**TAS eBilling SDD US2489**

System Design Document

IB\*2.0\*592



Department of Veterans Affairs

**July 2017**

Version 2.00

**User Story Number:** NO ID

**User Story Name:** Fix <UNDEFINED> errors

**Product Backlog ID:** n/a

**Rally ID:** US-2489

# Design/Resolution

This SDD is to fix two <UNDEFINED> errors intermittently occurring when the user tries to 1) “Print MRA” while in the MRA Management WorkList, which is error **<UNDEFINED>PMRA+3^IBCECOB2** and 2) “CANCEL/CLONE” a bill, which is error **<UNDEFINED>STEP1+5^IBCCC**.

This fix requires modifying 4 routines, namely IBCC, IBCCC, IBCCCB and IBCECOB2 as summarized and detailed below:

**Error #1 <UNDEFINED>PMRA+3^IBCECOB2:** This simply requires the use of $G to wrap an array element being referenced in PMRA^IBCECOB2 that was $G’d at a previous command in the same line of code.

**Error #2**  **<UNDEFINED>STEP1+5^IBCCC:** Is directly caused by the killing of a critical variable IBCCCC which is subsequently referenced. It is being subsequently referenced because the logic is caught in an endless loop due to the fact that the quit flag IBQUIT is not being flagged as ‘1’ after issuing a message directing the user to another activity. To resolve this error, the code at STEP1^IBCCC shall be modified to wrap variable IBCCCC with $G to prevent any future <UNDEFINED> errors at this place in the code. Wrapping IBCCC with $G, is really a failsafe since IBCCC should no longer be undefined at this point due to the changes to END1^IBCCC, which shall be modified so that the killing of IBCCC occurs after the loopback to STEP1 instead of prior to the loopback. The root cause of this error, however, shall be addressed by modifying routines IBCC and IBCCCB to properly set the IBQUIT flag after issuing the message directing the user to another activity.

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCC | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** |  | | | | | | | | |
| **Related Options** | None | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  |  | | | | |  | | | |
| **Data Dictionary (DD) References** |  | | | | | | | | |
| **Related Protocols** |  | | | | | | | | |
| **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** | | | | | | | | | |
| IBCC^INT^1^^0  IBCC ;ALB/MJB - CANCEL THIRD PARTY BILL ;14 JUN 88 10:12  ;;2.0;INTEGRATED BILLING;\*\*2,19,77,80,51,142,137,161,199,241,155,276,320,358,433,432,447,516,547\*\*;21-MAR-94;Build 119  ;;Per VA Directive 6402, this routine should not be modified.  ;  ;MAP TO DGCRC  ;  I '$D(IBCAN) S IBCAN=1  ASK ;  ;\*\*\*  ;I $D(XRT0) S:'$D(XRTN) XRTN="IBCC" D T1^%ZOSV ;stop rt clock  ;S XRTL=$ZU(0),XRTN="IBCC-1" D T0^%ZOSV ;start rt clock  ;  ; If called at entry point PROCESS, variable IBNOASK will exist.  ; First time through, IBNOASK=1  ; Second time through, IBNOASK=2 and it will quit  I $G(IBNOASK)=2 G Q  I $G(IBNOASK)=1 S IBNOASK=2  ;  G Q:$G(IBCE("EDI"))  D Q  S IBQUIT=0  N DPTNOFZY S DPTNOFZY=1 ;Suppress PATIENT file fuzzy lookups  I '$G(IBNOASK) S DIC="^DGCR(399,",DIC(0)="AEMQZ",DIC("A")="Enter BILL NUMBER or Patient NAME: " W !! D ^DIC I Y<1 S IBQUIT=1 G Q1  K IB364  NOPTF ; Note if IB364 is >0 it will be used as the ien to update in file 364  N DA,I  I '$G(IBNOASK) S IBIFN=+$G(Y)  I '$G(IBIFN) G ASK  I IBCAN>1 D NOPTF^IBCB2 I 'IBAC1 D NOPTF1^IBCB2 G ASK  ;  I $G(IBCNCRD)=1,$P($P($G(^DGCR(399,IBIFN,0)),U),"-",2)>98 D Q  .W !!,"Please note that you have exceeded the maximum number of iterations (99) for this claim."  .W "Copy and cancel (CLON) must be used to correct this bill."  .S IBQUIT=1 H 3  ; Check if bill has been referred to Counsel  I $P($G(^PRCA(430,IBIFN,6)),U,4) D G ASK  . W !,"This bill has been referred to Regional Counsel and cannot be 'CANCELLED' in"  . W !,"Integrated Billing. Please use the option 'TP Referred Follow-up'"  . W !,"[PRCA RC ACTION MENU] in Accounts Receivable to request that Regional"  . W !,"Counsel return the bill to your facility."  . Q  ;  F I=0,"S","U1" S IB(I)=$G(^DGCR(399,IBIFN,I))  S IBSTAT=$P(IB(0),U,13)  ; REMOVE New messages for CRD option IB\*2.0\*433 in IB\*2.0\*447 IA#5630  ;I $G(IBCNCRD)=1,IBSTAT'=2,'$$ACCK^PRCAACC(IBIFN) D Q  ;.W !!,"This option cannot be used to correct some Billing Rate Types (Example: TRICARE)"  ;.W "Copy and cancel (CLON) must be used to correct this bill."  ;.S IBQUIT=1 H 3  ;  ; Restrict access to this process for REQUEST MRA bills  I IBSTAT=2,'$G(IBCE("EDI")),$$MRAWL^IBCEMU2(IBIFN) D G ASK  . W !!?4,"This bill is in a status of REQUEST MRA and it does appear on the"  . W !?4,"MRA Management Work List. Please use the 'MRA Management Menu' options"  . W !?4,"for all processing related to this bill."  . Q  ;  ; IB\*2.0\*432 Restrict access to claims on the new CBW Worklist  I $P($G(^DGCR(399,IBIFN,"S1")),U,7)=1,$G(IBMRANOT)'=1 D G ASK  . W !!?4,"This bill appears on the CBW Management Work List. Please use the"  . W !?4,"'CBW Management Menu' options for all processing related to this bill."  . Q  ;  ; Check if this is a paper claim. If not, check for split EOB. If split, don't allow CRD unless more than 1 EOB has been returned  I $G(IBCNCRD)=1,$P($G(^DGCR(399,IBIFN,"TX")),U,8)'=1,$$SPLTMRA^IBCEMU1(IBIFN)=1 D Q  .W !!,"There is a split EOB associated with this claim. You cannot use this option to Correct this claim until the second EOB has been received."  .S IBQUIT=1 H 3  .Q  ;  ; Warning message if in a REQUEST MRA status with no MRA on file  ; IB\*2.0\*516/TAZ,MRD - Forbid the user from using the option CRD  ; (Correct Rejected/Denied Bill) on an MRA claim if the status is  ; REQUEST MRA (IBSTAT=2).  I IBSTAT=2,'$$MRACNT^IBCEMU1(IBIFN) D I $G(IBQUIT) H 3 Q  . N REJ  . D TXSTS^IBCEMU2(IBIFN,,.REJ)  . ;IB\*2.0\*516/TAZ - If CRD is from CSA allow a REJected claim to be CRD'ed without displaying a warning.  . I $G(IBCNCSA),REJ Q  . W \*7,!!?4,$S('$G(IBCNCRD):"Warning! ",1:""),"This bill is in a status of REQUEST MRA."  . W !?4,"No MRAs have been received"  . I REJ W ", but the most recent transmission of this",!?4,"MRA request bill was rejected."  . I 'REJ W " and there are no rejection messages on file",!?4,"for the most recent transmission of this MRA request bill."  . I $G(IBCNCRD) S IBQUIT=1  . Q  ;  I IBCAN=2,IB("S")]"",+$P(IB("S"),U,16),$P(IB("S"),U,17)]"" D G 1  . W !!,"This bill was cancelled on " S Y=$P(IB("S"),U,17) X ^DD("DD") W Y," by ",$S($P(IB("S"),U,18)']"":IBU,$D(^VA(200,$P(IB("S"),U,18),0)):$P(^(0),U,1),1:IBU),"."  . S IBQUIT=1  ;  ; IB\*2.0\*516/TAZ,MRD - Forbid the user from using the option CRD  ; (Correct Rejected/Denied Bill) on all but primary claims.  I $G(IBCNCRD),($$COB^IBCEF(IBIFN)'="P") D Q  . W !!,"Please note that COB data may exist for this bill."  . W !,"Copy and cancel (CLON) must be used to correct this bill."  . S IBQUIT=1  . H 3  . Q  ;  ; Notify if a payment has been posted to this bill before cancel  N PRCABILL  S PRCABILL=$$TPR^PRCAFN(IBIFN)  I PRCABILL=-1 W !!,"Please note: PRCA was unable to determine if a payment has been posted." I $G(IBCNCRD)=1 W !,"Copy and cancel (CLON) must be used to correct this bill." S IBQUIT=1 H 3 Q  I PRCABILL>0 W !!,"Please note a PAYMENT of \*\*$"\_$$TPR^PRCAFN(IBIFN)\_"\*\* has been POSTED to this bill."  ; New message for CRD option  I $G(IBCNCRD)=1,PRCABILL>0 W !,"Copy and cancel (CLON) must be used to correct this bill." S IBQUIT=1 H 3 Q  ;  ; If bill was created via Electronic claims process then notify  ; user that cancellation should occur using ECME package  I $$GET1^DIQ(399,IBIFN\_",",460)]"" D G:'Y ASK  . W !!!?5,"This bill was created by the"  . W !?5,"Electronic Claims Management Engine (ECME)."  . W !?5,"Cancellation needs to occur in the ECME package by"  . W !?5,"submitting a REVERSAL to the Payer.",!!  . K DIR S DIR("A",1)="Has a REVERSAL for this e-Claim already been",DIR("A")="submitted to the payer via the ECME package (Y/N)",DIR(0)="Y",DIR("B")="NO" D ^DIR  . I Y=0 W !!,"<PLEASE SUBMIT A REVERSAL USING THE APPROPRIATE OPTION IN THE ECME PACKAGE>",$C(7)  ;  CHK ;  ; if user came from CLON, make sure they know about the new CRD option IB\*2.0\*447 remove TRICARE msg.  I $G(IBCNCOPY)=1 D  .W !!,\*7,"Warning: This option should NOT be used to correct Rejected/Denied claims."  .W !," It should ONLY be used to correct DENIED claims which have payments"  .W !," posted against them.\*\*\*" ; and claims with certain Billing Rate Types (Example: TRICARE)."  ;  S (IBCCCC,IBQUIT)=0 I '$G(IBCEAUTO),'$G(IBMCSCAN) W !!,"ARE YOU SURE YOU WANT TO CANCEL THIS BILL" S %=2 D YN^DICN G:%=0 HELP I %'=1 S IBQUIT=1 G NO  ;  I '$G(IBCEAUTO) W !!,"LAST CHANCE TO CHANGE YOUR MIND..."  S DIE=399,DA=IBIFN,DIE("NO^")=""  S DR="16;S:'X Y=0;19;S IBCCCC=1;"  I $G(IBCEAUTO) S DR="16////1;19////EDI/MRA TURNED OFF;S IBCCCC=1;"  ;  ; esg - 8/23/06 - IB\*2\*358 - fix semi-colon in free text field  I $G(IBMCSRSC)'="" S DR="16;S:'X Y=0;19//^S X=IBMCSRSC;S IBCCCC=1;"  D ^DIE K DIE,DR  ;  NO I 'IBCCCC W !!,"<NO ACTION TAKEN>",\*7 S IBQUIT=1 G ASK:IBCAN<2,Q  S IBCCR=$P($G(^DGCR(399,IBIFN,"S")),U,19)  ; update claim # with new iteration IB\*2.0\*447 move to later in the process  ;D:$G(IBCNCRD)=1 CRD  W !!,"...Bill has been cancelled..." D BULL^IBCBULL,BSTAT^IBCDC(IBIFN),PRIOR^IBCCC2(IBIFN)  ;  ; cancelling in ingenix claimsmanager if ingenix is running  ; clean-up of variables is OK if not coming in from ListMan screen  I $$CM^IBCIUT1(IBIFN) S IBCISNT=4 D ST2^IBCIST I '$G(IBCICNCL) K IBCISNT,IBCISTAT,IBCIREDT,IBCIERR  ;  S IBEDI=$G(IB364)  I 'IBEDI S IBEDI=+$$LAST364^IBCEF4(IBIFN)  ; ib\*2.0\*547 don't cancel MRA if cloning a bill that is secondary to MRA (share the same claim#)  I IBEDI D UPDEDI^IBCEM(IBEDI,"C",,$S($$MRASEC^IBCEF4(IBIFN):2,1:"")) ;Update EDI files, if needed  ;  F I="S","U1" S IB(I)=$S($D(^DGCR(399,IBIFN,I)):^(I),1:"")  S PRCASV("ARREC")=IBIFN,PRCASV("AMT")=$S(IB("U1")']"":0,1:$P(IB("U1"),"^")),PRCASV("DATE")=$P(IB("S"),"^",17),PRCASV("BY")=$P(IB("S"),"^",18)  S PRCASV("COMMENT")=$S($P(IB("S"),U,19)]"":$P(IB("S"),U,19),$P(^IBE(350.9,1,2),"^",7)]"":$P(^(2),"^",7),1:"BILL CANCELLED IN MAS")  S PRCASV("BY")=$S($P(IB("S"),U,18)]"":$P(IB("S"),U,18),1:"")  ; IA#3374/IB\*2.0\*433 Pass the CRD flag so FMS knows to send a cancel record before the new E record is sent  ;S X=$$CANCEL^RCBEIB($G(PRCASV("ARREC")),$G(PRCASV("DATE")),$G(PRCASV("BY")),$G(PRCASV("AMT")),$G(PRCASV("COMMENT")))  S PRCASV("ARCRD")=$G(IBCNCRD)  S X=$$CANCEL^RCBEIB($G(PRCASV("ARREC")),$G(PRCASV("DATE")),$G(PRCASV("BY")),$G(PRCASV("AMT")),$G(PRCASV("COMMENT")),$G(PRCASV("ARCRD")))  W !,$S(X:">> The receivable associated with the claim was cancelled.",1:">> The receivable associated with the claim was not cancelled.")  I $P(X,U,2)]"" W !,">>> ",$P(X,U,2) ; The reason why the claim can not be cancelled.  I IBCAN<2 D RNB^IBCC1 ;assign a reason not billable  G ASK:IBCAN<2,Q  ;  HELP W !,?3,"Answer 'YES' or 'Y' if you wish to cancel this bill.",!,?3,"Answer 'NO' or 'N' if you want to abort." G CHK  Q  1 I $P(IB(0),U,13)=1 W !,"This record was re-opened on " S Y=$P(IB(0),U,14) X ^DD("DD") W Y,"." G CHK  G ASK  Q1 K:IBCAN=1 IBQUIT K IBCAN  Q K %,IBEPAR,IBSTAT,IBARST,IBAC1,IB,DFN,IBX,IBZ,DIC,DIE,DR,PRCASV,PRCASVC,X,Y,IBEDI  ;\*\*\*  ;I $D(XRT0) S:'$D(XRTN) XRTN="IBCC" D T1^%ZOSV ;stop rt clock  Q  CRD(IBIFN) ; entry to point to add iteration # to claim  N IBFDA  S IBITN=$$ITN^IBCCC(IBIFN)  S IBFDA(399,IBIFN\_",",.01)=IBITN  D FILE^DIE("","IBFDA")  ; this will re-open the claim, so reset to cancelled  S DIE=399,DA=IBIFN  S DR="16////1"  D ^DIE K DIE,DR  Q  ;  PROCESS(IBIFN,IBCAN) ;  ; Entry point when the bill number is already known. Use this when  ; you just want to try to cancel this bill and this bill only.  ; Input:  ; IBIFN - Internal bill# (Required)  ; IBCAN - Cancel Flag (optional, defaults to 1 if not included)  ;  NEW IBNOASK  S IBNOASK=1  S IBCAN=$G(IBCAN,1)  G ASK  ;  ;IBCC | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCC ;ALB/MJB - CANCEL THIRD PARTY BILL ;14 JUN 88 10:12  ;;2.0;INTEGRATED BILLING;\*\*2,19,77,80,51,142,137,161,199,241,155,276,320,358,433,432,447,516,547\*\*;21-MAR-94;Build 119  ;;Per VA Directive 6402, this routine should not be modified.  ;  ;MAP TO DGCRC  ;  I '$D(IBCAN) S IBCAN=1  ASK ;  ;\*\*\*  ;I $D(XRT0) S:'$D(XRTN) XRTN="IBCC" D T1^%ZOSV ;stop rt clock  ;S XRTL=$ZU(0),XRTN="IBCC-1" D T0^%ZOSV ;start rt clock  ;  ; If called at entry point PROCESS, variable IBNOASK will exist.  ; First time through, IBNOASK=1  ; Second time through, IBNOASK=2 and it will quit  I $G(IBNOASK)=2 G Q  I $G(IBNOASK)=1 S IBNOASK=2  ;  G Q:$G(IBCE("EDI"))  D Q  S IBQUIT=0  N DPTNOFZY S DPTNOFZY=1 ;Suppress PATIENT file fuzzy lookups  I '$G(IBNOASK) S DIC="^DGCR(399,",DIC(0)="AEMQZ",DIC("A")="Enter BILL NUMBER or Patient NAME: " W !! D ^DIC I Y<1 S IBQUIT=1 G Q1  K IB364  NOPTF ; Note if IB364 is >0 it will be used as the ien to update in file 364  N DA,I  I '$G(IBNOASK) S IBIFN=+$G(Y)  I '$G(IBIFN) G ASK  I IBCAN>1 D NOPTF^IBCB2 I 'IBAC1 D NOPTF1^IBCB2 G ASK  ;  I $G(IBCNCRD)=1,$P($P($G(^DGCR(399,IBIFN,0)),U),"-",2)>98 D Q  .W !!,"Please note that you have exceeded the maximum number of iterations (99) for this claim."  .W "Copy and cancel (CLON) must be used to correct this bill."  .S IBQUIT=1 H 3  ; Check if bill has been referred to Counsel  I $P($G(^PRCA(430,IBIFN,6)),U,4) D G ASK  . W !,"This bill has been referred to Regional Counsel and cannot be 'CANCELLED' in"  . W !,"Integrated Billing. Please use the option 'TP Referred Follow-up'"  . W !,"[PRCA RC ACTION MENU] in Accounts Receivable to request that Regional"  . W !,"Counsel return the bill to your facility."  . Q  ;  F I=0,"S","U1" S IB(I)=$G(^DGCR(399,IBIFN,I))  S IBSTAT=$P(IB(0),U,13)  ; REMOVE New messages for CRD option IB\*2.0\*433 in IB\*2.0\*447 IA#5630  ;I $G(IBCNCRD)=1,IBSTAT'=2,'$$ACCK^PRCAACC(IBIFN) D Q  ;.W !!,"This option cannot be used to correct some Billing Rate Types (Example: TRICARE)"  ;.W "Copy and cancel (CLON) must be used to correct this bill."  ;.S IBQUIT=1 H 3  ;  ; Restrict access to this process for REQUEST MRA bills  I IBSTAT=2,'$G(IBCE("EDI")),$$MRAWL^IBCEMU2(IBIFN) D G ASK  . W !!?4,"This bill is in a status of REQUEST MRA and it does appear on the"  . W !?4,"MRA Management Work List. Please use the 'MRA Management Menu' options"  . W !?4,"for all processing related to this bill."  . Q  ;  ; IB\*2.0\*432 Restrict access to claims on the new CBW Worklist  I $P($G(^DGCR(399,IBIFN,"S1")),U,7)=1,$G(IBMRANOT)'=1 D G ASK  . W !!?4,"This bill appears on the CBW Management Work List. Please use the"  . W !?4,"'CBW Management Menu' options for all processing related to this bill."  . S IBQUIT=1 ;JRA need to set quit flag after issuing this message  . Q  ;  ; Check if this is a paper claim. If not, check for split EOB. If split, don't allow CRD unless more than 1 EOB has been returned  I $G(IBCNCRD)=1,$P($G(^DGCR(399,IBIFN,"TX")),U,8)'=1,$$SPLTMRA^IBCEMU1(IBIFN)=1 D Q  .W !!,"There is a split EOB associated with this claim. You cannot use this option to Correct this claim until the second EOB has been received."  .S IBQUIT=1 H 3  .Q  ;  ; Warning message if in a REQUEST MRA status with no MRA on file  ; IB\*2.0\*516/TAZ,MRD - Forbid the user from using the option CRD  ; (Correct Rejected/Denied Bill) on an MRA claim if the status is  ; REQUEST MRA (IBSTAT=2).  I IBSTAT=2,'$$MRACNT^IBCEMU1(IBIFN) D I $G(IBQUIT) H 3 Q  . N REJ  . D TXSTS^IBCEMU2(IBIFN,,.REJ)  . ;IB\*2.0\*516/TAZ - If CRD is from CSA allow a REJected claim to be CRD'ed without displaying a warning.  . I $G(IBCNCSA),REJ Q  . W \*7,!!?4,$S('$G(IBCNCRD):"Warning! ",1:""),"This bill is in a status of REQUEST MRA."  . W !?4,"No MRAs have been received"  . I REJ W ", but the most recent transmission of this",!?4,"MRA request bill was rejected."  . I 'REJ W " and there are no rejection messages on file",!?4,"for the most recent transmission of this MRA request bill."  . I $G(IBCNCRD) S IBQUIT=1  . Q  ;  I IBCAN=2,IB("S")]"",+$P(IB("S"),U,16),$P(IB("S"),U,17)]"" D G 1  . W !!,"This bill was cancelled on " S Y=$P(IB("S"),U,17) X ^DD("DD") W Y," by ",$S($P(IB("S"),U,18)']"":IBU,$D(^VA(200,$P(IB("S"),U,18),0)):$P(^(0),U,1),1:IBU),"."  . S IBQUIT=1  ;  ; IB\*2.0\*516/TAZ,MRD - Forbid the user from using the option CRD  ; (Correct Rejected/Denied Bill) on all but primary claims.  I $G(IBCNCRD),($$COB^IBCEF(IBIFN)'="P") D Q  . W !!,"Please note that COB data may exist for this bill."  . W !,"Copy and cancel (CLON) must be used to correct this bill."  . S IBQUIT=1  . H 3  . Q  ;  ; Notify if a payment has been posted to this bill before cancel  N PRCABILL  S PRCABILL=$$TPR^PRCAFN(IBIFN)  I PRCABILL=-1 W !!,"Please note: PRCA was unable to determine if a payment has been posted." I $G(IBCNCRD)=1 W !,"Copy and cancel (CLON) must be used to correct this bill." S IBQUIT=1 H 3 Q  I PRCABILL>0 W !!,"Please note a PAYMENT of \*\*$"\_$$TPR^PRCAFN(IBIFN)\_"\*\* has been POSTED to this bill."  ; New message for CRD option  I $G(IBCNCRD)=1,PRCABILL>0 W !,"Copy and cancel (CLON) must be used to correct this bill." S IBQUIT=1 H 3 Q  ;  ; If bill was created via Electronic claims process then notify  ; user that cancellation should occur using ECME package  I $$GET1^DIQ(399,IBIFN\_",",460)]"" D G:'Y ASK  . W !!!?5,"This bill was created by the"  . W !?5,"Electronic Claims Management Engine (ECME)."  . W !?5,"Cancellation needs to occur in the ECME package by"  . W !?5,"submitting a REVERSAL to the Payer.",!!  . K DIR S DIR("A",1)="Has a REVERSAL for this e-Claim already been",DIR("A")="submitted to the payer via the ECME package (Y/N)",DIR(0)="Y",DIR("B")="NO" D ^DIR  . I Y=0 W !!,"<PLEASE SUBMIT A REVERSAL USING THE APPROPRIATE OPTION IN THE ECME PACKAGE>",$C(7)  ;  CHK ;  ; if user came from CLON, make sure they know about the new CRD option IB\*2.0\*447 remove TRICARE msg.  I $G(IBCNCOPY)=1 D  .W !!,\*7,"Warning: This option should NOT be used to correct Rejected/Denied claims."  .W !," It should ONLY be used to correct DENIED claims which have payments"  .W !," posted against them.\*\*\*" ; and claims with certain Billing Rate Types (Example: TRICARE)."  ;  S (IBCCCC,IBQUIT)=0 I '$G(IBCEAUTO),'$G(IBMCSCAN) W !!,"ARE YOU SURE YOU WANT TO CANCEL THIS BILL" S %=2 D YN^DICN G:%=0 HELP I %'=1 S IBQUIT=1 G NO  ;  I '$G(IBCEAUTO) W !!,"LAST CHANCE TO CHANGE YOUR MIND..."  S DIE=399,DA=IBIFN,DIE("NO^")=""  S DR="16;S:'X Y=0;19;S IBCCCC=1;"  I $G(IBCEAUTO) S DR="16////1;19////EDI/MRA TURNED OFF;S IBCCCC=1;"  ;  ; esg - 8/23/06 - IB\*2\*358 - fix semi-colon in free text field  I $G(IBMCSRSC)'="" S DR="16;S:'X Y=0;19//^S X=IBMCSRSC;S IBCCCC=1;"  D ^DIE K DIE,DR  ;  NO I 'IBCCCC W !!,"<NO ACTION TAKEN>",\*7 S IBQUIT=1 G ASK:IBCAN<2,Q  S IBCCR=$P($G(^DGCR(399,IBIFN,"S")),U,19)  ; update claim # with new iteration IB\*2.0\*447 move to later in the process  ;D:$G(IBCNCRD)=1 CRD  W !!,"...Bill has been cancelled..." D BULL^IBCBULL,BSTAT^IBCDC(IBIFN),PRIOR^IBCCC2(IBIFN)  ;  ; cancelling in ingenix claimsmanager if ingenix is running  ; clean-up of variables is OK if not coming in from ListMan screen  I $$CM^IBCIUT1(IBIFN) S IBCISNT=4 D ST2^IBCIST I '$G(IBCICNCL) K IBCISNT,IBCISTAT,IBCIREDT,IBCIERR  ;  S IBEDI=$G(IB364)  I 'IBEDI S IBEDI=+$$LAST364^IBCEF4(IBIFN)  ; ib\*2.0\*547 don't cancel MRA if cloning a bill that is secondary to MRA (share the same claim#)  I IBEDI D UPDEDI^IBCEM(IBEDI,"C",,$S($$MRASEC^IBCEF4(IBIFN):2,1:"")) ;Update EDI files, if needed  ;  F I="S","U1" S IB(I)=$S($D(^DGCR(399,IBIFN,I)):^(I),1:"")  S PRCASV("ARREC")=IBIFN,PRCASV("AMT")=$S(IB("U1")']"":0,1:$P(IB("U1"),"^")),PRCASV("DATE")=$P(IB("S"),"^",17),PRCASV("BY")=$P(IB("S"),"^",18)  S PRCASV("COMMENT")=$S($P(IB("S"),U,19)]"":$P(IB("S"),U,19),$P(^IBE(350.9,1,2),"^",7)]"":$P(^(2),"^",7),1:"BILL CANCELLED IN MAS")  S PRCASV("BY")=$S($P(IB("S"),U,18)]"":$P(IB("S"),U,18),1:"")  ; IA#3374/IB\*2.0\*433 Pass the CRD flag so FMS knows to send a cancel record before the new E record is sent  ;S X=$$CANCEL^RCBEIB($G(PRCASV("ARREC")),$G(PRCASV("DATE")),$G(PRCASV("BY")),$G(PRCASV("AMT")),$G(PRCASV("COMMENT")))  S PRCASV("ARCRD")=$G(IBCNCRD)  S X=$$CANCEL^RCBEIB($G(PRCASV("ARREC")),$G(PRCASV("DATE")),$G(PRCASV("BY")),$G(PRCASV("AMT")),$G(PRCASV("COMMENT")),$G(PRCASV("ARCRD")))  W !,$S(X:">> The receivable associated with the claim was cancelled.",1:">> The receivable associated with the claim was not cancelled.")  I $P(X,U,2)]"" W !,">>> ",$P(X,U,2) ; The reason why the claim can not be cancelled.  I IBCAN<2 D RNB^IBCC1 ;assign a reason not billable  G ASK:IBCAN<2,Q  ;  HELP W !,?3,"Answer 'YES' or 'Y' if you wish to cancel this bill.",!,?3,"Answer 'NO' or 'N' if you want to abort." G CHK  Q  1 I $P(IB(0),U,13)=1 W !,"This record was re-opened on " S Y=$P(IB(0),U,14) X ^DD("DD") W Y,"." G CHK  G ASK  Q1 K:IBCAN=1 IBQUIT K IBCAN  Q K %,IBEPAR,IBSTAT,IBARST,IBAC1,IB,DFN,IBX,IBZ,DIC,DIE,DR,PRCASV,PRCASVC,X,Y,IBEDI  ;\*\*\*  ;I $D(XRT0) S:'$D(XRTN) XRTN="IBCC" D T1^%ZOSV ;stop rt clock  Q  CRD(IBIFN) ; entry to point to add iteration # to claim  N IBFDA  S IBITN=$$ITN^IBCCC(IBIFN)  S IBFDA(399,IBIFN\_",",.01)=IBITN  D FILE^DIE("","IBFDA")  ; this will re-open the claim, so reset to cancelled  S DIE=399,DA=IBIFN  S DR="16////1"  D ^DIE K DIE,DR  Q  ;  PROCESS(IBIFN,IBCAN) ;  ; Entry point when the bill number is already known. Use this when  ; you just want to try to cancel this bill and this bill only.  ; Input:  ; IBIFN - Internal bill# (Required)  ; IBCAN - Cancel Flag (optional, defaults to 1 if not included)  ;  NEW IBNOASK  S IBNOASK=1  S IBCAN=$G(IBCAN,1)  G ASK  ;  ;IBCC | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCCC | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** |  | | | | | | | | |
| **Related Options** | None | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  |  | | | | |  | | | |
| **Data Dictionary (DD) References** |  | | | | | | | | |
| **Related Protocols** |  | | | | | | | | |
| **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** | | | | | | | | | |
| IBCCC^INT^1^^0  IBCCC ;ALB/AAS - CANCEL AND CLONE A BILL ;25-JAN-90  ;;2.0;INTEGRATED BILLING;\*\*80,109,106,51,320,433,432,447,516\*\*;21-MAR-94;Build 123  ;;Per VA Directive 6402, this routine should not be modified.  ;  ;MAP TO DGCRCC  ;  ;STEP 1 - cancel bill  ;STEP 1.5 - entry to clone previously cancelled bill. (must be cancel)  ;STEP 2 - build array of IBIDS call screen that asks ok  ;STEP 3 - pass stub entry to ar  ;STEP 4 - store stub data in MCCR then x-ref  ;STEP 4.5 - store claim clone info on "S1" node.  ;STEP 5 - get remainder of data to move and store in MCCR then x-ref  ;STEP 6 - go to screens, come out to IBB1  EN ;  N IBBCF,IBBCT,IBBCB,IBCCR,IBDBC,IBCNCOPY,IBNOCALC  S IBDBC=DT ;date claim was cloned  S IBBCB=DUZ ;user-id of person cloning the claim.  S IBCNCOPY=1 ; flag indicating this function is entered as the copy/cancel option  ;  ; MRD;IB\*2.0\*516 - Added the flag IBNOCALC. This flag is set here and  ; below. When charges are recalculated in BILL^IBCRBC, the first time  ; that procedure is entered for the new claim, if the IBNOCALC flag is  ; set, it will reset the flag and quit out. That is, it does not re-  ; calculate the charges the first time it otherwise would if the user  ; is doing either a CRD or CLON.  ;  S IBNOCALC=1  ;  STEP1 I $G(IBCE("EDI"))>1 G END1  S IBCAN=2,IBQUIT=0,IBAC=6,IBU="UNSPECIFIED"  I '$G(IBCE("EDI")) D ASK^IBCC  I $G(IBCE("EDI"))=1 S IB364="" D NOPTF^IBCC  G:IBQUIT END1  I 'IBCCCC!('$D(IBIFN)) G STEP1:'$G(IBCE("EDI")),END1  I $G(IBCE("EDI")) S IBCE("EDI")=2  EN1 ;  STEP1P5 I '$D(IBIFN) S IBCAN=2,IBQUIT=0,IBAC=6 W !,"Copy Previously Cancelled Bill.",!! S DIC="^DGCR(399,",DIC("S")="I $P(^(0),U,13)=7",DIC(0)="AEMQZ",DIC("A")="Enter BILL NUMBER or Patient NAME: " D ^DIC G:Y<1 END S IBIFN=+Y  ;  S IBBCF=IBIFN ;this is the claim we are copying FROM  S IBIDS(.15)=IBIFN K IBIFN  STEP2 S IBND0=^DGCR(399,IBIDS(.15),0) I $D(^("U")) S IBNDU=^("U")  ; \*\*\* Note - all these fields should also be included in WHERE^IBCCC1  ; IB\*2.0\*432 added check of variable IBSILENT to allow COB copy in background mode  F I=2:1:12 S:$P(IBND0,"^",I)]"" IBIDS(I/100)=$P(IBND0,"^",I)  F I=16:1:19,21:1:27 S:$P(IBND0,"^",I)]"" IBIDS(I/100)=$P(IBND0,"^",I)  F I=151,152,155 S IBIDS(I)=$P(IBNDU,"^",(I-150))  S IBIDS(159.5)=$P(IBNDU,U,20)  ; \*\*\*  D:$G(IBSILENT)="" HOME^%ZIS  S DFN=IBIDS(.02) D DEM^VADPT  I +$G(IBCTCOPY)!$G(IBCE("EDI")) G STEP3  D ^IBCA1  ASK S IBYN=0 W !!,"IS THE ABOVE INFORMATION CORRECT AS SHOWN" S %=1 D YN^DICN G END:%=2,STEP3:%=1 I % G END  W !!?4,"YES - If this information is correct as shown and you wish to file the bill.",!?4,"NO - If you wish to change this information prior to filing."  W !?4,"'^' - Enter the up-arrow character to DELETE this Bill at this time." G ASK  ;  STEP3 ;  S PRCASV("SER")=$P($G(^IBE(350.9,1,1)),"^",14)  S PRCASV("SITE")=$P($$SITE^VASITE,"^",3),IBNWBL="",PRCASV("ARCRD")=$G(IBCNCRD)  ; IA#386 & 1992 If user came from CRD option, need to pass old bill # and claim ien, as well as new iteration number  I $G(IBCNCRD)=1 D CRD^IBCC(IBBCF) S PRCASV("ARREC")=IBBCF,PRCASV("ARBIL")=PRCASV("SITE")\_"-"\_$P(IBITN,"-"),PRCASV("ARITN")=PRCASV("SITE")\_"-"\_IBITN  W:$G(IBSILENT)="" !,"Passing bill to Accounts Receivable Module..." D SETUP^PRCASVC3 I $S($P(PRCASV("ARREC"),"^")=-1:1,$P(PRCASV("ARBIL"),"^")=-1:1,1:0) W:$G(IBSILENT)="" \*7," ",$P(PRCASV("ARREC"),"^",2),$P(PRCASV("ARBIL"),"^",2) G END  S IBIDS(.01)=$P(PRCASV("ARBIL"),"-",2),IBIDS(.17)=$S($D(IBIDS(.17)):IBIDS(.17),1:PRCASV("ARREC"))  I '$G(IBCE("EDI")) W !,"Billing Record #",IBIDS(.01)," being established for '",VADM(1),"'..." S IBIDS(.02)=DFN,IBHV("IBIFN")=$S($G(IBIFN):IBIFN,1:$G(IBIDS(.15)))  G ^IBCCC1 ;go to step4  Q  ;  END W:$G(IBSILENT)="" !!,"No Billing Record Set up. You must manually enter the bill."  END1 K %,%DT,IBCAN,IBAC,IBND0,IBNDU,IBYN,IBCCCC,IBIFN,IB,IBA,IBNWBL,IBBT,IBIDS,IBU,I,J,VA,VADM,X,X1,X2,X3,X4,D,Y  ;I '$G(IBQUIT),$S(+$G(IBCNCOPY):1,1:'$G(IBCE("EDI"))) G STEP1  I '$G(IBQUIT),$S(+$G(IBCNCOPY)!(+$G(IBCNCRD)):1,1:'$G(IBCE("EDI"))) G STEP1  K IBQUIT,IBCNCOPY,IBCNCRD,IBNOCALC  Q  ;  ITN(IBX) ; determine iteration # for rejected or denied claim  N IBCF,IBCL  S IBCF=$P($G(^DGCR(399,IBX,"S1")),U,2)  ; if this claim has never been cloned, iteration # is -01  Q:IBCF="" $P($G(^DGCR(399,IBX,0)),U)\_"-01"  S IBCL=$P($G(^DGCR(399,IBCF,0)),U)  ; if claim was a CLON 1st and now a CORRECT, this is the 1st iteration  I $P(IBCL,"-")'=$P($P($G(^DGCR(399,IBX,0)),U),"-") Q $P($G(^DGCR(399,IBX,0)),U)\_"-01"  ; to determine iteration#, need to incriment from claim that was cloned from  S IBITN=$P(IBCL,"-",2),IBITN=IBITN+1 I $L(IBITN)=1 S IBITN="0"\_IBITN  Q $P(IBCL,"-")\_"-"\_IBITN  ;  CRD ; new entry point if user comes from CRD option instead of CLON  N IBBCF,IBBCT,IBBCB,IBCCR,IBDBC,IBCNCRD,IBITN,IBNOCALC  S IBDBC=DT ;date claim was cloned  S IBBCB=DUZ ;user-id of person cloning the claim.  S IBCNCRD=1 ; flag indicating this function is entered as the CRD option  ;  ; MRD;IB\*2.0\*516 - Added the flag IBNOCALC. This flag is set here and  ; above. When charges are recalculated in BILL^IBCRBC, the first time  ; that procedure is entered for the new claim, if the IBNOCALC flag is  ; set, it will reset the flag and quit out. That is, it does not re-  ; calculate the charges the first time it otherwise would if the user  ; is doing either a CRD or CLON.  ;  S IBNOCALC=1  ;  G STEP1  Q  ; | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCCC ;ALB/AAS - CANCEL AND CLONE A BILL ;25-JAN-90  ;;2.0;INTEGRATED BILLING;\*\*80,109,106,51,320,433,432,447,516\*\*;21-MAR-94;Build 123  ;;Per VA Directive 6402, this routine should not be modified.  ;  ;MAP TO DGCRCC  ;  ;STEP 1 - cancel bill  ;STEP 1.5 - entry to clone previously cancelled bill. (must be cancel)  ;STEP 2 - build array of IBIDS call screen that asks ok  ;STEP 3 - pass stub entry to ar  ;STEP 4 - store stub data in MCCR then x-ref  ;STEP 4.5 - store claim clone info on "S1" node.  ;STEP 5 - get remainder of data to move and store in MCCR then x-ref  ;STEP 6 - go to screens, come out to IBB1  EN ;  N IBBCF,IBBCT,IBBCB,IBCCR,IBDBC,IBCNCOPY,IBNOCALC  S IBDBC=DT ;date claim was cloned  S IBBCB=DUZ ;user-id of person cloning the claim.  S IBCNCOPY=1 ; flag indicating this function is entered as the copy/cancel option  ;  ; MRD;IB\*2.0\*516 - Added the flag IBNOCALC. This flag is set here and  ; below. When charges are recalculated in BILL^IBCRBC, the first time  ; that procedure is entered for the new claim, if the IBNOCALC flag is  ; set, it will reset the flag and quit out. That is, it does not re-  ; calculate the charges the first time it otherwise would if the user  ; is doing either a CRD or CLON.  ;  S IBNOCALC=1  ;  STEP1 I $G(IBCE("EDI"))>1 G END1  S IBCAN=2,IBQUIT=0,IBAC=6,IBU="UNSPECIFIED"  I '$G(IBCE("EDI")) D ASK^IBCC  I $G(IBCE("EDI"))=1 S IB364="" D NOPTF^IBCC  G:IBQUIT END1  ;JRA use $G to prevent <UNDEF> error when/if IBCCCC=""  ;I 'IBCCCC!('$D(IBIFN)) G STEP1:'$G(IBCE("EDI")),END1 ;JRA ';'  I '$G(IBCCCC)!('$D(IBIFN)) G STEP1:'$G(IBCE("EDI")),END1 ;JRA add $G for IBCCCC  I $G(IBCE("EDI")) S IBCE("EDI")=2  EN1 ;  STEP1P5 I '$D(IBIFN) S IBCAN=2,IBQUIT=0,IBAC=6 W !,"Copy Previously Cancelled Bill.",!! S DIC="^DGCR(399,",DIC("S")="I $P(^(0),U,13)=7",DIC(0)="AEMQZ",DIC("A")="Enter BILL NUMBER or Patient NAME: " D ^DIC G:Y<1 END S IBIFN=+Y  ;  S IBBCF=IBIFN ;this is the claim we are copying FROM  S IBIDS(.15)=IBIFN K IBIFN  STEP2 S IBND0=^DGCR(399,IBIDS(.15),0) I $D(^("U")) S IBNDU=^("U")  ; \*\*\* Note - all these fields should also be included in WHERE^IBCCC1  ; IB\*2.0\*432 added check of variable IBSILENT to allow COB copy in background mode  F I=2:1:12 S:$P(IBND0,"^",I)]"" IBIDS(I/100)=$P(IBND0,"^",I)  F I=16:1:19,21:1:27 S:$P(IBND0,"^",I)]"" IBIDS(I/100)=$P(IBND0,"^",I)  F I=151,152,155 S IBIDS(I)=$P(IBNDU,"^",(I-150))  S IBIDS(159.5)=$P(IBNDU,U,20)  ; \*\*\*  D:$G(IBSILENT)="" HOME^%ZIS  S DFN=IBIDS(.02) D DEM^VADPT  I +$G(IBCTCOPY)!$G(IBCE("EDI")) G STEP3  D ^IBCA1  ASK S IBYN=0 W !!,"IS THE ABOVE INFORMATION CORRECT AS SHOWN" S %=1 D YN^DICN G END:%=2,STEP3:%=1 I % G END  W !!?4,"YES - If this information is correct as shown and you wish to file the bill.",!?4,"NO - If you wish to change this information prior to filing."  W !?4,"'^' - Enter the up-arrow character to DELETE this Bill at this time." G ASK  ;  STEP3 ;  S PRCASV("SER")=$P($G(^IBE(350.9,1,1)),"^",14)  S PRCASV("SITE")=$P($$SITE^VASITE,"^",3),IBNWBL="",PRCASV("ARCRD")=$G(IBCNCRD)  ; IA#386 & 1992 If user came from CRD option, need to pass old bill # and claim ien, as well as new iteration number  I $G(IBCNCRD)=1 D CRD^IBCC(IBBCF) S PRCASV("ARREC")=IBBCF,PRCASV("ARBIL")=PRCASV("SITE")\_"-"\_$P(IBITN,"-"),PRCASV("ARITN")=PRCASV("SITE")\_"-"\_IBITN  W:$G(IBSILENT)="" !,"Passing bill to Accounts Receivable Module..." D SETUP^PRCASVC3 I $S($P(PRCASV("ARREC"),"^")=-1:1,$P(PRCASV("ARBIL"),"^")=-1:1,1:0) W:$G(IBSILENT)="" \*7," ",$P(PRCASV("ARREC"),"^",2),$P(PRCASV("ARBIL"),"^",2) G END  S IBIDS(.01)=$P(PRCASV("ARBIL"),"-",2),IBIDS(.17)=$S($D(IBIDS(.17)):IBIDS(.17),1:PRCASV("ARREC"))  I '$G(IBCE("EDI")) W !,"Billing Record #",IBIDS(.01)," being established for '",VADM(1),"'..." S IBIDS(.02)=DFN,IBHV("IBIFN")=$S($G(IBIFN):IBIFN,1:$G(IBIDS(.15)))  G ^IBCCC1 ;go to step4  Q  ;  END W:$G(IBSILENT)="" !!,"No Billing Record Set up. You must manually enter the bill."  END1 K %,%DT,IBCAN,IBAC,IBND0,IBNDU,IBYN,IBIFN,IB,IBA,IBNWBL,IBBT,IBIDS,IBU,I,J,VA,VADM,X,X1,X2,X3,X4,D,Y ;JRA remove IBCCCC from KILL  ;I '$G(IBQUIT),$S(+$G(IBCNCOPY):1,1:'$G(IBCE("EDI"))) G STEP1  I '$G(IBQUIT),$S(+$G(IBCNCOPY)!(+$G(IBCNCRD)):1,1:'$G(IBCE("EDI"))) G STEP1  K IBQUIT,IBCNCOPY,IBCNCRD,IBNOCALC,IBCCCC ;JRA Added IBCCCC to KILL  Q  ;  ITN(IBX) ; determine iteration # for rejected or denied claim  N IBCF,IBCL  S IBCF=$P($G(^DGCR(399,IBX,"S1")),U,2)  ; if this claim has never been cloned, iteration # is -01  Q:IBCF="" $P($G(^DGCR(399,IBX,0)),U)\_"-01"  S IBCL=$P($G(^DGCR(399,IBCF,0)),U)  ; if claim was a CLON 1st and now a CORRECT, this is the 1st iteration  I $P(IBCL,"-")'=$P($P($G(^DGCR(399,IBX,0)),U),"-") Q $P($G(^DGCR(399,IBX,0)),U)\_"-01"  ; to determine iteration#, need to incriment from claim that was cloned from  S IBITN=$P(IBCL,"-",2),IBITN=IBITN+1 I $L(IBITN)=1 S IBITN="0"\_IBITN  Q $P(IBCL,"-")\_"-"\_IBITN  ;  CRD ; new entry point if user comes from CRD option instead of CLON  N IBBCF,IBBCT,IBBCB,IBCCR,IBDBC,IBCNCRD,IBITN,IBNOCALC  S IBDBC=DT ;date claim was cloned  S IBBCB=DUZ ;user-id of person cloning the claim.  S IBCNCRD=1 ; flag indicating this function is entered as the CRD option  ;  ; MRD;IB\*2.0\*516 - Added the flag IBNOCALC. This flag is set here and  ; above. When charges are recalculated in BILL^IBCRBC, the first time  ; that procedure is entered for the new claim, if the IBNOCALC flag is  ; set, it will reset the flag and quit out. That is, it does not re-  ; calculate the charges the first time it otherwise would if the user  ; is doing either a CRD or CLON.  ;  S IBNOCALC=1  ;  G STEP1  Q  ; | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCCCB | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** |  | | | | | | | | |
| **Related Options** | None | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  |  | | | | |  | | | |
| **Data Dictionary (DD) References** |  | | | | | | | | |
| **Related Protocols** |  | | | | | | | | |
| **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** | | | | | | | | | |
| IBCCCB^INT^1^^0  IBCCCB ;ALB/ARH - COPY BILL FOR COB ;2/13/06 10:46am  ;;2.0;INTEGRATED BILLING;\*\*80,106,51,151,137,182,155,323,436,432,447,547\*\*;21-MAR-94;Build 119  ;;Per VA Directive 6402, this routine should not be modified.  ;  ; Copy bill for COB w/out cancelling, update some flds  ; Primary->Secondary->Tertiary  ASK ;  S IBCBCOPY=1 ; flag that copy function entered thru Copy COB option  ;  D KVAR S IBCAN=2,IBU="UNSPECIFIED"  ;  S IBX=$$PB^IBJTU2 S:+IBX=2 IBIFN=$P(IBX,U,2) I +IBX=1 S DFN=$P(IBX,U,2),IBV=1,IBAC=5 D DATE^IBCB  I '$G(IBIFN) G EXIT  ;  ; IB\*2.0\*432 Restrict access to only allow claims that are NOT on the new CBW Worklist  I $P($G(^DGCR(399,IBIFN,"S1")),U,7)=1,$G(IBMRANOT)'=1 D G ASK  . W !!?4,"This bill appears on the CBW Management Work List. Please use the"  . W !?4,"'CBW Management Menu' options for all processing related to this bill."  . Q  ; Restrict access to this process for REQUEST MRA bills in 2 Cases:  ; 1. No MRA EOB's on File for bill  I $P($G(^DGCR(399,IBIFN,0)),U,13)=2,'$$CHK^IBCEMU1(IBIFN) D G ASK  . W !!?4,"This bill is in a status of REQUEST MRA and it has No MRA EOB's"  . W !?4,"on file. Access to this bill is restricted."  ;  ; 2. At least one MRA EOB appears on the MRA management worklist  I $P($G(^DGCR(399,IBIFN,0)),U,13)=2,$$MRAWL^IBCEMU2(IBIFN) D G ASK  . W !!?4,"This bill is in a status of REQUEST MRA and it does appear on the"  . W !?4,"MRA Management Work List. Please use the 'MRA Management Menu' options"  . W !?4,"for all processing related to this bill."  . Q  ;  ; If MRA is Activated and bill is in Entered/Not Reviewed status and current insurance Co. is WNR -->  ; ask if user wants to continue  I $$EDIACTV^IBCEF4(2),$P($G(^DGCR(399,IBIFN,0)),U,13)=1,$$MCRWNR^IBEFUNC(+$$CURR^IBCEF2(IBIFN)) D I 'Y G ASK  . W !!?4,"This bill is in a status of ENTERED/NOT REVIEWED and current payer is "  . W !?4,"MEDICARE (WNR). No MRA has been requested for this bill."  . S DIR(0)="YA",DIR("B")="NO",DIR("A")=" Are you sure you want to continue to process this bill?: "  . D ^DIR K DIR  ;  ; Display related bills  D DSPRB^IBCCCB0(IBIFN)  ;  CHKB ; Entrypoint-COB processing via EDI's COB Mgmt  ; Ask if final EOB was received for previous bill  I '$$FINALEOB^IBCCCB0(IBIFN) S IBSECHK=1  I $G(IBSECHK)=1,$$MCRONBIL^IBEFUNC(IBIFN) G EXIT  ;  ; Warn if previous bill not at least authorized  I '$$MCRONBIL^IBEFUNC(IBIFN) I '$$COBOK^IBCCCB0(IBIFN) G EXIT  ;  CHKB1 ; Entry point for Automatic/Silent COB Processing.  ; No writes or reads can occur from this point forward if variable  ; IBSILENT=1. Any and all error messages should be processed with  ; the ERROR procedure below.  ;  S IBX=$G(^DGCR(399,+IBIFN,0)),DFN=$P(IBX,U,2),IBDT=$P(IBX,U,3)\1,IBER=""  I IBCAN>1 D NOPTF^IBCB2 I 'IBAC1 D NOPTF1^IBCB2 G ASK1  ;  F IBI=0,"S","U1","M","MP","M1" S IB(IBI)=$G(^DGCR(399,IBIFN,IBI))  I IB(0)="" S IBER="Invalid Bill Number" D ERROR G ASK1  ;  ; check to see if the bill has been cancelled  I $P(IB("S"),U,16),$P(IB("S"),U,17) D G ASK1  . N WHO  . S IBER="This bill was cancelled on "  . S IBER=IBER\_$$FMTE^XLFDT($P(IB("S"),U,17),"1Z")\_" by "  . S WHO="UNSPECIFIED"  . I $P(IB("S"),U,18) S WHO=$P($G(^VA(200,$P(IB("S"),U,18),0)),U,1)  . S IBER=IBER\_WHO\_"."  . D ERROR  . Q  ;  S IBCOB=$$COB^IBCEF(IBIFN),IBCOBN=$TR(IBCOB,"PSTA","12")  S IBMRAIO=+$$CURR^IBCEF2(IBIFN),IBMRAO=$$MCRWNR^IBEFUNC(IBMRAIO)  S IBNMOLD=$S(IBCOB="P":"Primary",IBCOB="S":"Secondary",IBCOB="T":"Tertiary",IBCOB="A":"Patient",1:"")\_$S(IBMRAO:"-MRA Only",1:"")  S IBINSOLD=$G(^DIC(36,$S(IB("MP"):+IB("MP"),IBMRAO:IBMRAIO,1:0),0))  ;  NEXTP ; If current bill=MEDICARE WNR and valid 'next payer', use same  ; bill for new payer  ; If next valid 'payer' is ins co or MEDICARE WNR, create new bill  S IBCOBN=IBCOBN+1,IBNM=$S(IBCOBN=2:"Secondary Payer",IBCOBN=3:"Tertiary Payer",1:"")  ;  I IBNM="" S IBER=$P(IB(0),U,1)\_" is a "\_IBNMOLD\_" bill, there is no next bill in the series." D ERROR G ASK1  ;  S IBX=+$P(IB("M1"),U,(4+IBCOBN)),IBY=$G(^DGCR(399,+IBX,0)),IBCOBIL(+IBIFN)=""  ;  I $P(IBY,U,13)=7 S IBER="The "\_$P(IBNM," ",1)\_" bill "\_$P(IBY,U,1)\_" has been cancelled." D ERROR S IBX=""  ;  I +IBX,$D(IBCOBIL(+IBX)) S IBER="Next bill in series can not be determined." D ERROR G ASK1  I +IBX S IBER=$P(IBNM," ",1)\_" bill already defined for this series: "\_$P(IBY,U,1) D ERROR S IBIFN=IBX G ASK1  ;  S IBINSN=$P(IB("M"),U,IBCOBN) I 'IBINSN S IBER="There is no "\_IBNM\_" for "\_$P(IB(0),U,1)\_"." D ERROR G ASK1  S IBINS=$G(^DIC(36,+IBINSN,0)) I IBINS="" S IBER="The "\_IBNM\_" for "\_$P(IB(0),U,1)\_" is not a valid Insurance Co." D ERROR G ASK1  ;  S IBMRA=0  I $P(IBINS,U,2)="N" S IBQ=0 D G:IBQ NEXTP  . I $$MCRWNR^IBEFUNC(IBINSN) D Q  .. ; Check if a valid tert ins if MCR WNR secondary  .. I IBCOBN'>2 D  ... N Z  ... S Z=+$P(IB("M"),U,IBCOBN+1)  ... I Z,$D(^DIC(36,Z,0)),$P(^(0),U,2)'="N" S IBMRA=1,IBNM=$P(IBNM," ")\_"-MRA.Only"  .. I 'IBMRA S IBER="MEDICARE will not reimburse and no further valid insurance for bill" D ERROR S IBQ=1  . S IBER=$P(IB(0),U,1)\_" "\_IBNM\_", "\_$P(IBINS,U,1)\_", will not Reimburse" D ERROR S IBQ=1  ;  ; If processing in silent mode, skip over the following reads  I $G(IBSILENT) G SKIP  ;  W !!  S DIR("?")="Enter Yes to "\_$S('$G(IBMRAO):"create a new bill in the bill series for this care. The new bill will be the "\_$P(IBNM," ")\_" bill ",1:"enter the MRA information and change the payer to the "\_$P($P(IBNM,"-")," ")\_" payer ")  S DIR("?")=DIR("?")\_$S('IBMRA:"with the "\_IBNM\_" responsible for payment.",1:"and will request an MRA from MEDICARE.")  S DIR(0)="YO",DIR("A")=$S('$G(IBMRAO):"Copy "\_$P(IB(0),U,1)\_" for a bill to the ",1:"Change payer on bill "\_$P(IB(0),U,1)\_" to ")\_IBNM\_", "\_$P(IBINS,U,1) D ^DIR K DIR I Y'=1 S IBSECHK=1 G ASK1  ;  W !  S IBQ=0  I '$G(IBMRAO) D G:IBQ ASK1  . N Z  . S DIR("?")="Enter the amount of the payment from the payer of the "\_IBNMOLD\_" bill."  . S DIR("?")=DIR("?")\_" This will be added to the new bill as a prior payment and subtracted from the charges due for the new bill."  . S DIR("A")="Prior Payment from "\_$P(IB(0),U,1)\_" "\_IBNMOLD\_" Payer, "\_$P(IBINSOLD,U,1)\_": "  . S Z=$$EOBTOT^IBCEU1(IBIFN,$$COBN^IBCEF(IBIFN))  . S:Z DIR("B")=Z  . S DIR(0)="NOA^0:99999999:2"  . D ^DIR K DIR I Y=""!$D(DIRUT) S IBQ=1  . K IBCOB  . S IBCOB("U2",IBCOBN+2)=Y  . Q  ;  SKIP ; Jump here if skipping over the preceeding reads  ;  ; If payer is Medicare (WNR) update payer sequence and quit  I IBMRAO!($G(IBSTSM)=1) D I $G(IBSTSM)'=1 G END  . N IBPRTOT,IBTOTCHG,IBPTRESP  . S IBTOTCHG=0  . ;  . ; Get Total Charges from BILLS/CLAIMS (#399) file  . S IBTOTCHG=$P($G(^DGCR(399,IBIFN,"U1")),U,1)  . ; Calculate Patient Responsibility for Bill  . ; IB\*2.0\*447 If claim's type of plan has effective date multiple, use those calculations  . ;S IBPTRESP=$$PREOBTOT^IBCEU0(IBIFN,$G(IBSTSM))  . ; Calculate Patient Primary/Secondary Prior Payment (field 218 or 219 of File 399)  . ; These fields are stored in DGCR(399,IBIFN,"U2") pieces 4 and 5 respectively  . ; Calculate: Prior Payment= Total Submitted Charges - Patient Responsibility  . S:$G(IBSTSM)'=1 IBPTRESP=$S($$MSEDT^IBCEMU4(IBIFN)'="":$$MSPRE^IBCEMU4(IBIFN),1:$$PREOBTOT^IBCEU0(IBIFN,$G(IBSTSM))),IBPRTOT=IBTOTCHG-IBPTRESP  . S:$G(IBSTSM)=1 IBPRTOT=$$EOBTOT^IBCEU1(IBIFN,$$COBN^IBCEF(IBIFN)) ;Pat Resp for non-medicare  . I IBPRTOT<0 S IBPRTOT=0 ; don't allow negative prior payment or offset  . S IBCOB("U2",IBCOBN+2)=IBPRTOT  . ; IB\*2.0\*547 don't change status back to 1.5 if auto-creating secondary or tertiary in silent mode  . ; D:$G(IBSTSM)'=1 COBCHG^IBCCC2(IBIFN,IBMRAIO,.IBCOB)  . ; D STAT^IBCEMU2(IBIFN,1.5,1) ; mra eob status update  . I $G(IBSTSM)'=1 D COBCHG^IBCCC2(IBIFN,IBMRAIO,.IBCOB),STAT^IBCEMU2(IBIFN,1.5,1) ; mra eob status update  . I $G(IBSILENT) S IBERRMSG=""  . Q  ;  ; We should NOT get to here in silent mode .... just in case  I $G(IBSILENT),$G(IBSTSM)'=1 G END ; currently only MCRWNR in silent mode  ;  ; Payer is not Medicare (WNR) - Perform additional steps  S IBCOB(0,15)=""  S IBCOB(0,21)=$S(IBCOBN=2:"S",IBCOBN=3:"T",1:"")  I IBCOB(0,21)="" G END  S IBCOB("M1",IBCOBN+3)=IBIFN  S IBIDS(.15)=IBIFN  D KVAR  G STEP2^IBCCC  ;  END ;  Q  ;  ;  ASK1 ; If entering thru EDI COB processing, don't ask for new bill, quit  I $G(IBCBASK) G EXIT  G ASK  ;  ERROR ; Display/Save error message  I '$G(IBSILENT) W !,IBER,!  E S IBERRMSG=IBER  S IBER=""  I $D(IBSECHK) S IBSECHK=1  Q  ;  EXIT K IBCAN,IBCOB,IBU  KVAR K IBX,IBY,IBI,IBIFN,DFN,IBDT,IB,IBCOBN,IBNMOLD,IBINSOLD,IBNM,IBINSN,IBINS,IBER,DIR,IBAC,IBAC1,IBV,X,Y,IBDATA,IBT,IBND0,DIRUT,IBCOBIL,IBMRA,IBMRAI,IBMRAO,IBMRAIO,IBCBCOPY  K ^UTILITY($J)  Q  ;  DSPRB(IBIFN) ; display related bills  ;  D DSPRB^IBCCCB0(IBIFN) ; Code moved for size too big  Q  ;  ; ==============  ;  ; Copy a bill for Reasonable Charges without cancelling it, update certain fields  ;  ; there is always both inpt inst (created first) and prof charges, always need both bills  ; there may be both outpt inst (created first) and prof charges, may not need both bills  ; if billing by episode rather than by day (current standard) then may need multiple prof bills per day  ;  ; Inst bills are copied to create prof Bills automatically  ; Subsequent prof bills may be created if the user wants them  ;  ; Only the first bill in the COB series of bills should be copied for the next prof bill  ; The primary inst bill should be copied to get the secondary inst bill  ; The primary prof bill should be copied to get the secondary prof bill  ;  CTCOPY(IBIFN,IBMRA) ; based on the type of bill, copy without cancelling  ; IBMRA = 1 if an MRA bill and copy for prof components is desired  ;  D CTCOPY^IBCCCB0(IBIFN,$G(IBMRA)) ;Moved due to routine size  Q  ; | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCCCB ;ALB/ARH - COPY BILL FOR COB ;2/13/06 10:46am  ;;2.0;INTEGRATED BILLING;\*\*80,106,51,151,137,182,155,323,436,432,447,547\*\*;21-MAR-94;Build 119  ;;Per VA Directive 6402, this routine should not be modified.  ;  ; Copy bill for COB w/out cancelling, update some flds  ; Primary->Secondary->Tertiary  ASK ;  S IBCBCOPY=1 ; flag that copy function entered thru Copy COB option  ;  D KVAR S IBCAN=2,IBU="UNSPECIFIED"  ;  S IBX=$$PB^IBJTU2 S:+IBX=2 IBIFN=$P(IBX,U,2) I +IBX=1 S DFN=$P(IBX,U,2),IBV=1,IBAC=5 D DATE^IBCB  I '$G(IBIFN) G EXIT  ;  ; IB\*2.0\*432 Restrict access to only allow claims that are NOT on the new CBW Worklist  I $P($G(^DGCR(399,IBIFN,"S1")),U,7)=1,$G(IBMRANOT)'=1 D G ASK  . W !!?4,"This bill appears on the CBW Management Work List. Please use the"  . W !?4,"'CBW Management Menu' options for all processing related to this bill."  . S IBQUIT=1 ;JRA need to set quit flag after issuing this message  . Q  ; Restrict access to this process for REQUEST MRA bills in 2 Cases:  ; 1. No MRA EOB's on File for bill  I $P($G(^DGCR(399,IBIFN,0)),U,13)=2,'$$CHK^IBCEMU1(IBIFN) D G ASK  . W !!?4,"This bill is in a status of REQUEST MRA and it has No MRA EOB's"  . W !?4,"on file. Access to this bill is restricted."  ;  ; 2. At least one MRA EOB appears on the MRA management worklist  I $P($G(^DGCR(399,IBIFN,0)),U,13)=2,$$MRAWL^IBCEMU2(IBIFN) D G ASK  . W !!?4,"This bill is in a status of REQUEST MRA and it does appear on the"  . W !?4,"MRA Management Work List. Please use the 'MRA Management Menu' options"  . W !?4,"for all processing related to this bill."  . Q  ;  ; If MRA is Activated and bill is in Entered/Not Reviewed status and current insurance Co. is WNR -->  ; ask if user wants to continue  I $$EDIACTV^IBCEF4(2),$P($G(^DGCR(399,IBIFN,0)),U,13)=1,$$MCRWNR^IBEFUNC(+$$CURR^IBCEF2(IBIFN)) D I 'Y G ASK  . W !!?4,"This bill is in a status of ENTERED/NOT REVIEWED and current payer is "  . W !?4,"MEDICARE (WNR). No MRA has been requested for this bill."  . S DIR(0)="YA",DIR("B")="NO",DIR("A")=" Are you sure you want to continue to process this bill?: "  . D ^DIR K DIR  ;  ; Display related bills  D DSPRB^IBCCCB0(IBIFN)  ;  CHKB ; Entrypoint-COB processing via EDI's COB Mgmt  ; Ask if final EOB was received for previous bill  I '$$FINALEOB^IBCCCB0(IBIFN) S IBSECHK=1  I $G(IBSECHK)=1,$$MCRONBIL^IBEFUNC(IBIFN) G EXIT  ;  ; Warn if previous bill not at least authorized  I '$$MCRONBIL^IBEFUNC(IBIFN) I '$$COBOK^IBCCCB0(IBIFN) G EXIT  ;  CHKB1 ; Entry point for Automatic/Silent COB Processing.  ; No writes or reads can occur from this point forward if variable  ; IBSILENT=1. Any and all error messages should be processed with  ; the ERROR procedure below.  ;  S IBX=$G(^DGCR(399,+IBIFN,0)),DFN=$P(IBX,U,2),IBDT=$P(IBX,U,3)\1,IBER=""  I IBCAN>1 D NOPTF^IBCB2 I 'IBAC1 D NOPTF1^IBCB2 G ASK1  ;  F IBI=0,"S","U1","M","MP","M1" S IB(IBI)=$G(^DGCR(399,IBIFN,IBI))  I IB(0)="" S IBER="Invalid Bill Number" D ERROR G ASK1  ;  ; check to see if the bill has been cancelled  I $P(IB("S"),U,16),$P(IB("S"),U,17) D G ASK1  . N WHO  . S IBER="This bill was cancelled on "  . S IBER=IBER\_$$FMTE^XLFDT($P(IB("S"),U,17),"1Z")\_" by "  . S WHO="UNSPECIFIED"  . I $P(IB("S"),U,18) S WHO=$P($G(^VA(200,$P(IB("S"),U,18),0)),U,1)  . S IBER=IBER\_WHO\_"."  . D ERROR  . Q  ;  S IBCOB=$$COB^IBCEF(IBIFN),IBCOBN=$TR(IBCOB,"PSTA","12")  S IBMRAIO=+$$CURR^IBCEF2(IBIFN),IBMRAO=$$MCRWNR^IBEFUNC(IBMRAIO)  S IBNMOLD=$S(IBCOB="P":"Primary",IBCOB="S":"Secondary",IBCOB="T":"Tertiary",IBCOB="A":"Patient",1:"")\_$S(IBMRAO:"-MRA Only",1:"")  S IBINSOLD=$G(^DIC(36,$S(IB("MP"):+IB("MP"),IBMRAO:IBMRAIO,1:0),0))  ;  NEXTP ; If current bill=MEDICARE WNR and valid 'next payer', use same  ; bill for new payer  ; If next valid 'payer' is ins co or MEDICARE WNR, create new bill  S IBCOBN=IBCOBN+1,IBNM=$S(IBCOBN=2:"Secondary Payer",IBCOBN=3:"Tertiary Payer",1:"")  ;  I IBNM="" S IBER=$P(IB(0),U,1)\_" is a "\_IBNMOLD\_" bill, there is no next bill in the series." D ERROR G ASK1  ;  S IBX=+$P(IB("M1"),U,(4+IBCOBN)),IBY=$G(^DGCR(399,+IBX,0)),IBCOBIL(+IBIFN)=""  ;  I $P(IBY,U,13)=7 S IBER="The "\_$P(IBNM," ",1)\_" bill "\_$P(IBY,U,1)\_" has been cancelled." D ERROR S IBX=""  ;  I +IBX,$D(IBCOBIL(+IBX)) S IBER="Next bill in series can not be determined." D ERROR G ASK1  I +IBX S IBER=$P(IBNM," ",1)\_" bill already defined for this series: "\_$P(IBY,U,1) D ERROR S IBIFN=IBX G ASK1  ;  S IBINSN=$P(IB("M"),U,IBCOBN) I 'IBINSN S IBER="There is no "\_IBNM\_" for "\_$P(IB(0),U,1)\_"." D ERROR G ASK1  S IBINS=$G(^DIC(36,+IBINSN,0)) I IBINS="" S IBER="The "\_IBNM\_" for "\_$P(IB(0),U,1)\_" is not a valid Insurance Co." D ERROR G ASK1  ;  S IBMRA=0  I $P(IBINS,U,2)="N" S IBQ=0 D G:IBQ NEXTP  . I $$MCRWNR^IBEFUNC(IBINSN) D Q  .. ; Check if a valid tert ins if MCR WNR secondary  .. I IBCOBN'>2 D  ... N Z  ... S Z=+$P(IB("M"),U,IBCOBN+1)  ... I Z,$D(^DIC(36,Z,0)),$P(^(0),U,2)'="N" S IBMRA=1,IBNM=$P(IBNM," ")\_"-MRA.Only"  .. I 'IBMRA S IBER="MEDICARE will not reimburse and no further valid insurance for bill" D ERROR S IBQ=1  . S IBER=$P(IB(0),U,1)\_" "\_IBNM\_", "\_$P(IBINS,U,1)\_", will not Reimburse" D ERROR S IBQ=1  ;  ; If processing in silent mode, skip over the following reads  I $G(IBSILENT) G SKIP  ;  W !!  S DIR("?")="Enter Yes to "\_$S('$G(IBMRAO):"create a new bill in the bill series for this care. The new bill will be the "\_$P(IBNM," ")\_" bill ",1:"enter the MRA information and change the payer to the "\_$P($P(IBNM,"-")," ")\_" payer ")  S DIR("?")=DIR("?")\_$S('IBMRA:"with the "\_IBNM\_" responsible for payment.",1:"and will request an MRA from MEDICARE.")  S DIR(0)="YO",DIR("A")=$S('$G(IBMRAO):"Copy "\_$P(IB(0),U,1)\_" for a bill to the ",1:"Change payer on bill "\_$P(IB(0),U,1)\_" to ")\_IBNM\_", "\_$P(IBINS,U,1) D ^DIR K DIR I Y'=1 S IBSECHK=1 G ASK1  ;  W !  S IBQ=0  I '$G(IBMRAO) D G:IBQ ASK1  . N Z  . S DIR("?")="Enter the amount of the payment from the payer of the "\_IBNMOLD\_" bill."  . S DIR("?")=DIR("?")\_" This will be added to the new bill as a prior payment and subtracted from the charges due for the new bill."  . S DIR("A")="Prior Payment from "\_$P(IB(0),U,1)\_" "\_IBNMOLD\_" Payer, "\_$P(IBINSOLD,U,1)\_": "  . S Z=$$EOBTOT^IBCEU1(IBIFN,$$COBN^IBCEF(IBIFN))  . S:Z DIR("B")=Z  . S DIR(0)="NOA^0:99999999:2"  . D ^DIR K DIR I Y=""!$D(DIRUT) S IBQ=1  . K IBCOB  . S IBCOB("U2",IBCOBN+2)=Y  . Q  ;  SKIP ; Jump here if skipping over the preceeding reads  ;  ; If payer is Medicare (WNR) update payer sequence and quit  I IBMRAO!($G(IBSTSM)=1) D I $G(IBSTSM)'=1 G END  . N IBPRTOT,IBTOTCHG,IBPTRESP  . S IBTOTCHG=0  . ;  . ; Get Total Charges from BILLS/CLAIMS (#399) file  . S IBTOTCHG=$P($G(^DGCR(399,IBIFN,"U1")),U,1)  . ; Calculate Patient Responsibility for Bill  . ; IB\*2.0\*447 If claim's type of plan has effective date multiple, use those calculations  . ;S IBPTRESP=$$PREOBTOT^IBCEU0(IBIFN,$G(IBSTSM))  . ; Calculate Patient Primary/Secondary Prior Payment (field 218 or 219 of File 399)  . ; These fields are stored in DGCR(399,IBIFN,"U2") pieces 4 and 5 respectively  . ; Calculate: Prior Payment= Total Submitted Charges - Patient Responsibility  . S:$G(IBSTSM)'=1 IBPTRESP=$S($$MSEDT^IBCEMU4(IBIFN)'="":$$MSPRE^IBCEMU4(IBIFN),1:$$PREOBTOT^IBCEU0(IBIFN,$G(IBSTSM))),IBPRTOT=IBTOTCHG-IBPTRESP  . S:$G(IBSTSM)=1 IBPRTOT=$$EOBTOT^IBCEU1(IBIFN,$$COBN^IBCEF(IBIFN)) ;Pat Resp for non-medicare  . I IBPRTOT<0 S IBPRTOT=0 ; don't allow negative prior payment or offset  . S IBCOB("U2",IBCOBN+2)=IBPRTOT  . ; IB\*2.0\*547 don't change status back to 1.5 if auto-creating secondary or tertiary in silent mode  . ; D:$G(IBSTSM)'=1 COBCHG^IBCCC2(IBIFN,IBMRAIO,.IBCOB)  . ; D STAT^IBCEMU2(IBIFN,1.5,1) ; mra eob status update  . I $G(IBSTSM)'=1 D COBCHG^IBCCC2(IBIFN,IBMRAIO,.IBCOB),STAT^IBCEMU2(IBIFN,1.5,1) ; mra eob status update  . I $G(IBSILENT) S IBERRMSG=""  . Q  ;  ; We should NOT get to here in silent mode .... just in case  I $G(IBSILENT),$G(IBSTSM)'=1 G END ; currently only MCRWNR in silent mode  ;  ; Payer is not Medicare (WNR) - Perform additional steps  S IBCOB(0,15)=""  S IBCOB(0,21)=$S(IBCOBN=2:"S",IBCOBN=3:"T",1:"")  I IBCOB(0,21)="" G END  S IBCOB("M1",IBCOBN+3)=IBIFN  S IBIDS(.15)=IBIFN  D KVAR  G STEP2^IBCCC  ;  END ;  Q  ;  ;  ASK1 ; If entering thru EDI COB processing, don't ask for new bill, quit  I $G(IBCBASK) G EXIT  G ASK  ;  ERROR ; Display/Save error message  I '$G(IBSILENT) W !,IBER,!  E S IBERRMSG=IBER  S IBER=""  I $D(IBSECHK) S IBSECHK=1  Q  ;  EXIT K IBCAN,IBCOB,IBU  KVAR K IBX,IBY,IBI,IBIFN,DFN,IBDT,IB,IBCOBN,IBNMOLD,IBINSOLD,IBNM,IBINSN,IBINS,IBER,DIR,IBAC,IBAC1,IBV,X,Y,IBDATA,IBT,IBND0,DIRUT,IBCOBIL,IBMRA,IBMRAI,IBMRAO,IBMRAIO,IBCBCOPY  K ^UTILITY($J)  Q  ;  DSPRB(IBIFN) ; display related bills  ;  D DSPRB^IBCCCB0(IBIFN) ; Code moved for size too big  Q  ;  ; ==============  ;  ; Copy a bill for Reasonable Charges without cancelling it, update certain fields  ;  ; there is always both inpt inst (created first) and prof charges, always need both bills  ; there may be both outpt inst (created first) and prof charges, may not need both bills  ; if billing by episode rather than by day (current standard) then may need multiple prof bills per day  ;  ; Inst bills are copied to create prof Bills automatically  ; Subsequent prof bills may be created if the user wants them  ;  ; Only the first bill in the COB series of bills should be copied for the next prof bill  ; The primary inst bill should be copied to get the secondary inst bill  ; The primary prof bill should be copied to get the secondary prof bill  ;  CTCOPY(IBIFN,IBMRA) ; based on the type of bill, copy without cancelling  ; IBMRA = 1 if an MRA bill and copy for prof components is desired  ;  D CTCOPY^IBCCCB0(IBIFN,$G(IBMRA)) ;Moved due to routine size  Q  ; | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCECOB2 | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** |  | | | | | | | | |
| **Related Options** | None | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  |  | | | | |  | | | |
| **Data Dictionary (DD) References** |  | | | | | | | | |
| **Related Protocols** |  | | | | | | | | |
| **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** | | | | | | | | | |
| IBCECOB2 ;ALB/CXW - IB COB MANAGEMENT SCREEN ;16-JUN-1999  ;;2.0;INTEGRATED BILLING;\*\*137,155,433,432,447,488,516\*\*;21-MAR-1994;Build 123  ;;Per VA Directive 6402, this routine should not be modified.  ;  EDI ;history detail display  N IBIFN,IBDA  D SEL(.IBDA,1)  S IBDA=+$O(IBDA(0)),IBIFN=+$G(IBDA(IBDA))  D EDI1(IBIFN)  S VALMBCK="R"  Q  ;  EDI1(IBIFN) ;  N DFN  Q:'IBIFN  S DFN=$P($G(^DGCR(399,IBIFN,0)),U,2)  D EN^VALM("IBJT EDI STATUS")  K:$D(IBFASTXT) IBFASTXT  Q  ;  EDI2(IBIFN) ;  N DFN  Q:'IBIFN  S DFN=$P($G(^DGCR(399,IBIFN,0)),U,2)  D EN^VALM("IBJT EDI STATUS ALONE")  K:$D(IBFASTXT) IBFASTXT  Q  ;  CSA ;claims status awaiting resolution  N IBDAX  D EN^IBCECSA  I $D(IBFASTXT) K IBFASTXT  S VALMBCK="R"  Q  ;  RVEOB ;Review EOB  D FULL^VALM1 W !  N IBDA,IBIFN,IBCMT,IBSEL  D SEL(.IBDA,1)  S IBSEL=+$O(IBDA(0))  S IBDA=$G(IBDA(IBSEL))  S IBIFN=$P(IBDA,U),IBDA=$P(IBDA,U,3)  I 'IBIFN G VEOBQ  S IBCMT=$G(^TMP("IBCECOB1",$J,IBSEL))  I IBCMT'="" D EN^VALM("IBCEM MRA REVIEW")  VEOBQ K ^TMP("IBCECOC",$J)  S VALMBCK="R"  Q  ;  TPJI ;Third Party joint Inquiry  N IBDA,IBIFN  D SEL(.IBDA,1)  S IBDA=+$O(IBDA(0)),IBIFN=+$G(IBDA(IBDA))  I IBDA="" G TPJIQ  D TPJI1(IBIFN)  TPJIQ S VALMBCK="R"  Q  ;  TPJI1(IBIFN) ;  N DFN,IBNOTPJI  Q:'IBIFN  S DFN=$P($G(^DGCR(399,IBIFN,0)),U,2),IBNOTPJI=1  D EN^VALM("IBJT CLAIM INFO")  K:$D(IBFASTXT) IBFASTXT  Q  ;  PBILL ;Print bill  N IBIFN,IBDA,IBRESUB  D SEL(.IBDA,1)  S IBDA=$O(IBDA(0)),IBIFN=+$G(IBDA(+IBDA))  I IBDA="" G PBOUT  S IBRESUB=$$RESUB^IBCECSA4(IBIFN,1,"P")  I IBRESUB'>0 W !,\*7,"This is not a transmittable bill or review not needed" D PAUSE^VALM1 G PBOUT  I IBRESUB=2 D G PBOUT  . N IB364  . S IB364=+$P($G(IBDA(IBDA)),U,2)  . D PRINT1^IBCEM03(IBIFN,.IBDA,IB364)  D PBILL1(IBIFN)  PBOUT S VALMBCK="R"  Q  ;  PMRA ;Print MRA  N IBIFN,IBDA,IBDAX  D SEL(.IBDA,1)  S IBDA=$O(IBDA(0)),IBIFN=+$G(IBDA(+IBDA)),IBDAX=$P(IBDA(+IBDA),U,3)  G:'IBIFN PRMQ  I '$G(IBMRANOT),$D(^IBM(361.1,IBDAX,"ERR")),'$$WARNMSE G PRMQ ; Claim contains Message Storage Errors  D MRA^IBCEMRAA(.IBIFN)  PRMQ S VALMBCK="R"  Q  PBILL1(IBIFN) ;  N IBAC1,IBAC,DFN  Q:'IBIFN  S DFN=$P($G(^DGCR(399,IBIFN,0)),U,2)  S IBAC=4,IBAC1=1  D 4^IBCB1  D FULL^VALM1,PAUSE^VALM1  Q  ;  CANCEL ;Cancel bill  ; IBDA(IBDA)=IBIFN^IB364^ien of 361.1^user selection seq^user name~duz#  ;  N IBIFN,IBDA,IB364,IBEOBIFN,X,IBDENCT  ;  ; Check for security key  I '$$KCHK^XUSRB("IB AUTHORIZE") D G CANCELQ  . D FULL^VALM1 S VALMBCK="R"  . W !!?5,"You don't hold the proper security key to access this function."  . W !?5,"The necessary key is IB AUTHORIZE. Please see your manager."  . D PAUSE^VALM1  . Q  ;  D SEL(.IBDA,1)  S IBDA=$O(IBDA(0)),IBIFN=+$G(IBDA(+IBDA)),IB364=$P($G(IBDA(+IBDA)),U,2)  S IBEOBIFN=$P($G(IBDA(+IBDA)),U,3)  ;  ; IB\*2.0\*432 - if not mra, only allow cancel of denied claims. If no EOB, check AR status instead  I 'IBEOBIFN,$G(IBMRANOT)=1,$P($$ARSTATA^IBJTU4(IBIFN),U)="COLLECTED/CLOSED" D G CANCELQ  . D FULL^VALM1 S VALMBCK="R"  . W !!,\*7,"You can only cancel denied claims. This claim is in a COLLECTED/CLOSED status"  . W !,"Use Remove Action to remove claim from this worklist."  . D PAUSE^VALM1  . Q  ;  ; IB\*2.0\*432 - if not mra, only allow cancel of claims with multiple EOBS if none have processed.  I $G(IBMRANOT)=1,'$$DENCHK(IBIFN,.IBDENCT),$G(IBDENCT)>1 D G CANCELQ  . D FULL^VALM1 S VALMBCK="R"  . W !!,\*7,"Multiple EOBs exist for this claim and at least one has EOB status of PROCESSED."  . W !,"Use Remove Action to remove claim from this worklist."  . D PAUSE^VALM1  . Q  ;  ; IB\*2.0\*432 - if not mra, only allow cancel of denied claims  I IBEOBIFN,$G(IBMRANOT)=1,$P($G(^IBM(361.1,IBEOBIFN,0)),U,13)'=2 D G CANCELQ  . D FULL^VALM1 S VALMBCK="R"  . W !!?5,\*7,"You can only cancel denied claims."  . D PAUSE^VALM1  . Q  ;  I IBDA D  . I '$$LOCK^IBCEU0(361.1,IBEOBIFN) Q  . D CANCEL^IBCEM3(.IBDA,IBIFN,IB364)  . D UNLOCK^IBCEU0(361.1,IBEOBIFN)  S VALMBCK="R"  ;  ; for non-MRA claims cancelled from worklist, set field 38  I $G(IBMRANOT)=1,$P($G(^DGCR(399,IBIFN,0)),U,13)=7 S X=$$WLRMVF^IBCECOB1($S($G(IBIFN)'="":IBIFN,1:+$G(IBDA(IBDA))),"CA")  I $G(IBDA)'="" D BLD^IBCECOB1  CANCELQ Q  ;  CRD ; Correct Rejected/Denied claim protocol action  N IBCNCRD  S IBCNCRD=1  CLONE ; 'Copy/cancel bill' protocol action  N IBDA,IBQ,IBEOBIFN,IBKEY,X,IBDENCT  ;  ; Check for security key  ;IB\*2.0\*516/TAZ - Remove check for IB CLON  ;I '$$KCHK^XUSRB("IB AUTHORIZE") D G CLONEQ  ;S IBKEY=$S($G(IBCNCRD)=1:"IB AUTHORIZE",1:"IB CLON")  S IBKEY="IB AUTHORIZE"  I '$$KCHK^XUSRB(IBKEY) D G CLONEQ  . D FULL^VALM1 S VALMBCK="R"  . ;W !!?5,"You don't hold the proper security key to access this function."  . ;W !?5,"The necessary key is IB AUTHORIZE. Please see your manager."  . W !!?5,"You must hold the "\_IBKEY\_" security key to access this function."  . W !?5,"Please see your manager."  . D PAUSE^VALM1  . Q  ;  D SEL(.IBDA,1)  S IBDA=$O(IBDA(""))  I IBDA="" G CLONEQ  ;  ; IB\*2.0\*432 - if not mra, only allow cancel of claims with multiple EOBS if none have processed.  I $G(IBMRANOT)=1,'$$DENCHK(+IBDA(IBDA),.IBDENCT),$G(IBDENCT)>1 D G CANCELQ  . D FULL^VALM1 S VALMBCK="R"  . W !!,\*7,"Multiple EOBs exist for this claim and at least one has EOB status of PROCESSED."  . W !,"Use Remove Action to remove claim from this worklist."  . D PAUSE^VALM1  . Q  ;  S IBEOBIFN=$P($G(IBDA(+IBDA)),U,3)  I '$$LOCK^IBCEU0(361.1,IBEOBIFN) G CLONEQ  D COPYCLON(+$G(IBDA(IBDA)),$P($G(IBDA(+IBDA)),U,2),.IBQ)  D UNLOCK^IBCEU0(361.1,IBEOBIFN)  ;  ; for non-MRA claims cloned or corrected from worklist, set field 38  I $G(IBMRANOT)=1,$G(IBQ)'="" S X=$$WLRMVF^IBCECOB1(+$G(IBDA(IBDA)),$S($G(IBCNCRD)=1:"CR",1:"CL"))  ;  CLONEQ ;  S VALMBCK="R"  D:$G(IBQ)'="" BLD^IBCECOB1  Q  ;  COPYCLON(IBIFN,IB364,IBQ) ; Generic entry point for clone a bill from EDI processing  ; IBIFN = original bill ien  ; IB364 = the ien of the transmission bill entry in file 364  ; IBQ = If bill is not cancelled, this is returned as null  ; - pass by reference -  ;  N IBQUIT,IBCCCC,IBHV,Y,IBCAN,IBCE,IBDA,IBCNCOPY  ;I '$$CANCKS^IBCEM3("CC",IBIFN) S IBQ="" G CCQ  I $G(IBCNCRD)'=1,'$$CANCKS^IBCEM3("CC",IBIFN) S IBQ="" G CCQ  ;  ;S IBCAN=2,IBCE("EDI")=1,Y=IBIFN,IBCCCC=0,IBHV("IBIFN")=IBIFN,IBHV("IBIFN1")="",IBCNCOPY=1  S IBCAN=2,IBCE("EDI")=1,Y=IBIFN,IBCCCC=0,IBHV("IBIFN")=IBIFN,IBHV("IBIFN1")=""  I $G(IBCNCRD)'=1 S IBCNCOPY=1 D ^IBCCC  I $G(IBCNCRD)=1 D CRD^IBCCC  ;D ^IBCCC  S IBIFN=IBHV("IBIFN")  K IBCE("EDI") S IBQ=1  I $P($G(^DGCR(399,IBIFN,0)),U,13)'=7 S IBQ=""  I IBHV("IBIFN1") D  . N IBU  . S IBU="R"  . S IBNIEN=+IBHV("IBIFN1")  . I "23"'[$P($G(^DGCR(399,+IBHV("IBIFN1"),0)),U,13) D  .. W:'$G(IBCEAUTO) !,\*7,"Please note: the new bill was not AUTHORIZED.",!,"It can only be accessed now via the normal, non-EDI functions.",!,"Status of new bill is ",$$EXPAND^IBTRE(399,.13,$P(^DGCR(399,IBHV("IBIFN1"),0),U,13)) S IBU="C"  . D UPDEDI^IBCEM(IB364,IBU)  ;  I '$G(IBCEAUTO) D PAUSE^VALM1  CCQ Q  ;  PRO ; Copy for secondary/tertiary bill  N VALMY,IBDA,Z,IBIFN,IBIFNH,IB364,IBCE,IBNCN  ;I '$P($G(^IBE(350.9,1,8)),U,12) D G PROQ  I '$P($G(^IBE(350.9,1,8)),U,12),$G(IBMRANOT)'=1 D G PROQ  . D FULL^VALM1  . W !!?5,"MRA's may not be processed at this time."  . W !?5,"The IB site parameter ""Allow MRA Processing?"" is set to NO."  . D PAUSE^VALM1  . Q  D SEL(.IBDA,1)  S Z=$O(IBDA(0)),Z=$G(IBDA(+Z)) G:'Z PROQ  S IBIFN=$P(Z,U),IB364=$P(Z,U,2),IBDA=$P(Z,U,3),IBIFNH=IBIFN  I 'IBIFN G PROQ  I '$G(IBMRANOT),$D(^IBM(361.1,IBDA,"ERR")),'$$WARNMSE G PROQ ; Claim contains Message Storage Errors  I '$$LOCK^IBCEU0(361.1,IBDA) G PROQ  D COBCOPY(IBIFN,IB364,2,IBDA,"BLD^IBCECOB1",.IBNCN)  D UNLOCK^IBCEU0(361.1,IBDA)  ;  ; for non-MRA claims copied from work list, set field 38  I $G(IBMRANOT)=1,$G(IBNCN)'="",($G(IBNCN)'=$G(IBIFN)) D  .S X=$$WLRMVF^IBCECOB1($G(IBIFN),"PC")  .;I $P($G(^DGCR(399,+IBNCN,"S")),U,9)'=1 D  .;.W:'$G(IBCEAUTO) !,\*7,"Please note: the new bill was not AUTHORIZED.",!,"It can only be accessed now via the normal, non-EDI functions.",!,"Status of new bill is ",$$EXPAND^IBTRE(399,.13,$P(^DGCR(399,IBNCN,0),U,13))  .;.D PAUSE^VALM1  .D:$G(IBMRANOT)=1 BLD^IBCECOB1  .Q  ;  PROQ S VALMBCK="R"  Q  ;  COBCOPY(IBIFN,IB364,IBFROM,IBIEN,IBBLD,IBNCN) ; Generic entry point for EDI COB copy  ; IBIFN = original bill ien  ; IB364 = the ien of the transmission bill entry in file 364  ; IBFROM = 1 if called from CSA, 2 if called from COB/EOB processing  ; IBIEN = entry in 361 (IBFROM=1) or 361.1 (IBFROM=2) being processed  ; IBBLD = the name of the entrypoint that will rebuild the display  ; IBNCN = by reference, returns the new claim ien if user completed the Copy process  ;  N IBCBASK,IBCBCOPY,IBCAN,IBIFNH,IBNSTAT,IBOSTAT,IBPRCOB,IBSECHK,IBLMVAR,IBAC,IBMRAIEN,IBDA,IBAUTO  N IBCOB,IBCOBIL,IBCOBN,IBINS,IBINSN,IBINSOLD,IBMRAIO,IBMRAO,IBNMOLD,IBQUIT  S (IBCBASK,IBCBCOPY,IBCAN,IBAUTO)=1,(IBPRCOB,IBSECHK)=0,(IBMRAIEN,IBDA)=IBIEN  I $G(IBMRANOT)'=1,'IB364!'IBIFN W !,"Transmission record is missing for this bill" D PAUSE^VALM1 G COBCOPX  ;  S IBIFNH=IBIFN  I IBFROM=2 S IBPRCOB=1  ; IB\*2.0\*447 Check PR to include excess and percentages where applicable  ;I $S($G(IBMRANOT)=1:$$TOT(IBIFN)'>0,1:$$PREOBTOT^IBCEU0(IBIFN,$G(IBMRANOT))'>0) D G COBCOPX  I $$TOT(IBIFN,$G(IBMRANOT))'>0 D G COBCOPX  . D FULL^VALM1  . W !!?5,"There is no "\_$S($G(IBMRANOT)=1:"balance remaining",1:"patient responsibility and/or excess charges")\_" for this claim."  . W !?5,"This claim may not be processed."  . D PAUSE^VALM1  . Q  ;  I $G(IBDA)'="",$P($G(^IBM(361.1,IBDA,0)),U,16)="1.5" D G COBCOPX  . W !!,"This claim has already been processed as a sec/tert claim."  . W !,"You will need to complete the authorization process for this claim."  . D PAUSE^VALM1  . D AUTH  . Q  ;  ; If multiple EOBs and one is processed, make sure collected closed.  I $G(IBMRANOT),$$CCCHK(IBIFN)<0 D G COBCOPX  . W !,"Multiple EOBs exist for this claim and at least one has EOB status of PROCESSED."  . W !,"Claim cannot be sent to next payer until AR status is Collected/Closed."  . D PAUSE^VALM1  . Q  ;  ; Get out if no next payer  I '$P($G(^DGCR(399,IBIFN,"I"\_($$COBN^IBCEF(IBIFN)+1))),U,1) D G COBCOPX  . W !,"There is no next payer for this bill"  . D PAUSE^VALM1  . Q  ;  D DSPRB^IBCCCB0(IBIFN) ; display related bills  S IBCE("EDI")=1  D CHKB^IBCCCB ; process COB, create secondary bill  S IBNCN=$G(IBCE("EDI","NEW")) ; get new claim ien  S IBIFN=IBIFNH  I IBSECHK G COBCOPX  ;  ; if user came from CBW, no need to view and authorize a 2nd time (already happens in IBCCCB)  Q:$G(IBMRANOT)=1  S IBV=1 D VIEW^IBCB2 ; display billing screens  D AUTH ; authorize bill  COBCOPX ;  Q  ;  AUTH ; procedure to authorize the claim and refresh the screen  K ^UTILITY($J) S IBAC=1,IBQUIT=0 D 3^IBCB1  I '$D(IOUON)!'$D(IORVON) D ENS^%ZISS  I $P($G(^IBM(361.1,IBMRAIEN,0)),U,16)=3 D UPDEDI^IBCEM(IB364,"Z")  I $G(IBBLD)'="" D @IBBLD  D PAUSE^VALM1  AUTHX ;  Q  ;  RES ;Resubmit bill by print  N IBDA,IBIFN,IB364  D SEL(.IBDA,1)  S IBDA=+$O(IBDA(0)),IBIFN=+$G(IBDA(+IBDA)),IB364=+$P($G(IBDA(IBDA)),U,2)  I 'IBIFN G RESQ  D PRINT1^IBCEM03(IBIFN,.IBDA,IB364)  D PAUSE^VALM1  I $G(IBDA)'="" D BLD^IBCECOB1  RESQ S VALMBCK="R"  Q  ;  EBI ;View an unauthorized transmitted bill  N IBFLG,IBDA,IBIFN,IB364,DFN  K ^TMP($J,"IBBILL")  D FULL^VALM1  ;  D SEL(.IBDA,1)  S IBDA=+$O(IBDA(""))  S IBIFN=+$G(IBDA(IBDA)),IB364=+$P($G(IBDA(IBDA)),U,2),DFN=$P($G(^DGCR(399,IBIFN,0)),U,2)  G:'IBIFN EDITQ  S IBV=1 D VIEW^IBCB2  I '$D(IOUON)!'$D(IORVON) D ENS^%ZISS  D BLD^IBCECOB1  EDITQ S VALMBCK="R"  Q  ;  SEL(IBDA,ONE) ; Select entry(s) from list  ; IBDA = array returned if selections made  ; IBDA(n)=ien of bill selected (file 399)  ; ONE = if set to 1, only one selection can be made at a time  N IB  K IBDA  D FULL^VALM1  D EN^VALM2($G(XQORNOD(0)),$S('$G(ONE):"",1:"S"))  S IBDA=0 F S IBDA=$O(VALMY(IBDA)) Q:'IBDA S IBDA(IBDA)=$P($G(^TMP("IBCECOB",$J,+IBDA)),U,2,6)  Q  ;  EXIT ; Exit out of COB  D FASTEXIT^IBCEFG4  I $G(IBFASTXT)=1 S IBFASTXT=5  Q  ;  TOT(IBIFN,IBMRANOT) ; calculate if any balance remaining on non-MRA claim  ; IBIFN = claim ien  ; IBMRANOT = MRW/CBW flag (1=user came from CBW) added with IB\*2.0\*447  N IBPRTOT,IBBLD,IBCBN,IBU2  I $G(IBMRANOT)'=1 Q $S($$MSEDT^IBCEMU4(IBIFN)'="":$$MSPRE^IBCEMU4(IBIFN),1:$$PREOBTOT^IBCEU0(IBIFN))  ; total up the payer paid amounts, if this is a 2ndary claim, be sure to account for what the primary paid also  S IBU2=$G(^DGCR(399,IBIFN,"U2")),IBCBN=$$COBN^IBCEF(IBIFN),IBPRTOT=$$EOBTOT^IBCEU1(IBIFN,IBCBN)  S:IBPRTOT<0 IBPRTOT=0 ; don't allow negative prior payment or offset  S:IBCBN=2 IBPRTOT=IBPRTOT+$P(IBU2,U,4)  S:IBCBN=3 IBPRTOT=IBPRTOT+$P(IBU2,U,4)+$P(IBU2,U,5)  S:IBPRTOT<0 IBPRTOT=0 ; don't allow negative prior payment or offset  ; Subtract payer paid amount from Total Charges from BILLS/CLAIMS (#399) file, don't allow neg  S IBBLD=$P($G(^DGCR(399,IBIFN,"U1")),U,1)-IBPRTOT  S:IBBLD<0 IBBLD=0  Q IBBLD  ;  CCCHK(IBIFN) ; If there are multiple EOBS on file for this claim, then one of them must be processed and AR status must be collected closed to process.  ; returns 1 if true  ; 0 if there are not multiple EOBs or mulitple EOBs and none are processed (all denials)  ; -1 if false  N IBDA,IBCT,IBPROC,IBARSTAT,IBEOBNDX,IBEOB  S IBCT=0,IBPROC=0  F IBEOBNDX="B","C" D  .S IBDA=0 F S IBDA=$O(^IBM(361.1,IBEOBNDX,IBIFN,IBDA)) Q:'+IBDA D  ..Q:$D(IBEOB(IBDA))  ..Q:$P($G(^IBM(361.1,IBDA,0)),U,4)=1 ; only count EOBs  ..S IBEOB(IBDA)="",IBCT=IBCT+1  ..I $P($G(^IBM(361.1,IBDA,0)),U,13)=1 S IBPROC=1  I IBCT<2 Q 0 ; less than 2 EOBs  I 'IBPROC Q 0 ; no EOBs with status processed  S IBARSTAT=$$ARSTATA^IBJTU4(IBIFN) ; get status of AR  I $P(IBARSTAT,U)="COLLECTED/CLOSED" Q 1  Q -1  ;  DENCHK(IBIFN,IBCT) ; Make sure all EOBs from this claim are denied.  ; Input: IBIFN - IEN to 399  ; IBCT - by reference. Return count of EOBs.  ; Output: returns 1 if there is at least one EOB and that none of the EOBS are processed.  ; otherwise 0  ;  N IBDA,IBPROC,IBEOBNDX,IBEOB  S IBCT=0,IBPROC=0  F IBEOBNDX="B","C" D  .S IBDA=0 F S IBDA=$O(^IBM(361.1,IBEOBNDX,IBIFN,IBDA)) Q:'+IBDA D  ..Q:$D(IBEOB(IBDA))  ..Q:$P($G(^IBM(361.1,IBDA,0)),U,4)=1 ; only count EOBs  ..S IBEOB(IBDA)="",IBCT=IBCT+1  ..I $P($G(^IBM(361.1,IBDA,0)),U,13)=1 S IBPROC=1  I IBCT,'IBPROC Q 1 ; there is at least one EOB and none of the EOBS are processed.  Q 0 ;  ;  WARNMSE() ; Display MSE Warning and check if we should continue.  D FULL^VALM1  N DIR,X,Y  S DIR("A",1)="WARNING : The MRA for this claim caused a Data Mismatch/Message Storage Error."  S DIR("A",2)="If you continue, the secondary claim may not contain the correct data."  S DIR("A")="Do you wish to continue? ",DIR("B")="NO",DIR(0)="YA" D ^DIR  I Y>0 Q 1 ; Okay to continue.  Q 0 ; | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCECOB2 ;ALB/CXW - IB COB MANAGEMENT SCREEN ;16-JUN-1999  ;;2.0;INTEGRATED BILLING;\*\*137,155,433,432,447,488,516\*\*;21-MAR-1994;Build 123  ;;Per VA Directive 6402, this routine should not be modified.  ;  EDI ;history detail display  N IBIFN,IBDA  D SEL(.IBDA,1)  S IBDA=+$O(IBDA(0)),IBIFN=+$G(IBDA(IBDA))  D EDI1(IBIFN)  S VALMBCK="R"  Q  ;  EDI1(IBIFN) ;  N DFN  Q:'IBIFN  S DFN=$P($G(^DGCR(399,IBIFN,0)),U,2)  D EN^VALM("IBJT EDI STATUS")  K:$D(IBFASTXT) IBFASTXT  Q  ;  EDI2(IBIFN) ;  N DFN  Q:'IBIFN  S DFN=$P($G(^DGCR(399,IBIFN,0)),U,2)  D EN^VALM("IBJT EDI STATUS ALONE")  K:$D(IBFASTXT) IBFASTXT  Q  ;  CSA ;claims status awaiting resolution  N IBDAX  D EN^IBCECSA  I $D(IBFASTXT) K IBFASTXT  S VALMBCK="R"  Q  ;  RVEOB ;Review EOB  D FULL^VALM1 W !  N IBDA,IBIFN,IBCMT,IBSEL  D SEL(.IBDA,1)  S IBSEL=+$O(IBDA(0))  S IBDA=$G(IBDA(IBSEL))  S IBIFN=$P(IBDA,U),IBDA=$P(IBDA,U,3)  I 'IBIFN G VEOBQ  S IBCMT=$G(^TMP("IBCECOB1",$J,IBSEL))  I IBCMT'="" D EN^VALM("IBCEM MRA REVIEW")  VEOBQ K ^TMP("IBCECOC",$J)  S VALMBCK="R"  Q  ;  TPJI ;Third Party joint Inquiry  N IBDA,IBIFN  D SEL(.IBDA,1)  S IBDA=+$O(IBDA(0)),IBIFN=+$G(IBDA(IBDA))  I IBDA="" G TPJIQ  D TPJI1(IBIFN)  TPJIQ S VALMBCK="R"  Q  ;  TPJI1(IBIFN) ;  N DFN,IBNOTPJI  Q:'IBIFN  S DFN=$P($G(^DGCR(399,IBIFN,0)),U,2),IBNOTPJI=1  D EN^VALM("IBJT CLAIM INFO")  K:$D(IBFASTXT) IBFASTXT  Q  ;  PBILL ;Print bill  N IBIFN,IBDA,IBRESUB  D SEL(.IBDA,1)  S IBDA=$O(IBDA(0)),IBIFN=+$G(IBDA(+IBDA))  I IBDA="" G PBOUT  S IBRESUB=$$RESUB^IBCECSA4(IBIFN,1,"P")  I IBRESUB'>0 W !,\*7,"This is not a transmittable bill or review not needed" D PAUSE^VALM1 G PBOUT  I IBRESUB=2 D G PBOUT  . N IB364  . S IB364=+$P($G(IBDA(IBDA)),U,2)  . D PRINT1^IBCEM03(IBIFN,.IBDA,IB364)  D PBILL1(IBIFN)  PBOUT S VALMBCK="R"  Q  ;  PMRA ;Print MRA  N IBIFN,IBDA,IBDAX  D SEL(.IBDA,1)  ;JRA Fix <UNDEFINED> error occurring when IBDA(+IBDA) does not exist. Also, ensure that IBDAX'=""  ; since it's used as a subscript to ^IBM.  ;S IBDA=$O(IBDA(0)),IBIFN=+$G(IBDA(+IBDA)),IBDAX=$P(IBDA(+IBDA),U,3) ;JRA ';'  S IBDA=$O(IBDA(0)),IBIFN=+$G(IBDA(+IBDA)),IBDAX=+$P($G(IBDA(+IBDA)),U,3) ;JRA Add $G to SET of IBDAX - also +$P  G:'IBIFN PRMQ  I '$G(IBMRANOT),$D(^IBM(361.1,IBDAX,"ERR")),'$$WARNMSE G PRMQ ; Claim contains Message Storage Errors  D MRA^IBCEMRAA(.IBIFN)  PRMQ S VALMBCK="R"  Q  PBILL1(IBIFN) ;  N IBAC1,IBAC,DFN  Q:'IBIFN  S DFN=$P($G(^DGCR(399,IBIFN,0)),U,2)  S IBAC=4,IBAC1=1  D 4^IBCB1  D FULL^VALM1,PAUSE^VALM1  Q  ;  CANCEL ;Cancel bill  ; IBDA(IBDA)=IBIFN^IB364^ien of 361.1^user selection seq^user name~duz#  ;  N IBIFN,IBDA,IB364,IBEOBIFN,X,IBDENCT  ;  ; Check for security key  I '$$KCHK^XUSRB("IB AUTHORIZE") D G CANCELQ  . D FULL^VALM1 S VALMBCK="R"  . W !!?5,"You don't hold the proper security key to access this function."  . W !?5,"The necessary key is IB AUTHORIZE. Please see your manager."  . D PAUSE^VALM1  . Q  ;  D SEL(.IBDA,1)  S IBDA=$O(IBDA(0)),IBIFN=+$G(IBDA(+IBDA)),IB364=$P($G(IBDA(+IBDA)),U,2)  S IBEOBIFN=$P($G(IBDA(+IBDA)),U,3)  ;  ; IB\*2.0\*432 - if not mra, only allow cancel of denied claims. If no EOB, check AR status instead  I 'IBEOBIFN,$G(IBMRANOT)=1,$P($$ARSTATA^IBJTU4(IBIFN),U)="COLLECTED/CLOSED" D G CANCELQ  . D FULL^VALM1 S VALMBCK="R"  . W !!,\*7,"You can only cancel denied claims. This claim is in a COLLECTED/CLOSED status"  . W !,"Use Remove Action to remove claim from this worklist."  . D PAUSE^VALM1  . Q  ;  ; IB\*2.0\*432 - if not mra, only allow cancel of claims with multiple EOBS if none have processed.  I $G(IBMRANOT)=1,'$$DENCHK(IBIFN,.IBDENCT),$G(IBDENCT)>1 D G CANCELQ  . D FULL^VALM1 S VALMBCK="R"  . W !!,\*7,"Multiple EOBs exist for this claim and at least one has EOB status of PROCESSED."  . W !,"Use Remove Action to remove claim from this worklist."  . D PAUSE^VALM1  . Q  ;  ; IB\*2.0\*432 - if not mra, only allow cancel of denied claims  I IBEOBIFN,$G(IBMRANOT)=1,$P($G(^IBM(361.1,IBEOBIFN,0)),U,13)'=2 D G CANCELQ  . D FULL^VALM1 S VALMBCK="R"  . W !!?5,\*7,"You can only cancel denied claims."  . D PAUSE^VALM1  . Q  ;  I IBDA D  . I '$$LOCK^IBCEU0(361.1,IBEOBIFN) Q  . D CANCEL^IBCEM3(.IBDA,IBIFN,IB364)  . D UNLOCK^IBCEU0(361.1,IBEOBIFN)  S VALMBCK="R"  ;  ; for non-MRA claims cancelled from worklist, set field 38  I $G(IBMRANOT)=1,$P($G(^DGCR(399,IBIFN,0)),U,13)=7 S X=$$WLRMVF^IBCECOB1($S($G(IBIFN)'="":IBIFN,1:+$G(IBDA(IBDA))),"CA")  I $G(IBDA)'="" D BLD^IBCECOB1  CANCELQ Q  ;  CRD ; Correct Rejected/Denied claim protocol action  N IBCNCRD  S IBCNCRD=1  CLONE ; 'Copy/cancel bill' protocol action  N IBDA,IBQ,IBEOBIFN,IBKEY,X,IBDENCT  ;  ; Check for security key  ;IB\*2.0\*516/TAZ - Remove check for IB CLON  ;I '$$KCHK^XUSRB("IB AUTHORIZE") D G CLONEQ  ;S IBKEY=$S($G(IBCNCRD)=1:"IB AUTHORIZE",1:"IB CLON")  S IBKEY="IB AUTHORIZE"  I '$$KCHK^XUSRB(IBKEY) D G CLONEQ  . D FULL^VALM1 S VALMBCK="R"  . ;W !!?5,"You don't hold the proper security key to access this function."  . ;W !?5,"The necessary key is IB AUTHORIZE. Please see your manager."  . W !!?5,"You must hold the "\_IBKEY\_" security key to access this function."  . W !?5,"Please see your manager."  . D PAUSE^VALM1  . Q  ;  D SEL(.IBDA,1)  S IBDA=$O(IBDA(""))  I IBDA="" G CLONEQ  ;  ; IB\*2.0\*432 - if not mra, only allow cancel of claims with multiple EOBS if none have processed.  I $G(IBMRANOT)=1,'$$DENCHK(+IBDA(IBDA),.IBDENCT),$G(IBDENCT)>1 D G CANCELQ  . D FULL^VALM1 S VALMBCK="R"  . W !!,\*7,"Multiple EOBs exist for this claim and at least one has EOB status of PROCESSED."  . W !,"Use Remove Action to remove claim from this worklist."  . D PAUSE^VALM1  . Q  ;  S IBEOBIFN=$P($G(IBDA(+IBDA)),U,3)  I '$$LOCK^IBCEU0(361.1,IBEOBIFN) G CLONEQ  D COPYCLON(+$G(IBDA(IBDA)),$P($G(IBDA(+IBDA)),U,2),.IBQ)  D UNLOCK^IBCEU0(361.1,IBEOBIFN)  ;  ; for non-MRA claims cloned or corrected from worklist, set field 38  I $G(IBMRANOT)=1,$G(IBQ)'="" S X=$$WLRMVF^IBCECOB1(+$G(IBDA(IBDA)),$S($G(IBCNCRD)=1:"CR",1:"CL"))  ;  CLONEQ ;  S VALMBCK="R"  D:$G(IBQ)'="" BLD^IBCECOB1  Q  ;  COPYCLON(IBIFN,IB364,IBQ) ; Generic entry point for clone a bill from EDI processing  ; IBIFN = original bill ien  ; IB364 = the ien of the transmission bill entry in file 364  ; IBQ = If bill is not cancelled, this is returned as null  ; - pass by reference -  ;  N IBQUIT,IBCCCC,IBHV,Y,IBCAN,IBCE,IBDA,IBCNCOPY  ;I '$$CANCKS^IBCEM3("CC",IBIFN) S IBQ="" G CCQ  I $G(IBCNCRD)'=1,'$$CANCKS^IBCEM3("CC",IBIFN) S IBQ="" G CCQ  ;  ;S IBCAN=2,IBCE("EDI")=1,Y=IBIFN,IBCCCC=0,IBHV("IBIFN")=IBIFN,IBHV("IBIFN1")="",IBCNCOPY=1  S IBCAN=2,IBCE("EDI")=1,Y=IBIFN,IBCCCC=0,IBHV("IBIFN")=IBIFN,IBHV("IBIFN1")=""  I $G(IBCNCRD)'=1 S IBCNCOPY=1 D ^IBCCC  I $G(IBCNCRD)=1 D CRD^IBCCC  ;D ^IBCCC  S IBIFN=IBHV("IBIFN")  K IBCE("EDI") S IBQ=1  I $P($G(^DGCR(399,IBIFN,0)),U,13)'=7 S IBQ=""  I IBHV("IBIFN1") D  . N IBU  . S IBU="R"  . S IBNIEN=+IBHV("IBIFN1")  . I "23"'[$P($G(^DGCR(399,+IBHV("IBIFN1"),0)),U,13) D  .. W:'$G(IBCEAUTO) !,\*7,"Please note: the new bill was not AUTHORIZED.",!,"It can only be accessed now via the normal, non-EDI functions.",!,"Status of new bill is ",$$EXPAND^IBTRE(399,.13,$P(^DGCR(399,IBHV("IBIFN1"),0),U,13)) S IBU="C"  . D UPDEDI^IBCEM(IB364,IBU)  ;  I '$G(IBCEAUTO) D PAUSE^VALM1  CCQ Q  ;  PRO ; Copy for secondary/tertiary bill  N VALMY,IBDA,Z,IBIFN,IBIFNH,IB364,IBCE,IBNCN  ;I '$P($G(^IBE(350.9,1,8)),U,12) D G PROQ  I '$P($G(^IBE(350.9,1,8)),U,12),$G(IBMRANOT)'=1 D G PROQ  . D FULL^VALM1  . W !!?5,"MRA's may not be processed at this time."  . W !?5,"The IB site parameter ""Allow MRA Processing?"" is set to NO."  . D PAUSE^VALM1  . Q  D SEL(.IBDA,1)  S Z=$O(IBDA(0)),Z=$G(IBDA(+Z)) G:'Z PROQ  S IBIFN=$P(Z,U),IB364=$P(Z,U,2),IBDA=$P(Z,U,3),IBIFNH=IBIFN  I 'IBIFN G PROQ  I '$G(IBMRANOT),$D(^IBM(361.1,IBDA,"ERR")),'$$WARNMSE G PROQ ; Claim contains Message Storage Errors  I '$$LOCK^IBCEU0(361.1,IBDA) G PROQ  D COBCOPY(IBIFN,IB364,2,IBDA,"BLD^IBCECOB1",.IBNCN)  D UNLOCK^IBCEU0(361.1,IBDA)  ;  ; for non-MRA claims copied from work list, set field 38  I $G(IBMRANOT)=1,$G(IBNCN)'="",($G(IBNCN)'=$G(IBIFN)) D  .S X=$$WLRMVF^IBCECOB1($G(IBIFN),"PC")  .;I $P($G(^DGCR(399,+IBNCN,"S")),U,9)'=1 D  .;.W:'$G(IBCEAUTO) !,\*7,"Please note: the new bill was not AUTHORIZED.",!,"It can only be accessed now via the normal, non-EDI functions.",!,"Status of new bill is ",$$EXPAND^IBTRE(399,.13,$P(^DGCR(399,IBNCN,0),U,13))  .;.D PAUSE^VALM1  .D:$G(IBMRANOT)=1 BLD^IBCECOB1  .Q  ;  PROQ S VALMBCK="R"  Q  ;  COBCOPY(IBIFN,IB364,IBFROM,IBIEN,IBBLD,IBNCN) ; Generic entry point for EDI COB copy  ; IBIFN = original bill ien  ; IB364 = the ien of the transmission bill entry in file 364  ; IBFROM = 1 if called from CSA, 2 if called from COB/EOB processing  ; IBIEN = entry in 361 (IBFROM=1) or 361.1 (IBFROM=2) being processed  ; IBBLD = the name of the entrypoint that will rebuild the display  ; IBNCN = by reference, returns the new claim ien if user completed the Copy process  ;  N IBCBASK,IBCBCOPY,IBCAN,IBIFNH,IBNSTAT,IBOSTAT,IBPRCOB,IBSECHK,IBLMVAR,IBAC,IBMRAIEN,IBDA,IBAUTO  N IBCOB,IBCOBIL,IBCOBN,IBINS,IBINSN,IBINSOLD,IBMRAIO,IBMRAO,IBNMOLD,IBQUIT  S (IBCBASK,IBCBCOPY,IBCAN,IBAUTO)=1,(IBPRCOB,IBSECHK)=0,(IBMRAIEN,IBDA)=IBIEN  I $G(IBMRANOT)'=1,'IB364!'IBIFN W !,"Transmission record is missing for this bill" D PAUSE^VALM1 G COBCOPX  ;  S IBIFNH=IBIFN  I IBFROM=2 S IBPRCOB=1  ; IB\*2.0\*447 Check PR to include excess and percentages where applicable  ;I $S($G(IBMRANOT)=1:$$TOT(IBIFN)'>0,1:$$PREOBTOT^IBCEU0(IBIFN,$G(IBMRANOT))'>0) D G COBCOPX  I $$TOT(IBIFN,$G(IBMRANOT))'>0 D G COBCOPX  . D FULL^VALM1  . W !!?5,"There is no "\_$S($G(IBMRANOT)=1:"balance remaining",1:"patient responsibility and/or excess charges")\_" for this claim."  . W !?5,"This claim may not be processed."  . D PAUSE^VALM1  . Q  ;  I $G(IBDA)'="",$P($G(^IBM(361.1,IBDA,0)),U,16)="1.5" D G COBCOPX  . W !!,"This claim has already been processed as a sec/tert claim."  . W !,"You will need to complete the authorization process for this claim."  . D PAUSE^VALM1  . D AUTH  . Q  ;  ; If multiple EOBs and one is processed, make sure collected closed.  I $G(IBMRANOT),$$CCCHK(IBIFN)<0 D G COBCOPX  . W !,"Multiple EOBs exist for this claim and at least one has EOB status of PROCESSED."  . W !,"Claim cannot be sent to next payer until AR status is Collected/Closed."  . D PAUSE^VALM1  . Q  ;  ; Get out if no next payer  I '$P($G(^DGCR(399,IBIFN,"I"\_($$COBN^IBCEF(IBIFN)+1))),U,1) D G COBCOPX  . W !,"There is no next payer for this bill"  . D PAUSE^VALM1  . Q  ;  D DSPRB^IBCCCB0(IBIFN) ; display related bills  S IBCE("EDI")=1  D CHKB^IBCCCB ; process COB, create secondary bill  S IBNCN=$G(IBCE("EDI","NEW")) ; get new claim ien  S IBIFN=IBIFNH  I IBSECHK G COBCOPX  ;  ; if user came from CBW, no need to view and authorize a 2nd time (already happens in IBCCCB)  Q:$G(IBMRANOT)=1  S IBV=1 D VIEW^IBCB2 ; display billing screens  D AUTH ; authorize bill  COBCOPX ;  Q  ;  AUTH ; procedure to authorize the claim and refresh the screen  K ^UTILITY($J) S IBAC=1,IBQUIT=0 D 3^IBCB1  I '$D(IOUON)!'$D(IORVON) D ENS^%ZISS  I $P($G(^IBM(361.1,IBMRAIEN,0)),U,16)=3 D UPDEDI^IBCEM(IB364,"Z")  I $G(IBBLD)'="" D @IBBLD  D PAUSE^VALM1  AUTHX ;  Q  ;  RES ;Resubmit bill by print  N IBDA,IBIFN,IB364  D SEL(.IBDA,1)  S IBDA=+$O(IBDA(0)),IBIFN=+$G(IBDA(+IBDA)),IB364=+$P($G(IBDA(IBDA)),U,2)  I 'IBIFN G RESQ  D PRINT1^IBCEM03(IBIFN,.IBDA,IB364)  D PAUSE^VALM1  I $G(IBDA)'="" D BLD^IBCECOB1  RESQ S VALMBCK="R"  Q  ;  EBI ;View an unauthorized transmitted bill  N IBFLG,IBDA,IBIFN,IB364,DFN  K ^TMP($J,"IBBILL")  D FULL^VALM1  ;  D SEL(.IBDA,1)  S IBDA=+$O(IBDA(""))  S IBIFN=+$G(IBDA(IBDA)),IB364=+$P($G(IBDA(IBDA)),U,2),DFN=$P($G(^DGCR(399,IBIFN,0)),U,2)  G:'IBIFN EDITQ  S IBV=1 D VIEW^IBCB2  I '$D(IOUON)!'$D(IORVON) D ENS^%ZISS  D BLD^IBCECOB1  EDITQ S VALMBCK="R"  Q  ;  SEL(IBDA,ONE) ; Select entry(s) from list  ; IBDA = array returned if selections made  ; IBDA(n)=ien of bill selected (file 399)  ; ONE = if set to 1, only one selection can be made at a time  N IB  K IBDA  D FULL^VALM1  D EN^VALM2($G(XQORNOD(0)),$S('$G(ONE):"",1:"S"))  S IBDA=0 F S IBDA=$O(VALMY(IBDA)) Q:'IBDA S IBDA(IBDA)=$P($G(^TMP("IBCECOB",$J,+IBDA)),U,2,6)  Q  ;  EXIT ; Exit out of COB  D FASTEXIT^IBCEFG4  I $G(IBFASTXT)=1 S IBFASTXT=5  Q  ;  TOT(IBIFN,IBMRANOT) ; calculate if any balance remaining on non-MRA claim  ; IBIFN = claim ien  ; IBMRANOT = MRW/CBW flag (1=user came from CBW) added with IB\*2.0\*447  N IBPRTOT,IBBLD,IBCBN,IBU2  I $G(IBMRANOT)'=1 Q $S($$MSEDT^IBCEMU4(IBIFN)'="":$$MSPRE^IBCEMU4(IBIFN),1:$$PREOBTOT^IBCEU0(IBIFN))  ; total up the payer paid amounts, if this is a 2ndary claim, be sure to account for what the primary paid also  S IBU2=$G(^DGCR(399,IBIFN,"U2")),IBCBN=$$COBN^IBCEF(IBIFN),IBPRTOT=$$EOBTOT^IBCEU1(IBIFN,IBCBN)  S:IBPRTOT<0 IBPRTOT=0 ; don't allow negative prior payment or offset  S:IBCBN=2 IBPRTOT=IBPRTOT+$P(IBU2,U,4)  S:IBCBN=3 IBPRTOT=IBPRTOT+$P(IBU2,U,4)+$P(IBU2,U,5)  S:IBPRTOT<0 IBPRTOT=0 ; don't allow negative prior payment or offset  ; Subtract payer paid amount from Total Charges from BILLS/CLAIMS (#399) file, don't allow neg  S IBBLD=$P($G(^DGCR(399,IBIFN,"U1")),U,1)-IBPRTOT  S:IBBLD<0 IBBLD=0  Q IBBLD  ;  CCCHK(IBIFN) ; If there are multiple EOBS on file for this claim, then one of them must be processed and AR status must be collected closed to process.  ; returns 1 if true  ; 0 if there are not multiple EOBs or mulitple EOBs and none are processed (all denials)  ; -1 if false  N IBDA,IBCT,IBPROC,IBARSTAT,IBEOBNDX,IBEOB  S IBCT=0,IBPROC=0  F IBEOBNDX="B","C" D  .S IBDA=0 F S IBDA=$O(^IBM(361.1,IBEOBNDX,IBIFN,IBDA)) Q:'+IBDA D  ..Q:$D(IBEOB(IBDA))  ..Q:$P($G(^IBM(361.1,IBDA,0)),U,4)=1 ; only count EOBs  ..S IBEOB(IBDA)="",IBCT=IBCT+1  ..I $P($G(^IBM(361.1,IBDA,0)),U,13)=1 S IBPROC=1  I IBCT<2 Q 0 ; less than 2 EOBs  I 'IBPROC Q 0 ; no EOBs with status processed  S IBARSTAT=$$ARSTATA^IBJTU4(IBIFN) ; get status of AR  I $P(IBARSTAT,U)="COLLECTED/CLOSED" Q 1  Q -1  ;  DENCHK(IBIFN,IBCT) ; Make sure all EOBs from this claim are denied.  ; Input: IBIFN - IEN to 399  ; IBCT - by reference. Return count of EOBs.  ; Output: returns 1 if there is at least one EOB and that none of the EOBS are processed.  ; otherwise 0  ;  N IBDA,IBPROC,IBEOBNDX,IBEOB  S IBCT=0,IBPROC=0  F IBEOBNDX="B","C" D  .S IBDA=0 F S IBDA=$O(^IBM(361.1,IBEOBNDX,IBIFN,IBDA)) Q:'+IBDA D  ..Q:$D(IBEOB(IBDA))  ..Q:$P($G(^IBM(361.1,IBDA,0)),U,4)=1 ; only count EOBs  ..S IBEOB(IBDA)="",IBCT=IBCT+1  ..I $P($G(^IBM(361.1,IBDA,0)),U,13)=1 S IBPROC=1  I IBCT,'IBPROC Q 1 ; there is at least one EOB and none of the EOBS are processed.  Q 0 ;  ;  WARNMSE() ; Display MSE Warning and check if we should continue.  D FULL^VALM1  N DIR,X,Y  S DIR("A",1)="WARNING : The MRA for this claim caused a Data Mismatch/Message Storage Error."  S DIR("A",2)="If you continue, the secondary claim may not contain the correct data."  S DIR("A")="Do you wish to continue? ",DIR("B")="NO",DIR(0)="YA" D ^DIR  I Y>0 Q 1 ; Okay to continue.  Q 0 ; | | | | | | | | | |