TX MONTHS COUNT  ; 2262 - seq 62, 5-ORTHO TX MTHS CT TRANS  ; 2263 - seq 62, 6-ORTHO TX INDICATOR  ; 2264 - seq 63, 1-RECORD ID 'DN2 '  ; 2266 - seq 63, 2-TOOTH NUMBER  ; 2267 - seq 63, 3-TOOTH STATUS CODE  ; 2268 - seq 63, 4-TOOTH CODE LIST QUALIFIER  ; 2269 - seq 101, 8-REFERRING PROVIDER CODE  ; 2270 - seq 101, 9-REFERRING PROVIDER CODE QUALIFIER  ; 2271 - seq 101, 10-REFERRING PROVIDER TAXONOMY  ; 2272 - seq 104.6, 1-RECORD ID 'OPRB'  ; 2273 - seq 104.6, 2-ASST SURGEON QUALIFIER  ; 2274 - seq 104.6, 3-ASST SURGEON TYPE  ; 2275 - seq 104.6, 4-ASST SURGEON LAST NAME  ; 2276 - seq 104.6, 5-ASST SURGEON FIRST NAME  ; 2277 - seq 104.6, 6-ASST SURGEON MIDDLE NAME  ; 2278 - seq 104.6, 7-ASST SURGEON NAME SUFFIX  ; 2279 - seq 104.6, 8-ASST SURGEON PRIMARY ID QUALIFIER  ; 2280 - seq 104.6, 9-ASST SURGEON PRIMARY ID  ; 2281 - seq 104.6, 10-ASST SURGEON TAXONOMY QUALIFIER  ; 2282 - seq 104.6, 11-ASST SURGEON TAXONOMY  ; 2283 - seq 104.6, 1.5-OPRB Setup  ; 2284 - seq 104.6, 99.5-OPRG Cleanup  ; 2285 - seq 104.61, 1-RECORD ID 'OPRC'  ; 2286 - seq 104.61, 2-ASST SURGEON SEC ID QUALIFIER(1)  ; 2287 - seq 104.61, 3-ASST SURGEON SEC ID (1)  ; 2288 - seq 104.61, 1.5-OPRC Setup  ; 2289 - seq 104.61, 4-ASST SURGEON SEC ID QIALIFIER(2)  ; 2290 - seq 104.61, 5-ASST SURGEON SEC ID (2)  ; 2291 - seq 104.61, 6-ASST SURGEON SEC ID QUALIFIER(3)  ; 2292 - seq 104.61, 7-ASST SURGEON SEC ID (3)  ; 2293 - seq 104.61, 8-ASST SURGEON SEC ID QIALIFIER(4)  ; 2294 - seq 104.61, 9-ASST SURGEON SEC ID (4)  ; 2295 - seq 104.61, 99.5-OPRC Cleanup  ; 2296 - seq 178.1, 1-RECORD ID 'OP10'  ; 2297 - seq 178.1, 1.5-OP10 Setup  ; 2298 - seq 178.1, 2-PAYER RESPONSIBILITY SEQ # CODE  ; 2299 - seq 178.1, 3-OTHER PAYER ASST SURGEON ENTITY ID  ; 2300 - seq 178.1, 4-OTHER PAYER ASST SURGEON ENTITY QUAL  ; 2301 - seq 178.1, 5-OTHER PAYER ASST SURGEON SEC ID QUAL(1)  ; 2302 - seq 178.1, 6-OTHER PAYER ASST SURGEON SEC ID(1)  ; 2303 - seq 178.1, 7-OTHER PAYER ASST SURGEON SEC ID QUAL(2)  ; 2304 - seq 178.1, 8-OTHER PAYER ASST SURGEON SEC ID(2)  ; 2305 - seq 178.1, 9-OTHER PAYER ASST SURGEON SEC ID QUAL(3)  ; 2306 - seq 178.1, 10-OTHER PAYER ASST SURGEON SEC ID(3)  ; 2307 - seq 178.1, 99.5-OP10 Cleanup  ; 2308 - seq 186, 1-RECORD ID 'DEN '  ; 2309 - seq 186, 2-SERVICE LINE #  ; 2310 - seq 186, 3-DATE/TIME QUALIFIER  ; 2311 - seq 186, 4-SERVICE DATE  ; 2312 - seq 186, 5-SERVICE ID QUALIFIER  ; 2313 - seq 186, 6-PROCEDURE CODE  ; 2314 - seq 186, 7-PROCEDURE MODIFIER(1)  ; 2315 - seq 186, 8-PROCEDURE MODIFIER(2)  ; 2316 - seq 186, 9-PROCEDURE MODIFIER(3)  ; 2317 - seq 186, 10-PROCEDURE MODIFIER(4)  ; 2318 - seq 186, 11-PROCEDURE CODE DESCRIPTION  ; 2319 - seq 186, 12-LINE ITEM CHARGE AMT  ; 2320 - seq 186, 13-PLACE OF SERVICE CODE  ; 2321 - seq 186, 14-ORAL CAVITY DESIGNATION(1)  ; 2322 - seq 186, 15-ORAL CAVITY DESIGNATION(2)  ; 2323 - seq 186, 16-ORAL CAVITY DESIGNATION(3)  ; 2324 - seq 186, 17-ORAL CAVITY DESIGNATION(4)  ; 2325 - seq 186, 18-ORAL CAVITY DESIGNATION(5)  ; 2326 - seq 186, 19-PROSTHESIS CROWN INLAY CODE  ; 2327 - seq 186.1, 1-RECORD ID 'DEN1'  ; 2328 - seq 186.1, 2-SERVICE LINE #  ; 2329 - seq 186.1, 3-PROCEDURE COUNT  ; 2330 - seq 186.1, 4-DIAGNOSIS CODE POINTER(1)  ; 2331 - seq 186.1, 5-DIAGNOSIS CODE POINTER(2)  ; 2332 - seq 186.1, 6-DIAGNOSIS CODE POINTER(3)  ; 2333 - seq 186.1, 7-DIAGNOSIS CODE POINTER(4)  ; 2334 - seq 186.1, 8-PRIOR PLACEMENT DATE QUALIFIER  ; 2335 - seq 186.1, 9-PRIOR PLACEMENT DATE  ; 2336 - seq 186.1, 10-ORTHO BANDING DATE QUALIFIER  ; 2337 - seq 186.1, 11-ORTHO BANDING DATE  ; 2338 - seq 186.1, 12-REPLACEMENT DATE QUALIFIER  ; 2339 - seq 186.1, 13-REPLACEMENT DATE  ; 2340 - seq 186.1, 14-TREATMENT START DATE QUALIFIER  ; 2341 - seq 186.1, 15-TREATMENT START DATE  ; 2342 - seq 186.1, 16-TREATMENT COMPLETION DATE QUALIFIER  ; 2343 - seq 186.1, 17-TREATMENT COMPLETION DATE  ; 2344 - seq 186.2, 1-RECORD ID 'DEN2'  ; 2345 - seq 186.2, 2-SERVICE LINE #  ; 2346 - seq 186.2, 3-TOOTH CODE QUALIFIER  ; 2347 - seq 186.2, 4-TOOTH CODE  ; 2348 - seq 186.2, 5-TOOTH SURFACE(1)  ; 2349 - seq 186.2, 6-TOOTH SURFACE(2)  ; 2350 - seq 186.2, 7-TOOTH SURFACE(3)  ; 2351 - seq 186.2, 8-TOOTH SURFACE(4)  ; 2352 - seq 186.2, 9-TOOTH SURFACE(5)  ; 2353 - seq 194.5, 1-RECORD ID 'LSUR'  ; 2354 - seq 194.5, 2-SERVICE LINE COUNTER  ; 2355 - seq 194.5, 3-ASST SURGEON QUALIFIER  ; 2356 - seq 194.5, 4-ASST SURGEON LAST NAME  ; 2357 - seq 194.5, 5-ASST SURGEON FIRST NAME  ; 2358 - seq 194.5, 6-ASST SURGEON MIDDLE NAME  ; 2359 - seq 194.5, 7-ASST SURGEON NAME SUFFIX  ; 2360 - seq 194.5, 8-ASST SURGEON PRIMARY ID QUALIFIER  ; 2361 - seq 194.5, 9-ASST SURGEON PRIMARY ID  ; 2370 - seq 194.5, 1.9-LSUR DATA EXTRACT  ; 2372 - seq 194.6, 1-RECORD ID 'LSUR1'  ; 2373 - seq 194.6, 1.9-LSUR1 DATA EXTRACT  ; 2374 - seq 194.6, 2-SERVICE LINE COUNTER  ; 2375 - seq 194.6, 3-ASST SURGEON SECONDARY ID QUALIFIER(1)  ; 2376 - seq 194.6, 4-ASST SURGEON SECONDARY ID(1)  ; 2377 - seq 194.6, 5-ASST SURGEON SECONDARY ID QUALIFIER(2)  ; 2378 - seq 194.6, 6-ASST SURGEON SECONDARY ID(2)  ; 2379 - seq 194.6, 7-ASST SURGEON SECONDARY ID QUALIFIER(3)  ; 2380 - seq 194.6, 8-ASST SURGEON SECONDARY ID(3)  ; 2381 - seq 194.6, 9-ASST SURGEON TAXONOMY QUALIFIER  ; 2382 - seq 194.6, 10-ASST SURGEON TAXONOMY CODE  ; ENT6 ; O.F. entries in file 364.6 to be included  ;  ;;^2258^2259^2260^2261^2262^2263^2264^2266^2267^2268^2269^2270^2271^2272^  ;;^2273^2274^2275^2276^2277^2278^2279^2280^2281^2282^2283^2284^2285^2286^  ;;^2287^2288^2289^2290^2291^2292^2293^2294^2295^2296^2297^2298^2299^2300^  ;;^2301^2302^2303^2304^2305^2306^2307^2308^2309^2310^2311^2312^2313^2314^  ;;^2315^2316^2317^2318^2319^2320^2321^2322^2323^2324^2325^2326^2327^2328^  ;;^2329^2330^2331^2332^2333^2334^2335^2336^2337^2338^2339^2340^2341^2342^  ;;^2343^2344^2345^2346^2347^2348^2349^2350^2351^2352^2353^2354^2355^2356^  ;;^2357^2358^2359^2360^2361^2370^2372^2373^2374^2375^2376^2377^2378^2379^  ;;^2380^2381^2382^  ;-----------------------------------------------------------------------  ; 364.7 entries modified:  ;  ; ENT7 ; O.F. entries in file 364.7 to be included  ;  ;;^1425^1426^1460^1461^1462^1463^1464^1465^1466^1467^1468^1469^1470^1471^  ;;^1472^1473^1474^1475^1476^1477^1478^1479^1480^1481^1482^1483^1484^1485^  ;;^1486^1487^1488^1489^1490^1491^1492^1493^1494^1495^1496^1497^1498^1499^  ;;^1505^1506^1507^1508^1509^1510^1511^1512^1513^1514^1515^1516^1517^1528^  ;;^1548^1549^1552^1553^1554^1555^1556^1557^1558^1559^1560^1580^1581^1582^  ;;^1583^1584^1585^1586^1587^1588^1589^1590^1591^1592^1593^1594^1595^1596^  ;;^1597^1598^1599^1600^1601^1602^1603^1604^1605^1606^1607^1608^1609^1614^  ;;^1688^1689^1728^1729^1730^1731^1957^1958^1959^1960^1961^1962^1963^1964^  ;;^1965^1966^1967^31^37^1006^1023^1015^16^127^75^1008^1727^854^  ;  ;-----------------------------------------------------------------------  ; 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)  ;  ;;  ;  ;----------------------------------------------------------------------- | | | | | | | | | |

1. The Output Formatter control files will be changed in order to create the 837D transaction. The following are entries that will need modified and new(\*) entries to file 364.5 IB DATA ELEMENT DEFINITION.

NUMBER: **83**

NAME: **N-ATT/REND PHYSICIAN NAME**

SECURITY LEVEL: **NATIONAL,NO EDIT**

TYPE OF ELEMENT: **EXTRACTED VIA CODE**

ELEMENT CATEGORY: **INDIVIDUAL ELEMENT**

BASE FILE: **BILL/CLAIMS**

EXTRACT CODE: **N IBZ,IBT S IBT=$S($$FT^IBCEF(IBXIEN)=2:3,$$FT^IBCEF(IBXIEN)=7:3**

**,1:4) D GETPRV^IBCEU(IBXIEN,IBT,.IBZ) S**

**IBXDATA=$P($G(IBZ(IBT,1)),U)\_U\_$P($G(IBZ**

**(IBT,1)),U,3)\_U\_$P($G(IBZ(IBT,1)),U,4)**

DESCRIPTION: **The name of the attending (inpatient) or rendering**

**(outpatient) physician for bill entry IBXIEN. Inpatient**

**type=4, outpatient=3 1st '^'-piece is name, 2nd is new**

**person file ien, 3rd is credentials**

NUMBER: **97**

NAME: **N-NON-INSTITUTIONAL CLAIM TYPE**

SECURITY LEVEL: **NATIONAL,NO EDIT**

TYPE OF ELEMENT: **EXTRACTED VIA CODE**

ELEMENT CATEGORY: **INDIVIDUAL ELEMENT**

BASE FILE: **BILL/CLAIMS**

EXTRACT CODE: **S IBXDATA="" I**

**$$FT^IBCEF(IBXIEN)=2!($$FT^IBCEF(IBXIEN)=7) S IBX**

**DATA=$S($O(^IBA(362.4,"AIFN"\_IBXIEN,0)):"RX",1:"MD")**

DESCRIPTION: **If an outpatient bill-'MD'; if a prescription bill-'RX'**

**for bill entry IBXIEN.**

NUMBER: **249**

NAME: **N-ATT/REND PHYSICIAN SPEC**

SECURITY LEVEL: **NATIONAL,NO EDIT**

TYPE OF ELEMENT: **EXTRACTED VIA CODE**

ELEMENT CATEGORY: **INDIVIDUAL ELEMENT**

BASE FILE: **BILL/CLAIMS**

EXTRACT CODE: **N IBZ,IBI S**

**IBI=$S($$FT^IBCEF(IBXIEN)=2!($$FT^IBCEF(IBXIEN)=7):3**

**,1:4) D GETPRV^IBCEU(IBXIEN,IBI,.IBZ) S**

**IBXDATA=$$SPEC^IBCEU($P($G(IBZ(IBI,1)),U**

**,3),+$G(^DGCR(399,IBXIEN,"U")))**

DESCRIPTION: **Find the specialty from the VA code of the person class**

**attached to the attending/rendering provider for bill**

**IBXIEN, if it can be determined.**

NUMBER: **261**

NAME: **N-ATT/REND PROVIDER ID**

SECURITY LEVEL: **NATIONAL,NO EDIT**

TYPE OF ELEMENT: **EXTRACTED VIA CODE**

ELEMENT CATEGORY: **INDIVIDUAL ELEMENT**

BASE FILE: **BILL/CLAIMS**

EXTRACT CODE: **N IBZ,IBI,IBS S**

**IBS=+$$COBN^IBCEF(IBXIEN),IBI=$S($$FT^IBCEF(IBXI**

**EN)=2!($$FT^IBCEF(IBXIEN)=7):3,1:4) D**

**GETPRV^IBCEU(IBXIEN,IBI,.IBZ) S IBXDATA=$S**

**($P($G(IBZ(IBI,1,IBS)),U)'="":$P(IBZ(IBI,1,IBS),U),1:$P($G(IBZ(IBI,1,1)),U))**

DESCRIPTION: **The id # of the attending/rendering physician for the**

**current insurance company for bill entry IBXIEN.**

NUMBER: **370**

NAME: **N-ATT/REND PHYSICIAN NAME BR**

SECURITY LEVEL: **NATIONAL,NO EDIT**

TYPE OF ELEMENT: **EXTRACTED VIA CODE**

ELEMENT CATEGORY: **INDIVIDUAL ELEMENT**

BASE FILE: **BILL/CLAIMS**

EXTRACT CODE: **N IBT S**

**IBT=$S($$FT^IBCEF(IBXIEN)=2!($$FT^IBCEF(IBXIEN)=7):3,1:4**

**) S IBXDATA=$$GETPRV^IBCEF83(IBXIEN,,IBT,"A1")\_U\_**

**$$GETPRV^IBCEF83(IBXIEN,,IBT,"A0")\_U\_**

**$$GETPRV^IBCEF83(IBXIEN,,IBT,"A6")**

DESCRIPTION: **The name of the ATTENDING or RENDERING provider for bill**

**entry IBXIEN. The pieces returned are: provider name^vp**

**ien for provider (200/355.93)^credentials.**

NUMBER: **378\***

NAME: **N-ORTHO BANDING QUALIFIER**

SECURITY LEVEL: **NATIONAL,NO EDIT**

TYPE OF ELEMENT: **CONSTANT VALUE**

ELEMENT CATEGORY: **INDIVIDUAL ELEMENT**

BASE FILE: **BILL/CLAIMS**

CONSTANT VALUE: **452**

NUMBER: **379\***

NAME: **N-ORTHO BANDING DATE**

SECURITY LEVEL: **NATIONAL,NO EDIT**

TYPE OF ELEMENT: **NON-MULTIPLE FILEMAN FIELD**

ELEMENT CATEGORY: **INDIVIDUAL ELEMENT**

BASE FILE: **BILL/CLAIMS**

FILEMAN FIELD REFERENCE: **BANDING DATE**

FILEMAN RETURN FORMAT: **INTERNAL**

NUMBER: **380\***

NAME: **N-ORTHO TX MTHS COUNT**

SECURITY LEVEL: **NATIONAL,NO EDIT**

TYPE OF ELEMENT: **NON-MULTIPLE FILEMAN FIELD**

ELEMENT CATEGORY: **INDIVIDUAL ELEMENT**

BASE FILE: **BILL/CLAIMS**

FILEMAN FIELD REFERENCE: **TREATMENT MONTHS COUNT**

FILEMAN RETURN FORMAT: **INTERNAL**

NUMBER: **381\***

NAME: **N-ORTHO TX MTHS COUNT TRANSFER**

SECURITY LEVEL: **NATIONAL,NO EDIT**

TYPE OF ELEMENT: **NON-MULTIPLE FILEMAN FIELD**

ELEMENT CATEGORY: **INDIVIDUAL ELEMENT**

BASE FILE: **BILL/CLAIMS**

FILEMAN FIELD REFERENCE: **TREATMENT MONTHS REMAINING**

FILEMAN RETURN FORMAT: **INTERNAL**

NUMBER: **382\***

NAME: **N-ORTHO TX INDICATOR**

SECURITY LEVEL: **NATIONAL,NO EDIT**

TYPE OF ELEMENT: **NON-MULTIPLE FILEMAN FIELD**

ELEMENT CATEGORY: **INDIVIDUAL ELEMENT**

BASE FILE: **BILL/CLAIMS**

FILEMAN FIELD REFERENCE: **TREATMENT INDICATOR**

FILEMAN RETURN FORMAT: **EXTERNAL**

NUMBER: **383\***

NAME: **N-TOOTH NUMBER**

SECURITY LEVEL: **NATIONAL,NO EDIT**

TYPE OF ELEMENT: **EXTRACTED VIA COD**

ELEMENT CATEGORY: **GROUP ELEMENT**

BASE FILE: **BILL/CLAIMS**

EXTRACT CODE: **S IBXDATA="" I $$FT^IBCEF(IBXIEN)=7 D TNUM^IBCEF12(IBXIEN)**

NUMBER: **384\***

NAME: **N-HCFA J430D SERV LINE (EDI)**

SECURITY LEVEL: **NATIONAL,NO EDIT**

TYPE OF ELEMENT: **EXTRACTED VIA CODE**

ELEMENT CATEGORY: **GROUP ELEMENT**

BASE FILE: **BILL/CLAIMS**

EXTRACT CODE: **S IBXDATA="" I $$FT^IBCEF(IBXIEN)=7 D OUTPT^IBCEF11(IBXIEN,0)**

DESCRIPTION: **The elements of the outpatient services line on the HCFA J430D**

**for bill entry IBXIEN. Format is begin date^end date^place of service**

**code^type of service code^procedure or revenue code^type of**

**code^diagnosis pointers^unit charge^units^modifier pointer ien(s)**

**separated by commas^purchased charge amount^anesthesia**

**minutes^emergency indicator^lap type of service flag. Data is**

**returned in an array IBXDATA(1-n). Also returns the arrays**

**IBXDATA(n,"AUX")=the 'AUX' node for the procedure entry on the**

**claim IBXDATA("ITEM",item type,item pointer)=n^ctr if this data can be**

**gatherd.**

1. The following are additional entries to file 364.6 IB FORM SKELETON DEFINITION. These entries are used by the VistA Output Formatter when generating the 837D transaction.

NUMBER: **2258** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL**: NATIONAL,NO EDIT** PAGE OR SEQUENCE: **62**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **1**  LENGTH: **4**

SHORT DESCRIPTION: **RECORD ID 'DN1 '**  TRANSMIT IGNORES IF NULL: **TRUE**

DATA REQUIRED FOR FIELD: **YES**

NUMBER: **2259** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **62**

FIRST LINE NUMBER: **1**  LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **2**  LENGTH: **3**

SHORT DESCRIPTION: **ORTHO BANDING QUALIFIER**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2260** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **62**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **3** LENGTH: **35**

SHORT DESCRIPTION: **ORTHO BANDING DATE**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2261** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **62**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **4** LENGTH: **15**

SHORT DESCRIPTION: **ORTHO TX MONTHS COUNT**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2262** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **62**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **5** LENGTH: **15**

SHORT DESCRIPTION: **ORTHO TX MTHS CT TRANS**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2263** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **62**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **6** LENGTH: **80**

SHORT DESCRIPTION: **ORTHO TX INDICATOR**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2264** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **63**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **1**  LENGTH: **4**

SHORT DESCRIPTION: **RECORD ID 'DN2 '** TRANSMIT IGNORES IF NULL: **TRUE**

DATA REQUIRED FOR FIELD: **YES**

NUMBER: **2266** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **63**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **2** LENGTH: **50**

SHORT DESCRIPTION: **TOOTH NUMBER** TRANSMIT IGNORES IF NULL: **TRUE**

DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2267** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **63**

FIRST LINE NUMBER: **1**  LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **3** LENGTH: **2**

SHORT DESCRIPTION: **TOOTH STATUS CODE** TRANSMIT IGNORES IF NULL: **TRUE**

DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2268** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **63**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **4** LENGTH: **3**

SHORT DESCRIPTION: **TOOTH CODE LIST QUALIFIER**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2269** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **101**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **8** LENGTH: **3**

SHORT DESCRIPTION: **REFERRING PROVIDER CODE**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2270** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **101**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **9** LENGTH: **3**

SHORT DESCRIPTION: **REFERRING PROVIDER CODE QUALIFIER**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2271** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **101**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **10** LENGTH: **50**

SHORT DESCRIPTION: **REFERRING PROVIDER TAXONOMY**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2272** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **104.6**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **1** LENGTH: **4**

SHORT DESCRIPTION: **RECORD ID 'OPRB'** TRANSMIT IGNORES IF NULL: **TRUE**

DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2283** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **104.6**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **1.5**  SHORT DESCRIPTION: **OPRB Setup**

CALCULATE ONLY OR OUTPUT: **CALCULATE ONLY**

NUMBER: **2273** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **104.6**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **2** LENGTH: **3**

SHORT DESCRIPTION: **ASST SURGEON QUALIFIER**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2274** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **104.6**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **3** LENGTH: **1**

SHORT DESCRIPTION: **ASST SURGEON TYPE** TRANSMIT IGNORES IF NULL: **TRUE**

DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2275** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **104.6**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **4** LENGTH: **60**

SHORT DESCRIPTION: **ASST SURGEON LAST NAME**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2276** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **104.6**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **5** LENGTH: **35**

SHORT DESCRIPTION: **ASST SURGEON FIRST NAME**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2277** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **104.6**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **6** LENGTH: **25**

SHORT DESCRIPTION: **ASST SURGEON MIDDLE NAME**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2278** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **104.6**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **7** LENGTH: **10**

SHORT DESCRIPTION: **ASST SURGEON NAME SUFFIX**

TRANSMIT IGNORES IF NULL: **TRUE**  DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2279** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **104.6**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **8** LENGTH: **2**

SHORT DESCRIPTION: **ASST SURGEON PRIMARY ID QUALIFIER**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2280** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **104.6**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **9** LENGTH: **80**

SHORT DESCRIPTION: **ASST SURGEON PRIMARY ID**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2281** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **104.6**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **10** LENGTH: **3**

SHORT DESCRIPTION: **ASST SURGEON TAXONOMY QUALIFIER**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2282** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **104.6**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **11** LENGTH: **10**

SHORT DESCRIPTION: **ASST SURGEON TAXONOMY**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2284** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **104.6**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **99.5** SHORT DESCRIPTION: **OPRB Cleanup**

CALCULATE ONLY OR OUTPUT: **CALCULATE ONLY**

NUMBER: **2285** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **104.61**

FIRST LINE NUMBER: **1**  LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **1**  LENGTH: **4**

SHORT DESCRIPTION: **RECORD ID 'OPRC'** TRANSMIT IGNORES IF NULL: **TRUE**

DATA REQUIRED FOR FIELD: **YES**

NUMBER: **2288** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **104.61**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **1.5** SHORT DESCRIPTION: **OPRC Setup**

CALCULATE ONLY OR OUTPUT: **CALCULATE ONLY**

NUMBER: **2286** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **104.61**

FIRST LINE NUMBER: **1**  LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **2** LENGTH: **3**

SHORT DESCRIPTION: **ASST SURGEON SEC ID QUALIFIER (1)**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2287** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **104.61**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **3** LENGTH: **50**

SHORT DESCRIPTION: **ASST SURGEON SEC ID (1)**

TRANSMIT IGNORES IF NULL: **TRUE**  DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2289** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **104.61**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **4** LENGTH: **3**

SHORT DESCRIPTION: **ASST SURGEON SEC ID QUALIFIER (2)**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2290**  BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **104.61**

FIRST LINE NUMBER: **1**  LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **5**  LENGTH: **50**

SHORT DESCRIPTION: **ASST SURGEON SEC ID (2)**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2291**  BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **104.61**

FIRST LINE NUMBER: **1**  LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **6** LENGTH: **3**

SHORT DESCRIPTION: **ASST SURGEON SEC ID QUALIFIER (3)**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2292** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **104.61**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **7** LENGTH: **50**

SHORT DESCRIPTION: **ASST SURGEON SEC ID (3)**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2293** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **104.61**

FIRST LINE NUMBER: **1**  LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **8**  LENGTH: **3**

SHORT DESCRIPTION: **ASST SURGEON SEC ID QUALIFIER (4)**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2294** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **104.61**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **9** LENGTH: **50**

SHORT DESCRIPTION: **ASST SURGEON SEC ID (4)**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2295** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **104.61**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **99.5** SHORT DESCRIPTION: **OPRC Cleanup**

CALCULATE ONLY OR OUTPUT: **CALCULATE ONLY**

NUMBER: **2296** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **178.1**

FIRST LINE NUMBER: **1**  LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **1**  LENGTH: **4**

SHORT DESCRIPTION: **RECORD ID 'OP10'** TRANSMIT IGNORES IF NULL: **TRUE**

DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2297** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **178.1**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **1.5** SHORT DESCRIPTION: **OP10 Setup**

CALCULATE ONLY OR OUTPUT: **CALCULATE ONLY**

NUMBER: **2298**  BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **178.1**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **2**  LENGTH: **1**

SHORT DESCRIPTION: **PAYER RESPONSIBILITY SEQ # CODE**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2299** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **178.1**

FIRST LINE NUMBER: **1**  LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **3** LENGTH: **3**

SHORT DESCRIPTION: **OTHER PAYER ASST SURGEON ENTITY ID**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2300** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **178.1**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **4** LENGTH: **1**

SHORT DESCRIPTION: **OTHER PAYER ASST SURGEON ENTITY QUAL**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2301** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **178.1**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **5** LENGTH: 3

SHORT DESCRIPTION: **OTHER PAYER ASST SURGEON SEC ID QUAL(1)**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2302** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **178.1**

FIRST LINE NUMBER: **1**  LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **6**  LENGTH: **50**

SHORT DESCRIPTION: **OTHER PAYER ASST SURGEON SEC ID(1)**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2303** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **178.1**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **7** LENGTH: **3**

SHORT DESCRIPTION: **OTHER PAYER ASST SURGEON SEC ID QUAL(2)**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2304** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **178.1**

FIRST LINE NUMBER: **1**  LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **8**  LENGTH: **50**

SHORT DESCRIPTION: **OTHER PAYER ASST SURGEON SEC ID(2)**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2305** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **178.1**

FIRST LINE NUMBER: **1**  LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **9** LENGTH: **3**

SHORT DESCRIPTION: **OTHER PAYER ASST SURGEON SEC ID QUAL(3)**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2306**  BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **178.1**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **10**  LENGTH: **50**

SHORT DESCRIPTION: **OTHER PAYER ASST SURGEON SEC ID(3)**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2307** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **178.1**

FIRST LINE NUMBER: **1**  LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **99.5** SHORT DESCRIPTION: **OP10 Cleanup**

CALCULATE ONLY OR OUTPUT: **CALCULATE ONLY**

NUMBER: **2308** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **186**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **1**  LENGTH: **4**

SHORT DESCRIPTION: **RECORD ID 'DEN '** TRANSMIT IGNORES IF NULL: **TRUE**

DATA REQUIRED FOR FIELD: **YES**

NUMBER: **2309** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **186**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **2** LENGTH: **6**

SHORT DESCRIPTION: **SERVICE LINE #** TRANSMIT IGNORES IF NULL: **TRUE**

DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2310** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **186**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **3**  LENGTH: **3**

SHORT DESCRIPTION: **DATE/TIME QUALIFIER**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2311** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **186**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **4**  LENGTH: **35**

SHORT DESCRIPTION: **SERVICE DATE** TRANSMIT IGNORES IF NULL: **TRUE**

DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2312** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **186**

FIRST LINE NUMBER: **1**  LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **5** LENGTH: **2**

SHORT DESCRIPTION: **SERVICE ID QUALIFIER**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2313** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **186**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **6** LENGTH: **48**

SHORT DESCRIPTION: **PROCEDURE CODE** TRANSMIT IGNORES IF NULL: **TRUE**

DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2314** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **186**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **7** LENGTH: **2**

SHORT DESCRIPTION: **PROCEDURE MODIFIER(1)**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2315** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **186**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **8** LENGTH: **2**

SHORT DESCRIPTION: **PROCEDURE MODIFIER(2)**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2316** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **186**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **9** LENGTH: **2**

SHORT DESCRIPTION: **PROCEDURE MODIFIER(3)**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2317** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **186**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **10** LENGTH: **2**

SHORT DESCRIPTION: **PROCEDURE MODIFIER(4)**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2318** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **186**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **11** LENGTH: **80**

SHORT DESCRIPTION: **PROCEDURE CODE DESCRIPTION**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2319** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **186**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **12** LENGTH: **18**

SHORT DESCRIPTION: **LINE ITEM CHARGE AMT**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2320** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **186**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **13** LENGTH: **2**

SHORT DESCRIPTION: **PLACE OF SERVICE CODE**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2321** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **186**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **14** LENGTH: **3**

SHORT DESCRIPTION: **ORAL CAVITY DESIGNATION(1)**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2322** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **186**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **15** LENGTH: **3**

SHORT DESCRIPTION: **ORAL CAVITY DESIGNATION(2)**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2323** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **186**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **16** LENGTH: **3**

SHORT DESCRIPTION: **ORAL CAVITY DESIGNATION(3)**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2324** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **186**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **17** LENGTH: **3**

SHORT DESCRIPTION: **ORAL CAVITY DESIGNATION(4)**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2325** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **186**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **18** LENGTH: **3**

SHORT DESCRIPTION: **ORAL CAVITY DESIGNATION(5)**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2326** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **186**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **19** LENGTH: **1**

SHORT DESCRIPTION: **PROSTHESIS CROWN INLAY CODE**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2327** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **186.1**

FIRST LINE NUMBER: **1**  LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **1** LENGTH: **4**

SHORT DESCRIPTION: **RECORD ID 'DEN1'** TRANSMIT IGNORES IF NULL: **TRUE**

DATA REQUIRED FOR FIELD: **YES**

NUMBER: **2328**  BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL**: NATIONAL,NO EDIT** PAGE OR SEQUENCE: **186.1**

FIRST LINE NUMBER: **1**  LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **2** LENGTH: **6**

SHORT DESCRIPTION: **SERVICE LINE #** TRANSMIT IGNORES IF NULL: **TRUE**

DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2329** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **186.1**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **3** LENGTH: **15**

SHORT DESCRIPTION: **PROCEDURE COUNT** TRANSMIT IGNORES IF NULL: **TRUE**

DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2330** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **186.1**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **4** LENGTH: **2**

SHORT DESCRIPTION: **DIAGNOSIS CODE POINTER(1)**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2331** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **186.1**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **5**  LENGTH: **2**

SHORT DESCRIPTION: **DIAGNOSIS CODE POINTER(2)**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2332** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **186.1**

FIRST LINE NUMBER: **1**  LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **6** LENGTH: **2**

SHORT DESCRIPTION: **DIAGNOSIS CODE POINTER(3)**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2333** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **186.1**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **7** LENGTH: **2**

SHORT DESCRIPTION: **DIAGNOSIS CODE POINTER(4)**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2334** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **186.1**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **8**  LENGTH: **3**

SHORT DESCRIPTION: **PRIOR PLACEMENT DATE QUALIFIER**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2335** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **186.1**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **9** LENGTH: **35**

SHORT DESCRIPTION: **PRIOR PLACEMENT DATE**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2336** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **186.1**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **10** LENGTH: 3

SHORT DESCRIPTION: **ORTHO BANDING DATE QUALIFIER**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2337**  BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **186.1**

FIRST LINE NUMBER: **1**  LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **11** LENGTH: **35**

SHORT DESCRIPTION: **ORTHO BANDING DATE**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2338** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **186.1**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **12** LENGTH: **3**

SHORT DESCRIPTION: **REPLACEMENT DATE QUALIFIER**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2339** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **186.1**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **13** LENGTH: **35**

SHORT DESCRIPTION: **REPLACEMENT DATE** TRANSMIT IGNORES IF NULL: **TRUE**

DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2340** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **186.1**

FIRST LINE NUMBER: **1**  LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **14** LENGTH: **3**

SHORT DESCRIPTION: **TREATMENT START DATE QUALIFIER**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2341** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **186.1**

FIRST LINE NUMBER: **1**  LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **15**  LENGTH: **35**

SHORT DESCRIPTION: **TREATMENT START DATE**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2342**  BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **186.1**

FIRST LINE NUMBER: **1**  LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **16**  LENGTH: **3**

SHORT DESCRIPTION: **TREATMENT COMPLETION DATE QUALIFIER**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2343** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **186.1**

FIRST LINE NUMBER: **1**  LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **17** LENGTH: **35**

SHORT DESCRIPTION: **TREATMENT COMPLETION DATE**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2344** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **186.2**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **1** LENGTH: **4**

SHORT DESCRIPTION: **RECORD ID 'DEN2'** TRANSMIT IGNORES IF NULL: **TRUE**

DATA REQUIRED FOR FIELD: **YES**

NUMBER: **2345** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **186.2**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **2** LENGTH: **6**

SHORT DESCRIPTION: **SERVICE LINE #** TRANSMIT IGNORES IF NULL: **TRUE**

DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2346** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **186.2**

FIRST LINE NUMBER: **1**  LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **3** LENGTH: **3**

SHORT DESCRIPTION: **TOOTH CODE QUALIFIER**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2347** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **186.2**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **4** LENGTH: **30**

SHORT DESCRIPTION: **TOOTH CODE** TRANSMIT IGNORES IF NULL: **TRUE**

DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2348** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **186.2**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **5** LENGTH: **2**

SHORT DESCRIPTION: **TOOTH SURFACE(1)** TRANSMIT IGNORES IF NULL: **TRUE**

DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2349** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **186.2**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **6** LENGTH: **2**

SHORT DESCRIPTION: **TOOTH SURFACE(2**) TRANSMIT IGNORES IF NULL: **TRUE**

DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2350** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **186.2**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **7** LENGTH: **2**

SHORT DESCRIPTION: **TOOTH SURFACE(3)** TRANSMIT IGNORES IF NULL: **TRUE**

DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2351** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **186.2**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **8** LENGTH: **2**

SHORT DESCRIPTION: **TOOTH SURFACE(4)** TRANSMIT IGNORES IF NULL: **TRUE**

DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2352** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **186.2**

FIRST LINE NUMBER: **1**  LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **9**  LENGTH: **2**

SHORT DESCRIPTION: **TOOTH SURFACE(5)** TRANSMIT IGNORES IF NULL: **TRUE**

DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2353** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **194.5**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **1** LENGTH: **4**

SHORT DESCRIPTION: **N-RECORD 'LSUR'** TRANSMIT IGNORES IF NULL: **TRUE**

DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2370** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **194.5**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **1.9**

SHORT DESCRIPTION: **LSUR DATA EXTRACT**

CALCULATE ONLY OR OUTPUT: **CALCULATE ONLY**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2354** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **194.5**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **2** LENGTH: **6**

SHORT DESCRIPTION: **SERVICE LINE COUNTER**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2355** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **194.5**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **3** LENGTH**: 3**

SHORT DESCRIPTION: **ASST SURGEON QUALIFIER**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2356** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **194.5**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **4** LENGTH: **60**

SHORT DESCRIPTION: **ASST SURGEON LAST NAME**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2357** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **194.5**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **5** LENGTH: **35**

SHORT DESCRIPTION: **ASST SURGEON FIRST NAME**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2358** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **194.5**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: 6 LENGTH: **25**

SHORT DESCRIPTION: **ASST SURGEON MIDDLE NAME**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2359** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **194.5**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **7** LENGTH: **10**

SHORT DESCRIPTION: **ASST SURGEON NAME SUFFIX**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2360** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **194.5**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **8** LENGTH: **2**

SHORT DESCRIPTION: **ASST SURGEON PRIMARY ID QUALIFIER**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2361** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: NATIONAL,NO EDIT PAGE OR SEQUENCE: **194.5**

FIRST LINE NUMBER: **1** LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **9** LENGTH: **10**

SHORT DESCRIPTION: **ASST SURGEON PRIMARY ID**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2372**  BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **194.6**

FIRST LINE NUMBER: **1**  LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **1**  LENGTH: **5**

SHORT DESCRIPTION: **RECORD ID 'LSUR1'** TRANSMIT IGNORES IF NULL: **TRUE**

DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2373**  BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **194.6**

FIRST LINE NUMBER: **1**  LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **1.9**

SHORT DESCRIPTION: **LSUR1 DATA EXTRACT**

CALCULATE ONLY OR OUTPUT: **CALCULATE ONLY**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2374**  BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **194.6**

FIRST LINE NUMBER: **1**  LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **2**  LENGTH: **6**

SHORT DESCRIPTION: **SERVICE LINE COUNTER**

TRANSMIT IGNORES IF NULL: **TRUE**  DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2375** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **194.6**

FIRST LINE NUMBER: **1**  LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **3**  LENGTH: **3**

SHORT DESCRIPTION: **ASST SURGEON SECONDARY ID QUALIFIER(1)**

TRANSMIT IGNORES IF NULL: **TRUE**  DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2376** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **194.6**

FIRST LINE NUMBER: **1**  LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **4**  LENGTH: **15**

SHORT DESCRIPTION: **ASST SURGEON SECONDARY ID(1)**

TRANSMIT IGNORES IF NULL: **TRUE**  DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2377** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **194.6**

FIRST LINE NUMBER: **1**  LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **5**  LENGTH: **3**

SHORT DESCRIPTION: **ASST SURGEON SECONDARY ID QUALIFIER(2)**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2378** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **194.6**

FIRST LINE NUMBER: **1**  LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **6**  LENGTH: **15**

SHORT DESCRIPTION: **ASST SURGEON SECONDARY ID(2)**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2379**  BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **194.6**

FIRST LINE NUMBER: **1**  LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **7**  LENGTH: **3**

SHORT DESCRIPTION: **ASST SURGEON SECONDARY ID QUALIFIER(3)**

TRANSMIT IGNORES IF NULL: **TRUE**  DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2380**  BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **194.6**

FIRST LINE NUMBER: **1**  LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **8**  LENGTH: **15**

SHORT DESCRIPTION: **ASST SURGEON SECONDARY ID(3)**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2381**  BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **194.6**

FIRST LINE NUMBER: **1**  LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **9**  LENGTH: **2**

SHORT DESCRIPTION: **ASST SURGEON TAXONOMY QUALIFIER**

TRANSMIT IGNORES IF NULL: **TRUE**  DATA REQUIRED FOR FIELD: **NO**

NUMBER: **2382** BILL FORM: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** PAGE OR SEQUENCE: **194.6**

FIRST LINE NUMBER: **1**  LOCAL OVERRIDE ALLOWED: **NO**

STARTING COLUMN OR PIECE: **10**  LENGTH: **10**

SHORT DESCRIPTION: **ASST SURGEON TAXONOMY CODE**

TRANSMIT IGNORES IF NULL: **TRUE** DATA REQUIRED FOR FIELD: **NO**

1. The following are additional entries to file 364.7 IB FORM FIELD CONTENT. These entries are used by the VistA Output Formatter when generating the 837D transaction.

NUMBER: **1425**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT** DATA ELEMENT: **N-RECORD ID**

PAD CHARACTER: **NO PAD REQUIRED** FORMAT CODE: **S IBXDATA="DN1 "**

FORMAT CODE DESCRIPTION: **Output RECORD ID for DN1 record.**

NUMBER: **1426**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-ORTHO BANDING QUALIFIER**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7,$P($G(^DGCR(399,IBXIEN,"DEN")),U**

**)'="" S IDXDATA=452**

FORMAT CODE DESCRIPTION: **Orthodontic Banding Qualifier always will be 452.**

NUMBER: **1460**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-ORTHO BANDING DATE**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **S:$$FT^IBCEF(IBXIEN)'=7 IBXDATA="" I $$FT^IBCEF(IBXIEN)=7 S IBXDA**

**TA=$$DT^IBCEFG1(IBXDATA,"","D8")**

FORMAT CODE DESCRIPTION: **Format date is CCYYMMDD**

NUMBER: **1461**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-ORTHO TX MTHS COUNT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **I $$FT^IBCEF(IBXIEN)'=7 K IBXDATA**

FORMAT CODE DESCRIPTION: **Ortho Treatment Months count for Dental Claim**

NUMBER: **1462**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-ORTHO TX MTHS COUNT TRANSFER**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **I $$FT^IBCEF(IBXIEN)'=7 K IBXDATA**

FORMAT CODE DESCRIPTION: **Ortho Treatment months remaining for a transfer**

**patient.**

NUMBER: **1463**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-ORTHO TX INDICATOR**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **I $$FT^IBCEF(IBXIEN)=7 D TRANS^IBCEF12**

NUMBER: **1464**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-RECORD ID**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **S IBXDATA="DN2 "**

FORMAT CODE DESCRIPTION: **Output the record id for 'DN2' record.**

NUMBER: **1465**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-TOOTH NUMBER**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7 N Z S Z=0 F S Z=$O(^TMP("IBXSAV**

**E",$J,"TO",IBXIEN,Z)) Q:'Z S Z1=^(Z),IBXSAVE("OUTPT",Z)=Z1,IBXDATA(Z)=$P(Z1,U)**

**I Z>1 D ID^IBCEF2(Z,"DN2 ")**

FORMAT CODE DESCRIPTION: **Tooth number of treatment**

NUMBER: **1466**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7 N Z S Z=0 F S Z=$O(IBXSAVE("OUT**

**PT",Z)) Q:'Z S IBXDATA(Z)=$P(IBXSAVE("OUTPT",Z),U,2)**

FORMAT CODE DESCRIPTION: **Tooth Status code, either E (to be extracted) or M**

**(Missing)**

NUMBER: **1467**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7 N Z S Z=0 F S Z=$O(IBXSAVE("OUT**

**PT",Z)) Q:'Z S IBXDATA(Z)=$P(IBXSAVE("OUTPT",Z),U,3)**

FORMAT CODE DESCRIPTION: **Code List Qualifier Code - will always be JP for**

**Dental claim.**

NUMBER: **1468**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7,$G(IBXSAVE("PROVINF",IBXIEN,"C",**

**1,1,"TAXONOMY"))'="" S IBXDATA="RF"**

FORMAT CODE DESCRIPTION: **Code identifing the type of provider. For Dental,**

**always RF for Referring.**

NUMBER: **1469**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7,$G(IBXSAVE("PROVINF",IBXIEN,"C",**

**1,1,"TAXONOMY"))'="" S IBXDATA="PXC"**

FORMAT CODE DESCRIPTION: **Code qualifing the Reference Identification. For**

**Dental, it will always be PXC for Taxonomy Code.**

NUMBER: **1470**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7 S IBXDATA=$G(IBXSAVE("PROVINF",I**

**BXIEN,"C",1,1,"TAXONOMY"))**

FORMAT CODE DESCRIPTION: **Reference information as specified by the Reference**

**Identification Qualifier. For Dental, this will be the Taxonomy Code.**

NUMBER: **1471**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-RECORD ID**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **S IBXDATA="OPRB"**

NUMBER: **1472**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **D CLEANUP^IBCEFP1(.IBXSAVE),ALLIDS^IBCEFP(IBXIEN,.IBXSAVE,1)**

FORMAT CODE DESCRIPTION: **Setup IBXSAVE array for OPRB record.**

NUMBER: **1473**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7,$D(IBXSAVE("PROVINF",IBXIEN,"C",**

**1,6,"NAME")) S IBXDATA="DD"**

NUMBER: **1474**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7,$D(IBXSAVE("PROVINF",IBXIEN,"C",**

**1,6,"NAME")) S IBXDATA=1**

NUMBER: **1475**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7 S IBXDATA=$P($G(IBXSAVE("PROVINF**

**",IBXIEN,"C",1,6,"NAME")),U)**

NUMBER: **1476**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7 S IBXDATA=$P($G(IBXSAVE("PROVINF**

**",IBXIEN,"C",1,6,"NAME")),U,2)**

NUMBER: **1477**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7 S IBXDATA=$P($G(IBXSAVE("PROVINF**

**",IBXIEN,"C",1,6,"NAME")),U,3)**

NUMBER: **1478**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7 S IBXDATA=$P($G(IBXSAVE("PROVINF**

**",IBXIEN,"C",1,6,"NAME")),U,5)**

NUMBER: **1479**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7 S IBXDATA=$P($G(IBXSAVE("PROVINF**

**",IBXIEN,"C",1,6,0)),U,3)**

NUMBER: **1480**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7 S IBXDATA=$P($G(IBXSAVE("PROVINF**

**",IBXIEN,"C",1,6,0)),U,4)**

NUMBER: **1481**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7,$G(IBXSAVE("PROVINF",IBXIEN,"C",**

**1,6,"TAXONOMY"))'="" S IBXDATA="AS"**

NUMBER: **1482**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7 S IBXDATA=$G(IBXSAVE("PROVINF",I**

**BXIEN,"C",1,6,"TAXONOMY"))**

NUMBER: **1483**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **D CLEANUP^IBCEFP1(.IBXSAVE)**

NUMBER: **1484**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-RECORD ID**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **I $$FT^IBCEF(IBXIEN)=7 S IBXDATA="OPRC"**

NUMBER: **1485**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **D CLEANUP^IBCEFP1(.IBXSAVE),ALLIDS^IBCEFP(IBXIEN,.IBXSAVE,1)**

FORMAT CODE DESCRIPTION: **Set up IBXSAVE array for OPRC record.**

NUMBER: **1486**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7 S IBXDATA=$P($G(IBXSAVE("PROVINF",IBXIEN,"C",1,6,1)),U,3)**

NUMBER: **1487**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7 S IBXDATA=$$NOPUNCT^IBCEF($P($G(IBXSAVE("PROVINF",IBXIEN,"C",1,6,1)),"^",4),1)**

NUMBER: **1488**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7 S IBXDATA=$P($G(IBXSAVE("PROVINF",IBXIEN,"C",1,6,2)),U,3)**

NUMBER: **1489**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7 S IBXDATA=$$NOPUNCT^IBCEF($P($G(IBXSAVE("PROVINF",IBXIEN,"C",1,6,2)),U,4),1)**

NUMBER: **1490**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7 S IBXDATA=$P($G(IBXSAVE("PROVINF",IBXIEN,"C",1,6,3)),U,3)**

NUMBER: **1491**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7 S IBXDATA=$$NOPUNCT^IBCEF($P($G(IBXSAVE("PROVINF",IBXIEN,"C",1,6,3)),U,4),1)**

NUMBER: **1492**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7 S IBXDATA=$P($G(IBXSAVE("PROVINF",IBXIEN,"C",1,6,4)),U,3)**

NUMBER: **1493**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7 S IBXDATA=$$NOPUNCT^IBCEF($P($G(IBXSAVE("PROVINF",IBXIEN,"C",1,6,4)),U,4),1)**

NUMBER: **1494**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **D CLEANUP^IBCEFP1(.IBXSAVE)**

NUMBER: **1495**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-RECORD ID**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **S IBXDATA="OP10"**

NUMBER: **1496**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

FORMAT CODE**: I $$FT^IBCEF(IBXIEN)=7 D CLEANUP^IBCEFP1(.IBXSAVE),ALLIDS^IBCEFP(IBXIEN,.IBXSAVE,1)**

FORMAT CODE DESCRIPTION: **Setup IBXSAVE array for the OP10 record.**

NUMBER: **1497**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXSAVE("OSQ") N C,Z,Q,OK M Q=IBXSAVE("PROVINF",IBXIEN,"O") S (C,Z)=0 F S Z=$O(Q(Z)) Q:'Z S OK=0 X "N A F A=1:1 Q:'$D(Q(Z,6,A)) I $P(Q(Z,6,A),U**

**,4)'="""" S OK=1 Q" I OK S C=C+1,IBXDATA(C)=$G(Q(Z)),IBXSAVE("OSQ",Z)=C D:C>1 ID**

**^IBCEF2(C,"OP10")**

NUMBER: **1498**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7 D OTHPAYC^IBCEF71(IBXIEN,.IBXSAVE,.IBXDATA,6,"ZZ")**

NUMBER: **1499**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7 D OTHPAYC^IBCEF71(IBXIEN,.IBXSAVE,.IBXDATA,6,"1")**

NUMBER: **1505**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7 D OTHPAYV^IBCEF71(IBXIEN,.IBXSAVE,.IBXDATA,6,"Q",1)**

NUMBER: **1506**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7 D OTHPAYV^IBCEF71(IBXIEN,.IBXSAVE,.IBXDATA,6,"I",1) I $D(IBXDATA) D NOPUNCT^IBCEF73(.IBXDATA,1)**

NUMBER: **1507**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7 D OTHPAYV^IBCEF71(IBXIEN,.IBXSAVE,.IBXDATA,6,"Q",2)**

NUMBER: **1508**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA D OTHPAYV^IBCEF71(IBXIEN,.IBXSAVE,.IBXDATA,6,"I",2) I $D(IBXDATA) D NOPUNCT^IBCEF73(.IBXDATA,1)**

NUMBER: **1509**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7 D OTHPAYV^IBCEF71(IBXIEN,.IBXSAVE,.IBXDATA,6,"Q",3)**

NUMBER: **1510**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7 D OTHPAYV^IBCEF71(IBXIEN,.IBXSAVE,.IBXDATA,6,"I",3) I $D(IBXDATA) D NOPUNCT^IBCEF73(.IBXDATA,1)**

NUMBER: **1511**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **D CLEANUP^IBCEFP1(.IBXSAVE)**

FORMAT CODE DESCRIPTION: **Clean up IBXSAVE arry after OP10 record.**

NUMBER: **1512**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-RECORD ID**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **S IBXDATA="DEN "**

FORMAT CODE DESCRIPTION: **Output the record id for 'DEN' record.**

NUMBER: **1513**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-HCFA J430D SERV LINE (EDI)**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **I $$FT^IBCEF(IBXIEN)=7 D DEN^IBCEF12**

NUMBER: **1514**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7 D SRVDTQ^IBCEF12**

NUMBER: **1515**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7 D SRVDT^IBCEF12**

NUMBER: **1516**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7 N Z S Z=0 F S Z=$O(IBXSAVE("OUT**

**PT",Z)) Q:'Z S IBXDATA(Z)="AD"**

NUMBER: **1517**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **I $$FT^IBCEF(IBXIEN)=7 S IBXNOREQ=$$NFT^IBCEF1(2,IBXIEN) K IBXDATA N Z S Z=0 F S Z=$O(IBXSAVE("OUTPT",Z)) Q:'Z S:$P(IBXSAVE("OUTPT",Z),U,5)'="" IBXDATA(Z)=$P(IBXSAVE("OUTPT",Z),U,5)**

NUMBER: **1528**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **I $$FT^IBCEF(IBXIEN)=7 N Z,Z0 S Z=0 F S Z=$O(IBXDATA(Z)) Q:'Z D**

**SETMODS^IBCVA1(IBXDATA(Z),Z,.IBXSAVE) S Z0=$P($G(IBXSAVE("PROCMODS",Z)),",") I Z0'="" S IBXDATA(Z)=Z0**

NUMBER: **1548**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7 N Z,Z0 S Z=0 F S Z=$O(IBXSAVE("**

**OUTPT",Z)) Q:'Z S Z0=$P($G(IBXSAVE("PROCMODS",Z)),",",2) I Z0'="" S IBXDATA(Z)=**

**Z0**

NUMBER: **1549**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7 N Z,Z0 S Z=0 F S Z=$O(IBXSAVE("**

**OUTPT",Z)) Q:'Z S Z0=$P($G(IBXSAVE("PROCMODS",Z)),",",3) I Z0'="" S IBXDATA(Z)=**

**Z0**

NUMBER: **1552**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7 N Z,Z0 S Z=0 F S Z=$O(IBXSAVE("**

**OUTPT",Z)) Q:'Z S Z0=$P($G(IBXSAVE("PROCMODS",Z)),",",4) K IBXSAVE("PROCMODS",Z) I Z0'="" S IBXDATA(Z)=Z0**

NUMBER: **1553**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

NUMBER: **1554**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7 N Z S Z=0 F S Z=$O(IBXSAVE("OUT**

**PT",Z)) Q:'Z S IBXDATA(Z)=$S($P(IBXSAVE("OUTPT",Z),U,8)="":$$DOLLAR^IBCEFG1("0.**

**00"),1:$$DOLLAR^IBCEFG1($P(IBXSAVE("OUTPT",Z),U,8)\*$P(IBXSAVE("OUTPT",Z),U,9)))**

NUMBER: **1555**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7 N Z S Z=0 F S Z=$O(IBXSAVE("OUT**

**PT",Z)) Q:'Z S:$P(IBXSAVE("OUTPT",Z),U,3)'="" IBXDATA(Z)=$P(IBXSAVE("OUTPT",Z),**

**U,3)**

NUMBER: **1556**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7 N Z S Z=0 F S Z=$O(IBXSAVE("OUT**

**PT",Z)) Q:'Z I $D(IBXSAVE("OUTPT",Z,"DEN")) S IBXDATA(Z)=$P(IBXSAVE("OUTPT",Z,"**

**DEN"),U)**

NUMBER: **1557**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER**: NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7 N Z S Z=0 F S Z=$O(IBXSAVE("OUT**

**PT",Z)) Q:'Z I $D(IBXSAVE("OUTPT",Z,"DEN")) S IBXDATA(Z)=$P(IBXSAVE("OUTPT",Z,"**

**DEN"),U,2)**

NUMBER: **1558**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7 N Z S Z=0 F S Z=$O(IBXSAVE("OUT**

**PT",Z)) Q:'Z I $D(IBXSAVE("OUTPT",Z,"DEN")) S IBXDATA(Z)=$P(IBXSAVE("OUTPT",Z,"**

**DEN"),U,3)**

NUMBER: **1559**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7 N Z S Z=0 F S Z=$O(IBXSAVE("OUT**

**PT",Z)) Q:'Z I $D(IBXSAVE("OUTPT",Z,"DEN")) S IBXDATA(Z)=$P(IBXSAVE("OUTPT",Z,"**

**DEN"),U,4)**

NUMBER: **1560**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7 N Z S Z=0 F S Z=$O(IBXSAVE("OUT**

**PT",Z)) Q:'Z I $D(IBXSAVE("OUTPT",Z,"DEN")) S IBXDATA(Z)=$P(IBXSAVE("OUTPT",Z,"**

**DEN"),U,4)**

NUMBER: **1580**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7 N Z S Z=0 F S Z=$O(IBXSAVE("OUT**

**PT",Z)) Q:'Z I $D(IBXSAVE("OUTPT",Z,"DEN")) S IBXDATA(Z)=$P(IBXSAVE("OUTPT",Z,"**

**DEN"),U,6)**

NUMBER: **1581**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-RECORD ID**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **S IBXDATA="DEN1"**

FORMAT CODE DESCRIPTION: **Output the record id for 'DEN1' record.**

NUMBER: **1582**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-HCFA 1500 SERVICE LINE (EDI)**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **I $$FT^IBCEF(IBXIEN)=7 D DEN1^IBCEF12**

NUMBER: **1583**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **I $$FT^IBCEF(IBXIEN)=7 D PROC^IBCEF12**

NUMBER: **1584**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7 N Z S Z=0 F S Z=$O(IBXSAVE("OUT**

**PT",Z)) Q:'Z S Z0=$P(IBXSAVE("OUTPT",Z),U,7) S:$P(Z0,",")'="" IBXDATA(Z)=$P(Z0,**

**",")**

NUMBER: **1585**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7 N Z,Z0 F S Z=$O(IBXSAVE("OUTPT"**

**,Z)) Q:'Z S Z0=$P(IBXSAVE("OUTPT",Z),U,7) S:$P(Z0,",",2) IBXDATA(Z)=$P(Z0,",",2**

**)**

NUMBER: **1586**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE**: K IBXDATA I $$FT^IBCEF(IBXIEN)=7 N Z,Z0 S Z=0 F S Z=$O(IBXSAVE("**

**OUTPT",Z)) Q:'Z S Z0=$P(IBXSAVE("OUTPT",Z),U,7) S:$P(Z0,",",3) IBXDATA(Z)=$P(Z0**

**,",",3)**

NUMBER: **1587**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7 N Z,Z0 S Z=0 F S Z=$O(IBXSAVE("**

**OUTPT",Z)) Q:'Z S Z0=$P(IBXSAVE("OUTPT",Z),U,7) S:$P(Z0,",",4) IBXDATA(Z)=$P(Z0**

**,",",4)**

NUMBER: **1588**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7 N Z S Z=0 F S Z=$O(IBXSAVE("OUT**

**PT",Z)) Q:'Z I $D(IBXSAVE("OUTPT",Z,"DEN")) S IBXDATA(Z)=$P(IBXSAVE("OUTPT",Z,"**

**DEN"),U,7)**

NUMBER: **1589**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7 N Z,DT S Z=0 F S Z=$O(IBXSAVE("**

**OUTPT",Z)) Q:'Z I $D(IBXSAVE("OUTPT",Z,"DEN")) S DT=$P(IBXSAVE("OUTPT",Z,"DEN"),U,8) I DT'="" S IBXDATA(Z)=$$DT^IBCEFG1(DT,"","D8")**

NUMBER: **1590**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7 N Z S Z=0 F S Z=$O(IBXSAVE("OUT**

**PT",Z)) Q:'Z I $P($G(IBXSAVE("OUTPT",Z,"DEN")),U,9)'="" S IBXDATA(Z)=452**

NUMBER: **1591**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7 N Z,DT S Z=0 F S Z=$O(IBXSAVE("**

**OUTPT",Z)) Q:'Z I $D(IBXSAVE("OUTPT",Z,"DEN")) S DT=$P(IBXSAVE("OUTPT",Z,"DEN"),U,9) I DT'="" S IBXDATA(Z)=$$DT^IBCEFG1(DT,"","D8")**

NUMBER: **1592**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7 N Z S Z=0 F S Z=$O(IBXSAVE("OUT**

**PT",Z)) Q:'Z I $P($G(IBXSAVE("OUTPT",Z,"DEN")),U,10)'="" S IBXDATA(Z)=446**

NUMBER: **1593**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7 N Z,DT S Z=0 F S Z=$O(IBXSAVE("**

**OUTPT",Z)) Q:'Z I $D(IBXSAVE("OUTPT",Z,"DEN")) S DT=$P(IBXSAVE("OUTPT",Z,"DEN"),U,10) I DT'="" S IBXDATA(Z)=$$DT^IBCEFG1(DT,"","D8")**

NUMBER: **1594**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7 N Z S Z=0 F S Z=$O(IBXSAVE("OUT**

**PT",Z)) Q:'Z I $P($G(IBXSAVE("OUTPT",Z,"DEN")),U,11)'="" S IBXDATA(Z)=196**

NUMBER: **1595**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7 N Z,DT S Z=0 F S Z=$O(IBXSAVE("**

**OUTPT",Z)) Q:'Z I $D(IBXSAVE("OUTPT",Z,"DEN")) S DT=$P(IBXSAVE("OUTPT",Z,"DEN"),U,11) I DT'="" S IBXDATA(Z)=$$DT^IBCEFG1(DT,"","D8")**

NUMBER: **1596**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7 N Z S Z=0 F S Z=$O(IBXSAVE("OUT**

**PT",Z)) Q:'Z I $P($G(IBXSAVE("OUTPT",Z,"DEN")),U,12)'="" S IBXDATA(Z)=198**

NUMBER: **1597**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7 N Z,DT S Z=0 F S Z=$O(IBXSAVE("**

**OUTPT",Z)) Q:'Z I $D(IBXSAVE("OUTPT",Z,"DEN")) S DT=$P(IBXSAVE("OUTPT",Z,"DEN"),U,12) I DT'="" S IBXDATA(Z)=$$DT^IBCEFG1(DT,"","D8")**

NUMBER: **1598**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-RECORD ID**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **S IBXDATA="DEN2"**

NUMBER: **1599**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-HCFA 1500 SERVICE LINE (EDI)**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **I $$FT^IBCEF(IBXIEN)=7 D DEN2^IBCEF12**

NUMBER: **1600**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7 D DEN23^IBCEF12**

NUMBER: **1601**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7 D DEN24^IBCEF12**

NUMBER: **1602**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7 D DEN25^IBCEF12**

NUMBER: **1603**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7 D DEN26^IBCEF12**

NUMBER: **1604**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7 D DEN27^IBCEF12**

NUMBER: **1605**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7 D DEN28^IBCEF12**

NUMBER: **1606**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)=7 D DEN29^IBCEF12**

NUMBER: **1607**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-RECORD ID**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **S IBXDATA="LSUR"**

NUMBER: **1609**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **I $$FT^IBCEF(IBXIEN)=7 D CLEANUP^IBCEFP1(.IBXSAVE),ALLIDS^IBCEFP(IBXIEN,.IBXSAVE,1,"LSUR")**

NUMBER: **1608**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **I $$FT^IBCEF(IBXIEN)=7 N Z K IBXDATA S Z=0 F S Z=$O(IBXSAVE("SLP**

**RV",Z)) Q:'Z S IBXDATA(Z)=$G(IBXSAVE("SLPRV",Z,"SLC")) D:Z>1 ID^IBCEF2(Z,"LSUR"**

**)**

NUMBER: **1614**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **I $$FT^IBCEF(IBXIEN)=7 N Z K IBXDATA S Z=0 F S Z=$O(IBXSAVE("SLP**

**RV",Z)) Q:'Z S IBXDATA(Z)="" S:$P(IBXSAVE("SLPRV",Z),U)'="" IBXDATA(Z)="DD"**

NUMBER: **1688**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **I $$FT^IBCEF(IBXIEN)=7 N Z K IBXDATA S Z=0 F S Z=$O(IBXSAVE("SLP**

**RV",Z)) Q:'Z S IBXDATA(Z)=$P(IBXSAVE("SLPRV",Z),U)**

NUMBER: **1689**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: NO PAD REQUIRED

FORMAT CODE: **I $$FT^IBCEF(IBXIEN)=7 N Z K IBXDATA S Z=0 F S Z=$O(IBXSAVE("SLP**

**RV",Z)) Q:'Z S IBXDATA(Z)=$P(IBXSAVE("SLPRV",Z),U,2)**

NUMBER: **1728**

FORM FIELD REFERENCE: IB 837 **TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **I $$FT^IBCEF(IBXIEN)=7 N Z K IBXDATA S Z=0 F S Z=$O(IBXSAVE("SLP**

**RV",Z)) Q:'Z S IBXDATA(Z)=$P(IBXSAVE("SLPRV",Z),U,3)**

NUMBER: **1729**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **I $$FT^IBCEF(IBXIEN)=7 N Z K IBXDATA S Z=0 F S Z=$O(IBXSAVE("SLP**

**RV",Z)) Q:'Z S IBXDATA(Z)=$P(IBXSAVE("SLPRV",Z),U,4)**

NUMBER: **1730**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **I $$FT^IBCEF(IBXIEN)=7 N Z K IBXDATA S Z=0 F S Z=$O(IBXSAVE("SLP**

**RV",Z)) Q:'Z S IBXDATA(Z)="" S:$P(IBXSAVE("SLPRV",Z),U,6)'="" IBXDATA(Z)="XX"**

NUMBER: **1731**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **I $$FT^IBCEF(IBXIEN)=7 N Z K IBXDATA S Z=0 F S Z=$O(IBXSAVE("SLP**

**RV",Z)) Q:'Z S IBXDATA(Z)=$P(IBXSAVE("SLPRV",Z),U,6)**

NUMBER: **1957**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-RECORD ID**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **S IBXDATA="LSUR1"**

NUMBER: **1958**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **I $$FT^IBCEF(IBXIEN)=7 D**

**CLEANUP^IBCEFP1(.IBXSAVE),ALLIDS^IBCEFP(**

**IBXIEN,.IBXSAVE,1,"LSUR1")**

NUMBER: **1959**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **I $$FT^IBCEF(IBXIEN)=7 N Z K IBXDATA S Z=0 F S Z=$O(IBXSAVE("SLP**

**RV",Z)) Q:'Z S IBXDATA(Z)=$G(IBXSAVE("SLPRV",Z,"SLC")) D:Z>1**

**ID^IBCEF2(Z,"LSUR1")**

NUMBER: **1960**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **I $$FT^IBCEF(IBXIEN)=7 N Z K IBXDATA S Z=0 F S Z=$O(IBXSAVE("SLP**

**RV",Z)) Q:'Z S IBXDATA(Z)="" S:$P(IBXSAVE("SLPRV",Z),U,7)'="" IBXDATA(Z)=$P(IBXSAVE("SLPRV",Z),U,8)**

NUMBER: **1961**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **I $$FT^IBCEF(IBXIEN)=7 N Z K IBXDATA S Z=0 F S Z=$O(IBXSAVE("SLP**

**RV",Z)) Q:'Z S IBXDATA(Z)=$P(IBXSAVE("SLPRV",Z),U,7)**

NUMBER: **1962**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **I $$FT^IBCEF(IBXIEN)=7 N Z K IBXDATA S Z=0 F S Z=$O(IBXSAVE("SLP**

**RV",Z)) Q:'Z S IBXDATA(Z)="" S:$P(IBXSAVE("SLPRV",Z),U,9)'="" IBXDATA(Z)=$P(IBXSAVE("SLPRV",Z),U,10)**

NUMBER: **1963**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **I $$FT^IBCEF(IBXIEN)=7 N Z K IBXDATA S Z=0 F S Z=$O(IBXSAVE("SLP**

**RV",Z)) Q:'Z S IBXDATA(Z)=$P(IBXSAVE("SLPRV",Z),U,9)**

NUMBER: **1964**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **I $$FT^IBCEF(IBXIEN)=7 N Z K IBXDATA S Z=0 F S Z=$O(IBXSAVE("SLP**

**RV",Z)) Q:'Z S IBXDATA(Z)="" S:$P(IBXSAVE("SLPRV",Z),U,11)'="" IBXDATA(Z)=$P(IBXSAVE("SLPRV",Z),U,12)**

NUMBER: **1965**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **I $$FT^IBCEF(IBXIEN)=7 N Z K IBXDATA S Z=0 F S Z=$O(IBXSAVE("SLP**

**RV",Z)) Q:'Z S IBXDATA(Z)=$P(IBXSAVE("SLPRV",Z),U,11)**

NUMBER: **1966**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **I $$FT^IBCEF(IBXIEN)=7 N Z K IBXDATA S Z=0 F S Z=$O(IBXSAVE("SLP**

**RV",Z)) Q:'Z S IBXDATA(Z)="" S:$P(IBXSAVE("SLPRV",Z),U,5)'="" IBXDATA(Z)="AS"**

NUMBER: **1967**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **I $$FT^IBCEF(IBXIEN)=7 N Z K IBXDATA S Z=0 F S Z=$O(IBXSAVE("SLP**

**RV",Z)) Q:'Z S IBXDATA(Z)=$P(IBXSAVE("SLPRV",Z),U,5)**

1. The following are modified entries to file 364.7 IB FORM FIELD CONTENT. These entries are used by the VistA Output Formatter when generating the 837D transaction.

NUMBER: **31**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-SPECIAL PROGRAM**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **I IBXDATA="",$$WNRBILL^IBEFUNC(IBXIEN),$$FT^IBCEF(IBXIEN)=2**

**!($$FT^IBCEF(IBXIEN)=7) S IBXDATA="31"**

NUMBER: **37**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-UB-04 PROCEDURES**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K:$$FT^IBCEF(IBXIEN)=2!($$FT^IBCEF(IBXIEN)=7)!'$$INPAT^IBCEF(IBXI**

**EN) IBXDATA N Z S Z=0 F S Z=$O(IBXDATA(Z)) K:'Z IBXDATA Q:'Z I '$D(IBXDATA(Z,"M")) S IBXSAVE("PROC",Z)=IBXDATA(Z) I Z'<25 K IBXDATA Q**

FORMAT CODE DESCRIPTION: **This is a group data element so more than 1**

**occurrence of a value is possible for the data element in the IBXDATA array. If an institutional bill or if the bill is professional and the procedure being extracted was not added as a result of being a modifier with a 5-digit code starting with 0 ("M" node does not exist), save in an IBXSAVE array for later use. No output.**

NUMBER: **1006**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-OTHER INSURANCE CO TYPES**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **N A,Q,Z S Q=$G(IBXDATA) K IBXDATA I $$FT^IBCEF(IBXIEN)=2!($$FT^IB**

**CEF(IBXIEN)=7) F Z=1,2 S A=$P(Q,U,Z) I $D(^DGCR(399,IBXIEN,"I"\_(Z+1))) S IBXDATA(Z)=$S(A=1:"HM",A=2:"C1",A=3:"MB",A=4:"MC",A=5:"GP",1:"OT")**

FORMAT CODE DESCRIPTION: **This is a group data element so more than 1**

**occurrence of a value is possible for the data element in**

**the IBXDATA array. If any 'other' insurance company**

**data is found, the data is formatted as the**

**insurance type of the insurance company in X12 format**

**(see codes below or refer to the 837 V4010 field**

**2320/SBR/05 - professional guide)**

**Type codes: = 1:HMO (HM)**

**2:COMMERCIAL (C1)**

**3:MEDICARE (MB)**

**4:MEDICAID (MC)**

**5:GROUP POLICY (GP)**

**9:OTHER (OT)**

NUMBER: **1023**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **D OTH^IBCEF76(**

**IBXIEN,.IBXSAVE,.IBXDATA,($$FT^IBCEF(IBXIEN)=2!($$F**

**T^IBCEF(IBXIEN)=7)),"OP7 ")**

FORMAT CODE DESCRIPTION: **OP7-2 other payer sequence. Call provider ID**

**function only when claim is a 1500 claim**.

NUMBER: **1015**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **S IBXDATA="592.0" I '$$PROD^XUPROD(1) S $E(IBXDATA,11)="D"**

FORMAT CODE DESCRIPTION: **This field contains the VistA IB patch# that**

**pertains to the applicable changes in the claim map. Austin uses this field to determine which set of EDI claim maps to use when processing the EDI claims. If the EDI claims are being transmitted from a non-production account, then the "D" in the 11th position indicates to Austin that it is test data.**

NUMBER: **16**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **I $$FT^IBCEF(IBXIEN)'=7 S IBXDATA=$P($G(IBXSAVE("EMP")),U)**

FORMAT CODE DESCRIPTION: **Retrieves insured employment info from IBXSAVE**

**array previously extracted. If data element's value is null, do not output.**

NUMBER: **127**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K IBXDATA I $$FT^IBCEF(IBXIEN)'=7 N Z F Z=1,2 I $D(^DGCR(399,IBXI**

**EN,"I"\_(Z+1))),$P($G(IBXSAVE("OIEMP",Z)),U)'="" S IBXDATA(Z)=$P(IBXSAVE("OIEMP",Z),U)**

FORMAT CODE DESCRIPTION: **This is a group data element so more than 1**

**occurrence of a value is possible for the data element in the previously extract IBXSAVE array. If any other insurance employer data is found, the data is output.**

NUMBER: **75**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-HCFA 1500 BALANCE DUE BOX**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **I $$FT^IBCEF(IBXIEN)'=7 S IBXNOREQ='$$REQ^IBCEF1(2,"",IBXIEN) S I**

**BXDATA=$$DOLLAR^IBCEFG1(IBXDATA)**

FORMAT CODE DESCRIPTION: **Format data element so it is numeric including**

**cents, without the decimal. If data element's value is null, do not output.**

NUMBER: **1008**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **I $$FT^IBCEF(IBXIEN)'=7 S**

**IBXDATA=$P($G(IBXSAVE("PROVINF",IBXIEN, "C",1,3,"NAME")),U,4)**

FORMAT CODE DESCRIPTION: Credentials should be in the 4th piece of the

"NAME" node of the Rendering Provider IBXSAVE array.

NUMBER: **1727**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-GET FROM PREVIOUS EXTRACT**

FORMAT CODE: **N Z,Z0,IBZ K IBXDATA I $$FT^IBCEF(IBXIEN)'=7 S (IBZ,Z0)=0 F S IB**

**Z=$O(IBXSAVE("CCOB",IBZ)) Q:'IBZ S Z0=Z0+1,Z=0 F S Z=$O(IBXSAVE("CCOB",IBZ,Z)) Q:'Z I $P($G(IBXSAVE("CCOB",IBZ,Z,0)),U,21)'="" S IBXDATA(Z0)=$P(IBXSAVE("CCOB",IBZ,Z,0),U,21)**

NUMBER: **854**

FORM FIELD REFERENCE: **IB 837 TRANSMISSION**

SECURITY LEVEL: **NATIONAL,NO EDIT**

DATA ELEMENT: **N-DRG USED**

PAD CHARACTER: **NO PAD REQUIRED**

FORMAT CODE: **K:$$FT^IBCEF(IBXIEN)=7 IBXDATA I +$G(IBXDATA)=0 K IBXDATA**

1. The following field definitions need to be added or modified in order to process Dental claims. These fields are located in the Insurance Company file (#36) and are accessible via the INSURANCE COMPANY ENTRY/EDIT option. These new fields are necessary to address the need for the new Dental Payer ID and the Provider ID requirements for Dental form J430D.

36,3.15 EDI ID NUMBER - DENTAL 3;15 FREE TEXT

INPUT TRANSFORM: K:$L(X)>30!($L(X)<1)!

($$UP^XLFSTR(X)["PRNT") X

MAXIMUM LENGTH: 30

LAST EDITED: JUL 21, 2017

HELP-PROMPT: Answer must be 1-30 characters in

length. PRNT values are not allowed.

DESCRIPTION: This is the ID number used to identify

The Payer on Dental claim

transmissions. PRNT values are not

valid Payer IDs.

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

PROGRAMMER

36,.191 CLAIMS (DENTAL) STREET ADDR 1 .19;1 FREE TEXT

INPUT TRANSFORM: K:$L(X)>30!($L(X)<3)!'$G(IBCNS) X

MAXIMUM LENGTH: 30

LAST EDITED: JUL 25, 2017

HELP-PROMPT: If the Dental claims process address of this

company is different from its main address,

enter Line 1 of the Dental claims street

address. Answer must be 3-30 characters in

length.

DESCRIPTION: If the Dental claims process address of this

company is different from its main address,

enter Line 1 of the Dental claims street

address. Answer must be 3-30 characters in

length.

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

CROSS-REFERENCE: ^^TRIGGER^36^.192

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

Y(1)=$S($D(^DIC(36,D0,.19)):^(.19),1:"")

S X=$P(Y(1),U,2),X=X S DIU=X K Y S X=""

S DIH=$G(^DIC(36,DIV(0),.19)),DIV=X

S $P(^(.19),U,2)=DIV,DIH=36,DIG=.192

D ^DICR

2)= K DIV S DIV=X,D0=DA,DIV(0)=D0 S

Y(1)=$S($D(^DIC(36,D0,.19)):^(.19),1:"")

S X=$P(Y(1),U,2),X=X S DIU=X K Y S X=""

S DIH=$G(^DIC(36,DIV(0),.19)),DIV=X

S $P(^(.19),U,2)=DIV,DIH=36,DIG=.192

D ^DICR

3)= When changing or deleting CLAIMS (DENTAL)

STREET ADDR 1 delete CLAIMS (DENTAL)

STREET ADDR2.

CREATE VALUE)= @

DELETE VALUE)= @

FIELD)= CLAIMS (DENTAL) STREET ADDR 2

When changing or deleting CLAIMS (DENTAL)

STREET ADDR 1 delete CLAIMS (DENTAL) STREET

ADDR 2.

36,.1911 CLAIMS (DENTAL) PHONE NUMBER .19;11 FREE TEXT

INPUT TRANSFORM: K:$L(X)>20!($L(X)<7) X

MAXIMUM LENGTH: 20

LAST EDITED: JUL 25, 2017

HELP-PROMPT: Enter the telephone number of the Dental

claims office with 7-20 characters, e.g.

777-8888, 415 111 2222 x123.

DESCRIPTION: Enter the telephone number at which this

insurance carrier's Dental claims office can

be reached.

36,.192 CLAIMS (DENTAL) STREET ADDR 2 .19;2 FREE TEXT

INPUT TRANSFORM: K:$L(X)>30!($L(X)<3)!

$$DUPADDRL^IBCNSU(X,+$G(IBCNS),.191) X

MAXIMUM LENGTH: 30

LAST EDITED: JUL 25, 2017

HELP-PROMPT: If the Dental Claims Process Address is

longer than one line, enter a second line

between 3-30 characters. The response can

not be the same as Line 1.

DESCRIPTION: If the Dental Claims Process Address is

longer than one line, enter a second line

between 3-30 characters. The response can

not be the same as line 1.

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

TRIGGERED by the CLAIMS (DENTAL) STREET ADDR1

field of the INSURANCE COMPANY File

36,.193 BLANK .19;3 FREE TEXT

INPUT TRANSFORM: K:$L(X)>30!($L(X)<1) X

MAXIMUM LENGTH: 30

LAST EDITED: JUL 25, 2017

HELP-PROMPT: Answer must be 1-30 characters in length.

DESCRIPTION: This is a place holder for a 3rd address

line, if needed.

36,.194 CLAIMS (DENTAL) PROCESS CITY .19;4 FREE TEXT

INPUT TRANSFORM: K:$L(X)>25!($L(X)<2) X

MAXIMUM LENGTH: 25

LAST EDITED: JUL 25, 2017

HELP-PROMPT: If the Dental claims process address of this

company is different from its main address,

enter city of the Dental claims process

address. Answer must be 2-25 characters in

length.

DESCRIPTION: Enter the state in which this insurance

company's Dental claims office is located.

Enter state even if it is the same as the

state of the company's main address.

36,.195 CLAIMS (DENTAL) PROCESS STATE .19;5 POINTER TO STATE

FILE (#5)

LAST EDITED: JUL 25, 2017

HELP-PROMPT: If the Dental claims process address of this

company is different from its main address,

enter state of the Dental claims process

address.

DESCRIPTION: Enter the state in which this insurance

company's Dental claims office is located.

Enter state even if it is the same as the

State of the company's main address.

36,.196 CLAIMS (DENTAL) PROCESS ZIP .19;6 FREE TEXT

INPUT TRANSFORM: K:$L(X)>10!($L(X)<9) X

I $D(X) K:'$$ZIPCHK9^IBCNSU(X) X

I $D(X) D ZIPIN^VAFADDR

MAXIMUM LENGTH: 10

LAST EDITED: JUL 25, 2017

HELP-PROMPT: Answer must be nine (999999999) or ten

characters (99999-9999) in length. The last

4 cannot be '0000' or '9999'.

DESCRIPTION: This is the ZIP code for the address of the

Dental claims processing location when it

differs from the payer's main mailing

address. Enter a 9 or 10 character ZIP code.

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

36,.197 CLAIMS (DENTAL) COMPANY NAME .19;7 POINTER TO INSURANCE

COMPANY FILE (#36)

INPUT TRANSFORM: S DIC(0)=DIC(0)\_"F",DIC("S")=

"I '$P(^(0),U,5),'$P($G(^(.19)),U,7),Y'=DA"

D ^DIC K DIC S DIC=DIE,X=+Y K:Y<0 X

LAST EDITED: JUL 25, 2017

DESCRIPTION: You can only select a company that processes

claims. The company specified in this field

must be an active insurance company, not the

same company as the entry being edited, and

must not have another company specified as

handling Dental Claims for it.

SCREEN: S DIC("S")="I $P(^(0),U,5),

'$P($G(^(.19)),U,7),Y'=DA"

EXPLANATION: Select a company that processes Dental claims

for this company. Must be active, not this

company, and process its own Dental claims.

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

TRIGGERED by the ANOTHER CO. PROC DENT

CLAIMS? field of the INSURANCE COMPANY File

36,.198 ANOTHER CO. PROC DENT CLAIMS? .19;8 SET

'0' FOR NO;

'1' FOR YES;

LAST EDITED: JUL 25, 2017

DESCRIPTION: Enter 'Yes' if another insurance company

processes Dental Claims.

CROSS-REFERENCE: ^^TRIGGER^36^.197

1)= K DIV S DIV=X,D0=DA,DIV(0)=D0 S Y(0)=X

S X='$P($G(^DIC(36,DA,.19)),"^",8)

I X S X=DIV S Y(1)=$S($D(^DIC(36,D0,.19))

:^(.19),1:"") S X=$P(Y(1),U,7),X=X

S DIU=X K Y S X="" X ^DD(36,.198,1,1,1.4)

1.4)= S DIH=$G(^DIC(36,DIV(0),.19)),DIV=X

S $P(^(.19),U,7)=DIV,DIH=36,DIG=.197

D ^DICR

2)= Q

CREATE CONDITION)= S X=

'$P($G(^DIC(36,DA,.19)),

"^",8)

CREATE VALUE)= @

DELETE VALUE)= NO EFFECT

FIELD)= CLAIMS (DENTAL) COMPANY NAME

Enter 'Yes' if another insurance company

processes Dental Claims.

36,.199 CLAIMS (DENTAL) FAX .19;9 FREE TEXT

INPUT TRANSFORM: K:$L(X)>20!($L(X)<7) X

MAXIMUM LENGTH: 20

LAST EDITED: JUL 25, 2017

HELP-PROMPT: Enter the fax number of the inpatient claims

office with 7-20 characters, e.g. 444-8888,

614-333-9999.

DESCRIPTION: Enter the fax number of this insurance

carrier's Dental claims office.

FILES POINTED TO FIELDS

INSURANCE COMPANY (#36) CLAIMS (DENTAL) COMPANY NAME (#.197)

STATE (#5) CLAIMS (DENTAL) PROCESS STATE (#.195)

36,4.14       PERF PROV SECOND ID TYPE J430D 4;14 POINTER TO

IB PROVIDER ID # TYPE FILE (#355.97)

              INPUT TRANSFORM:  S DIC("S")="I $P(^(0),U,3)'=""1A"",

$P(^(0),U,3)'=""TJ"",$$RAINS^IBCEPU(Y)

!($$RAOWN^IBCEPU(Y))" D ^DIC K DIC S

DIC=DIE,X=+Y K:Y<0 X

              LAST EDITED:      MAY 30, 2017

              HELP-PROMPT:      Enter the type of performing provider

id # the insurance co requires on its

J430D bills.

              DESCRIPTION:      This is the type of performing provider

                                secondary id # that the insurance

company expects on J430D bills received

from the V.A. When the payer-specific

provider id is extracted, this field is

used to determine where to get the

default data from if another secondary

id is not entered for the claim.

              SCREEN:           S DIC("S")="I $P(^(0),U,3)'=""1A"",

$P(^(0),U,3)'=""TJ"",$$RAINS^IBCEPU(Y)

!($$RAOWN^IBCEPU(Y))"

              EXPLANATION:      Must be valid id type for performing

provider id

36,4.15       REF PROV SEC ID DEF J430D 4;15 POINTER TO

IB PROVIDER ID # TYPE FILE (#355.97)

              INPUT TRANSFORM:  S DIC("S")="I $P(^(0),U,3)'=""1A"",

$$RAINS^IBCEPU(Y)!($$RAOWN^IBCEPU(Y))"

D ^DIC K DIC S DIC=DIE,X=+Y K:Y<0 X

              LAST EDITED:      MAY 30, 2017

              HELP-PROMPT:      Enter the referring providers secondary ID

type to be used on outgoing claims.

              DESCRIPTION:      This is the default qualifier for a referring

                                provider if there is a referring provider and

                                the form type is J430D.

              SCREEN:           S DIC("S")="I $P(^(0),U,3)'=""1A"",

$$RAINS^IBCEPU(Y)!($$RAOWN^IBCEPU(Y))"

              EXPLANATION:      Must be an allowable ID for a person

36,4.16       ATT/REND ID BILL SEC ID J430D 4;16 SET

                                '0' FOR NO;

                                '1' FOR YES;

              LAST EDITED:      MAY 30, 2017

              HELP-PROMPT:      Enter 1 if att/rend ID should be used as

                                Billing Provider's secondary ID for Dental

                                J430D claims.

              DESCRIPTION:      This flag is set for insurance companies that

                                wish to have the attending/rendering provider

                                secondary ID used as a billing provider

                                secondary ID.  This applies to J430D claims

                                only.

      FILES POINTED TO                      FIELDS

IB PROVIDER ID # TYPE (#355.97)   PERF PROV SECOND ID TYPE J430D (#4.14)

                                  REF PROV SEC ID DEF J430D (#4.15)

36,4.17 ASSIST SURG SEC ID DEF J430D 4;17 POINTER TO IB PROVIDER ID #

TYPE FILE (#355.97)

INPUT TRANSFORM: S DIC("S")="I $P(^(0),U,3)'=""1A"",

$$RAINS^IBCEPU(Y)!($$RAOWN^IBCEPU(Y))"

D ^DIC K DIC S DIC=DIE,X=+Y K:Y<0 X

LAST EDITED: JUL 21, 2017

HELP-PROMPT: Enter the assistant surgeon's secondary ID

type to be used on outgoing claims.

DESCRIPTION: This is the default qualifier for an

Assistant surgeon if there is an assistant

surgeon and the form type is J430D.

SCREEN: S DIC("S")="I $P(^(0),U,3)'=""1A"",

$$RAINS^IBCEPU(Y)!($$RAOWN^IBCEPU(Y))"

EXPLANATION: Must be an allowable ID for a person

FILES POINTED TO FIELDS

IB PROVIDER ID # TYPE (#355.97) ASSIST SURG SEC ID DEF J430D (#4.17)

355.9,.04 FORM TYPE APPLIED TO 0;4 SET (Required)

'0' FOR UB-04, CMS-1500 and J430D FORMS;

'1' FOR UB-04 FORMS ONLY;

'2' FOR CMS-1500 FORMS ONLY;

'4' FOR J430D FORMS ONLY;

LAST EDITED: MAY 12, 2017

HELP-PROMPT: Enter the form types that this id # will apply

to.

DESCRIPTION: This designates whether the id number is to be

used for just UB-04 form types, just CMS-1500,

just J430D, or all form types.

RECORD INDEX: AUNIQ (#115) MUMPS IR SORTING ONLY

Short Descr: Xref by provider,ins co(or ALL),care unit,form

type,pt stat,prov type

Description: This cross reference provides an index to insure

that each record in this file has a unique combination of provider,insurance company, form type, patient status and provider id type.

Set Logic: S ^IBA(355.9,"AUNIQ",X(1),$E(X(2),1,30),$E(X(3)

,1,30),X(4),X(5),X(6),DA)=""

Kill Logic: K ^IBA(355.9,"AUNIQ",X(1),$E(X(2),1,30),$E(X(3)

,1,30),X(4),X(5),X(6),DA)

Whole Kill: K ^IBA(355.9,"AUNIQ")

X(1): PRACTITIONER (355.9,.01) (Subscr 1)

(forwards)

X(2): INDEX VALUE INSURANCE CO (355.9,.15)

(Subscr 2) (Len 30) (forwards)

X(3): INDEX VALUE CARE UNIT (355.9,.16) (Subscr 3)

(Len 30) (forwards)

X(4): FORM TYPE APPLIED TO (355.9,.04) (Subscr 4)

(forwards)

X(5): BILL CARE TYPE (355.9,.05) (Subscr 5)

(forwards)

X(6): PROVIDER ID TYPE (355.9,.06) (Subscr 6)

(forwards)

355.91,.04 FORM TYPE APPLIED TO 0;4 SET (Required)

'0' FOR UB-04, CMS-1500 and J430D FORMS;

'1' FOR UB-04 FORMS ONLY;

'2' FOR CMS-1500 FORMS ONLY;

'4' FOR J430D FORMS ONLY;

LAST EDITED: MAY 12, 2017

HELP-PROMPT: Select one or all forms that the provider id

will apply to.

DESCRIPTION: This designates whether the id number is to be

used for just UB-04 form types, just CMS-1500,

just J430D, or all form types.

RECORD INDEX: AUNIQ (#117) MUMPS IR SORTING ONLY

Short Descr: Xref by ins co,care unit,form

type,pt stat,prov type

Description: This cross reference provides an index to

insure that each record in this file has a

unique combination of insurance company, form

type, patient status and provider id type.

Set Logic: S ^IBA(355.91,"AUNIQ",X(1),$S(X(2):$E(X(2),1,30)

,1:"\*N/A\*"),X(3),X(4),X(5),DA)=""

Kill Logic: K ^IBA(355.91,"AUNIQ",X(1),$S(X(2):$E(X(2),1,30

),1:"\*N/A\*"),X(3),X(4),X(5),DA)

Whole Kill: K ^IBA(355.91,"AUNIQ")

X(1): INSURANCE CO (355.91,.01) (Subscr 1)

(forwards)

X(2): INDEX VALUE CARE UNIT (355.91,.1) (Subscr 2)

(Len 30) (forwards)

X(3): FORM TYPE APPLIED TO (355.91,.04) (Subscr 3)

(forwards)

X(4): BILL CARE TYPE (355.91,.05) (Subscr 4)

(forwards)

X(5): PROVIDER ID TYPE (355.91,.06) (Subscr 5)

(forwards)

355.92,.04 FORM TYPE APPLIED TO 0;4 SET (Required)

'0' FOR UB-04, CMS-1500 and J430D FORMS;

'1' FOR UB-04 FORMS ONLY;

'2' FOR CMS-1500 ONLY;

'3' FOR PRESCRIPTION ONLY;

'4' FOR J430D ONLY;

LAST EDITED: MAY 12, 2017

HELP-PROMPT: Select the forms that the facility id will

apply to.

DESCRIPTION: This designates whether the id number is to be

used for just UB-04 form types, CMS-1500 form

types, J430D form types, or all form types.

RECORD INDEX: AUNIQ (#140) MUMPS IR SORTING ONLY

Short Descr: Xref by ins co,care unit,form type,division,prov id type

Description: This cross reference allows edits to the additonal provider id's to be

replicated to linked insurance companies.

Set Logic: S ^IBA(355.92,"AUNIQ",X(1),$E(X(2),1,30),X(3),X

(4),X(5),DA)=""

Set Cond: S X=0 I X(1)]"",X(2)]"",X(3)]"",X(4)]"",X(5)]""

,$P($G(^IBA(355.92,DA,0)),U,8)="A" S X=1

Kill Logic: K ^IBA(355.92,"AUNIQ",X(1),$E(X(2),1,30),X(3),X

(4),X(5),DA)

Kill Cond: S X=0 I X(1)]"",X(2)]"",X(3)]"",X(4)]"",X(5)]""

S X=1

Whole Kill: K ^IBA(355.92,"AUNIQ")

X(1): INSURANCE COMPANY (355.92,.01) (forwards)

X(2): INDEX VALUE CARE UNIT (355.92,.1) (Len 30)

(forwards)

X(3): FORM TYPE APPLIED TO (355.92,.04) (forwards)

X(4): INDEX VALUE DIVISION (355.92,.11) (forwards)

X(5): PROVIDER ID TYPE (355.92,.06) (forwards)

350.9,1.22 MULTIPLE FORM TYPES 1;22 SET

'1' FOR YES;

'0' FOR NO;

LAST EDITED: APR 28, 2017

HELP-PROMPT: Enter 'Y'es if your facility uses the CMS-1500

& J430D as well as the UB claim form.

DESCRIPTION: Set this field to 'YES' if the facility uses

more than one health insurance form type.

Therefore, if your site uses the UB form and

the CMS-1500 & J430D forms, this should be

answered 'YES'. If your site is only using the

UB form, then answer 'NO'. If this is set to

'NO' or left blank then only the UB type claim

forms will be allowed.

355.96,.04 FORM TYPE APPLIED TO 0;4 SET (Required)

'0' FOR UB-04, CMS-1500 and J430D FORMS;

'1' FOR UB-04 FORM ONLY;

'2' FOR CMS-1500 FORM ONLY;

'4' FOR J430D FORM ONLY;

LAST EDITED: MAY 12, 2017

HELP-PROMPT: Select one or both forms that the provider id

will apply to.

DESCRIPTION: This designates whether the id number is to be

used for the UB-04, CMS-1500, and J430D form

types, just the UB-04 form type, just the

CMS-1500 form type or just the J430D form type.

355.97,.07 ALLOWABLE FORM TYPE 0;7 SET

'I' FOR INSTITUTIONAL;

'P' FOR PROFESSIONAL;

'B' FOR BOTH INSTITUTIONAL AND PROFESSIONAL;

LAST EDITED: APR 28, 2017

HELP-PROMPT: Enter I if this is used on UB type forms, P if

used on CMS & J430D type forms, or B if used on

either type.

DESCRIPTION: This is a flag used to determine what type of

form this qualifier is valid for. It is used

to validate provider id file set-up.

364.4,.05 FORM TYPE 0;5 SET (Required)

'1' FOR INSTITUTIONAL (UB-04) only;

'2' FOR PROFESSIONAL (CMS-1500 and J430D) only;

'3' FOR Both INST (UB-04) and PROF (CMS-1500 and

J430D);

LAST EDITED: APR 28, 2017

HELP-PROMPT: Select the form type this rule will apply to

DESCRIPTION: This is the form type that the rule will be

executed for.

RECORD INDEX: AD (#133) MUMPS IR SORTING ONLY

Short Descr: Sets the first level of xref to determine which

edits are most general.

Set Logic: S ^IBE(364.4,"AD",X2(1),X2(2),$S(X2(3)=3:X2(3),

X2(6)=0:X2(3)#2+1,1:X2(3)),X2(4),DA)=+X2(5)

Set Cond: I X2(1)=""!(X2(2)="")!(X2(3)="")!(X2(4)="") S X

=0

Kill Logic: K ^IBE(364.4,"AD",X1(1),X1(2),$S(X1(3)=3:X1(3),

X1(6)=0:X1(3)#2+1,1:X1(3)),X1(4),DA)

Kill Cond: I X1(1)=""!(X1(2)="")!(X1(3)="")!(X1(4)="") S X

=0

Whole Kill: K ^IBE(364.4,"AD")

X(1): TRANSMISSION TYPE (364.4,.03) (Subscr 1)

(forwards)

X(2): TYPE OF CARE (364.4,.04) (Subscr 2)

(forwards)

X(3): FORM TYPE (364.4,.05) (Subscr 3) (forwards)

X(4): INSURANCE COMPANY OPTION (364.4,.07)

(Subscr 4) (forwards)

X(5): INACTIVE DATE (364.4,.06) (forwards)

X(6): TYPE OF RULE (364.4,.11) (forwards)

364.1,.06 BILL TYPE 0;6 SET

'2' FOR CMS-1500 (PROF);

'3' FOR UB-04 (INST);

'7' FOR J430D (DENT);

LAST EDITED: MAR 09, 2017

HELP-PROMPT: ENTER THE TYPE OF BILL CONTAINED IN THIS BATCH

DESCRIPTION: THIS IS THE TYPE OF BILL THAT IS CONTAINED IN

THIS BATCH

36,4.03 SECONDARY ID REQUIREMENTS 4;3 SET

'0' FOR NONE REQUIRED;

'1' FOR CMS-1500 REQUIRED;

'2' FOR UB-04 REQUIRED;

'3' FOR BOTH UB-04 AND CMS-1500 REQUIRED;

'4' FOR J430D REQUIRED;

'5' FOR BOTH CMS-1500 AND J430D REQUIRED;

'6' FOR UB-04, CMS-1500 AND J430D REQUIRED;

LAST EDITED: MAY 30, 2017

HELP-PROMPT: Enter the code to specify the secondary

performing provider id requirement for this ins co

by form type

DESCRIPTION: This field is used to identify if the insurance

company requires the performing provider

secondary id on the UB-04, the CMS-1500 or the

J430D.

36,4.05 REF PROV SEC ID REQ ON CLAIMS 4;5 SET

'1' FOR CMS-1500 & J430D;

'0' FOR NONE;

LAST EDITED: MAY 30, 2017

HELP-PROMPT: Enter 1 if this qualifier is required on

CMS-1500 claims that have a referring provider

DESCRIPTION: Set this field to CMS-1500 & J430D if the

default ID for a Referring Provider is REQUIRED

on a claim.

NOTES: TRIGGERED by the NAME field of the INSURANCE

COMPANY File

399.0222,.01 FUNCTION 0;1 SET (Required) (Multiply asked)

'1' FOR REFERRING;

'2' FOR OPERATING;

'3' FOR RENDERING;

'4' FOR ATTENDING;

'5' FOR SUPERVISING;

'9' FOR OTHER OPERATING;

'6' FOR ASSISTANT SURGEON;

LAST EDITED: MAR 07, 2017

HELP-PROMPT: Select the function performed by a provider for

this bill.

DESCRIPTION: There are providers who performed specific

functions for the services on this bill. These

providers are needed to enable the V.A. to

collect reimbursement when more than one

provider function is involved in the billable

episode (like an operating physician or

referring provider). This data identifies the

type of function that was performed by a

provider. There can only be 1 provider recorded

for each function on a claim.

SCREEN: S DIC("S")="I $$PRVOK^IBCEU(+Y,$S($G(D0):D0,1

:$G(DA)))"

EXPLANATION: Function must match bill form type. Use '??'

to see the function definitions.

EXECUTABLE HELP: D PRVHELP^IBCEU5

CROSS-REFERENCE: 399.0222^B

1)= S ^DGCR(399,DA(1),"PRV","B",$E(X,1,30),DA)=""

2)= K ^DGCR(399,DA(1),"PRV","B",$E(X,1,30),DA)

CROSS-REFERENCE: ^^TRIGGER^399.0222^.04

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

DIV(1)=D1 S Y(0)=X S X=Y(0),X=X S X=X'=1

I X S X=DIV S (1)=$S($D(^DGCR(399,D0,"PRV",D1,0)):

^(0),1:"") S X=$P(Y(1),U,4),X=X S DIU=X K Y S X=""

X ^DD(399.0222,.01,1,2,1.4)

1.4)= S DIH=$S($D(^DGCR(399,DIV(0),"PRV",DIV(1)

,0)):^(0),1:""),DIV=X S $P(^(0),U,4)=DIV,

DIH=399.0222,DIG=.04 D ^DICR:$O(^DD(DIH,DIG,1,0))>0

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

DIV(1)=D1 S Y(1)=$S($D(^DGCR(399,D0,"PRV",D1,0)):

^(0),1:"") S X=$P(Y(1),U,4),X=X S DIU=X K Y

S X="" X ^DD(399.0222,.01,1,2,2.4)

2.4)= S DIH=$S($D(^DGCR(399,DIV(0),"PRV",DIV(1)

,0)):^(0),1:""),DIV=X S $P(^(0),U,4)=DIV,

DIH=399.0222,DIG=.04 D ^DICR:$O(^DD(DIH,DIG,1,0))>0

3)= Do not delete

CREATE CONDITION)= INTERNAL(PROVIDER FUNCTION)'=1

CREATE VALUE)= @

DELETE VALUE)= @

FIELD)= PROVIDER STATE

This xref is used to delete the state field if

the provider is not a referring provider

function type.

CROSS-REFERENCE: 399.0222^C^MUMPS

1)= S ^DGCR(399,DA(1),"PRV","C",$E(

$$EXTERNAL^DILFD(399.0222,.01,,X),1,30),DA)=""

2)= K ^DGCR(399,DA(1),"PRV","C",$E(

$$EXTERNAL^DILFD(399.0222,.01,,X),1,30),DA)

CROSS-REFERENCE: 399.0222^CA^MUMPS

1)= S ^DGCR(399,DA(1),"PRV","C",$$LOW^XLFSTR(

$E($$EXTERNAL^DILFD(399.0222,.01,,X),1,30)),DA)=""

2)= K ^DGCR(399,DA(1),"PRV","C",$$LOW^XLFSTR(

$E($$EXTERNAL^DILFD(399.0222,.01,,X),1,30)),DA)

399.0404,.01 LINE FUNCTION 0;1 SET (Multiply asked)

'1' FOR REFERRING;

'2' FOR OPERATING;

'3' FOR RENDERING;

'4' FOR ATTENDING;

'5' FOR SUPERVISING;

'9' FOR OTHER OPERATING;

'6' FOR ASSISTANT SURGEON;

LAST EDITED: MAR 01, 2017

HELP-PROMPT: Select the function performed by a provider for

this claim line.

DESCRIPTION: There are providers who performed specific

functions for the services on this claim line.

These providers are needed to enable the V.A.

to collect reimbursement when more than one

provider function is involved in the billable

episode (like an operating physician or

referring provider). This data identifies the

type of function that was performed by a

provider. There can only be 1 provider

recorded for each function on a claim line.

SCREEN: S DIC("S")="I $$LNPRVOK^IBCEU7(+Y,$G(DA(2)))"

EXPLANATION: Function must match bill form type. Use '??' to

see the function definitions.

EXECUTABLE HELP: D LNPRVHLP^IBCEU7

CROSS-REFERENCE: 399.0404^B

1)= S ^DGCR(399,DA(2),"CP",DA(1),"LNPRV","B",

$E(X,1,30),DA)=""

2)= K ^DGCR(399,DA(2),"CP",DA(1),"LNPRV","B",

$E(X,1,30),DA)

CROSS-REFERENCE: ^^TRIGGER^399.0404^.04

1)= K DIV S DIV=X,D0=DA(2),DIV(0)=D0,D1=DA(1),

DIV(1)=D1,D2=DA,DIV(2)=D2 S Y(0)=X S X=Y(0),X=X

S X=X'=1 I X S X=DIV S Y(1)=$S($D(^DGCR(399,D0,

"CP",D1,"LNPRV",D2,0)):^(0),1:"") S X=$P(Y(1),U,4),X=X

S DIU=X K Y S X="" X ^DD(399.0404,.01,1,2,1.4)

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

"LNPRV",DIV(2),0)),DIV=X S $P(^(0),U,4)=DIV,

IH=399.0404,DIG=.04 D ^DICR

2)= K DIV S DIV=X,D0=DA(2),DIV(0)=D0,D1=DA(1),

DIV(1)=D1,D2=DA,DIV(2)=D2 S Y(1)=$S(

$D(^DGCR(399,D0,"CP",D1,"LNPRV",D2,0)):^(0),1:"")

S X=$P(Y(1),U,4),X=X S DIU=X K Y S X=""

X ^DD(399.0404,.01,1,2,2.4)

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

DIV(2),0)),DIV=X S $P(^(0),U,4)=DIV,DIH=399.0404,

DIG=.04 D ^DICR

3)= Do Not Delete

CREATE CONDITION)= INTERNAL(LINE PROVIDER LINE

FUNCTION)'=1

CREATE VALUE)= @

DELETE VALUE)= @

FIELD)= LINE PROVIDER STATE

This xref is used to delete the state field if the provider is not a referring provider

function type.

FIELD INDEX: C (#172) REGULAR IR LOOKUP & SORTING

Short Descr: External value of LINE FUNCTION field.

Set Logic: S ^DGCR(399,DA(2),"CP",DA(1),"LNPRV","C",

$E(X(2),1,30),DA)=""

Kill Logic: K ^DGCR(399,DA(2),"CP",DA(1),"LNPRV","C",

$E(X(2),1,30),DA)

Whole Kill: K ^DGCR(399,DA(2),"CP",DA(1),"LNPRV","C")

X(1): LINE FUNCTION (399.0404,.01) (Len 30)

(forwards)

X(2): Computed Code: S X=$$EXTERNAL^DILFD(399.0404,.0

1,,X(1))

(Subscr 1) (Len 30)

1. The Input Template IBEDIT INS CO1 for the INSURANCE COMPANY (#36) file needs to be modified in order to process Dental claims. This Input Template will be modified to allow for the entry of the new Insurance Payer ID for Dental and also the fields related to Provider Secondary Insurance IDs.

NUMBER: 1838 NAME: IBEDIT INS CO1

DATE CREATED: JUL 21, 2017@13:11 FILE: INSURANCE COMPANY

USER #: 520824637 DATE LAST USED: AUG 07, 2017

EDIT FIELDS (c)

: S:",6,"'[IBY Y="@0"

: NAME

: @0

: S:",0,1,6,12,"'[IBY Y="@10"

: S:",12,"[IBY Y="@18"

: SIGNATURE REQUIRED ON BILL?

: REIMBURSE?

: ALLOW MULTIPLE BEDSECTIONS

: DIFFERENT REVENUE CODES TO USE

: ONE OPT. VISIT ON BILL ONLY

: AMBULATORY SURG. REV. CODE

: PRESCRIPTION REFILL REV. CODE

: STANDARD FTF;"STANDARD FILING TIME FRAME"

: I 'X S Y="@016"

: I '$$FTFV^IBCNSU31(X) S Y="@016"

: STANDARD FTF VALUE;"STANDARD FILING TIME FRAME VALUE"

: @016

: FILING TIME FRAME

: TYPE OF COVERAGE

: BILLING PHONE NUMBER

: VERIFICATION PHONE NUMBER

: ANOTHER CO. PROCESS PRECERTS?;T

: S:'X Y="@11"

: PRECERT COMPANY NAME

: S Y="@16"

: @11

: PRECERTIFICATION PHONE NUMBER

: @16

: I '$$KCHK^XUSRB("IB EDI INSURANCE EDIT") S Y="@171"

: TRANSMIT ELECTRONICALLY;"EDI - Transmit?"

: S DIPA("IBTX")=X

: I X=$G(IBEDIKEY(1))!$$KCHK^XUSRB("IB EDI INSURANCE EDIT") S Y="

@1721"

: TRANSMIT ELECTRONICALLY////^S X=$G(IBEDIKEY(1))

: I $$EDIKEY^IBCNSC()

: S Y="@171"

: @1721

: I '$G(DIPA("IBTX")) S Y="@17"

: EDI ID NUMBER - INST;"EDI - Inst Payer Primary ID"

: I X=$G(IBEDIKEY(4))!$$KCHK^XUSRB("IB EDI INSURANCE EDIT") S Y="

@17211"

: EDI ID NUMBER - INST////^S X=$G(IBEDIKEY(4))

: I $$EDIKEY^IBCNSC()

: S Y="@171"

: @17211

: 15;"EDI - Alt Inst Payer Primary ID Type"

: .01;"EDI - Alt Inst Payer Primary ID Type"

: .02;"EDI - Alt Inst Payer Primary ID"

: I '$G(DIPA("IBTX")) S Y="@17"

: EDI INST SECONDARY ID QUAL(1);"EDI - 1ST Inst Payer Sec. ID Qua

lifier"

: I X=""&($G(IBEDIKEY(3,6))="")&$$KCHK^XUSRB("IB EDI INSURANCE ED

IT") S Y="@1722"

: I X=$G(IBEDIKEY(1,6))!$$KCHK^XUSRB("IB EDI INSURANCE EDIT") S Y

="@17212"

: EDI INST SECONDARY ID QUAL(1)////^S X=$G(IBEDIKEY(1,6))

: EDI INST SECONDARY ID(1)////^S X=$G(IBEDIKEY(2,6))

: I $$EDIKEY^IBCNSC()

: S Y="@171"

: @17212

: I '$G(DIPA("IBTX")) S Y="@17"

: EDI INST SECONDARY ID(1);"EDI - 1ST Inst Payer Sec. ID"

: I X=$G(IBEDIKEY(2,6))!$$KCHK^XUSRB("IB EDI INSURANCE EDIT") S Y

="@17213"

: EDI INST SECONDARY ID(1)////^S X=$G(IBEDIKEY(2,6))

: EDI INST SECONDARY ID QUAL(1)////^S X=$G(IBEDIKEY(1,6))

: I $$EDIKEY^IBCNSC()

: S Y="@171"

: @17213

: I '$G(DIPA("IBTX")) S Y="@17"

: EDI INST SECONDARY ID QUAL(2);"EDI - 2ND Inst Payer Sec. ID Qua

lifier"

: I X=""&$$KCHK^XUSRB("IB EDI INSURANCE EDIT") S Y="@1722"

: I X=$G(IBEDIKEY(3,6))!$$KCHK^XUSRB("IB EDI INSURANCE EDIT") S Y

="@17214"

: EDI INST SECONDARY ID QUAL(2)////^S X=$G(IBEDIKEY(3,6))

: EDI INST SECONDARY ID(2)////^S X=$G(IBEDIKEY(4,6))

: I $$EDIKEY^IBCNSC()

: S Y="@171"

: @17214

: I '$G(DIPA("IBTX")) S Y="@17"

: EDI INST SECONDARY ID(2);"EDI - 2ND Inst Payer Sec. ID"

: I X=$G(IBEDIKEY(4,6))!$$KCHK^XUSRB("IB EDI INSURANCE EDIT") S Y

="@1722"

: EDI INST SECONDARY ID(2)////^S X=$G(IBEDIKEY(4,6))

: EDI INST SECONDARY ID QUAL(2)////^S X=$G(IBEDIKEY(3,6))

: I $$EDIKEY^IBCNSC()

: S Y="@171"

: @1722

: EDI ID NUMBER - PROF;"EDI - Prof Payer Primary ID"

: I X=$G(IBEDIKEY(2))!$$KCHK^XUSRB("IB EDI INSURANCE EDIT") S Y="

@17221"

: EDI ID NUMBER - PROF////^S X=$G(IBEDIKEY(2))

: I $$EDIKEY^IBCNSC()

: S Y="@171"

: @17221

: 16;"EDI - Alt Prof Payer Primary ID Type"

: .01;"EDI - Alt Prof Payer Primary ID Type"

: .02;"EDI - Alt Prof Payer Primary ID"

: I '$G(DIPA("IBTX")) S Y="@17"

: EDI PROF SECONDARY ID QUAL(1);"EDI - 1ST Prof Payer Sec. ID Qua

lifier"

: I X=""&($G(IBEDIKEY(7,6))="")&$$KCHK^XUSRB("IB EDI INSURANCE ED

IT") S Y="@1723"

: I X=$G(IBEDIKEY(5,6))!$$KCHK^XUSRB("IB EDI INSURANCE EDIT") S Y

="@17222"

: EDI PROF SECONDARY ID QUAL(1)////^S X=$G(IBEDIKEY(5,6))

: EDI PROF SECONDARY ID(1)////^S X=$G(IBEDIKEY(6,6))

: I $$EDIKEY^IBCNSC()

: S Y="@171"

: @17222

: I '$G(DIPA("IBTX")) S Y="@17"

: EDI PROF SECONDARY ID(1);"EDI - 1ST Prof Payer Sec. ID"

: I X=$G(IBEDIKEY(6,6))!$$KCHK^XUSRB("IB EDI INSURANCE EDIT") S Y

="@17223"

: EDI PROF SECONDARY ID(1)////^S X=$G(IBEDIKEY(6,6))

: EDI PROF SECONDARY ID QUAL(1)////^S X=$G(IBEDIKEY(5,6))

: I $$EDIKEY^IBCNSC()

: S Y="@171"

: @17223

: I '$G(DIPA("IBTX")) S Y="@17"

: EDI PROF SECONDARY ID QUAL(2);"EDI - 2ND Prof Payer Sec. ID Qua

lifier"

: I X=""&$$KCHK^XUSRB("IB EDI INSURANCE EDIT") S Y="@1723"

: I X=$G(IBEDIKEY(7,6))!$$KCHK^XUSRB("IB EDI INSURANCE EDIT") S Y

="@17224"

: EDI PROF SECONDARY ID QUAL(2)////^S X=$G(IBEDIKEY(7,6))

: EDI PROF SECONDARY ID(2)////^S X=$G(IBEDIKEY(8,6))

: I $$EDIKEY^IBCNSC()

: S Y="@171"

: @17224

: I '$G(DIPA("IBTX")) S Y="@17"

: EDI PROF SECONDARY ID(2);"EDI - 2ND Prof Payer Sec. ID"

: I X=$G(IBEDIKEY(8,6))!$$KCHK^XUSRB("IB EDI INSURANCE EDIT") S Y

="@1723"

: EDI PROF SECONDARY ID(2)////^S X=$G(IBEDIKEY(8,6))

: EDI PROF SECONDARY ID QUAL(2)////^S X=$G(IBEDIKEY(7,6))

: I $$EDIKEY^IBCNSC()

: S Y="@171"

: @1723

**: EDI ID NUMBER - DENTAL;"EDI - Dental Payer Primary ID"**

**: I X=$G(IBEDIKEY(15))!$$KCHK^XUSRB("IB EDI INSURANCE EDIT") S Y=**

**"@17225"**

**: EDI ID NUMBER - DENTAL////^S X=$G(IBEDIKEY(15))**

**: I $$EDIKEY^IBCNSC()**

**: S Y="@171"**

**: @17225**

: @17

: ELECTRONIC INSURANCE TYPE;"EDI - Insurance Type"

: I X=$G(IBEDIKEY(9))!$$KCHK^XUSRB("IB EDI INSURANCE EDIT") S Y="

@1724"

: ELECTRONIC INSURANCE TYPE////^S X=$G(IBEDIKEY(9))

: I $$EDIKEY^IBCNSC()

: S Y="@171"

: @1724

: @171

: BIN NUMBER;"EDI - Bin Number"

: I '$$KCHK^XUSRB("IB EDI INSURANCE EDIT") S Y="@1725"

: EDI - UMO (278) ID

: @1725

: PRINT SEC/TERT AUTO CLAIMS?;"EDI - Print Sec/Tert Auto Claims?"

: PRINT SEC MED CLAIMS W/O MRA;"EDI - Print Medicare Sec Claims w

/o MRA?"

: I $G(DIPA("IBTX"))'=2 S Y="@18"

: MAX NUMBER TEST BILLS PER DAY;"MAX # TEST BILLS TO TRANSMIT PER

DAY"

: @18

: S:",6,12,"'[IBY Y="@181"

: W !!,"Attending/Rendering Provider Secondary ID"

: PERF PROV SECOND ID TYPE 1500;"Default ID (1500)"

**: PERF PROV SECOND ID TYPE J430D;"Default ID (J430D)"**

: PERF PROV SECOND ID TYPE UB;"Default ID (UB)"

: SECONDARY ID REQUIREMENTS;"Require ID on Claim"

: W !!,"Referring Provider Secondary ID"

: REF PROV SEC ID DEF CMS-1500//UPIN;"Default ID (1500)"

**: REF PROV SEC ID DEF J430D//UPIN;"Default ID (J430D)"**

: REF PROV SEC ID REQ ON CLAIMS;"Require ID on Claim"

: W !!,"Billing Provider Secondary IDs"

: ATT/REND ID BILL SEC ID PROF//NO;"Use Att/Rend ID as Billing Pr

ovider Sec. ID (1500)?"

**: ATT/REND ID BILL SEC ID J430D//NO;"Use Att/Rend ID as Billing P**

**rovider Sec. ID (J430D)?"**

: ATT/REND ID BILL SEC ID INST//NO;"Use Att/Rend ID as Billing Pr

ovider Sec. ID (UB)?"

: W !!,"Billing Provider/Service Facility"

: @181

: S:IBY["1" Y="@99"

: @10

: S:",0,2,6,"'[IBY Y="@20"

: STREET ADDRESS [LINE 1]

: S:X="" Y="@1"

: STREET ADDRESS [LINE 2]

: S:X="" Y="@1"

: STREET ADDRESS [LINE 3]

: @1

: CITY

: STATE

: ZIP CODE

: PHONE NUMBER

: FAX NUMBER

: S:(IBY["0")!(IBY["2") Y="@99"

: @20

: S:",3,6,"'[IBY Y="@30"

: ANOTHER CO. PROCESS IP CLAIMS?;T

: S:'X Y="@21"

: CLAIMS (INPT) COMPANY NAME

: S Y="@26"

: @21

: CLAIMS (INPT) STREET ADDRESS 1

: S:X="" Y="@2"

: CLAIMS (INPT) STREET ADDRESS 2

: S:X="" Y="@2"

: CLAIMS (INPT) STREET ADDRESS 3

: @2

: CLAIMS (INPT) PROCESS CITY

: CLAIMS (INPT) PROCESS STATE

: CLAIMS (INPT) PROCESS ZIP

: CLAIMS (INPT) PHONE NUMBER

: CLAIMS (INPT) FAX

: @26

: S:IBY["3" Y="@99"

: @30

: S:",10,6,"'[IBY Y="@80"

: ANOTHER CO. PROCESS OP CLAIMS?;T

: S:'X Y="@31"

: CLAIMS (OPT) COMPANY NAME

: S Y="@36"

: @31

: CLAIMS (OPT) STREET ADDRESS 1

: S:X="" Y="@5"

: CLAIMS (OPT) STREET ADDRESS 2

: S:X="" Y="@5"

: CLAIMS (OPT) STREET ADDRESS 3

: @5

: CLAIMS (OPT) PROCESS CITY

: CLAIMS (OPT) PROCESS STATE

: CLAIMS (OPT) PROCESS ZIP

: CLAIMS (OPT) PHONE NUMBER

: CLAIMS (OPT) FAX

: @36

: S:IBY["10" Y="@99"

: @80

: S:",11,6,"'[IBY Y="@90"

: ANOTHER CO. PROCESS RX CLAIMS?;T

: S:'X Y="@81"

: CLAIMS (RX) COMPANY NAME

: S Y="@86"

: @81

: CLAIMS (RX) STREET ADDRESS 1

: S:X="" Y="@6"

: CLAIMS (RX) STREET ADDRESS 2

: S:X="" Y="@6"

: CLAIMS (RX) STREET ADDRESS 3

: @6

: CLAIMS (RX) CITY

: CLAIMS (RX) STATE

: CLAIMS (RX) ZIP

: CLAIMS (RX) PHONE NUMBER

: CLAIMS (RX) FAX

: @86

: S:IBY["11" Y="@99"

: @90

: S:",4,6,"'[IBY Y="@40"

: ANOTHER CO. PROCESS APPEALS?;T

: S:'X Y="@41"

: APPEALS COMPANY NAME

: S Y="@46"

: @41

: APPEALS ADDRESS ST. [LINE 1]

: S:X="" Y="@3"

: APPEALS ADDRESS ST. [LINE 2]

: S:X="" Y="@3"

: APPEALS ADDRESS ST. [LINE 3]

: @3

: APPEALS ADDRESS CITY

: APPEALS ADDRESS STATE

: APPEALS ADDRESS ZIP

: APPEALS PHONE NUMBER

: APPEALS FAX

: @46

: S:IBY["4" Y="@99"

: @40

: S:",5,6,"'[IBY Y="@55"

: ANOTHER CO. PROCESS INQUIRIES?;T

: S:'X Y="@51"

: INQUIRY COMPANY NAME

: S Y="@56"

: @51

: INQUIRY ADDRESS ST. [LINE 1]

: S:X="" Y="@4"

: INQUIRY ADDRESS ST. [LINE 2]

: S:X="" Y="@4"

: INQUIRY ADDRESS ST. [LINE 3]

: @4

: INQUIRY ADDRESS CITY

: INQUIRY ADDRESS STATE

: INQUIRY ADDRESS ZIP CODE

: INQUIRY PHONE NUMBER

: INQUIRY FAX

: @56

: S:IBY["5" Y="@99"

: @55

: S:",13,6,"'[IBY Y="@60"

: I '$$KCHK^XUSRB("IB EDI INSURANCE EDIT") S Y="@551"

: I $D(^DIC(36,"APC",+$G(DA))),$P($G(^DIC(36,+$G(DA),3)),U,13)="P

" S Y="@551"

: INS COMPANY LINK TYPE;T

: S DIPA("IBLNK")=X

: I X=$G(IBEDIKEY(13))!$$KCHK^XUSRB("IB EDI INSURANCE EDIT") S Y=

"@552"

: INS COMPANY LINK TYPE////^S X=$G(IBEDIKEY(13))

: I $$EDIKEY^IBCNSC()

: S Y="@551"

: @552

: I $G(DIPA("IBLNK"))'="C" S Y="@551"

: INS COMPANY LINK PARENT;T

: I X=$G(IBEDIKEY(14))!$$KCHK^XUSRB("IB EDI INSURANCE EDIT") S Y=

"@553"

: INS COMPANY LINK PARENT////^S X=$G(IBEDIKEY(14))

: I $$EDIKEY^IBCNSC()

: S Y="@551"

: @553

: D COPY^IBCEPCID(+$G(DA))

: @551

: S:IBY=",13," Y="@99"

: @60

: S:",7,6,"'[IBY Y="@50"

: PAYER

: @50

: S:",8,6,"'[IBY Y="@70"

: REMARKS

: S:IBY["8" Y="@99"

: @70

: S:",9,6,"'[IBY Y="@99"

: SYNONYM

: ALL

: @99 COMPILED (c): NO

EDIT FIELDS (c): S:",6,"'[IBY Y="@0"

: NAME

: @0

: S:",0,1,6,12,"'[IBY Y="@10"

: S:",12,"[IBY Y="@18"

: SIGNATURE REQUIRED ON BILL?

: REIMBURSE?

: ALLOW MULTIPLE BEDSECTIONS

: DIFFERENT REVENUE CODES TO USE

: ONE OPT. VISIT ON BILL ONLY

: AMBULATORY SURG. REV. CODE

: PRESCRIPTION REFILL REV. CODE

: STANDARD FTF;"STANDARD FILING TIME FRAME"

: I 'X S Y="@016"

: I '$$FTFV^IBCNSU31(X) S Y="@016"

: STANDARD FTF VALUE;"STANDARD FILING TIME FRAME VALUE"

: @016

: FILING TIME FRAME

: TYPE OF COVERAGE

: BILLING PHONE NUMBER

: VERIFICATION PHONE NUMBER

: ANOTHER CO. PROCESS PRECERTS?;T

: S:'X Y="@11"

: PRECERT COMPANY NAME

: S Y="@16"

: @11

: PRECERTIFICATION PHONE NUMBER

: @16

: I '$$KCHK^XUSRB("IB EDI INSURANCE EDIT") S Y="@171"

: TRANSMIT ELECTRONICALLY;"EDI - Transmit?"

: S DIPA("IBTX")=X

: I X=$G(IBEDIKEY(1))!$$KCHK^XUSRB("IB EDI INSURANCE EDIT") S Y="

@1721"

: 3.01////^S X=$G(IBEDIKEY(1))

: I $$EDIKEY^IBCNSC()

: S Y="@171"

: @1721

: I '$G(DIPA("IBTX")) S Y="@17"

: EDI ID NUMBER - INST;"EDI - Inst Payer Primary ID"

: I X=$G(IBEDIKEY(4))!$$KCHK^XUSRB("IB EDI INSURANCE EDIT") S Y="

@17211"

: 3.04////^S X=$G(IBEDIKEY(4))

: I $$EDIKEY^IBCNSC()

: S Y="@171"

: @17211

: 15;"EDI - Alt Inst Payer Primary ID Type"

: .01;"EDI - Alt Inst Payer Primary ID Type"

: .02;"EDI - Alt Inst Payer Primary ID"

: I '$G(DIPA("IBTX")) S Y="@17"

: EDI INST SECONDARY ID QUAL(1);"EDI - 1ST Inst Payer Sec. ID Qua

lifier"

: I X=""&($G(IBEDIKEY(3,6))="")&$$KCHK^XUSRB("IB EDI INSURANCE ED

IT") S Y="@1722"

: I X=$G(IBEDIKEY(1,6))!$$KCHK^XUSRB("IB EDI INSURANCE EDIT") S Y

="@17212"

: 6.01////^S X=$G(IBEDIKEY(1,6))

: 6.02////^S X=$G(IBEDIKEY(2,6))

: I $$EDIKEY^IBCNSC()

: S Y="@171"

: @17212

: I '$G(DIPA("IBTX")) S Y="@17"

: EDI INST SECONDARY ID(1);"EDI - 1ST Inst Payer Sec. ID"

: I X=$G(IBEDIKEY(2,6))!$$KCHK^XUSRB("IB EDI INSURANCE EDIT") S Y

="@17213"

: 6.02////^S X=$G(IBEDIKEY(2,6))

: 6.01////^S X=$G(IBEDIKEY(1,6))

: I $$EDIKEY^IBCNSC()

: S Y="@171"

: @17213

: I '$G(DIPA("IBTX")) S Y="@17"

: EDI INST SECONDARY ID QUAL(2);"EDI - 2ND Inst Payer Sec. ID Qua

lifier"

: I X=""&$$KCHK^XUSRB("IB EDI INSURANCE EDIT") S Y="@1722"

: I X=$G(IBEDIKEY(3,6))!$$KCHK^XUSRB("IB EDI INSURANCE EDIT") S Y

="@17214"

: 6.03////^S X=$G(IBEDIKEY(3,6))

: 6.04////^S X=$G(IBEDIKEY(4,6))

: I $$EDIKEY^IBCNSC()

: S Y="@171"

: @17214

: I '$G(DIPA("IBTX")) S Y="@17"

: EDI INST SECONDARY ID(2);"EDI - 2ND Inst Payer Sec. ID"

: I X=$G(IBEDIKEY(4,6))!$$KCHK^XUSRB("IB EDI INSURANCE EDIT") S Y

="@1722"

: 6.04////^S X=$G(IBEDIKEY(4,6))

: 6.03////^S X=$G(IBEDIKEY(3,6))

: I $$EDIKEY^IBCNSC()

: S Y="@171"

: @1722

: EDI ID NUMBER - PROF;"EDI - Prof Payer Primary ID"

: I X=$G(IBEDIKEY(2))!$$KCHK^XUSRB("IB EDI INSURANCE EDIT") S Y="

@17221"

: 3.02////^S X=$G(IBEDIKEY(2))

: I $$EDIKEY^IBCNSC()

: S Y="@171"

: @17221

: 16;"EDI - Alt Prof Payer Primary ID Type"

: .01;"EDI - Alt Prof Payer Primary ID Type"

: .02;"EDI - Alt Prof Payer Primary ID"

: I '$G(DIPA("IBTX")) S Y="@17"

: EDI PROF SECONDARY ID QUAL(1);"EDI - 1ST Prof Payer Sec. ID Qua

lifier"

: I X=""&($G(IBEDIKEY(7,6))="")&$$KCHK^XUSRB("IB EDI INSURANCE ED

IT") S Y="@1723"

: I X=$G(IBEDIKEY(5,6))!$$KCHK^XUSRB("IB EDI INSURANCE EDIT") S Y

="@17222"

: 6.05////^S X=$G(IBEDIKEY(5,6))

: 6.06////^S X=$G(IBEDIKEY(6,6))

: I $$EDIKEY^IBCNSC()

: S Y="@171"

: @17222

: I '$G(DIPA("IBTX")) S Y="@17"

: EDI PROF SECONDARY ID(1);"EDI - 1ST Prof Payer Sec. ID"

: I X=$G(IBEDIKEY(6,6))!$$KCHK^XUSRB("IB EDI INSURANCE EDIT") S Y

="@17223"

: 6.06////^S X=$G(IBEDIKEY(6,6))

: 6.05////^S X=$G(IBEDIKEY(5,6))

: I $$EDIKEY^IBCNSC()

: S Y="@171"

: @17223

: I '$G(DIPA("IBTX")) S Y="@17"

: EDI PROF SECONDARY ID QUAL(2);"EDI - 2ND Prof Payer Sec. ID Qua

lifier"

: I X=""&$$KCHK^XUSRB("IB EDI INSURANCE EDIT") S Y="@1723"

: I X=$G(IBEDIKEY(7,6))!$$KCHK^XUSRB("IB EDI INSURANCE EDIT") S Y

="@17224"

: 6.07////^S X=$G(IBEDIKEY(7,6))

: 6.08////^S X=$G(IBEDIKEY(8,6))

: I $$EDIKEY^IBCNSC()

: S Y="@171"

: @17224

: I '$G(DIPA("IBTX")) S Y="@17"

: EDI PROF SECONDARY ID(2);"EDI - 2ND Prof Payer Sec. ID"

: I X=$G(IBEDIKEY(8,6))!$$KCHK^XUSRB("IB EDI INSURANCE EDIT") S Y

="@1723"

: 6.08////^S X=$G(IBEDIKEY(8,6))

: 6.07////^S X=$G(IBEDIKEY(7,6))

: I $$EDIKEY^IBCNSC()

: S Y="@171"

: @1723

**: EDI ID NUMBER - DENTAL;"EDI - Dental Payer Primary ID"**

**: I X=$G(IBEDIKEY(15))!$$KCHK^XUSRB("IB EDI INSURANCE EDIT") S Y=**

**"@17225"**

**: 3.15////^S X=$G(IBEDIKEY(15))**

**: I $$EDIKEY^IBCNSC()**

**: S Y="@171"**

**: @17225**

: @17

: ELECTRONIC INSURANCE TYPE;"EDI - Insurance Type"

: I X=$G(IBEDIKEY(9))!$$KCHK^XUSRB("IB EDI INSURANCE EDIT") S Y="

@1724"

: 3.09////^S X=$G(IBEDIKEY(9))

: I $$EDIKEY^IBCNSC()

: S Y="@171"

: @1724

: @171

: BIN NUMBER;"EDI - Bin Number"

: I '$$KCHK^XUSRB("IB EDI INSURANCE EDIT") S Y="@1725"

: EDI - UMO (278) ID

: @1725

: PRINT SEC/TERT AUTO CLAIMS?;"EDI - Print Sec/Tert Auto Claims?"

: PRINT SEC MED CLAIMS W/O MRA;"EDI - Print Medicare Sec Claims w

/o MRA?"

: I $G(DIPA("IBTX"))'=2 S Y="@18"

: MAX NUMBER TEST BILLS PER DAY;"MAX # TEST BILLS TO TRANSMIT PER

DAY"

: @18

: S:",6,12,"'[IBY Y="@181"

: W !!,"Attending/Rendering Provider Secondary ID"

: PERF PROV SECOND ID TYPE 1500;"Default ID (1500)"

**: PERF PROV SECOND ID TYPE J430D;"Default ID (J430D)"**

: PERF PROV SECOND ID TYPE UB;"Default ID (UB)"

: SECONDARY ID REQUIREMENTS;"Require ID on Claim"

: W !!,"Referring Provider Secondary ID"

: REF PROV SEC ID DEF CMS-1500//UPIN;"Default ID (1500)"

**: REF PROV SEC ID DEF J430D//UPIN;"Default ID (J430D)"**

: REF PROV SEC ID REQ ON CLAIMS;"Require ID on Claim"

: W !!,"Billing Provider Secondary IDs"

: ATT/REND ID BILL SEC ID PROF//NO;"Use Att/Rend ID as Billing Pr

ovider Sec. ID (1500)?"

**: ATT/REND ID BILL SEC ID J430D//NO;"Use Att/Rend ID as Billing P**

**rovider Sec. ID (J430D)?"**

: ATT/REND ID BILL SEC ID INST//NO;"Use Att/Rend ID as Billing Pr

ovider Sec. ID (UB)?"

: W !!,"Billing Provider/Service Facility"

: @181

: S:IBY["1" Y="@99"

: @10

: S:",0,2,6,"'[IBY Y="@20"

: STREET ADDRESS [LINE 1]

: S:X="" Y="@1"

: STREET ADDRESS [LINE 2]

: S:X="" Y="@1"

: STREET ADDRESS [LINE 3]

: @1

: CITY

: STATE

: ZIP CODE

: PHONE NUMBER

: FAX NUMBER

: S:(IBY["0")!(IBY["2") Y="@99"

: @20

: S:",3,6,"'[IBY Y="@30"

: ANOTHER CO. PROCESS IP CLAIMS?;T

: S:'X Y="@21"

: CLAIMS (INPT) COMPANY NAME

: S Y="@26"

: @21

: CLAIMS (INPT) STREET ADDRESS 1

: S:X="" Y="@2"

: CLAIMS (INPT) STREET ADDRESS 2

: S:X="" Y="@2"

: CLAIMS (INPT) STREET ADDRESS 3

: @2

: CLAIMS (INPT) PROCESS CITY

: CLAIMS (INPT) PROCESS STATE

: CLAIMS (INPT) PROCESS ZIP

: CLAIMS (INPT) PHONE NUMBER

: CLAIMS (INPT) FAX

: @26

: S:IBY["3" Y="@99"

: @30

: S:",10,6,"'[IBY Y="@80"

: ANOTHER CO. PROCESS OP CLAIMS?;T

: S:'X Y="@31"

: CLAIMS (OPT) COMPANY NAME

: S Y="@36"

: @31

: CLAIMS (OPT) STREET ADDRESS 1

: S:X="" Y="@5"

: CLAIMS (OPT) STREET ADDRESS 2

: S:X="" Y="@5"

: CLAIMS (OPT) STREET ADDRESS 3

: @5

: CLAIMS (OPT) PROCESS CITY

: CLAIMS (OPT) PROCESS STATE

: CLAIMS (OPT) PROCESS ZIP

: CLAIMS (OPT) PHONE NUMBER

: CLAIMS (OPT) FAX

: @36

: S:IBY["10" Y="@99"

: @80

: S:",11,6,"'[IBY Y="@90"

: ANOTHER CO. PROCESS RX CLAIMS?;T

: S:'X Y="@81"

: CLAIMS (RX) COMPANY NAME

: S Y="@86"

: @81

: CLAIMS (RX) STREET ADDRESS 1

: S:X="" Y="@6"

: CLAIMS (RX) STREET ADDRESS 2

: S:X="" Y="@6"

: CLAIMS (RX) STREET ADDRESS 3

: @6

: CLAIMS (RX) CITY

: CLAIMS (RX) STATE

: CLAIMS (RX) ZIP

: CLAIMS (RX) PHONE NUMBER

: CLAIMS (RX) FAX

: @86

: S:IBY["11" Y="@99"

: @90

: S:",4,6,"'[IBY Y="@40"

: ANOTHER CO. PROCESS APPEALS?;T

: S:'X Y="@41"

: APPEALS COMPANY NAME

: S Y="@46"

: @41

: APPEALS ADDRESS ST. [LINE 1]

: S:X="" Y="@3"

: APPEALS ADDRESS ST. [LINE 2]

: S:X="" Y="@3"

: APPEALS ADDRESS ST. [LINE 3]

: @3

: APPEALS ADDRESS CITY

: APPEALS ADDRESS STATE

: APPEALS ADDRESS ZIP

: APPEALS PHONE NUMBER

: APPEALS FAX

: @46

: S:IBY["4" Y="@99"

: @40

: S:",5,6,"'[IBY Y="@55"

: ANOTHER CO. PROCESS INQUIRIES?;T

: S:'X Y="@51"

: INQUIRY COMPANY NAME

: S Y="@56"

: @51

: INQUIRY ADDRESS ST. [LINE 1]

: S:X="" Y="@4"

: INQUIRY ADDRESS ST. [LINE 2]

: S:X="" Y="@4"

: INQUIRY ADDRESS ST. [LINE 3]

: @4

: INQUIRY ADDRESS CITY

: INQUIRY ADDRESS STATE

: INQUIRY ADDRESS ZIP CODE

: INQUIRY PHONE NUMBER

: INQUIRY FAX

: @56

: S:IBY["5" Y="@99"

: @55

: S:",13,6,"'[IBY Y="@60"

: I '$$KCHK^XUSRB("IB EDI INSURANCE EDIT") S Y="@551"

: I $D(^DIC(36,"APC",+$G(DA))),$P($G(^DIC(36,+$G(DA),3)),U,13)="P

" S Y="@551"

: INS COMPANY LINK TYPE;T

: S DIPA("IBLNK")=X

: I X=$G(IBEDIKEY(13))!$$KCHK^XUSRB("IB EDI INSURANCE EDIT") S Y=

"@552"

: 3.13////^S X=$G(IBEDIKEY(13))

: I $$EDIKEY^IBCNSC()

: S Y="@551"

: @552

: I $G(DIPA("IBLNK"))'="C" S Y="@551"

: INS COMPANY LINK PARENT;T

: I X=$G(IBEDIKEY(14))!$$KCHK^XUSRB("IB EDI INSURANCE EDIT") S Y=

"@553"

: 3.14////^S X=$G(IBEDIKEY(14))

: I $$EDIKEY^IBCNSC()

: S Y="@551"

: @553

: D COPY^IBCEPCID(+$G(DA))

: @551

: S:IBY=",13," Y="@99"

: @60

: S:",7,6,"'[IBY Y="@50"

: PAYER

: @50

: S:",8,6,"'[IBY Y="@70"

: REMARKS

: S:IBY["8" Y="@99"

: @70

: S:",9,6,"'[IBY Y="@99"

: SYNONYM

: ALL

: @99

BUILD(S) (c): IB\*2.0\*52

: IB\*2.0\*137

: IB\*2.0\*191

: IB\*2.0\*184

: IB\*2.0\*232

: IB\*2.0\*291

: IB\*2.0\*320

: IB\*2.0\*348

: IB\*2.0\*349

: IB\*2.0\*371

: IB\*2.0\*399

: IB\*2.0\*400

: IB\*2.0\*432

: IB\*2.0\*516

: IB\*2.0\*547

: IB\*2.0\*592

1. The Insurance Company Editor screens will need to be modified in order to allow for the additional Dental Claim Payer ID Mailing Address. The screen changes below have been proposed for the menu changes.

**Insurance Company Editor**      Jul 25, 2017@15:20:33          Page:    1 of    9

Insurance Company Information for: AETNA

Type of Company: HEALTH INSURANCE                     Currently Active

                           Billing Parameters

  Signature Required?: NO                       Type Of Coverage: HEALTH INSURAN

           Reimburse?: WILL REIMBURSE              Billing Phone: 888 632-3862

    Mult. Bedsections:                        Verification Phone: 877.277.3368

One Opt. Visit: NO                     Precert Comp. Name:

Diff. Rev. Codes:                             Precert Phone: 877 277-3368

Amb. Sur. Rev. Code:

Rx Refill Rev. Code:

Filing Time Frame:  (27 MONTH(S))

BP  Billing/EDI Param     PA  Payer                 AI  (In)Activate Company

**AD  Billing Addresses**     RE  Remarks               CC  Change Insurance Co.

AC  Associate Companies   SY  Synonyms              DC  Delete Company

ID  Prov IDs/ID Param     EA  Edit All              VP  View Plans

EX  Exit ID

Select Action: Next Screen// **AD  Billing Addresses**

New Address Listman

Claims Office Addresses          Jul 25, 2017@15:35:50          Page:    1 of    1

Insurance Co: AETNA

Street:                                City/State:

Street 2:                                     Phone:

                                        Fax:

                    Prescription Claims Office Information

  Company Name: AETNA                            Street 3:

        Street:                                City/State:

      Street 2:                                     Phone:

                                                      Fax:

+         Enter ?? for more actions                                          >>>

MM  Main Mailing Address    AO  Appeals Office        **DE Dental Claims Office**

IC  Inpt Claims Office      PC  Prescr Claims Of      EX Exit

OC  Opt Claims Office       IO  Inquiry Office

Select Action: Next Screen//

In order to accomplish the task of getting the screens to appear like the mockup and store a new Dental address the following needs to be accomplished:

9. Know the current menu option order set up to check against when finished so ensure proper execution order and that nothing is missing.

Select OPTION NAME: INSURANCE CO

1 INSURANCE COMPANY EDI PARAMETE IBCN INSURANCE EDI REPORT Insurance

Company EDI Parameter Report

2 INSURANCE COMPANY ENTRY/EDIT DG INSURANCE COMPANY EDIT Insurance Company Entry/Edit

3 INSURANCE COMPANY ENTRY/EDIT IBCN INSURANCE CO EDIT Insurance Comp

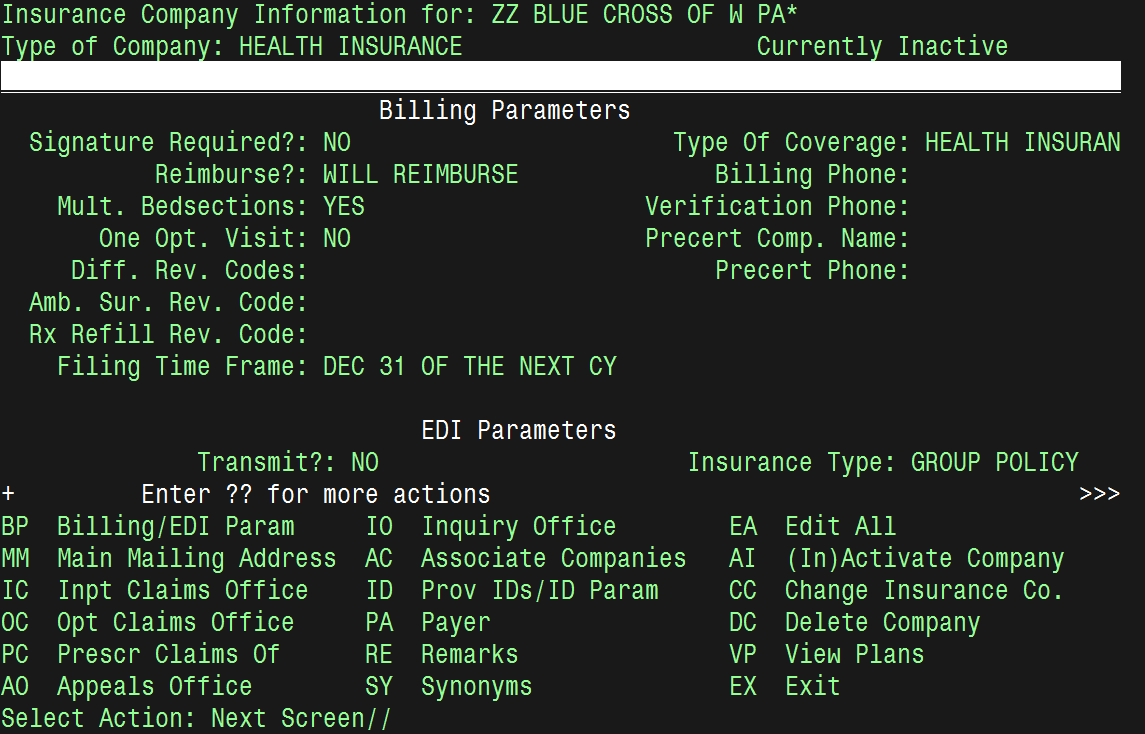
any Entry/Edit

CHOOSE 1-3: 3 IBCN INSURANCE CO EDIT Insurance Company Entry/Edit

Insurance Company Entry/Edit

Select INSURANCE COMPANY NAME: ZZ BLUE CROSS OF W PA\* PO BOX 1249

PITTSBURGH PENNSYLVANIA Y...................



10. Create a protocol action for the new action option DE- Dental Address

Using the List manager tool ^VALMWB

Template: IBCNS INSURANCE COMPANY

Demographics List Region

Template Name: IBCNS INSURANCE COMPANY Top Margin: 5

Entity Name: Insurance Company Bottom " : 16

Screen Title: Insurance Company Editor Right " : 90

Protocol Information Other Fields

Type of List: PROTOCOL OK to Transport?: OK

Protocol Menu: IBCNSC INSURANCE CO Use Cursor Control: YES

Print Protocol: Allowable Number of Actions:

Hidden Menu: VALM HIDDEN ACTIONS Date Range Limit:

Automatic Defaults:

+ Enter ?? for more actions >>>

DE Demographic Edit MC MUMPS Code Edit PE Protocol Edit

PI Protocol Information CE Caption Edit RN Run List

LR List Region Edit CL Change List Template IT Input Template

OF Other Fields EA Edit All RO Routine Editor

Select Tool:Next Screen// pe Protocol Edit

Select PROTOCOL NAME: zz new action item

11. Create a new protocol menu – option AD- Billing Addresses to contain the protocol actions of the DE- Dental Address option as well as the actions to be moved, MM, IC, OC, AO, PC, IO

Using the List manager tool ^VALMWB

**List Manager Workbench** Aug 15, 2017@20:00:52 Page: 1 of 3

Template: IBCNS INSURANCE COMPANY

Demographics List Region

Template Name: IBCNS INSURANCE COMPANY Top Margin: 5

Entity Name: Insurance Company Bottom " : 16

Screen Title: Insurance Company Editor Right " : 90

Protocol Information Other Fields

Type of List: PROTOCOL OK to Transport?: OK

Protocol Menu: IBCNSC INSURANCE CO Use Cursor Control: YES

Print Protocol: Allowable Number of Actions:

Hidden Menu: VALM HIDDEN ACTIONS Date Range Limit:

Automatic Defaults:

+ Enter ?? for more actions >>>

DE Demographic Edit MC MUMPS Code Edit PE Protocol Edit

PI Protocol Information CE Caption Edit RN Run List

LR List Region Edit CL Change List Template IT Input Template

OF Other Fields EA Edit All RO Routine Editor

Select Tool:Next Screen// PE Protocol Edit

Select PROTOCOL NAME: Set up new AD- Billing Addresses

TYPE: menu//

ENTRY ACTION: ?

ENTER STANDARD MUMPS CODE

ENTRY ACTION:

EXIT ACTION: I $G(IBFASTXT)=1 S VALMBCK="Q" Replace

Select ITEM: IBCNSC INS CO ASSOCIATION// ?

Answer with ITEM

Do you want the entire 35-Entry ITEM List? YES (Yes)

Choose from:

IBCNSC INS CO EDIT ALL

IBCNSC INS CO BILLING PARAMETERS

IBCNSC INS CO MAIN MAILING ADDRESS

IBCNSC INS CO APPEALS OFFICE

IBCNSC INS CO INQUIRY OFFICE

IBCNSC INS CO CHANGE COMPANY

IBCNSC INS CO REMARKS

IBCNSC INS CO SYNONYMS

IBCNSC INS CO (IN)ACTIVATE COMPANY

IBCNSC INS CO INPT CLAIMS

IBCNSC INS CO OPT CLAIMS

IBCNSC INS CO RX CLAIMS

IBCNS QUIT

IBCNSC INS CO DELETE COMPANY

IBCNSJ INS CO PLANS

IBCNSC PROVIDER ID PARAMETERS

IBCNSC INS CO PAYER

IBCNSC INS CO ASSOCIATION

You may enter a new ITEM, if you wish

Type <Enter> to continue or '^' to exit:

12. Modify the Template IBCNS INSURANCE COMPANY to add in the new menu item AD- Billing addresses by doing option PE- protocol file edit IBCNSC INSURANCE CO

Using the List manager tool ^VALMWB

**List Manager Workbench** Aug 15, 2017@20:00:52 Page: 1 of 3

Template: IBCNS INSURANCE COMPANY

Demographics List Region

Template Name: IBCNS INSURANCE COMPANY Top Margin: 5

Entity Name: Insurance Company Bottom " : 16

Screen Title: Insurance Company Editor Right " : 90

Protocol Information Other Fields

Type of List: PROTOCOL OK to Transport?: OK

Protocol Menu: IBCNSC INSURANCE CO Use Cursor Control: YES

Print Protocol: Allowable Number of Actions:

Hidden Menu: VALM HIDDEN ACTIONS Date Range Limit:

Automatic Defaults:

+ Enter ?? for more actions >>>

DE Demographic Edit MC MUMPS Code Edit PE Protocol Edit

PI Protocol Information CE Caption Edit RN Run List

LR List Region Edit CL Change List Template IT Input Template

OF Other Fields EA Edit All RO Routine Editor

Select Tool:Next Screen// PE Protocol Edit

Select PROTOCOL NAME: IBCNSC INSURANCE CO Insurance Company Edit

NAME: IBCNSC INSURANCE CO ITEM TEXT: Insurance Company Edit

TYPE: menu CREATOR: SHURMAN,JILLIAN A

PACKAGE: INTEGRATED BILLING COLUMN WIDTH: 26

MNEMONIC WIDTH: 4

ITEM: IBCNSC INS CO EDIT ALL MNEMONIC: EA

SEQUENCE: 26

ITEM: IBCNSC INS CO APPEALS OFFICE MNEMONIC: AO

SEQUENCE: 21

ITEM: IBCNSC INS CO INQUIRY OFFICE MNEMONIC: IO

SEQUENCE: 22

ITEM: IBCNSC INS CO MAIN MAILING ADDRESS

MNEMONIC: MM SEQUENCE: 12

ITEM: IBCNSC INS CO BILLING PARAMETERS MNEMONIC: BP

SEQUENCE: 11

ITEM: IBCNSC INS CO CHANGE COMPANY MNEMONIC: CC

SEQUENCE: 32

ITEM: IBCNSC INS CO REMARKS MNEMONIC: RE

SEQUENCE: 24

ITEM: IBCNSC INS CO SYNONYMS MNEMONIC: SY

Type <Enter> to continue or '^' to exit:

SEQUENCE: 25

ITEM: IBCNSC INS CO (IN)ACTIVATE COMPANY

MNEMONIC: AI SEQUENCE: 31

ITEM: IBCNSC INS CO INPT CLAIMS MNEMONIC: IC

SEQUENCE: 13

ITEM: IBCNSC INS CO OPT CLAIMS MNEMONIC: OC

SEQUENCE: 14

ITEM: IBCNSC INS CO RX CLAIMS MNEMONIC: PC

SEQUENCE: 15

ITEM: IBCNSC INS CO DELETE COMPANY MNEMONIC: DC

SEQUENCE: 33

ITEM: IBCNS QUIT MNEMONIC: EX

SEQUENCE: 35

ITEM: IBCNSJ INS CO PLANS MNEMONIC: VP

SEQUENCE: 34

ITEM: IBCNSC PROVIDER ID PARAMETERS MNEMONIC: ID

SEQUENCE: 23

ITEM: IBCNSC INS CO PAYER MNEMONIC: PA

SEQUENCE: 23

ITEM: IBCNSC INS CO ASSOCIATION MNEMONIC: AC

SEQUENCE: 22

Type <Enter> to continue or '^' to exit:

EXIT ACTION: I $G(IBFASTXT)=1 S VALMBCK="Q"

SCREEN: I 1 X:$D(^ORD(101,+$P(^ORD(101,DA(1),10,DA,0),"^"),24)) ^(24)

HEADER: D SHOW^VALM MENU PROMPT: Select Action:

TIMESTAMP: 62028,29972

NAME: IBCNSC INSURANCE CO//

Select PROTOCOL NAME: IBCNSC IN

1 IBCNSC INS CO (IN)ACTIVATE COMPANY (In)Activate Company AI

2 IBCNSC INS CO APPEALS OFFICE Appeals Office AA

3 IBCNSC INS CO ASSOCIATION Associate Companies

4 IBCNSC INS CO BILLING PARAMETERS Billing/EDI Param EP

5 IBCNSC INS CO CHANGE COMPANY Change Insurance Co. CC

Press <Enter> to see more, '^' to exit this list, OR

CHOOSE 1-5:

6 IBCNSC INS CO DELETE COMPANY Delete Company

7 IBCNSC INS CO EDIT ALL Edit All EA

8 IBCNSC INS CO INPT CLAIMS Inpt Claims Office IC

9 IBCNSC INS CO INQUIRY OFFICE Inquiry Office IA

10 IBCNSC INS CO MAIN MAILING ADDRESS Main Mailing Address EM

Press <Enter> to see more, '^' to exit this list, OR

CHOOSE 1-10:

11 IBCNSC INS CO OPT CLAIMS Opt Claims Office OC

12 IBCNSC INS CO PAYER Payer PA

13 IBCNSC INS CO REMARKS Remarks RE

14 IBCNSC INS CO RX CLAIMS Prescr Claims Of PC

15 IBCNSC INS CO SYNONYMS Synonyms ES

Press <Enter> to see more, '^' to exit this list, OR

CHOOSE 1-15:

16 IBCNSC INS CO TELEPHONE Telephone Numbers ET

17 IBCNSC INSURANCE CO Insurance Company Edit

CHOOSE 1-17: 17 IBCNSC INSURANCE CO Insurance Company Edit

NAME: IBCNSC INSURANCE CO ITEM TEXT: Insurance Company Edit

TYPE: menu CREATOR: SHURMAN,JILLIAN A

PACKAGE: INTEGRATED BILLING COLUMN WIDTH: 26

MNEMONIC WIDTH: 4

ITEM: IBCNSC INS CO EDIT ALL MNEMONIC: EA

SEQUENCE: 26

ITEM: IBCNSC INS CO APPEALS OFFICE MNEMONIC: AO

SEQUENCE: 21

ITEM: IBCNSC INS CO INQUIRY OFFICE MNEMONIC: IO

SEQUENCE: 22

ITEM: IBCNSC INS CO MAIN MAILING ADDRESS

MNEMONIC: MM SEQUENCE: 12

ITEM: IBCNSC INS CO BILLING PARAMETERS MNEMONIC: BP

SEQUENCE: 11

ITEM: IBCNSC INS CO CHANGE COMPANY MNEMONIC: CC

SEQUENCE: 32

ITEM: IBCNSC INS CO REMARKS MNEMONIC: RE

SEQUENCE: 24

ITEM: IBCNSC INS CO SYNONYMS MNEMONIC: SY

Type <Enter> to continue or '^' to exit: ^

NAME: IBCNSC INSURANCE CO//

PACKAGE: INTEGRATED BILLING//

ITEM TEXT: Insurance Company Edit Replace

TYPE: menu//

ENTRY ACTION: ?

ENTER STANDARD MUMPS CODE

ENTRY ACTION:

EXIT ACTION: I $G(IBFASTXT)=1 S VALMBCK="Q" Replace

Select ITEM: IBCNSC INS CO ASSOCIATION// ?

Answer with ITEM

Do you want the entire 35-Entry ITEM List? YES (Yes)

Choose from:

IBCNSC INS CO EDIT ALL

IBCNSC INS CO BILLING PARAMETERS

IBCNSC INS CO MAIN MAILING ADDRESS

IBCNSC INS CO APPEALS OFFICE

IBCNSC INS CO INQUIRY OFFICE

IBCNSC INS CO CHANGE COMPANY

IBCNSC INS CO REMARKS

IBCNSC INS CO SYNONYMS

IBCNSC INS CO (IN)ACTIVATE COMPANY

IBCNSC INS CO INPT CLAIMS

IBCNSC INS CO OPT CLAIMS

IBCNSC INS CO RX CLAIMS

IBCNS QUIT

IBCNSC INS CO DELETE COMPANY

IBCNSJ INS CO PLANS

IBCNSC PROVIDER ID PARAMETERS

IBCNSC INS CO PAYER

IBCNSC INS CO ASSOCIATION

You may enter a new ITEM, if you wish

Type <Enter> to continue or '^' to exit:

13. Create the code behind the DE- Dental Address for the storage of the address to be like the OC- opt Claims office functionality. Can copy code from routine CLAIMS2^IBCNSC0 for Dental display but the dental address data elements are to be stored in the Insurance file #36, .19 section. New fields have been created for Dental address in .191 through .1911

* 1. Select Action: Next Screen// OC Opt Claims Office

Are Outpatient Claims Processed by Another Insurance Co.?: YES

// NO NO

CLAIMS (OPT) STREET ADDRESS 1: DENTAL ADD1

CLAIMS (OPT) STREET ADDRESS 2: DENTAL ADD2

CLAIMS (OPT) STREET ADDRESS 3:

CLAIMS (OPT) PROCESS CITY: DCITY

CLAIMS (OPT) PROCESS STATE: DELAWARE

CLAIMS (OPT) PROCESS ZIP: 19709??

Answer must be nine (999999999) or ten characters (99999-9999) in length.

The last 4 cannot be '0000' or '9999'.

CLAIMS (OPT) PROCESS ZIP: 197095552

CLAIMS (OPT) PHONE NUMBER: 123456789

CLAIMS (OPT) FAX: 999-3100.......................

* 1. CLAIMS2 ; display Outpatient Claims information  
      ;  
      N OFFSET,START,IBCNS16,IBADD  
      ;WCJ;IB\*2.0\*547  
      ;S START=34,OFFSET=2  
      S START=35+(2\*$G(IBACMAX)),OFFSET=2  
      D SET^IBCNSP(START,OFFSET+20," Outpatient Claims Office Information ",IORVON,IORVOFF)  
      ;  
      ;WCJ;IB\*2.0\*547;Call New API  
      ;S IBCNS16=$$ADDRESS(IBCNS,.16,6)  
      S IBCNS16=$$ADD2(IBCNS,.16,6)  
      ;  
      D SET^IBCNSP(START+1,OFFSET," Company Name: "\_$P($G(^DIC(36,+$P(IBCNS16,"^",7),0)),"^",1))  
      D SET^IBCNSP(START+2,OFFSET," Street: "\_$P(IBCNS16,"^",1))  
      D SET^IBCNSP(START+3,OFFSET," Street 2: "\_$P(IBCNS16,"^",2))  
      N OFFSET S OFFSET=45  
      D SET^IBCNSP(START+1,OFFSET," Street 3: "\_$P(IBCNS16,"^",3)) S IBADD=1  
      D SET^IBCNSP(START+1+IBADD,OFFSET," City/State: "\_$E($P(IBCNS16,"^",4),1,15)\_$S($P(IBCNS16,"^",4)="":"",1:", ")\_$P($G(^DIC(5,+$P(IBCNS16,"^",5),0)),"^",2)\_" "\_$E($P(IBCNS16,"^",6),1,5))  
      D SET^IBCNSP(START+2+IBADD,OFFSET," Phone: "\_$P(IBCNS16,"^",8))  
      D SET^IBCNSP(START+3+IBADD,OFFSET," Fax: "\_$P(IBCNS16,"^",9))  
      Q  
      ;

14. Modify the input Template for the insurance file #36 to include the DE Dental items like the OC opt claims:

Select OPTION: ?

Answer with OPTION NUMBER, or NAME

Choose from:

1 ENTER OR EDIT FILE ENTRIES

2 PRINT FILE ENTRIES

3 SEARCH FILE ENTRIES

4 MODIFY FILE ATTRIBUTES

5 INQUIRE TO FILE ENTRIES

6 UTILITY FUNCTIONS

7 OTHER OPTIONS

8 DATA DICTIONARY UTILITIES

9 TRANSFER ENTRIES

Select OPTION: 6 UTILITY FUNCTIONS

Select UTILITY OPTION: ?

Answer with UTILITY OPTION NUMBER, or NAME

Do you want the entire 11-Entry UTILITY OPTION List? yes (Yes)

Choose from:

1 VERIFY FIELDS

2 CROSS-REFERENCE A FIELD OR FILE

3 IDENTIFIER

4 RE-INDEX FILE

5 INPUT TRANSFORM (SYNTAX)

6 EDIT FILE

7 OUTPUT TRANSFORM

8 TEMPLATE EDIT

9 UNEDITABLE DATA

10 MANDATORY/REQUIRED FIELD CHECK

11 KEY DEFINITION

Select UTILITY OPTION: 8 TEMPLATE EDIT

Modify what File: 36 INSURANCE COMPANY (1558 entries)

Select TEMPLATE File: IBEDIT ??

Select TEMPLATE File: > ??

Select TEMPLATE File: ?

Answer with TEMPLATE FILE NUMBER, or NAME

Choose from:

.4 PRINT TEMPLATE

.401 SORT TEMPLATE

.402 INPUT TEMPLATE

Select TEMPLATE File: .402 INPUT TEMPLATE

Select INPUT TEMPLATE: ?

Answer with INPUT TEMPLATE

Do you want the entire INPUT TEMPLATE List? YES (Yes)

Choose from:

AJKEDIT COLLECTION AMOUNT AJKEDIT COLLECTION AMOUNT

(SEP 20, 2001@10:50) User #452 File #36

IBEDIT INS CO1 IBEDIT INS CO1

(AUG 10, 2017@16:43) User #520824637 File #36

TEST BILL TEST BILL (MAR 06, 2017@09:48) User #520824635 File #36

Select INPUT TEMPLATE: IBEDIT INS CO1 IBEDIT INS CO1

(AUG 10, 2017@16:43) User #520824637 File #36

Do you want to use the screen-mode version? YES//

TEMPLATE NAME: **IBEDIT INS CO1**

DATE LAST MODIFIED: **AUG 10,2017@16:43**

DATE LAST USED: **AUG 15,2017**

READ ACCESS:

WRITE ACCESS:

USER #: **520824637**

DESCRIPTION...  CANONIC FOR FILE 36:

(Edit Fields on Next Page...)

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

Exit Save Next Page Refresh Quit

Enter a COMMAND, or "^" followed by the CAPTION of a FIELD to jump to.

COMMAND: **N**  **Press <PF1>H for help** **Insert**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

**Editing Input Template "IBEDIT INS CO1"**

============[INSERT ]===============< (File 36) >====[Press <PF1>H for help]====

@30

S:",10,6,"'[IBY Y="@80"

ANOTHER CO. PROCESS OP CLAIMS?;T

S:'X Y="@31"

CLAIMS (OPT) COMPANY NAME

S Y="@36"

@31

CLAIMS (OPT) STREET ADDRESS 1

S:X="" Y="@5"

CLAIMS (OPT) STREET ADDRESS 2

S:X="" Y="@5"

CLAIMS (OPT) STREET ADDRESS 3

@5

CLAIMS (OPT) PROCESS CITY

CLAIMS (OPT) PROCESS STATE

CLAIMS (OPT) PROCESS ZIP

CLAIMS (OPT) PHONE NUMBER

CLAIMS (OPT) FAX

@36

<=======T=======T=======T=======T=======T=======T=======T=======T=======T=======

15. Modify routine ^IBCNSC to add in the new Dental option code and reflect the correct menu sequence order.

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCNSC | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** |  | | | | | | | | |
| **Related Options** | None | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  |  | | | | |  | | | |
| **Data Dictionary (DD) References** |  | | | | | | | | |
| **Related Protocols** | None | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Current Logic** | | | | | | | | | |
| IBCNSC ;ALB/NLR - INSURANCE COMPANY EDIT ;6/1/05 9:42am  ;;2.0;INTEGRATED BILLING;\*\*46,137,184,276,320,371,400,488,547\*\*;21-MAR-94;Build 119  ;;Per VA Directive 6402, this routine should not be modified.  ;  ;also used for IA #4694  ; EN ; -- main entry point for IBCNS INSURANCE COMPANY, IBCNS VIEW INS CO  NEW IB1ST  K IBFASTXT,VALMQUIT,VALMEVL,XQORS,^TMP("XQORS",$J),IBCNS  S IBCHANGE="OKAY"  I '$G(IBVIEW) D EN^VALM("IBCNS INSURANCE COMPANY") G ENQ  D EN^VALM("IBCNS VIEW INS CO") ENQ Q  ; HDR ; -- header code  S VALMHDR(1)="Insurance Company Information for: "\_$E($P(^DIC(36,IBCNS,0),"^"),1,30)  S VALMHDR(2)="Type of Company: "\_$E($P($G(^IBE(355.2,+$P($G(^DIC(36,+IBCNS,0)),"^",13),0)),"^"),1,20)\_" Currently "\_$S(+($P($G(^DIC(36,+IBCNS,0)),"^",5)):"Inactive",1:"Active")  Q  ; INIT ; -- init variables and list array  K VALMQUIT  S VALMCNT=0,VALMBG=1  I '$D(IBCNS) D INSCO Q:$D(VALMQUIT)  D BLD,HDR  Q BLD ; -- list builder  ;WCJ;IB\*2.0\*547  ;NEW BLNKI  NEW BLNKI,IBACMAX ; new variable set in PARAM section and needed throughout for display  ;  K ^TMP("IBCNSC",$J)  D KILL^VALM10() ; delete all video attributes  F BLNKI=1:1:62 D BLANK(.BLNKI) ; 62 blank lines to start with  D PARAM^IBCNSC01 ; billing parameters  D MAIN^IBCNSC01 ; main mailing address  D CLAIMS1^IBCNSC0 ; inpatient claims office  D CLAIMS2^IBCNSC0 ; outpatient claims office  D PRESCR^IBCNSC1 ; prescription claims office  D APPEALS             ; appeals office  D INQUIRY             ; inquiry office  D DISP^IBCNSC02 ; parent/child associations (ESG 11/3/05)  D PROVID^IBCNSC1 ; provider IDs  D PAYER^IBCNSC01 ; payer/payer apps (ESG 7/29/02 IIV project)  D REMARKS^IBCNSC01 ; remarks  D SYN^IBCNSC01 ; synonyms  S VALMCNT=+$O(^TMP("IBCNSC",$J,""),-1)  Q  ; APPEALS ;  N OFFSET,START,IBCNS14,IBADD  ;  ;WCJ;IB\*2.0\*547;Call new API  ;S IBCNS14=$$ADDRESS^IBCNSC0(IBCNS,.14,7)  S IBCNS14=$$ADD2^IBCNSC0(IBCNS,.14,7)  ;  ;WCJ;IB\*2.0\*547  ;S START=48,OFFSET=2  S START=49+(2\*$G(IBACMAX)),OFFSET=2  D SET^IBCNSP(START,OFFSET+25," Appeals Office Information ",IORVON,IORVOFF)  D SET^IBCNSP(START+1,OFFSET," Company Name: "\_$P($G(^DIC(36,+$P(IBCNS14,"^",7),0)),"^",1))  D SET^IBCNSP(START+2,OFFSET," Street: "\_$P(IBCNS14,"^",1))  D SET^IBCNSP(START+3,OFFSET," Street 2: "\_$P(IBCNS14,"^",2))  N OFFSET S OFFSET=45  D SET^IBCNSP(START+1,OFFSET," Street 3: "\_$P(IBCNS14,"^",3)) S IBADD=1  D SET^IBCNSP(START+1+IBADD,OFFSET," City/State: "\_$E($P(IBCNS14,"^",4),1,15)\_$S($P(IBCNS14,"^",4)="":"",1:", ")\_$P($G(^DIC(5,+$P(IBCNS14,"^",5),0)),"^",2)\_" "\_$E($P(IBCNS14,"^",6),1,5))  D SET^IBCNSP(START+2+IBADD,OFFSET," Phone: "\_$P(IBCNS14,"^",8))  D SET^IBCNSP(START+3+IBADD,OFFSET," Fax: "\_$P(IBCNS14,"^",9))  Q  ; INQUIRY ;  ;  N OFFSET,START,IBCNS15,IBADD  ;  ;WCJ;IB\*2.0\*547;Call new API  ;S IBCNS15=$$ADDRESS^IBCNSC0(IBCNS,.15,8)  S IBCNS15=$$ADD2^IBCNSC0(IBCNS,.15,8)  ;  ;WCJ;IB\*2.0\*547  ;S START=55,OFFSET=2  S START=56+(2\*$G(IBACMAX)),OFFSET=2  D SET^IBCNSP(START,OFFSET+25," Inquiry Office Information ",IORVON,IORVOFF)  D SET^IBCNSP(START+1,OFFSET," Company Name: "\_$P($G(^DIC(36,+$P(IBCNS15,"^",7),0)),"^",1))  D SET^IBCNSP(START+2,OFFSET," Street: "\_$P(IBCNS15,"^"))  D SET^IBCNSP(START+3,OFFSET," Street 2: "\_$P(IBCNS15,"^",2))  N OFFSET S OFFSET=45  D SET^IBCNSP(START+1,OFFSET," Street 3: "\_$P(IBCNS15,"^",3)) S IBADD=1  D SET^IBCNSP(START+1+IBADD,OFFSET," City/State: "\_$E($P(IBCNS15,"^",4),1,15)\_$S($P(IBCNS15,"^",4)="":"",1:", ")\_$P($G(^DIC(5,+$P(IBCNS15,"^",5),0)),"^",2)\_" "\_$E($P(IBCNS15,"^",6),1,5))  D SET^IBCNSP(START+2+IBADD,OFFSET," Phone: "\_$P(IBCNS15,"^",8))  D SET^IBCNSP(START+3+IBADD,OFFSET," Fax: "\_$P(IBCNS15,"^",9))  Q  ; HELP ; -- help code  S X="?" D DISP^XQORM1 W !!  Q  ; EXIT ; -- exit code  K VALMQUIT,IBCNS,IBCHANGE,IBFASTXT  D CLEAN^VALM10  Q  ; INSCO ; -- select insurance company  NEW DLAYGO,DIC,X,Y,DTOUT,DUOUT,IBCNS3  I '$D(IBCNS) D  G:$D(VALMQUIT) INSCOQ  .S DIC="^DIC(36,",DIC(0)="AEQMZ",DIC("S")="I '$G(^(5))"  .I '$G(IBVIEW) S DLAYGO=36,DIC(0)=DIC(0)\_"L"  .D ^DIC K DIC  .S IBCNS=+Y  .;/Beginning of IB\*2.0\*488 (vd)  .I +IBCNS I $P($G(^DIC(36,+IBCNS,3)),"^",1)="" D     ; Set default for EDI=Transmit? to YES-LIVE  ..S DR="3.01////1",DIE="^DIC(36,",DA=IBCNS D ^DIE K DIE  ..;/End of IB\*2.0\*488 (vd)  I $G(IBCNS)<1 K IBCNS S VALMQUIT="" G INSCOQ INSCOQ ;  K DIC  Q  ; BLANK(LINE) ; -- Build blank line  D SET^VALM10(.LINE,$J("",80))  Q  ; EDIKEY() ; input transform code to determine if user is allowed to edit  ; certain fields in the insurance company file  NEW OK S OK=0  I $$KCHK^XUSRB("IB EDI INSURANCE EDIT") S OK=1 G EDIKEYX  D EN^DDIOL("You must hold the IB EDI INSURANCE EDIT security key to edit this field.",,"!!")  D EN^DDIOL("",,"!!?5") EDIKEYX ;  Q OK  ; DUPQUAL(IBCNS,QUAL,FIELD) ; input transform to make sure that the same qualifier is not used twice for  ; payer secondary IDs. There are two sets of fields in file 36 that can not be duplicated.  ; 6.01 EDI INST SECONDARY ID QUAL(1) can not be the same as 6.03 EDI INST SECONDARY ID QUAL(2)  ; 6.05 EDI PROF SECONDARY ID QUAL(1) can not be the same as 6.07 EDI PROF SECONDARY ID QUAL(2)  ;   ; Input:  ; IBCNS is the insurance company internal number  ; QUAL is the internal code of the value being input.  ; FIELD is the field it is being compare with.  ;  ; Returns:  ; TRUE/1 if they are the same (duplicate)  ; FALSE/0 if they are not  ;  Q:$G(QUAL)="" 0 ; should not happen because this is invoked as an input transform  Q:'+$G(IBCNS) 1 ; stop from editing through fileman  N DUP  S DUP=$$GET1^DIQ(36,+$G(IBCNS)\_",",+$G(FIELD),"I")  D CLEAN^DILF  Q QUAL=DUP  ;  ;WCJ;IB\*2.0\*547 ALLOWED(IBAC) ; input transform to make sure that Administrative Contractor is set up in the site parameters.  ; it will be set up for either commercial or medicare. Since the type is defined my the plan and we are at a higher  ; level in the Insurance Company, we have to allow both.  ; called from ^DD(36.015,.01,0) and ^DD(36.016,.01,0)  ;  ;3/17/2016 - A decision was made to limit which type is allowed by using the TYPE OF COVERAGE field. (TAZ)  ;  ;  ; Input:  ; IBAC is the internal code of the value being input.  ;  ; Returns:  ; TRUE/1 if allowed (set up in site parameters)  ; FALSE/0 if they are not  ;   Q:$D(^IBE(350.9,1,$S($$GET1^DIQ(36,IBCNS\_",","TYPE OF COVERAGE")="MEDICARE":81,1:82),"B",IBAC)) 1  Q 0  ;  ; WCJ;IB\*2.0\*547  ; This is to clean up any extraneous nodes if a user entered an alternate ID type, but not an actual ID. CLEANIDS(INSIEN) ;  ; INSIEN=Insurance Company IEN  ;  N NODE,LOOP,DATA,CLEANUP  F NODE=15,16 D  .S LOOP=0 F  S LOOP=$O(^DIC(36,INSIEN,NODE,LOOP)) Q:'+LOOP  S DATA=$G(^(LOOP,0)) I DATA]"",$P(DATA,U,2)="" D  ..N DIK,DA  ..S DA=LOOP,DA(1)=INSIEN  ..S DIK="^DIC(36,"\_INSIEN\_","\_NODE\_","  ..D ^DIK  ..S CLEANUP=1  I $G(CLEANUP) D  . N DIR  . S DIR("A",1)="Payer ID Types without corresponding ID# were deleted."  . S DIR(0)="EA",DIR("A")="PRESS ENTER TO CONTINUE "  . D ^DIR  .Q  Q | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
|  | | | | | | | | | |