**TAS eBilling SDD**

**US-1108**

**US-2488**

**V2.00**

System Design Document

IB\*2.0\*592



Department of Veterans Affairs

**August 2017**

Version 2.00

**User Story Number:** US-1108, US-2488

**User Story Name:** Enter/Edit Dental Claims, Update Reports – Form Type J430D

**Product Backlog ID:** n/a

# Design/Assumptions:

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

1. The dental specific claim information that is not available via the patient encounter records will be available to the billing clerks for manual entry.
2. Dental claims will not be printable locally.
3. VistA will provide the non-X12 data element VAMC Side/Div ID to the clearinghouse so they can create their claims reports that they return to VistA.
4. There will be a way in VistA for the IB software to identify the event as a dental event.
5. TJPI is covered in US14.
6. The IB system will provide the ability for users to view/input the additional Form Type J403D or Form Type designation (I/P/D) when one of the following reports/options searches or displays the form type:
   * View/Print EOB
   * EDI Claim Status Report
   * View/Resubmit Claims – Live or Test
   * Ready for Extract Status Report
   * HCCH Payer ID Report
   * View/Print EDI Bill Extract Data
   * Provider ID Query (CPAC)

# Resolution Summary:

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

1. Modify the Enter/Edit Billing Information [IB EDIT BILLING INFO] option to provide the ability to Create and maintain a new Form Type J430D for Dental Claims.
2. Modify the Enter/Edit Billing Information [IB EDIT BILLING INFO] option to retrieve the available data from the Patient’s PCE (Patient Care Encounter) and make it available to the user to add to the claim.
3. Modify the Enter/Edit Billing Information [IB EDIT BILLING INFO] option to default the CHARGE TYPE for Dental Claim to that of Professional.
4. Modify the Enter/Edit Billing Information [IB EDIT BILLING INFO] option to provide a NEW PROVIDER TYPE to be added to Dental claims at either the line level or the claim level equal to ASSISTANT SURGEON with the qualifier equal to DD.
5. Modify the Enter/Edit Billing Information [IB EDIT BILLING INFO] option to provide the following new Line Level Data Fields for Dental Procedures:

* Oral Cavity Designation [Up to 5 Procedures]
* Prosthesis/Crown/Inlay Code; define field as a SET with the following values:
  + I = Initial Placement
  + R = Replacement
* Prior Placement Date and Qualifier; REQUIRED when Prosthesis/Crown/Inlay Code is equal to “R”eplacement. This field should be defined as a SET with the following values:
  + 139 = Estimated
  + 441 = Prior Placement
* Tooth Code [New File containing the 32 different Teeth]
* Tooth Surface Code; define field as a SET with the following values:
  + B = Buccal
  + D = Distal
  + F = Facial
  + I = Incisal
  + L = Lingual
  + M = Mesial
  + O = Occlusal
* Orthodontic Banding Date; this is the date the patient’s orthodontic appliances were placed.
* Orthodontic Banding Replacement Date
* Treatment Start Date
* Treatment Completion Date

1. Modify the Enter/Edit Billing Information [IB EDIT BILLING INFO] option to provide the following new Line Claim Data Fields for Dental Claims:

* Tooth Number
* Tooth Status Code
* Orthodontic Banding Date
* Orthodontic Treatment Months Count
* Orthodontic Treatment Months Remaining
* Treatment Indicator; this is a YES or NO field
* Attachment Report Type; define field as a SET with the following values:
  + B4 = Referral Form
  + DA = Dental Models
  + DG = Diagnostic Report
  + EB = EOB (COB o Medicare Secondary Report)
  + OZ = Support Data for Claim
  + P6 = Periodontal Charts
  + RB = Radiology Films
  + RR = Radiology Reports
* Transmission Method; the REQUIRED logic is the same as the current Attachment Report fields in Screen 8.
* Attachment Control Number; the REQUIRED logic is the same as the current Attachment Report fields in Screen 8.
* Claim Note; this should be an 80 character free text field.

1. Modify the Enter/Edit Billing Information [IB EDIT BILLING INFO] option to create a Dental 837 transaction for Dental Events even though the Charge Type is Professional.
2. The Enter/Edit Billing Information [IB EDIT BILLING INFO] option should prevent the Local Printing of Dental Claims.
3. The Enter/Edit Billing Information [IB EDIT BILLING INFO] option will prevent the creation of dental claims to the insurance company, Medicare (WNR).
4. The IB System will provide the ability for users to view/input the additional Form Type J430D or Form Type designation (I/P/D) when one of the following reports/options searches or displays the form type:
   * View/Print EOB
   * EDI Claim Status Report
   * View/Resubmit Claims – Live or Test
   * Ready for Extract Status Report
   * HCCH Payer ID Report
   * View/Print EDI Bill Extract Data
   * Provider ID Query (CPAC)
5. The IB System will provide the ability for users to continue to use the GEN Print Bill option, [IB PRINT BILL], to view, the screens of previously transmitted dental claims while preventing their ability to print those claims.

# Design Constraints:

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

* US131 (Create 837D Transaction)
* US1109 (Create Dental Form/Update Autobiller)
* US2487 (Insurance Company Entry/Edit – Dental)
* US2503 (Provider ID Maintenance Dental)

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. Create a new Form Type “J430D” in Bill Form Type file [#353]. Form type must be setup as a printable form, even though it will not be printable, but for screen entry purposes, it needs to be a printable form. National Form field needs to be set to Yes, to allow billing screen entry.

NUMBER: 7 NAME: J430D

FORMAT TYPE: PRINTED FORM NATIONAL FORM: YES

SHORT DESCRIPTION: Dental Form

1. The Bill/Claims File [#399] requires the following new fields to be defined.

DATA NAME GLOBAL DATA

ELEMENT TITLE LOCATION TYPE

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

399.0304,90.01 ORAL CAVITY DESIGNATION (1) DEN;1 SET

'00' FOR Entire Oral Cavity;

'01' FOR Maxillary Arch;

'02' FOR Mandibular Arch;

'10' FOR Upper Right Quadrant;

'20' FOR Upper Left Quadrant;

'30' FOR Lower Left Quadrant;

'40' FOR Lower Right Quadrant;

LAST EDITED: MAR 02, 2017

HELP-PROMPT: Enter a valid Cavity Designation code. The

entered code must not already be present in

Oral Cavity Designations #2, #3, #4 or #5.

DESCRIPTION: The first Oral Cavity Designation code. You

can enter up to five codes.

SCREEN: S DIC("S")="I $$ORALCAV^IBCU7(90.01)"

EXPLANATION: Only allows Oral Cavity Designation Codes

that are not already present in Oral Cavity

Designations #2, #3, #4 or #5.

399.0304,90.02 ORAL CAVITY DESIGNATION (2) DEN;2 SET

'00' FOR Entire Oral Cavity;

'01' FOR Maxillary Arch;

'02' FOR Mandibular Arch;

'10' FOR Upper Right Quadrant;

'20' FOR Upper Left Quadrant;

'30' FOR Lower Left Quadrant;

'40' FOR Lower Right Quadrant;

LAST EDITED: MAR 02, 2017

HELP-PROMPT: Enter a valid Oral Cavity Designation Code.

The entered code must not already be present

in Oral Cavity Designations #1, #3, #4 or #5.

DESCRIPTION: The second Oral Cavity Designation code. You

can enter up to five codes.

SCREEN: S DIC("S")="I $$ORALCAV^IBCU7(90.02)"

EXPLANATION: Only allows Oral Cavity Designation Codes

that are not already present in Oral Cavity

Designations #1, #3, #4 or #5.

399.0304,90.03 ORAL CAVITY DESIGNATION (3) DEN;3 SET

'00' FOR Entire Oral Cavity;

'01' FOR Maxillary Arch;

'02' FOR Mandibular Arch;

'10' FOR Upper Right Quadrant;

'20' FOR Upper Left Quadrant;

'30' FOR Lower Left Quadrant;

'40' FOR Lower Right Quadrant;

LAST EDITED: MAR 02, 2017

HELP-PROMPT: Enter a valid Oral Cavity Designation Code.

The entered code must not already be present

in Oral Cavity Designations #1, #2, #4 or #5.

DESCRIPTION: The third Oral Cavity Designation code. You

can enter up to five codes.

SCREEN: S DIC("S")="I $$ORALCAV^IBCU7(90.03)"

EXPLANATION: Only allows Oral Cavity Designation Codes

That are not already present in Oral Cavity

Designations #1, #2, #4 or #5.

399.0304,90.04 ORAL CAVITY DESIGNATION (4) DEN;4 SET

'00' FOR Entire Oral Cavity;

'01' FOR Maxillary Arch;

'02' FOR Mandibular Arch;

'10' FOR Upper Right Quadrant;

'20' FOR Upper Left Quadrant;

'30' FOR Lower Left Quadrant;

'40' FOR Lower Right Quadrant;

LAST EDITED: MAR 02, 2017

HELP-PROMPT: Enter a valid Oral Cavity Designation code.

The entered code must not already be present

in Oral Cavity Designations #1, #2, #3 or #5.

DESCRIPTION: The fourth Oral Cavity Designation code. You

can enter up to five codes.

SCREEN: S DIC("S")="I $$ORALCAV^IBCU7(90.04)"

EXPLANATION: Only allows Oral Cavity Designation Codes

that are not already present in Oral Cavity

Designations #1, #2, #3 or #5.

399.0304,90.05 ORAL CAVITY DESIGNATION (5) DEN;5 SET

'00' FOR Entire Oral Cavity;

'01' FOR Maxillary Arch;

'02' FOR Mandibular Arch;

'10' FOR Upper Right Quadrant;

'20' FOR Upper Left Quadrant;

'30' FOR Lower Left Quadrant;

'40' FOR Lower Right Quadrant;

LAST EDITED: MAR 02, 2017

HELP-PROMPT: Enter a valid Oral Cavity Designation code.

The entered code must not already be present

in Oral Cavity Designations #1, #2, #3 or #4.

DESCRIPTION: The fifth Oral Cavity Designation code. You

can enter up to five codes.

SCREEN: S DIC("S")="I $$ORALCAV^IBCU7(90.05)"

EXPLANATION: Only allows Oral Cavity Designation Codes

that are not already present in Oral Cavity

Designations #1, #2, #3 and #4.

399.0304,90.06 PROSTHESIS/CROWN/INLAY CODE DEN;6 SET

'I' FOR Initial Placement;

'R' FOR Replacement;

LAST EDITED: JUN 28, 2017

HELP-PROMPT: Select a code that indicates the placement

status of the prosthesis, crown or inlay. DESCRIPTION: This is the placement status of the

prosthesis.

399.0304,90.07 PRIOR PLACEMENT DATE QUALIFIER DEN;7 SET

'139' FOR Estimated;

'441' FOR Prior Placement;

LAST EDITED: JUN 14, 2017

HELP-PROMPT: Select a qualifier that indicates whether or

not the Prior Placement Date is known or just

estimated.

DESCRIPTION: This is the date that indicates whether the

Prior Placement Date is known or is

estimated.

399.0304,90.08 PRIOR PLACEMENT DATE DEN;8 DATE

INPUT TRANSFORM: S %DT="EX" D ^%DT S X=Y K:Y<1 X

LAST EDITED: JUN 14, 2017

HELP-PROMPT: Enter the date when the prosthesis, crown or

inlay was replaced. Date is REQUIRED when

Prosthesis/Crown/Inlay code equals

Replacement.

DESCRIPTION: This date indicates when a prosthesis, crown

or inlay was replaced.

399.0304,90.09 ORTHODONTIC BANDING DATE DEN;9 DATE

INPUT TRANSFORM: S %DT="EX" D ^%DT S X=Y K:Y<1 X

LAST EDITED: JUN 28, 2017

HELP-PROMPT: Enter the date the patient's orthodontic

appliances were placed if different from the

claim level date.

DESCRIPTION: This is the date the patient's orthodontic

appliances were placed if different from the

claim level date.

399.0304,90.1 ORTHO BANDING REPLACEMENT DATE DEN;10 DATE

INPUT TRANSFORM: S %DT="EX" D ^%DT S X=Y K:Y<1 X

LAST EDITED: JUN 28, 2017

HELP-PROMPT: Enter the date the patient's orthodontic

appliances were replaced.

DESCRIPTION: This is the date the patient's orthodontic

appliances were replaced.

399.0304,90.11 TREATMENT START DATE DEN;11 DATE

INPUT TRANSFORM: S %DT="EX" D ^%DT S X=Y K:Y<1 X

LAST EDITED: JUN 28, 2017

HELP-PROMPT: Enter the date for initial impression or

preparation for a crown or dentures or

initial endodontic treatment or the implant

fixture placement.

DESCRIPTION: This is the date for initial impression or

preparation for a crown or dentures or

initial endodontic treatment or the implant

fixture placement.

399.0304,90.12 TREATMENT COMPLETION DATE DEN;12 DATE

INPUT TRANSFORM: S %DT="EX" D ^%DT S X=Y K:Y<1 X

LAST EDITED: JUN 28, 2017

HELP-PROMPT: Enter the date that a course of treatment was

completed.

DESCRIPTION: This is the date that a course of treatment

was completed.

399.0304,91 TOOTH INFORMATION DEN1;0 POINTER

Multiple #399.30491

(Add New Entry without Asking)

DESCRIPTION: This multiple holds tooth information for the

dental service line.

399.30491,.01 TOOTH CODE 0;1 POINTER TO X12 278 DENTAL NUMBERING

SYSTEM FILE (#356.022) (Multiply asked)

LAST EDITED: MAR 02, 2017

HELP-PROMPT: Enter a valid Tooth Code.

DESCRIPTION:

This identifies the tooth that requires work.

CROSS-REFERENCE:399.30491^B

1)=

S ^DGCR(399,DA(2),"CP",DA(1),"DEN1","B",$E(X,1,30),DA)=""

2)=

K ^DGCR(399,DA(2),"CP",DA(1),"DEN1","B",$E(X,1,30),DA)

399.30491,.02 TOOTH SURFACE (1) 0;2 SET

'B' FOR Buccal;

'D' FOR Distal;

'F' FOR Facial;

'I' FOR Incisal;

'L' FOR Lingual;

'M' FOR Mesial;

'O' FOR Occlusal;

LAST EDITED: MAR 02, 2017

HELP-PROMPT: Enter a valid Tooth Surface code. The

entered code must not already be present in

Tooth Surfaces #2, #3, #4 or #5.

DESCRIPTION: The code that best describes the area of the

tooth that was treated. Up to five Tooth

Surfaces are allowed.

SCREEN: S DIC("S")="I $$TOOTHS^IBCU7(.02)"

EXPLANATION: Only allow Tooth Surface Codes that are not

already present in Tooth Surfaces #2, #3, #4

or #5.

399.30491,.03 TOOTH SURFACE (2) 0;3 SET

'B' FOR Buccal;

'D' FOR Distal;

'F' FOR Facial;

'I' FOR Incisal;

'L' FOR Lingual;

'M' FOR Mesial;

'O' FOR Occlusal;

LAST EDITED: MAR 02, 2017

HELP-PROMPT: Enter a valid Tooth Surface code. The

Entered code must not already be present in

Tooth Surfaces #1, #3, #4 or #5.

DESCRIPTION: The code that best describes the area of the

tooth that was treated. Up to five Tooth

Surfaces are allowed.

SCREEN: S DIC("S")="I $$TOOTHS^IBCU7(.03)"

EXPLANATION: Only allow Tooth Surface Codes that are not

already present in Tooth Surfaces #1, #3, #4

or #5.

399.30491,.04 TOOTH SURFACE (3) 0;4 SET

'B' FOR Buccal;

'D' FOR Distal;

'F' FOR Facial;

'I' FOR Incisal;

'L' FOR Lingual;

'M' FOR Mesial;

'O' FOR Occlusal;

LAST EDITED: MAR 02, 2017

HELP-PROMPT: Enter a valid Tooth Surface code. The

entered code must not already be present in

Tooth Surfaces #1, #2, #4 or #5.

DESCRIPTION: The code that best describes the area of the

tooth that was treated. Up to five Tooth

Surfaces are allowed.

SCREEN: S DIC("S")="I $$TOOTHS^IBCU7(.04)"

EXPLANATION: Only allow Tooth Surface Codes that are not

already present in Tooth Surfaces #1, #2, #4

or #5.

399.30491,.05 TOOTH SURFACE (4) 0;5 SET

'B' FOR Buccal;

'D' FOR Distal;

'F' FOR Facial;

'I' FOR Incisal;

'L' FOR Lingual;

'M' FOR Mesial;

'O' FOR Occlusal;

LAST EDITED: MAR 02, 2017

HELP-PROMPT: Enter a valid Tooth Surface code. The

entered code must not already be present in

Tooth Surfaces #1, #2, #3 or #5.

DESCRIPTION: The code that best describes the area of the

tooth that was treated. Up to five Tooth

Surfaces are allowed.

SCREEN: S DIC("S")="I $$TOOTHS^IBCU7(.05)"

EXPLANATION: Only allow Tooth Surface codes that are not

already present in Tooth Surfaces #1, #2, #3

or #5.

399.30491,.06 TOOTH SURFACE (5) 0;6 SET

'B' FOR Buccal;

'D' FOR Distal;

'F' FOR Facial;

'I' FOR Incisal;

'L' FOR Lingual;

'M' FOR Mesial;

'O' FOR Occlusal;

LAST EDITED: MAR 02, 2017

HELP-PROMPT: Enter a valid Tooth Surface code. The

entered code must not already be present in

Tooth Surfaces #1, #2, #3 or #4.

DESCRIPTION: The code that best describes the area of the

tooth that was treated. Up to five Tooth

Surfaces are allowed.

SCREEN: S DIC("S")="I $$TOOTHS^IBCU7(.06)"

EXPLANATION: Only allow Tooth Surface codes that are not

already present in Tooth Surfaces #1, #2, #3

or #4.

399,92 BANDING DATE DEN;1 DATE

INPUT TRANSFORM: S %DT="EX" D ^%DT S X=Y K:Y<1 X

LAST EDITED: JUN 28, 2017

HELP-PROMPT: Enter the date the patient's orthodontic

appliances were placed.

DESCRIPTION: This is the date the patient's orthodontic

appliances were placed.

399,93 TREATMENT MONTHS COUNT DEN;2 NUMBER

INPUT TRANSFORM:

K:+X'=X!(X>999999999999999)!(X<0)!(X?.E1"."1N.N) X

LAST EDITED: JUN 28, 2017

HELP-PROMPT: Enter the estimated number of treatment

months in whole numbers.

DESCRIPTION: This is the estimated number of treatment

months.

399,94 TREATMENT MONTHS REMAINING DEN;3 NUMBER

INPUT TRANSFORM:

K:+X'=X!(X>999999999999999)!(X<0)!(X?.E1"."1N.N) X

LAST EDITED: JUN 28, 2017

HELP-PROMPT: Enter the number of months remaining required

for a transfer patient, in whole numbers.

DESCRIPTION: This is the number of months remaining

required for a transfer patient.

399,95 TREATMENT INDICATOR DEN;4 SET

'0' FOR NO;

'1' FOR YES;

LAST EDITED: JUN 28, 2017

HELP-PROMPT: Enter 'YES' if services reported on this

claim are for orthodontic purposes.

Otherwise, enter 'NO'. REQUIRED when neither

Treatment Months nor Treatment Months

Remaining are present.

DESCRIPTION: This field indicates that services reported

on this claim are for orthodontic purposes.

REQUIRED when neither Treatment Months nor

Treatment Months Remaining are present.

399,96 TOOTH STATUS DEN1;0 SET Multiple #399.096

DESCRIPTION: This is a multiple field defining the teeth

that the dental services were related to.

IDENTIFIED BY: STATUS CODE(#.02)

399.096,.01 TOOTH NUMBER 0;1 NUMBER (Multiply asked)

INPUT TRANSFORM: K:+X'=X!(X>32)!(X<0)!(X?.E1"."1N.N) X

LAST EDITED: MAR 13, 2017

HELP-PROMPT: Type a number between 1 and 32, 0 decimal

digits.

DESCRIPTION: This is the tooth number that is either

missing or is to be extracted.

CROSS-REFERENCE: 399.096^B

1)= S ^DGCR(399,DA(1),"DEN1","B",$E(X,1,30),DA)=""

2)= K ^DGCR(399,DA(1),"DEN1","B",$E(X,1,30),DA)

399.096,.02 STATUS CODE 0;2 SET

'E' FOR EXTRACTED;

'M' FOR MISSING;

LAST EDITED: JUN 28, 2017

HELP-PROMPT: Select the code that indicates whether a

tooth will be extracted or is missing.

DESCRIPTION: This code indicates whether a tooth will be

extracted or is missing.

399,97 DENTAL CLAIM NOTE DEN2;1 FREE TEXT

INPUT TRANSFORM: K:$L(X)>80!($L(X)<1) X

LAST EDITED: JUN 28, 2017

HELP-PROMPT: Enter information that is needed to

substantiate the medical treatment, 1 to 80

characters.

DESCRIPTION: This is an 80 character free text field to

allow for the entry of information that is

needed to substantiate medical treatment.

1. Screens 5 and 10 require modifications to allow for the view / entry of the necessary Dental field values. File 399, sub-file 399.0304, sub-file 399.0404, field .01 LINE FUNCTION and File 399, sub-file 399.0222, field .01 FUNCTION are a set of codes that will need to have an additional option added, 6 ASSISTANT SURGEON.

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

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

1. Screen 8 will be modified to display a different set of fields if the Claim is a Dental Claim (See recommended changes to the routine IBCSC8). The following data dictionary modifications are required to allow for the view/entry of the necessary Dental field values:

* Field #399, 285 (Attachment Report Type), needs to have the following SCREEN added to its field definition:

399,285 ATTACHMENT REPORT TYPE U8;2 POINTER TO IB ATTACHMENT

REPORT TYPE FILE (#353.3)

LAST EDITED: AUG 16, 2010

HELP-PROMPT: Select an Attachment Report Type.

DESCRIPTION: This is a Report Type to describe the type of

documentation that will provide additional

information for this claim. This applies to

the entire claim.

SCREEN: S DIC("S")=”I $$RTYPOK^IBCEU(X,DA)"

This will allow for the following:

1. For all Claims that are not Dental, the Screen will prevent the option of P6 (Periodontal Charts) from being a selected value.
2. For those claims that are Dental, the Screen will only allow the selection of the following values from file #353.3 (IB ATTACHMENT REPORT TYPE):

B4 = Referral Form

DA = Dental Models

DG = Diagnostic Report

EB = EOB (COB o Medicare Secondary Report)

OZ = Support Data for Claim

P6 = Periodontal Charts

RB = Radiology Films

RR = Radiology Reports

1. The following routines need to be modified to allow for the entry/edit of the new Dental fields.

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCB | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** |  | | | | | | | | |
| **Related Options** | None | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  |  | | | | |  | | | |
| **Data Dictionary (DD) References** | CLAIMS TRACKING File [#356] | | | | | | | | |
| **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** | | | | | | | | | |
| IBCB ;ALB/MRL - BILLING BEGINNING POINT/SELECT BILL OR PATIENT ;01 JUN 88 12:00  ;;2.0;INTEGRATED BILLING;\*\*52,80,106,51,137,161,199,348\*\*;21-MAR-94;Build 5  ;;Per VHA Directive 10-93-142, this routine should not be modified.  ;  ;MAP TO DGCRB  ; EN ;  D HOME^%ZIS Q:'$D(IBAC)  ;\*\*\*  ;I $D(XRT0) S:'$D(XRTN) XRTN="IBCB" D T1^%ZOSV ;stop rt clock  ;S XRTL=$ZU(0),XRTN="IBCB-"\_$G(IBAC) D T0^%ZOSV ;start rt clock  ;  S:'$D(IBV) IBV=1 L  K ^UTILITY($J),DFN,IBIFN,DIC,IBPOPOUT S DIC(0)="EQMZ" R !!,"Enter BILL NUMBER or PATIENT NAME: ",IBX:DTIME I IBX["^"!(IBX="") S IBAC1=0 Q  K ^TMP("IBCRRX",$J)  S IBAC1=1  N DPTNOFZY S DPTNOFZY=1 ;Suppress PATIENT file fuzzy lookups  I IBX?1A4N!(IBX?2A.AP)!(IBX?2.A1",".AP)!(IBX?1A1P.AP) S DIC="^DPT(",X=IBX D ^DIC G EN:Y'>0 S DFN=+Y D HINQ S X=$S('$D(^DGCR(399,"C",DFN)):1,'$D(^DGCR(399,"AOP",DFN)):2,1:0)  I $D(DFN),X,IBAC<4 W !!,"No ",$S(X=1:"",1:"OPEN "),"billing records on file for this patient." D ASK I '$D(IBIFN) G EN  I $D(DFN) D  G EN  . D DATE:'$D(IBIFN),ASK:'$D(IBIFN)  . I $D(IBIFN) D ST  S DIC("S")=$S(IBAC'=4&(IBAC'=4.1):"I $P(^(0),U,13)<3 D EN^DDIOL($P(^(0),U))",1:"I $P(^(""S""),U,17)="""""\_$S(IBAC=4.1:",$P(^(0),U,13)=3,+$$LAST364^IBCEF4(+Y),""PX""[$P($G(^IBA(364,+$$LAST364^IBCEF4(+Y),0)),U,3)",1:""))  S DIC="^DGCR(399,",X=IBX  D ^DIC G:Y'>0 EN S IBIFN=+Y,DFN=$P(Y(0),"^",2)  ;  D HINQ,ST G EN  G EN HINQ I $S('$D(^DPT(DFN,.361)):1,$P(^(.361),"^",1)'="V":1,1:0) W !?17,"\*\*\* ELIGIBILITY NOT VERIFIED \*\*\*" D HINQ1 MT ;I $D(DFN) D ^DGMT1 K DGMTLL  I $D(DFN) D DIS^DGMTU(DFN)  Q HINQ1 I $P($G(^IBE(350.9,1,1)),"^",16) S X="DVBHQZ4" X ^%ZOSF("TEST") K X I $T W ! D EN^DVBHQZ4 Q  ;I $P($G(^IBE(350.9,1,1)),"^",16) F X="DVBHQZ4","DGHINQZ4" X ^%ZOSF("TEST") I $T S DGROUT=X K X W ! D @("EN^"\_DGROUT) K DGROUT Q  K Y Q ASK I IBAC'=1 K IBIFN Q  W !!,"DO YOU WANT TO ESTABLISH A NEW BILLING RECORD FOR '",$P(^DPT(DFN,0),"^",1),"'" S %=2 D YN^DICN  I '% W !!?4,"YES - To establish a new billing record in the billing file.",!?4,"NO - To discontinue this process immediately." G ASK  I %'=1 K IBIFN Q  K DA,Y,DINUM,IBIFN S (IBNEW,IBYN)=1 D ^IBCA Q DATE I $D(^DGCR(399,"C",DFN)) S DA="" F I=1:1 S DA=$O(^DGCR(399,"APDT",DFN,DA)) Q:DA=""  D DATE1  I IBAC=4,'$D(^UTILITY($J,"IB")) W !,"No ",$S($D(^DGCR(399,"C",DFN)):"UNCANCELLED ",1:""),"billing records on file for this patient." Q  S CT=0,CT1=1,IBT="" F J=1:1 S IBT=$O(^UTILITY($J,"IB",IBT)) Q:IBT=""  F J1=0:0 S J1=$O(^UTILITY($J,"IB",IBT,J1)) Q:J1=""  S X=J1 D SET CT W ! S G="",CT2=$S(CT<(CT1+4):CT,1:(CT1+4)) F K=CT1:1:CT2 I $D(^UTILITY($J,"UB",K)) D WRLINE  S X="" D WDATE Q:X["^"  I '$D(IB),$D(^UTILITY($J,"UB",K+1)) S CT1=K+1 G CT  K CT,CT1,CT2,K,^UTILITY($J,"UB") Q WRLINE N IBX S IBDATA=^UTILITY($J,"UB",K),IBX=$G(^DGCR(399,+$P(IBDATA,"^",2),0))  W !?2,K,?6 S Y=+IBDATA X ^DD("DD") W Y,?27,$P(IBX,"^",1),?35,$S($P(IBX,U,21)="S":"s",$P(IBX,U,21)="T":"t",1:""),?38,$P(IBDATA,"^",3),?59,$E($P(IBDATA,"^",4),1,10),?70,$E($P(IBDATA,"^",5),1,10)  Q DATE1 S IBT=$O(^DGCR(399,"APDT",DFN,DA,0)) I $D(^DGCR(399,+DA,0)),$S(IBAC<3:$P(^(0),U,13)<2,IBAC=3:$P(^(0),U,13)<3,'$D(^("S")):0,$P(^("S"),"^",17)]"":0,1:1) S ^UTILITY($J,"IB",IBT,DA)=""  Q WDATE Q:'CT  W !! W:K<CT "PRESS <RETURN> TO CONTINUE, OR",! W "CHOOSE 1",$S(CT=1:"",1:"-"\_K),": " R X:DTIME Q:X["^"!(X="") I X["?" W !!,"Select one of the above or <RETURN> to establish a new billing record." G WDATE  I $S('$D(^UTILITY($J,"UB",+X)):1,+X>K:1,+X<1:1,'(X?.N):1,1:0) W !!,"NOT A VALID CHOICE!!",\*7 G WDATE  S IBIFN=$P(^UTILITY($J,"UB",X),"^",2),IB=1  Q  ; KEYOK(IBIFN,DUZ) ; Check if COB bill, does user have key  ; IBIFN = ien of bill (file 399)  ;  N IBCOB,IBOK,DIR,X,Y  S IBOK=1,IBCOB=$$COBN^IBCEF(IBIFN)  I IBCOB>1 D  . S IBCOB=$P("^SECONDARY^TERTIARY",U,IBCOB)  . S DIR(0)="YA",DIR("A",1)="YOU ARE ABOUT TO EDIT A "\_IBCOB\_" BILL",DIR("A")="ARE YOU SURE YOU WANT TO CONTINUE?: ",DIR("B")="NO" W ! D ^DIR K DIR W !  . I Y'=1 S IBOK=0  Q IBOK  ; SET I $S(IBV:1,$P(^DGCR(399,+X,0),"^",13):1,1:0) S CT=CT+1 D SET2  Q SET2 S IBND0=^DGCR(399,+X,0)  N IBFTP  S IBFTP=$S($$FT^IBCEF(+X)=3:"/UB",$$FT^IBCEF(+X)=2:"/1500",1:"")  S ^UTILITY($J,"UB",CT)=9999999-IBT\_"^"\_+X\_"^"\_$P($G(^DGCR(399.3,+$P(IBND0,"^",7),0)),"^",4)\_"-"\_$$BCHGTYPE^IBCU(+X)\_"^"\_$P($P($P($P(^DD(399,.13,0),"^",3),$P(IBND0,"^",13)\_":",2),";",1),"/",1)  S ^UTILITY($J,"UB",CT)=^UTILITY($J,"UB",CT)\_"^"\_$S($P(IBND0,U,27)=1:"INST"\_IBFTP,$P(IBND0,U,27)=2:"PROF"\_IBFTP,1:"")  Q ST ; Do not use the variable IBH when calling this entry point  L ^DGCR(399,IBIFN):5 I '$T W !,"No further processing of this record permitted at this time.",!,"Record locked by another user. Try again later." Q  D RECALL^DILFD(399,IBIFN\_",",DUZ)  D NOPTF^IBCB2 I 'IBAC1 D NOPTF1^IBCB2 Q  I IBAC'=1&(IBAC'=4.1) G ST2 ST1 K ^UTILITY($J) S IBPOPOUT=0  ; Only allow view of bill waiting for MRA or pending extract  I $P($G(^DGCR(399,IBIFN,0)),U,13)=2 D  G Q  . W !,"This bill is requesting an MRA - can only view bill data"  . S IBV=1 D VIEW^IBCB2  I IBAC=4.1 D  G Q  . Q:$P($G(^DGCR(399,IBIFN,0)),U,13)'=3  . N Z  . S Z=$P($G(^IBA(364,+$$LAST364^IBCEF4(IBIFN),0)),U,3)  . I Z'="X"&(Z'="P") Q  . W !,"This bill has a transmit status of ",$$EXPAND^IBTRE(364,.03,Z)," - can only view bill data"  . S IBV=1 D VIEW^IBCB2  D ^IBCSCU,^IBCSC1 G Q:'$T!($G(IBPOPOUT)) ST2 K IBTXPRT,IBPOPOUT  D ^IBCB1 ; perform IB edits/authorize the bill  I $G(IBCIREDT) G ST1     ; Re-edit the bill  KILL IBCIREDT            ; clean up  QUIT  ; Q ;  K IBIFN,IBV,IBAC  ;\*\*\*  ;I $D(XRT0) S:'$D(XRTN) XRTN="IBCB" D T1^%ZOSV ;stop rt clock  Q  ; EDI S IBAC=1,IBV=0 D EN G Q:'IBAC1,EDI REV G Q AUT S IBAC=3,IBV=0 D EN G Q:'IBAC1,AUT GEN S IBAC=4,IBV=1 D EN G Q:'IBAC1,GEN VIEW S IBAC=4.1,IBV=1 D EN G Q:'IBAC1,VIEW  Q  ; | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCB ;ALB/MRL - BILLING BEGINNING POINT/SELECT BILL OR PATIENT ;01 JUN 88 12:00  ;;2.0;INTEGRATED BILLING;\*\*52,80,106,51,137,161,199,348,592\*\*;21-MAR-94;Build 5  ;;Per VHA Directive 10-93-142, this routine should not be modified.  ;  ;MAP TO DGCRB  ; EN ;  D HOME^%ZIS Q:'$D(IBAC)  ;\*\*\*  ;I $D(XRT0) S:'$D(XRTN) XRTN="IBCB" D T1^%ZOSV ;stop rt clock  ;S XRTL=$ZU(0),XRTN="IBCB-"\_$G(IBAC) D T0^%ZOSV ;start rt clock  ;  S:'$D(IBV) IBV=1 L  K ^UTILITY($J),DFN,IBIFN,DIC,IBPOPOUT S DIC(0)="EQMZ" R !!,"Enter BILL NUMBER or PATIENT NAME: ",IBX:DTIME I IBX["^"!(IBX="") S IBAC1=0 Q  K ^TMP("IBCRRX",$J)  S IBAC1=1  N DPTNOFZY S DPTNOFZY=1 ;Suppress PATIENT file fuzzy lookups  I IBX?1A4N!(IBX?2A.AP)!(IBX?2.A1",".AP)!(IBX?1A1P.AP) S DIC="^DPT(",X=IBX D ^DIC G EN:Y'>0 S DFN=+Y D HINQ S X=$S('$D(^DGCR(399,"C",DFN)):1,'$D(^DGCR(399,"AOP",DFN)):2,1:0)  I $D(DFN),X,IBAC<4 W !!,"No ",$S(X=1:"",1:"OPEN "),"billing records on file for this patient." D ASK I '$D(IBIFN) G EN  I $D(DFN) D  G EN  . D DATE:'$D(IBIFN),ASK:'$D(IBIFN)  . I $D(IBIFN) D ST  S DIC("S")=$S(IBAC'=4&(IBAC'=4.1):"I $P(^(0),U,13)<3 D EN^DDIOL($P(^(0),U))",1:"I $P(^(""S""),U,17)="""""\_$S(IBAC=4.1:",$P(^(0),U,13)=3,+$$LAST364^IBCEF4(+Y),""PX""[$P($G(^IBA(364,+$$LAST364^IBCEF4(+Y),0)),U,3)",1:""))  S DIC="^DGCR(399,",X=IBX  D ^DIC G:Y'>0 EN S IBIFN=+Y,DFN=$P(Y(0),"^",2)  ;  D HINQ,ST G EN  G EN HINQ I $S('$D(^DPT(DFN,.361)):1,$P(^(.361),"^",1)'="V":1,1:0) W !?17,"\*\*\* ELIGIBILITY NOT VERIFIED \*\*\*" D HINQ1 MT ;I $D(DFN) D ^DGMT1 K DGMTLL  I $D(DFN) D DIS^DGMTU(DFN)  Q HINQ1 I $P($G(^IBE(350.9,1,1)),"^",16) S X="DVBHQZ4" X ^%ZOSF("TEST") K X I $T W ! D EN^DVBHQZ4 Q  ;I $P($G(^IBE(350.9,1,1)),"^",16) F X="DVBHQZ4","DGHINQZ4" X ^%ZOSF("TEST") I $T S DGROUT=X K X W ! D @("EN^"\_DGROUT) K DGROUT Q  K Y Q ASK I IBAC'=1 K IBIFN Q  W !!,"DO YOU WANT TO ESTABLISH A NEW BILLING RECORD FOR '",$P(^DPT(DFN,0),"^",1),"'" S %=2 D YN^DICN  I '% W !!?4,"YES - To establish a new billing record in the billing file.",!?4,"NO - To discontinue this process immediately." G ASK  I %'=1 K IBIFN Q  K DA,Y,DINUM,IBIFN S (IBNEW,IBYN)=1 D ^IBCA Q DATE I $D(^DGCR(399,"C",DFN)) S DA="" F I=1:1 S DA=$O(^DGCR(399,"APDT",DFN,DA)) Q:DA=""  D DATE1  I IBAC=4,'$D(^UTILITY($J,"IB")) W !,"No ",$S($D(^DGCR(399,"C",DFN)):"UNCANCELLED ",1:""),"billing records on file for this patient." Q  S CT=0,CT1=1,IBT="" F J=1:1 S IBT=$O(^UTILITY($J,"IB",IBT)) Q:IBT=""  F J1=0:0 S J1=$O(^UTILITY($J,"IB",IBT,J1)) Q:J1=""  S X=J1 D SET CT W ! S G="",CT2=$S(CT<(CT1+4):CT,1:(CT1+4)) F K=CT1:1:CT2 I $D(^UTILITY($J,"UB",K)) D WRLINE  S X="" D WDATE Q:X["^"  I '$D(IB),$D(^UTILITY($J,"UB",K+1)) S CT1=K+1 G CT  K CT,CT1,CT2,K,^UTILITY($J,"UB") Q WRLINE N IBX S IBDATA=^UTILITY($J,"UB",K),IBX=$G(^DGCR(399,+$P(IBDATA,"^",2),0))  W !?2,K,?6 S Y=+IBDATA X ^DD("DD") W Y,?27,$P(IBX,"^",1),?35,$S($P(IBX,U,21)="S":"s",$P(IBX,U,21)="T":"t",1:""),?38,$P(IBDATA,"^",3),?59,$E($P(IBDATA,"^",4),1,10),?70,$E($P(IBDATA,"^",5),1,10)  Q DATE1 S IBT=$O(^DGCR(399,"APDT",DFN,DA,0)) I $D(^DGCR(399,+DA,0)),$S(IBAC<3:$P(^(0),U,13)<2,IBAC=3:$P(^(0),U,13)<3,'$D(^("S")):0,$P(^("S"),"^",17)]"":0,1:1) S ^UTILITY($J,"IB",IBT,DA)=""  Q WDATE Q:'CT  W !! W:K<CT "PRESS <RETURN> TO CONTINUE, OR",! W "CHOOSE 1",$S(CT=1:"",1:"-"\_K),": " R X:DTIME Q:X["^"!(X="") I X["?" W !!,"Select one of the above or <RETURN> to establish a new billing record." G WDATE  I $S('$D(^UTILITY($J,"UB",+X)):1,+X>K:1,+X<1:1,'(X?.N):1,1:0) W !!,"NOT A VALID CHOICE!!",\*7 G WDATE  S IBIFN=$P(^UTILITY($J,"UB",X),"^",2),IB=1  Q  ; KEYOK(IBIFN,DUZ) ; Check if COB bill, does user have key  ; IBIFN = ien of bill (file 399)  ;  N IBCOB,IBOK,DIR,X,Y  S IBOK=1,IBCOB=$$COBN^IBCEF(IBIFN)  I IBCOB>1 D  . S IBCOB=$P("^SECONDARY^TERTIARY",U,IBCOB)  . S DIR(0)="YA",DIR("A",1)="YOU ARE ABOUT TO EDIT A "\_IBCOB\_" BILL",DIR("A")="ARE YOU SURE YOU WANT TO CONTINUE?: ",DIR("B")="NO" W ! D ^DIR K DIR W !  . I Y'=1 S IBOK=0  Q IBOK  ; SET I $S(IBV:1,$P(^DGCR(399,+X,0),"^",13):1,1:0) S CT=CT+1 D SET2  Q SET2 S IBND0=^DGCR(399,+X,0)  N IBFTP  ; JWS;IB\*2.0\*592 US1108 - Dental EDI 837D / form J430D  S IBFTP=$S($$FT^IBCEF(+X)=3:"/UB",$$FT^IBCEF(+X)=2:"/1500",$$FT^IBCEF(+X)=7:"/J430D",1:"")  S ^UTILITY($J,"UB",CT)=9999999-IBT\_"^"\_+X\_"^"\_$P($G(^DGCR(399.3,+$P(IBND0,"^",7),0)),"^",4)\_"-"\_$$BCHGTYPE^IBCU(+X)\_"^"\_$P($P($P($P(^DD(399,.13,0),"^",3),$P(IBND0,"^",13)\_":",2),";",1),"/",1)  S ^UTILITY($J,"UB",CT)=^UTILITY($J,"UB",CT)\_"^"\_$S($P(IBND0,U,27)=1:"INST"\_IBFTP,$P(IBND0,U,27)=2:"PROF"\_IBFTP,1:"")  Q ST ; Do not use the variable IBH when calling this entry point  L ^DGCR(399,IBIFN):5 I '$T W !,"No further processing of this record permitted at this time.",!,"Record locked by another user. Try again later." Q  D RECALL^DILFD(399,IBIFN\_",",DUZ)  D NOPTF^IBCB2 I 'IBAC1 D NOPTF1^IBCB2 Q  I IBAC'=1&(IBAC'=4.1) G ST2 ST1 K ^UTILITY($J) S IBPOPOUT=0  ; Only allow view of bill waiting for MRA or pending extract  I $P($G(^DGCR(399,IBIFN,0)),U,13)=2 D  G Q  . W !,"This bill is requesting an MRA - can only view bill data"  . S IBV=1 D VIEW^IBCB2  I IBAC=4.1 D  G Q  . Q:$P($G(^DGCR(399,IBIFN,0)),U,13)'=3  . N Z  . S Z=$P($G(^IBA(364,+$$LAST364^IBCEF4(IBIFN),0)),U,3)  . I Z'="X"&(Z'="P") Q  . W !,"This bill has a transmit status of ",$$EXPAND^IBTRE(364,.03,Z)," - can only view bill data"  . S IBV=1 D VIEW^IBCB2  D ^IBCSCU,^IBCSC1 G Q:'$T!($G(IBPOPOUT)) ST2 K IBTXPRT,IBPOPOUT  D ^IBCB1 ; perform IB edits/authorize the bill  I $G(IBCIREDT) G ST1     ; Re-edit the bill  KILL IBCIREDT            ; clean up  QUIT  ; Q ;  K IBIFN,IBV,IBAC  ;\*\*\*  ;I $D(XRT0) S:'$D(XRTN) XRTN="IBCB" D T1^%ZOSV ;stop rt clock  Q  ; EDI S IBAC=1,IBV=0 D EN G Q:'IBAC1,EDI REV G Q AUT S IBAC=3,IBV=0 D EN G Q:'IBAC1,AUT GEN S IBAC=4,IBV=1 D EN G Q:'IBAC1,GEN VIEW S IBAC=4.1,IBV=1 D EN G Q:'IBAC1,VIEW  Q  ; | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCB1 | | | | | | | | |
| **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** | | | | | | | | | |
| IBCB1 ;ALB/AAS - Process bill after enter/edited ;2-NOV-89  ;;2.0;INTEGRATED BILLING;\*\*70,106,51,137,161,182,155,327,432\*\*;21-MAR-94;Build 192  ;;Per VHA Directive 10-93-142, this routine should not be modified.  ;  ;MAP TO DGCRB1  ;  ;IBQUIT = Flag to stop processing  ;IBVIEW = Flag for Bill has been viewed  ;IBDISP = Flag for Bill entering display been viewed.  ;  K ^UTILITY($J) I $D(IBAC),IBAC>1 G @IBAC 1 ;complete bill  D END,EDITS^IBCB2 G:IBQUIT END  ;  I '$$IICM^IBCB2(IBIFN) G END ; Ingenix ClaimsManager  I '$$IIQMED^IBCB2(IBIFN) G END ; DSS QuadraMed Claims Scrubber  ; 3 ;authorize bill/request MRA  I '$D(^XUSEC("IB AUTHORIZE",DUZ))!('$D(IBIFN)) W !!,"You do not hold the Authorize Key.",! G END  I '$P($G(^IBE(350.9,1,1)),"^",23),DUZ=$P(^DGCR(399,IBIFN,"S"),"^",2) W !!,"Entering user can not authorize.",! G END  I $P(^DGCR(399,IBIFN,"S"),"^",9) W !,"Already Approved, Can't change" G END  D:'$G(IBAC)!($G(IBAC)>1) EDITS^IBCB2 G:IBQUIT END  ;  I $G(IBAC)'=1,'$$IICM^IBCB2(IBIFN) G END ; Ingenix ClaimsManager  I $G(IBAC)'=1,'$$IIQMED^IBCB2(IBIFN) G END ; DSS QuadraMed Claims Scrubber  ; AUTH S IBMRA=$$REQMRA^IBEFUNC(IBIFN)  S IBEND=0  I IBMRA["R" D AUTH^IBCB11 G:IBEND END ;MRA normally required, but MEDIGAP ins co  ; doesn't want/need it or MRA parameter off  ;  W !!,"THIS BILL WILL "\_$P("NOT ^",U,$$TXMT^IBCEF4(IBIFN)+1)\_"BE TRANSMITTED ELECTRONICALLY"  W !!,"WANT TO ",$S('IBMRA:"AUTHORIZE BILL",1:"REQUEST AN MRA")," AT THIS TIME" S %=2 D YN^DICN G:%=-1!(%=2) END  I '% W !?4,"YES - If finished entering bill information and to allow bill to be printed or transmitted",!?4,"No - To take no action" G AUTH  S (DIC,DIE)=399,IBYY=$S('IBMRA:"@90",1:"@901"),DA=IBIFN,DR="[IB STATUS]" D ^DIE K DIC,DIE,IBYY D:$D(IBX3) DISAP^IBCBULL  I $S('IBMRA:'$P(^DGCR(399,IBIFN,"S"),"^",9),1:'$P($G(^DGCR(399,IBIFN,"TX")),U,6)) G END  ;  ; Update the review status for all EOB's on file  D STAT^IBCEMU2(IBIFN,3) ; Accepted - Complete EOB  ;  D AUTOCK^IBCEU2(IBIFN) ; Checks for need to add any codes to bill based on information already on bill, specifically for EDI purposes  S IBTXSTAT=$$TXMT^IBCEF4(IBIFN,,1) ;Determine transmit, whether live/test  I IBTXSTAT D  I IBMRA D CTCOPY^IBCCCB(IBIFN,1) G END  .W !," Adding "  .W:+IBTXSTAT=2 "test " W "bill to BILL TRANSMISSION File"\_$S('IBMRA:"",1:" for MRA submission")\_".",!  .W:+IBTXSTAT=1&IBMRA " Bill is no longer editable unless returned in error from Medicare."  .S Y=$$ADDTBILL(IBIFN,+IBTXSTAT)  .W ! W:'$P(Y,U,3) \*7 W $S($P(Y,U,3):" Bill will be submitted electronically",1:" Error loading into transmit file - bill can not be transmitted.")  .;  ;  W !,"Passing completed Bill to Accounts Receivable. Bill is no longer editable."  D ARPASS(IBIFN,1)  G:'$G(PRCASV("OKAY")) END  W !,"Completed Bill Successfully sent to Accounts Receivable." D FIND^IBOHCK(DFN,IBIFN)  ;  ; Check to see if any unreviewed status messages or EOBs on file and  ; what to do about them  N IBTXBARR  S IBRESUB=$$RESUB^IBCECSA4($S($G(IBCNCOPY):$P($G(^DGCR(399,IBIFN,0)),U,15),1:IBIFN),+IBTXSTAT,"E",.IBTXBARR)  I IBRESUB=2 D         ; update review statuses to be 'review complete'  . N IBDA S IBDA=0  . F  S IBDA=$O(IBTXBARR(IBDA)) Q:'IBDA  D UPDEDI^IBCEM(IBDA,$S($G(IBCNCOPY):"R",1:"E"))  . Q  ;  K IBTXPRT  ; 4 ;generate/print bill  G:'$D(IBIFN) END  S:'$D(IBMRA) IBMRA=+$$NEEDMRA^IBEFUNC(IBIFN)  I 'IBMRA,'$P(^DGCR(399,IBIFN,"S"),"^",9) W !!,\*7,"Not Authorized, Can Not Print!" G END  I IBMRA,'$P(^DGCR(399,IBIFN,"TX"),"^",6) W !!,\*7,"Not Ready For MRA Submission, Can Not Print!" G END  S IBTXSTAT=$$TXMT^IBCEF4(IBIFN)  I IBMRA,$$NEEDMRA^IBEFUNC(IBIFN)'["R" W !!,\*7,"MRA Submission not yet confirmed by Austin, Can Not Print!" Q:$S('IBTXSTAT:1,1:"XP"'[$P($G(^IBA(364,+$$LAST364^IBCEF4(IBIFN),0)),U,3))  I +IBTXSTAT,$D(^IBA(364,"ABDT",IBIFN)) S IBTXOK="" D  I 'IBTXOK S %=2 G GENTX  . N IBX,IBTST  . S IBX=+$$LAST364^IBCEF4(IBIFN),IBTST=""  . I $$TEST^IBCEF4(IBIFN) S (IBTXOK,IBTST)=1  . I "XP"[$P($G(^IBA(364,IBX,0)),U,3) D:'IBTST  Q  .. W !!,\*7,"This Bill Can Not Be Printed Until Transmit Confirmed" W:IBMRA " (to request an MRA)" D:'$D(IBVIEW) VIEW^IBCB2  . W !!,"This Bill Has Already Been Transmitted" W:IBMRA " (to request an MRA)"  . S DIR("B")="Y",DIR("A")="WANT TO PRINT IT ANYWAY",DIR(0)="Y" D ^DIR K DIR Q:$D(DTOUT)!$D(DUOUT)!'Y  S IBTXOK=1  D DISP^IBCB2  S:'$D(IBQUIT) IBQUIT=0  D:'$D(IBVIEW) VIEW^IBCB2 G:IBQUIT END  S IBPNT=$P(^DGCR(399,IBIFN,"S"),"^",12) GEN I $$TEST^IBCEF4(IBIFN) W !!,"THIS BILL IS BEING USED AS A TRANSMISSION TEST BILL"  W !!,"WANT TO ",$S(IBPNT]"":"RE-",1:""),"PRINT BILL AT THIS TIME" S %=2 D YN^DICN I %=-1 D:+$G(IBAC)=1 END,CTCOPY^IBCCCB(IBIFN) G END  I '% W !?4,"YES - to print the bill now",!?4,"NO - To take no action" G GEN GENTX I %'=1 D:+$G(IBAC)=1 END,CTCOPY^IBCCCB(IBIFN) G END  ;  ; Bill has never been printed. First time print.  I 'IBPNT D  G END  . I $D(IBTXPRT) D TXPRTS  . D EN1^IBCF  . I $D(IBTXPRT) D TXPRT  . ;D MRA^IBCEMU1(IBIFN) ; Printing the MRA ;WCJ;IB\*2.0\*432;MRA may have a diffierent claim number if this is tertiary  . D MRA^IBCEMU1($$GETMRACL^IBCAPR(IBIFN)) ;WCJ;IB\*2.0\*432;see above  . I $G(IBMRANOT) D EOBALL^IBCAPR2(IBIFN) ;WCJ;IB\*2.0\*432 print all the EOBs (ask device once)  . I +$G(IBAC)=1 D END,CTCOPY^IBCCCB(IBIFN)  . Q  ;  ; Below section is for re-prints RPNT G:$$NEEDMRA^IBEFUNC(IBIFN) END  R !!,"(2)nd Notice, (3)rd Notice, (C)opy or (O)riginal: C// ",IBPNT:DTIME S:IBPNT="" IBPNT="C" G:IBPNT["^" END  S IBPNT=$E(IBPNT,1) I "23oOcC"'[IBPNT W !?5,"Enter 'O' to reprint the original bill or",!?5,"Enter 'C' to reprint the bill as a duplicate copy or",!?5,"Enter '2' or '3' to print 2nd or 3rd follow-up notices." S IBPNT=1 G RPNT  W " (",$S("cC"[IBPNT:"COPY","oO"[IBPNT:"ORIGINAL",IBPNT=2:"2nd NOTICE",IBPNT=3:"3rd NOTICE",1:""),")"  I $D(IBTXPRT) D  . D TXPRTS  . I "oOcC"[IBPNT S IBRESUB=$$RESUB^IBCECSA4(IBIFN,1,"P")  S IBPNT=$S("oO"[IBPNT:1,"cC"[IBPNT:0,1:IBPNT)  D EN1X^IBCF D:$D(IBTXPRT) TXPRT  D MRA^IBCEMU1(IBIFN) ; Printing the MRA  ;  ; END K IBER,IBEND D END^IBCBB1 K IBQUIT,IBVIEW,IBDISP,IBST,IB,PRCAERCD,PRCAERR,PRCASVC,PRCAT,DGRA2,IBBT,IBCH,IBNDS,IBOA,IBREV,IBX,DGXRF1,PRCAORA,IBX3,DGBILLBS,DGII,DGVISCNT,DGFIL,DGTE,IBTXOK,IBTXSTAT,IBMRA,IBNOFIX  K %DT,DIC,DIE,I,J,X,Y,Y1,Y2,IBER,IBDFN,IBDSDT,IBJ,IBNDI1,IBZZ,VA,IBMA,IBXDT,DI,PRCAPAYR,DGBS,DGCNT,DGDA,DGPAG,DGREVC,DGRV,DGTEXT,DGTOTPAG,IBOPV,DGLCNT,DGTEXT1,DGRSPAC,DGSM,IBPNT,DGINPT,DGLL,IBCPTN,IBFL  K IBRESUB,IBOPV1,IBOPV2,IBCHG,DGBIL1,DGU,DDH,IBA1,IBINS,IBPROC,PRCARI K:'$D(PRCASV("NOTICE")) PRCASV  K ^TMP("IBXDATA",$J),^TMP("IBXEDIT",$J)  K IBCISNT,IBCISTAT,IBCIERR   ; remove ClaimsManager variables  Q  ; TX1(IBX,RESUB) ; Transmit a single bill from file 364 entry # IBX  ; RESUB = flag (1 = resubmitting a bill, 0 = submitting bill 1st time)  ; Returns 1 if successfully extracted to mailman queue for transmission,  ; 0 if extract not successful  N IBTXOK,IBVVSAVE  K ^TMP("IBRESUBMIT",$J),^TMP("IBONE",$J)  S IBVVSAVE("IBX")=IBX,^TMP("IBONE",$J)=+$G(RESUB),^($J,IBX)=""  D ONE^IBCE837  S IBX=IBVVSAVE("IBX")  I $P($G(^IBA(364,IBX,0)),U,3)="P" S IBTXOK=1  K ^TMP("IBONE",$J)  Q $G(IBTXOK)  ; ARONLY(IBIFN) ; Pass bill to A/R, but that's all  D ARPASS(IBIFN,0)  Q  ; ARPASS(IBIFN,UPDOK) ;Pass bill to A/R as NEW BILL  ;IBIFN = bill entry #  ;UPDOK = flag 1: if error going to A/R, allow interactive edit  ; 0: send bulletin to IB EDI for error going to A/R  Q:+$$STA^PRCAFN(+IBIFN)'=201 ;Must not have been sent previously  D GVAR^IBCBB  ;Can't be an ins co that won't reimburse  Q:$S($P($G(^DGCR(399,IBIFN,0)),U,11)="i":'IBNDMP,1:0)  D ARRAY^IBCBB1,^PRCASVC6  D REL^PRCASVC:$G(PRCASV("OKAY"))  I '$G(PRCASV("OKAY")) D  . N IBQUIT,IBQUIT1  . S IBQUIT=0  . I $G(UPDOK) D  Q  .. F  D  Q:IBQUIT  ... D DSPARERR^IBCB2("")  ... Q:IBQUIT  ... I $$ASKEDIT^IBCB2($G(IBAC)) D VIEW1^IBCB2 Q  ... S IBQUIT=1  . N XMSUB,XMY,XMTEXT,XMDUZ,IBT  . S XMSUB="ERROR PASSING BILL TO A/R ON CONFIRMATION",XMTEXT="IBT(",XMY="G.IB EDI",XMDUZ=.5  . S IBT(1)="A problem has been detected while trying to pass bill "\_$P($G(^DGCR(399,IBIFN,0)),U)\_" to"  . S IBT(2)="Accounts Receivable when updating the bill's electronic confirmation."  . S IBT(3)="Please use the option PASS BILL TO A/R to complete this process."  . D ^XMD  Q  ; ADDTBILL(IBIFN,TXST) ; Add new transmit bill rec to file 364 for bill IBIFN  ; TXST = test flag 1=live, 2=test  N COB,DD,DO,DIC,DLAYGO,X  S TXST=($G(TXST)/2\1),COB=$$COB^IBCEF(IBIFN)  S DIC(0)="L",DIC="^IBA(364,",DLAYGO=364,X=IBIFN,DIC("DR")=".03///X;.04///NOW;.07////"\_TXST\_";.08////"\_COB D FILE^DICN  Q Y  ; TXPRTS ; Save off last print date to see if bill was reprinted without queueing  I '$$NEEDMRA^IBEFUNC(IBIFN) S IBTXPRT("PRT")=$P($G(^DGCR(399,IBIFN,"S")),U,14)  Q  ; TXPRT ; Set variable if print was tasked or bill was printed (last print date changed)  I '$$NEEDMRA^IBEFUNC(IBIFN),$S($G(ZTSK):1,1:IBTXPRT("PRT")'=$P($G(^DGCR(399,IBIFN,"S")),U,14)) S IBTXPRT=1  Q  ; | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCB1 ;ALB/AAS - Process bill after enter/edited ;2-NOV-89  ;;2.0;INTEGRATED BILLING;\*\*70,106,51,137,161,182,155,327,432,**592**\*\*;21-MAR-94;Build 192  ;;Per VHA Directive 10-93-142, this routine should not be modified.  ;  ;MAP TO DGCRB1  ;  ;IBQUIT = Flag to stop processing  ;IBVIEW = Flag for Bill has been viewed  ;IBDISP = Flag for Bill entering display been viewed.  ;  K ^UTILITY($J) I $D(IBAC),IBAC>1 G @IBAC 1 ;complete bill  D END,EDITS^IBCB2 G:IBQUIT END  ;  I '$$IICM^IBCB2(IBIFN) G END ; Ingenix ClaimsManager  I '$$IIQMED^IBCB2(IBIFN) G END ; DSS QuadraMed Claims Scrubber  ; 3 ;authorize bill/request MRA  I '$D(^XUSEC("IB AUTHORIZE",DUZ))!('$D(IBIFN)) W !!,"You do not hold the Authorize Key.",! G END  I '$P($G(^IBE(350.9,1,1)),"^",23),DUZ=$P(^DGCR(399,IBIFN,"S"),"^",2) W !!,"Entering user can not authorize.",! G END  I $P(^DGCR(399,IBIFN,"S"),"^",9) W !,"Already Approved, Can't change" G END  D:'$G(IBAC)!($G(IBAC)>1) EDITS^IBCB2 G:IBQUIT END  ;  I $G(IBAC)'=1,'$$IICM^IBCB2(IBIFN) G END ; Ingenix ClaimsManager  I $G(IBAC)'=1,'$$IIQMED^IBCB2(IBIFN) G END ; DSS QuadraMed Claims Scrubber  ; AUTH S IBMRA=$$REQMRA^IBEFUNC(IBIFN)  S IBEND=0  I IBMRA["R" D AUTH^IBCB11 G:IBEND END ;MRA normally required, but MEDIGAP ins co  ; doesn't want/need it or MRA parameter off  ;  W !!,"THIS BILL WILL "\_$P("NOT ^",U,$$TXMT^IBCEF4(IBIFN)+1)\_"BE TRANSMITTED ELECTRONICALLY"  W !!,"WANT TO ",$S('IBMRA:"AUTHORIZE BILL",1:"REQUEST AN MRA")," AT THIS TIME" S %=2 D YN^DICN G:%=-1!(%=2) END  I '% W !?4,"YES - If finished entering bill information and to allow bill to be printed or transmitted",!?4,"No - To take no action" G AUTH  S (DIC,DIE)=399,IBYY=$S('IBMRA:"@90",1:"@901"),DA=IBIFN,DR="[IB STATUS]" D ^DIE K DIC,DIE,IBYY D:$D(IBX3) DISAP^IBCBULL  I $S('IBMRA:'$P(^DGCR(399,IBIFN,"S"),"^",9),1:'$P($G(^DGCR(399,IBIFN,"TX")),U,6)) G END  ;  ; Update the review status for all EOB's on file  D STAT^IBCEMU2(IBIFN,3) ; Accepted - Complete EOB  ;  D AUTOCK^IBCEU2(IBIFN) ; Checks for need to add any codes to bill based on information already on bill, specifically for EDI purposes  S IBTXSTAT=$$TXMT^IBCEF4(IBIFN,,1) ;Determine transmit, whether live/test  I IBTXSTAT D  I IBMRA D CTCOPY^IBCCCB(IBIFN,1) G END  .W !," Adding "  .W:+IBTXSTAT=2 "test " W "bill to BILL TRANSMISSION File"\_$S('IBMRA:"",1:" for MRA submission")\_".",!  .W:+IBTXSTAT=1&IBMRA " Bill is no longer editable unless returned in error from Medicare."  .S Y=$$ADDTBILL(IBIFN,+IBTXSTAT)  .W ! W:'$P(Y,U,3) \*7 W $S($P(Y,U,3):" Bill will be submitted electronically",1:" Error loading into transmit file - bill can not be transmitted.")  .;  ;  W !,"Passing completed Bill to Accounts Receivable. Bill is no longer editable."  D ARPASS(IBIFN,1)  G:'$G(PRCASV("OKAY")) END  W !,"Completed Bill Successfully sent to Accounts Receivable." D FIND^IBOHCK(DFN,IBIFN)  ;  ; Check to see if any unreviewed status messages or EOBs on file and  ; what to do about them  N IBTXBARR  S IBRESUB=$$RESUB^IBCECSA4($S($G(IBCNCOPY):$P($G(^DGCR(399,IBIFN,0)),U,15),1:IBIFN),+IBTXSTAT,"E",.IBTXBARR)  I IBRESUB=2 D         ; update review statuses to be 'review complete'  . N IBDA S IBDA=0  . F  S IBDA=$O(IBTXBARR(IBDA)) Q:'IBDA  D UPDEDI^IBCEM(IBDA,$S($G(IBCNCOPY):"R",1:"E"))  . Q  ;  K IBTXPRT  ; 4 ;generate/print bill  G:'$D(IBIFN) END  S:'$D(IBMRA) IBMRA=+$$NEEDMRA^IBEFUNC(IBIFN)  I 'IBMRA,'$P(^DGCR(399,IBIFN,"S"),"^",9) W !!,\*7,"Not Authorized, Can Not Print!" G END  I IBMRA,'$P(^DGCR(399,IBIFN,"TX"),"^",6) W !!,\*7,"Not Ready For MRA Submission, Can Not Print!" G END  S IBTXSTAT=$$TXMT^IBCEF4(IBIFN)  I IBMRA,$$NEEDMRA^IBEFUNC(IBIFN)'["R" W !!,\*7,"MRA Submission not yet confirmed by Austin, Can Not Print!" Q:$S('IBTXSTAT:1,1:"XP"'[$P($G(^IBA(364,+$$LAST364^IBCEF4(IBIFN),0)),U,3))  I +IBTXSTAT,$D(^IBA(364,"ABDT",IBIFN)) S IBTXOK="" D  I 'IBTXOK S %=2 G GENTX  . N IBX,IBTST  . S IBX=+$$LAST364^IBCEF4(IBIFN),IBTST=""  . I $$TEST^IBCEF4(IBIFN) S (IBTXOK,IBTST)=1  . I "XP"[$P($G(^IBA(364,IBX,0)),U,3) D:'IBTST  Q **.. ;JWS;IB\*2.0\*592  .. I $$FT^IBCEF(IBIFN)=7 W !!,\*7,"This Bill Can Not Be Printed"**  .. E  W !!,\*7,"This Bill Can Not Be Printed Until Transmit Confirmed"  .. W:IBMRA " (to request an MRA)" D:'$D(IBVIEW) VIEW^IBCB2  . W !!,"This Bill Has Already Been Transmitted" W:IBMRA " (to request an MRA)"  . S DIR("B")="Y",DIR("A")="WANT TO PRINT IT ANYWAY",DIR(0)="Y" D ^DIR K DIR Q:$D(DTOUT)!$D(DUOUT)!'Y  S IBTXOK=1  D DISP^IBCB2  S:'$D(IBQUIT) IBQUIT=0  D:'$D(IBVIEW) VIEW^IBCB2 G:IBQUIT END  S IBPNT=$P(^DGCR(399,IBIFN,"S"),"^",12) GEN I $$TEST^IBCEF4(IBIFN) W !!,"THIS BILL IS BEING USED AS A TRANSMISSION TEST BILL"  W !!,"WANT TO ",$S(IBPNT]"":"RE-",1:""),"PRINT BILL AT THIS TIME" S %=2 D YN^DICN I %=-1 D:+$G(IBAC)=1 END,CTCOPY^IBCCCB(IBIFN) G END  I '% W !?4,"YES - to print the bill now",!?4,"NO - To take no action" G GEN  **;JWS;IB\*2.0\*592**  **I %=1,$$FT^IBCEF(IBIFN)=7 W !!,\*7,"Dental Claims can not be printed." G END**  GENTX I %'=1 D:+$G(IBAC)=1 END,CTCOPY^IBCCCB(IBIFN) G END  ;  ; Bill has never been printed. First time print.  I 'IBPNT D  G END  . I $D(IBTXPRT) D TXPRTS  . D EN1^IBCF  . I $D(IBTXPRT) D TXPRT  . ;D MRA^IBCEMU1(IBIFN) ; Printing the MRA ;WCJ;IB\*2.0\*432;MRA may have a diffierent claim number if this is tertiary  . D MRA^IBCEMU1($$GETMRACL^IBCAPR(IBIFN)) ;WCJ;IB\*2.0\*432;see above  . I $G(IBMRANOT) D EOBALL^IBCAPR2(IBIFN) ;WCJ;IB\*2.0\*432 print all the EOBs (ask device once)  . I +$G(IBAC)=1 D END,CTCOPY^IBCCCB(IBIFN)  . Q  ;  ; Below section is for re-prints RPNT G:$$NEEDMRA^IBEFUNC(IBIFN) END  R !!,"(2)nd Notice, (3)rd Notice, (C)opy or (O)riginal: C// ",IBPNT:DTIME S:IBPNT="" IBPNT="C" G:IBPNT["^" END  S IBPNT=$E(IBPNT,1) I "23oOcC"'[IBPNT W !?5,"Enter 'O' to reprint the original bill or",!?5,"Enter 'C' to reprint the bill as a duplicate copy or",!?5,"Enter '2' or '3' to print 2nd or 3rd follow-up notices." S IBPNT=1 G RPNT  W " (",$S("cC"[IBPNT:"COPY","oO"[IBPNT:"ORIGINAL",IBPNT=2:"2nd NOTICE",IBPNT=3:"3rd NOTICE",1:""),")"  I $D(IBTXPRT) D  . D TXPRTS  . I "oOcC"[IBPNT S IBRESUB=$$RESUB^IBCECSA4(IBIFN,1,"P")  S IBPNT=$S("oO"[IBPNT:1,"cC"[IBPNT:0,1:IBPNT)  D EN1X^IBCF D:$D(IBTXPRT) TXPRT  D MRA^IBCEMU1(IBIFN) ; Printing the MRA  ;  ; END K IBER,IBEND D END^IBCBB1 K IBQUIT,IBVIEW,IBDISP,IBST,IB,PRCAERCD,PRCAERR,PRCASVC,PRCAT,DGRA2,IBBT,IBCH,IBNDS,IBOA,IBREV,IBX,DGXRF1,PRCAORA,IBX3,DGBILLBS,DGII,DGVISCNT,DGFIL,DGTE,IBTXOK,IBTXSTAT,IBMRA,IBNOFIX  K %DT,DIC,DIE,I,J,X,Y,Y1,Y2,IBER,IBDFN,IBDSDT,IBJ,IBNDI1,IBZZ,VA,IBMA,IBXDT,DI,PRCAPAYR,DGBS,DGCNT,DGDA,DGPAG,DGREVC,DGRV,DGTEXT,DGTOTPAG,IBOPV,DGLCNT,DGTEXT1,DGRSPAC,DGSM,IBPNT,DGINPT,DGLL,IBCPTN,IBFL  K IBRESUB,IBOPV1,IBOPV2,IBCHG,DGBIL1,DGU,DDH,IBA1,IBINS,IBPROC,PRCARI K:'$D(PRCASV("NOTICE")) PRCASV  K ^TMP("IBXDATA",$J),^TMP("IBXEDIT",$J)  K IBCISNT,IBCISTAT,IBCIERR   ; remove ClaimsManager variables  Q  ; TX1(IBX,RESUB) ; Transmit a single bill from file 364 entry # IBX  ; RESUB = flag (1 = resubmitting a bill, 0 = submitting bill 1st time)  ; Returns 1 if successfully extracted to mailman queue for transmission,  ; 0 if extract not successful  N IBTXOK,IBVVSAVE  K ^TMP("IBRESUBMIT",$J),^TMP("IBONE",$J)  S IBVVSAVE("IBX")=IBX,^TMP("IBONE",$J)=+$G(RESUB),^($J,IBX)=""  D ONE^IBCE837  S IBX=IBVVSAVE("IBX")  I $P($G(^IBA(364,IBX,0)),U,3)="P" S IBTXOK=1  K ^TMP("IBONE",$J)  Q $G(IBTXOK)  ; ARONLY(IBIFN) ; Pass bill to A/R, but that's all  D ARPASS(IBIFN,0)  Q  ; ARPASS(IBIFN,UPDOK) ;Pass bill to A/R as NEW BILL  ;IBIFN = bill entry #  ;UPDOK = flag 1: if error going to A/R, allow interactive edit  ; 0: send bulletin to IB EDI for error going to A/R  Q:+$$STA^PRCAFN(+IBIFN)'=201 ;Must not have been sent previously  D GVAR^IBCBB  ;Can't be an ins co that won't reimburse  Q:$S($P($G(^DGCR(399,IBIFN,0)),U,11)="i":'IBNDMP,1:0)  D ARRAY^IBCBB1,^PRCASVC6  D REL^PRCASVC:$G(PRCASV("OKAY"))  I '$G(PRCASV("OKAY")) D  . N IBQUIT,IBQUIT1  . S IBQUIT=0  . I $G(UPDOK) D  Q  .. F  D  Q:IBQUIT  ... D DSPARERR^IBCB2("")  ... Q:IBQUIT  ... I $$ASKEDIT^IBCB2($G(IBAC)) D VIEW1^IBCB2 Q  ... S IBQUIT=1  . N XMSUB,XMY,XMTEXT,XMDUZ,IBT  . S XMSUB="ERROR PASSING BILL TO A/R ON CONFIRMATION",XMTEXT="IBT(",XMY="G.IB EDI",XMDUZ=.5  . S IBT(1)="A problem has been detected while trying to pass bill "\_$P($G(^DGCR(399,IBIFN,0)),U)\_" to"  . S IBT(2)="Accounts Receivable when updating the bill's electronic confirmation."  . S IBT(3)="Please use the option PASS BILL TO A/R to complete this process."  . D ^XMD  Q  ; ADDTBILL(IBIFN,TXST) ; Add new transmit bill rec to file 364 for bill IBIFN  ; TXST = test flag 1=live, 2=test  N COB,DD,DO,DIC,DLAYGO,X  S TXST=($G(TXST)/2\1),COB=$$COB^IBCEF(IBIFN)  S DIC(0)="L",DIC="^IBA(364,",DLAYGO=364,X=IBIFN,DIC("DR")=".03///X;.04///NOW;.07////"\_TXST\_";.08////"\_COB D FILE^DICN  Q Y  ; TXPRTS ; Save off last print date to see if bill was reprinted without queueing  I '$$NEEDMRA^IBEFUNC(IBIFN) S IBTXPRT("PRT")=$P($G(^DGCR(399,IBIFN,"S")),U,14)  Q  ; TXPRT ; Set variable if print was tasked or bill was printed (last print date changed)  I '$$NEEDMRA^IBEFUNC(IBIFN),$S($G(ZTSK):1,1:IBTXPRT("PRT")'=$P($G(^DGCR(399,IBIFN,"S")),U,14)) S IBTXPRT=1  Q  ; | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCB2 | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** |  | | | | | | | | |
| **Related Options** | None | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  |  | | | | |  | | | |
| **Data Dictionary (DD) References** | CLAIMS TRACKING File [#356] | | | | | | | | |
| **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** | | | | | | | | | |
| IBCB2 ;ALB/AAS - Process bill after enter/edited ;13-DEC-89  ;;2.0;INTEGRATED BILLING;\*\*52,51,161,182,155,447\*\*;21-MAR-94;Build 80  ;;Per VHA Directive 10-93-142, this routine should not be modified.  ;  ;MAP TO DGCRB2  ;  ;IBQUIT = Flag to stop processing  ;IBVIEW = Flag showing Bill has been viewed  ;IBDISP = Flag showing Bill entering display has been viewed.  ;IBNOFIX = Flag to indicate do not ask to edit or review bill screens  ;IBREEDIT = Flag to indicate Bill has been re-edited  ; VIEW ;View screens; if status allows editing , allow editing  N Y,DIR  S IBPOPOUT=0  S IBVIEW=1,IBV=$S($D(IBV):IBV,1:1)  S DIR(0)="YA",DIR("B")="NO",DIR("A")="WANT TO "\_$S('IBV:"EDIT",1:"REVIEW")\_" SCREENS? ",DIR("?",1)=" YES - to "\_$S('IBV:"EDIT",1:"REVIEW")\_" the screens",DIR("?")=" NO - To take no action"  D ^DIR K DIR  S:$D(DTOUT) IBQUIT=1  Q:Y'=1  I $G(IBREEDIT)=1,'IBV S IBREEDIT=2 ; set flag indicating re-edit VIEW1 S IBVIEW=1,IBEDIT=0  D SCREENS  S:$G(IBPOPOUT) IBQUIT=1  Q  ; DISP S IB("S")=$S($D(^DGCR(399,IBIFN,"S")):^("S"),1:"")  W ! D DISP^IBCNQ W !  S IBDISP=1 Q  Q  ; EDITS ; Perform edits on bill prior to authorization/transmission  N IBREEDIT ED1 ;  S IBQUIT=0  I '$D(IBER)!('$D(PRCASV)) D ALLED(.IBQUIT)  ;  ; If the user is wanting to quit, but there are some unresolved  ; errors reported by ClaimsManager, then capture the user's Exit  ; comments.  ;  I $$CM^IBCIUT1(IBIFN),IBQUIT,$P($G(^IBA(351.9,IBIFN,0)),U,2)=4 D COMMENT^IBCIUT7(IBIFN,1)  ;  Q:IBQUIT  D:'$D(IBDISP) DISP  ;  ; If claim re-edit, then call the IB edit checks again  I '$D(IBVIEW) S IBREEDIT=1 D VIEW I $G(IBREEDIT)=2 K IBER,IBDISP,IBVIEW G ED1  Q  ; ALLED(IBQUIT) ; Billing edit/correction  N IBQUIT1,IBDONE1,IBDONE,IBEDIT,IBCORR,IBER,IBPRT,IBXERR  S (IBQUIT,IBDONE,IBCORR)=0,IBER=""  ; IBDONE = 1 ==> exit, no errors   ; IBQUIT = 1 ==> exit, errors not corrected  I $$FT^IBCEF(IBIFN)=2,'$G(IBNOFIX) D DISP24(IBIFN,.IBCORR,.IBQUIT)  ;  F  D  Q:IBQUIT!IBDONE  D VIEW1 I $$FT^IBCEF(IBIFN)=2,'$G(IBNOFIX),'IBQUIT S IBCORR=0 D DISP24(IBIFN,.IBCORR,.IBQUIT)  . I $G(IBPOPOUT) S IBQUIT=1  . Q:IBQUIT!IBCORR  . I $G(IBNOFIX) D  .. W !!,"... Checking claim validity"  . E  D  .. W !!,"... Executing national IB edits"  . D EN^IBCBB,LOCERR  . ;  . I $G(IBER)'=""!$D(IBXERR) D  Q:'IBDONE  .. D DSPLERR ; Displays warnings/errors  .. K IBXERR  .. Q:IBQUIT!(IBDONE)  .. I $G(IBNOFIX) S IBDONE=1 Q  .. I '$$ASKEDIT($G(IBAC)) W ! S IBQUIT=1 ; Don't want to re-edit  .. ;  . I $G(IBNOFIX) S IBDONE=1 Q  . S IBEDIT=0  . I $S($P($G(^DGCR(399,IBIFN,0)),U,13)>2:1,$D(PRCASV):'$D(PRCASV("OKAY")),1:0) D  S:'IBQUIT&'IBEDIT IBDONE=1 Q  .. N IBQUIT1  .. S IBQUIT1=0  .. W !!!,"... Executing A/R edits"  .. I $P($G(^DGCR(399,IBIFN,0)),U,13)>2 D GVAR^IBCBB,ARRAY^IBCBB1  .. D ARCHK($G(IBNOFIX),0,.IBQUIT1,.IBQUIT,.IBEDIT,.PRCASV)  . S IBDONE=1 ; No errors  . S:$G(IBPRT("PRT"))'<0 IBQUIT=0  Q  ; ARCHK(IBNOFIX,IBNOPRT,IBQUIT1,IBQUIT,IBEDIT,PRCASV) ; A/R Verification  ; Returns IBEDIT, IBQUIT1, IBQUIT,PRCASV array if passed by reference  ; IBNOFIX = 1 if no editing needed  ; IBNOPRT = 1 if no printing needed  F  D ^PRCASVC6 D  Q:IBQUIT1!IBEDIT  D GVAR^IBCBB,ARRAY^IBCBB1  . I '$G(IBNOPRT) Q:$G(IBPRT("PRT"))<0  . I PRCASV("OKAY") W:'$G(IBNOPRT) !!,"No A/R errors found" S IBQUIT1=1 Q  . I 'PRCASV("OKAY") D  Q  .. D DSPARERR($G(IBNOPRT)) ; Display A/R errors  .. Q:IBQUIT  .. I $G(IBNOFIX) S IBQUIT1=1 Q  .. I '$$ASKEDIT($G(IBAC)) W !,"There is an unresolved A/R error - cannot authorize bill" D PAUSE^VALM1 S (IBQUIT,IBQUIT1)=1 Q  .. S IBEDIT=1  ;  Q  ; DSPLERR ; Display national/local edits failed  N Z  D PRTH(.IBPRT)  I IBPRT("PRT")<0 S IBQUIT=1 Q  S Z=0 F  S Z=$O(^TMP($J,"BILL-WARN",Z)) Q:'Z  W !,^(Z) W:'$O(^(Z)) !  S Y2=""  I IBER'="WARN" F I=1:1 S X=$P(IBER,";",I) Q:X=""  W:I=1 !?5,"\*\*Errors\*\*:" I $D(^IBE(350.8,+$O(^IBE(350.8,"AC",X,0)),0)) S Y=^(0),Y1=$P(Y,"^",5),Y2=Y2\_Y1 I Y1<5 W !?5,$E($P(Y,"^",2),1,80)  ; IBXERR = local edits return error array  ; If IBXERR returns = 1 then we have at least one error  ; = "" or 0, then we have only local warnings  ; undefined = no local errors or warnings  I $D(IBXERR) D  . S I="" W !!,?3,"Local Edits:"  . S:$G(IBXERR) Y2=3,IBER="L"  . F  S I=$O(IBXERR(I)) Q:I=""  W !,?5,$E(IBXERR(I),1,75)  I $G(IBPRT("PRT")) D CLOSE(.IBPRT)  G:$G(IBNOFIX) Q  I $G(IBER)="WARN"!($G(IBXERR)=0) D  ;Warnings only - make biller stop and look  . W !  . N DIR,X,Y  . S DIR(0)="YA",DIR("B")="NO",DIR("A",1)="THIS BILL STILL HAS ONE OR MORE WARNINGS - PLEASE REVIEW THEM CAREFULLY",DIR("A")="ARE YOU SURE IT'S OK TO CONTINUE? "  . D ^DIR K DIR  . I Y'=1 S Y2=3 Q  . S IBER="",IBDONE=1 K IBXERR  I $S(Y2'["3"&'$G(IBXERR):0,1:1) K IBXERR Q K ^TMP($J,"BILL-WARN")  Q  ; DSPARERR(IBNOPRT) ; Displays A/R errors  N I,J,Y,X,ERRPRT  I '$G(IBNOPRT) D PRTH(.IBPRT) I IBPRT("PRT")<0 S IBQUIT=1 Q  I $P($G(PRCAERR),U,2)'="" D  . N Z  . S Z=+$O(^IBE(350.8,"C",$P(PRCAERR,U,2),0)),Z=$P($G(^IBE(350.8,+Z,0)),U,2)  . W !,?5,"An A/R error has been reported - bill cannot be authorized",!!,?5,$P(PRCAERR,U,2)," - ",$S(Z'="":Z,1:"??")  E  D  . W !,?5,"An undetermined A/R error was found - "\_$G(PRCAERR)  I $G(IBPRT("PRT")) D CLOSE(.IBPRT)  Q  ; NOPTF S IBAC1=1 I $D(^DGCR(399,IBIFN,0)),$P(^(0),"^",8),'$D(^DGPT($P(^(0),"^",8),0)) S IBAC1=0  Q  ; NOPTF1 W !!,\*7,"PTF Record for this Bill was DELETED!",!,"Further processing not allowed. Cancel and re-enter." Q  ; LOCERR ; Check for local edits  ; Execute screen post-processor for bills with local scrn 9 affiliations  N IBZ,IBXIEN,IBPRT  K IBXERR  S IBZ=$$LOCSCRN^IBCSC11(IBIFN) ; IB\*2.0\*447 BI  I IBZ S IBXIEN=IBIFN W !!,"... Executing local IB edits" D FPOST^IBCEFG7(IBZ,0,.IBXERR) I '$D(IBXERR) W !!,"No errors found for local edits"  Q  ; PRTH(IBPRT,IBA) ; Print a heading for error/warnings sent to a printer  ; Returns IBPRT = 1 if valid pritner selected  ; IBPRT = -1 if '^' entered  ; IBPRT = 0 if home device  N POP,%ZIS,POP  S %ZIS("A")="ERROR/WARNING OUTPUT DEVICE: "  D ^%ZIS  I POP S IBPRT("PRT")=-1 Q  I IO=IO(0) S IBPRT("PRT")=0 Q  S IBPRT("PRT")=1  U IO  W !,"INCONSISTENCIES LIST FOR BILL #: ",$P($G(^DGCR(399,IBIFN,0)),U),!,$J("",29),"AT: ",$$FMTE^XLFDT($$NOW^XLFDT,2),!,$J("",19),"GENERATED BY: ",$P($G(^VA(200,DUZ,0)),U),!!  Q  ; CLOSE(IBPRT) ; Close device, reset printer flag  D ^%ZISC  S IBPRT("PRT")=0  D HOME^%ZIS  Q  ; ASKEDIT(IBAC) ; Ask if edit/review of bill is desired  ; FUNCTION returns 0/1 for NO/YES  ; IBAC = flag for function being performed - to determine edit/review  N DIR,X,Y  S DIR(0)="YA"  S DIR("A",1)=" ",DIR("A",2)=" ",DIR("A")="Do you wish to "\_$S($G(IBAC)<4:"edit",1:"review")\_" the inconsistencies now? ",DIR("B")="NO"  S DIR("?",1)=" ",DIR("?",2)=" ",DIR("?",3)=" YES - To edit inconsistent fields",DIR("?")=" NO - To discontinue this process."  D ^DIR K DIR  Q (Y=1)  ; SCREENS ;  N IBH  D ^IBCSCU,^IBCSC1  I $G(IBV) K IBPOPOUT  Q  ; DISP24(IBIFN,IBCORR,IBQUIT) ;  W @IOF D BL24^IBCSCH(IBIFN,0)  S DIR("A",1)=" ",DIR("A")="Are the above charges correct for this bill? ",DIR("B")="YES",DIR(0)="YA" D ^DIR K DIR  I Y'=1 D  . I Y=0,$$ASKEDIT($G(IBAC)) S IBCORR=1 Q  . S IBQUIT=1  Q  ; IICM(IBIFN) ; Ingenix ClaimsManager: Claim Scrubber  ; Send the bill to ClaimsManager, the IBCISTAT variable returned from ClaimsManager indicates  ; 3 - Passed CM with no errors  ; 5 - User overriding the CM errors  ; 7 - the CM interface isn't working  ; 11 - User overriding the CM errors (CM not updated)  ;   ; Returns False (0) if the bill fails the ClaimsManager Scrubber/errors found  ; Returns True (1) if the bill passed the ClaimsManager Scrubber/no errors found or ClaimsManager not On at site  ;  N IBOK S IBOK=1  I +$G(IBIFN),$$CM^IBCIUT1(IBIFN) S IBCISNT=1 D ST2^IBCIST I '$F(".3.5.7.11.","."\_IBCISTAT\_".") S IBOK=0  Q IBOK  ; IIQMED(IBIFN) ; DSS QuadraMed Interface: QuadraMed Claim Scrubber  ; Send the bill to the QuadraMed Claim Scrubber  ; Returns False (0) if the bill fails the QuadraMed Scrubber/errors found  ; Returns True (1) if the bill passed the QuadraMed Scrubber/no errors found or QuadraMed not On at site  ;  ; QuadraMed Scrubber EN^VEJDIBSC returns IBQMED = 1 if no error found, returns 0 if error found  ;  N IBQMED S IBQMED=1  I +$G(IBIFN),$$QMED^IBCU1("EN^VEJDIBSC",IBIFN) D EN^VEJDIBSC(IBIFN)  Q IBQMED | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCB2 ;ALB/AAS - Process bill after enter/edited ;13-DEC-89  ;;2.0;INTEGRATED BILLING;\*\*52,51,161,182,155,447,592\*\*;21-MAR-94;Build 80  ;;Per VHA Directive 10-93-142, this routine should not be modified.  ;  ;MAP TO DGCRB2  ;  ;IBQUIT = Flag to stop processing  ;IBVIEW = Flag showing Bill has been viewed  ;IBDISP = Flag showing Bill entering display has been viewed.  ;IBNOFIX = Flag to indicate do not ask to edit or review bill screens  ;IBREEDIT = Flag to indicate Bill has been re-edited  ; VIEW ;View screens; if status allows editing , allow editing  N Y,DIR  S IBPOPOUT=0  S IBVIEW=1,IBV=$S($D(IBV):IBV,1:1)  S DIR(0)="YA",DIR("B")="NO",DIR("A")="WANT TO "\_$S('IBV:"EDIT",1:"REVIEW")\_" SCREENS? ",DIR("?",1)=" YES - to "\_$S('IBV:"EDIT",1:"REVIEW")\_" the screens",DIR("?")=" NO - To take no action"  D ^DIR K DIR  S:$D(DTOUT) IBQUIT=1  Q:Y'=1  I $G(IBREEDIT)=1,'IBV S IBREEDIT=2 ; set flag indicating re-edit VIEW1 S IBVIEW=1,IBEDIT=0  D SCREENS  S:$G(IBPOPOUT) IBQUIT=1  Q  ; DISP S IB("S")=$S($D(^DGCR(399,IBIFN,"S")):^("S"),1:"")  W ! D DISP^IBCNQ W !  S IBDISP=1 Q  Q  ; EDITS ; Perform edits on bill prior to authorization/transmission  N IBREEDIT ED1 ;  S IBQUIT=0  I '$D(IBER)!('$D(PRCASV)) D ALLED(.IBQUIT)  ;  ; If the user is wanting to quit, but there are some unresolved  ; errors reported by ClaimsManager, then capture the user's Exit  ; comments.  ;  I $$CM^IBCIUT1(IBIFN),IBQUIT,$P($G(^IBA(351.9,IBIFN,0)),U,2)=4 D COMMENT^IBCIUT7(IBIFN,1)  ;  Q:IBQUIT  D:'$D(IBDISP) DISP  ;  ; If claim re-edit, then call the IB edit checks again  I '$D(IBVIEW) S IBREEDIT=1 D VIEW I $G(IBREEDIT)=2 K IBER,IBDISP,IBVIEW G ED1  Q  ; ALLED(IBQUIT) ; Billing edit/correction  N IBQUIT1,IBDONE1,IBDONE,IBEDIT,IBCORR,IBER,IBPRT,IBXERR  S (IBQUIT,IBDONE,IBCORR)=0,IBER=""  ; IBDONE = 1 ==> exit, no errors   ; IBQUIT = 1 ==> exit, errors not corrected  ;JWS;IB\*2.0\*592:Dental form #7 don't display Box 24 info for dental  I $$FT^IBCEF(IBIFN)=2,'$G(IBNOFIX) D DISP24(IBIFN,.IBCORR,.IBQUIT)  ;JWS;IB\*2.0\*592:Dental form #7 do same as CMS-1500  F  D  Q:IBQUIT!IBDONE  D VIEW1 I $$FT^IBCEF(IBIFN)=2!($$FT^IBCEF(IBIFN)=7),'$G(IBNOFIX),'IBQUIT S IBCORR=0 D:$$FT^IBCEF(IBIFN)'=7 DISP24(IBIFN,.IBCORR,.IBQUIT)  . I $G(IBPOPOUT) S IBQUIT=1  . Q:IBQUIT!IBCORR  . I $G(IBNOFIX) D  .. W !!,"... Checking claim validity"  . E  D  .. W !!,"... Executing national IB edits"  . D EN^IBCBB,LOCERR  . ;  . I $G(IBER)'=""!$D(IBXERR) D  Q:'IBDONE  .. D DSPLERR ; Displays warnings/errors  .. K IBXERR  .. Q:IBQUIT!(IBDONE)  .. I $G(IBNOFIX) S IBDONE=1 Q  .. I '$$ASKEDIT($G(IBAC)) W ! S IBQUIT=1 ; Don't want to re-edit  .. ;  . I $G(IBNOFIX) S IBDONE=1 Q  . S IBEDIT=0  . I $S($P($G(^DGCR(399,IBIFN,0)),U,13)>2:1,$D(PRCASV):'$D(PRCASV("OKAY")),1:0) D  S:'IBQUIT&'IBEDIT IBDONE=1 Q  .. N IBQUIT1  .. S IBQUIT1=0  .. W !!!,"... Executing A/R edits"  .. I $P($G(^DGCR(399,IBIFN,0)),U,13)>2 D GVAR^IBCBB,ARRAY^IBCBB1  .. D ARCHK($G(IBNOFIX),0,.IBQUIT1,.IBQUIT,.IBEDIT,.PRCASV)  . S IBDONE=1 ; No errors  . S:$G(IBPRT("PRT"))'<0 IBQUIT=0  Q  ; ARCHK(IBNOFIX,IBNOPRT,IBQUIT1,IBQUIT,IBEDIT,PRCASV) ; A/R Verification  ; Returns IBEDIT, IBQUIT1, IBQUIT,PRCASV array if passed by reference  ; IBNOFIX = 1 if no editing needed  ; IBNOPRT = 1 if no printing needed  F  D ^PRCASVC6 D  Q:IBQUIT1!IBEDIT  D GVAR^IBCBB,ARRAY^IBCBB1  . I '$G(IBNOPRT) Q:$G(IBPRT("PRT"))<0  . I PRCASV("OKAY") W:'$G(IBNOPRT) !!,"No A/R errors found" S IBQUIT1=1 Q  . I 'PRCASV("OKAY") D  Q  .. D DSPARERR($G(IBNOPRT)) ; Display A/R errors  .. Q:IBQUIT  .. I $G(IBNOFIX) S IBQUIT1=1 Q  .. I '$$ASKEDIT($G(IBAC)) W !,"There is an unresolved A/R error - cannot authorize bill" D PAUSE^VALM1 S (IBQUIT,IBQUIT1)=1 Q  .. S IBEDIT=1  ;  Q  ; DSPLERR ; Display national/local edits failed  N Z  D PRTH(.IBPRT)  I IBPRT("PRT")<0 S IBQUIT=1 Q  S Z=0 F  S Z=$O(^TMP($J,"BILL-WARN",Z)) Q:'Z  W !,^(Z) W:'$O(^(Z)) !  S Y2=""  I IBER'="WARN" F I=1:1 S X=$P(IBER,";",I) Q:X=""  W:I=1 !?5,"\*\*Errors\*\*:" I $D(^IBE(350.8,+$O(^IBE(350.8,"AC",X,0)),0)) S Y=^(0),Y1=$P(Y,"^",5),Y2=Y2\_Y1 I Y1<5 W !?5,$E($P(Y,"^",2),1,80)  ; IBXERR = local edits return error array  ; If IBXERR returns = 1 then we have at least one error  ; = "" or 0, then we have only local warnings  ; undefined = no local errors or warnings  I $D(IBXERR) D  . S I="" W !!,?3,"Local Edits:"  . S:$G(IBXERR) Y2=3,IBER="L"  . F  S I=$O(IBXERR(I)) Q:I=""  W !,?5,$E(IBXERR(I),1,75)  I $G(IBPRT("PRT")) D CLOSE(.IBPRT)  G:$G(IBNOFIX) Q  I $G(IBER)="WARN"!($G(IBXERR)=0) D  ;Warnings only - make biller stop and look  . W !  . N DIR,X,Y  . S DIR(0)="YA",DIR("B")="NO",DIR("A",1)="THIS BILL STILL HAS ONE OR MORE WARNINGS - PLEASE REVIEW THEM CAREFULLY",DIR("A")="ARE YOU SURE IT'S OK TO CONTINUE? "  . D ^DIR K DIR  . I Y'=1 S Y2=3 Q  . S IBER="",IBDONE=1 K IBXERR  I $S(Y2'["3"&'$G(IBXERR):0,1:1) K IBXERR Q K ^TMP($J,"BILL-WARN")  Q  ; DSPARERR(IBNOPRT) ; Displays A/R errors  N I,J,Y,X,ERRPRT  I '$G(IBNOPRT) D PRTH(.IBPRT) I IBPRT("PRT")<0 S IBQUIT=1 Q  I $P($G(PRCAERR),U,2)'="" D  . N Z  . S Z=+$O(^IBE(350.8,"C",$P(PRCAERR,U,2),0)),Z=$P($G(^IBE(350.8,+Z,0)),U,2)  . W !,?5,"An A/R error has been reported - bill cannot be authorized",!!,?5,$P(PRCAERR,U,2)," - ",$S(Z'="":Z,1:"??")  E  D  . W !,?5,"An undetermined A/R error was found - "\_$G(PRCAERR)  I $G(IBPRT("PRT")) D CLOSE(.IBPRT)  Q  ; NOPTF S IBAC1=1 I $D(^DGCR(399,IBIFN,0)),$P(^(0),"^",8),'$D(^DGPT($P(^(0),"^",8),0)) S IBAC1=0  Q  ; NOPTF1 W !!,\*7,"PTF Record for this Bill was DELETED!",!,"Further processing not allowed. Cancel and re-enter." Q  ; LOCERR ; Check for local edits  ; Execute screen post-processor for bills with local scrn 9 affiliations  N IBZ,IBXIEN,IBPRT  K IBXERR  S IBZ=$$LOCSCRN^IBCSC11(IBIFN) ; IB\*2.0\*447 BI  I IBZ S IBXIEN=IBIFN W !!,"... Executing local IB edits" D FPOST^IBCEFG7(IBZ,0,.IBXERR) I '$D(IBXERR) W !!,"No errors found for local edits"  Q  ; PRTH(IBPRT,IBA) ; Print a heading for error/warnings sent to a printer  ; Returns IBPRT = 1 if valid pritner selected  ; IBPRT = -1 if '^' entered  ; IBPRT = 0 if home device  N POP,%ZIS,POP  S %ZIS("A")="ERROR/WARNING OUTPUT DEVICE: "  D ^%ZIS  I POP S IBPRT("PRT")=-1 Q  I IO=IO(0) S IBPRT("PRT")=0 Q  S IBPRT("PRT")=1  U IO  W !,"INCONSISTENCIES LIST FOR BILL #: ",$P($G(^DGCR(399,IBIFN,0)),U),!,$J("",29),"AT: ",$$FMTE^XLFDT($$NOW^XLFDT,2),!,$J("",19),"GENERATED BY: ",$P($G(^VA(200,DUZ,0)),U),!!  Q  ; CLOSE(IBPRT) ; Close device, reset printer flag  D ^%ZISC  S IBPRT("PRT")=0  D HOME^%ZIS  Q  ; ASKEDIT(IBAC) ; Ask if edit/review of bill is desired  ; FUNCTION returns 0/1 for NO/YES  ; IBAC = flag for function being performed - to determine edit/review  N DIR,X,Y  S DIR(0)="YA"  S DIR("A",1)=" ",DIR("A",2)=" ",DIR("A")="Do you wish to "\_$S($G(IBAC)<4:"edit",1:"review")\_" the inconsistencies now? ",DIR("B")="NO"  S DIR("?",1)=" ",DIR("?",2)=" ",DIR("?",3)=" YES - To edit inconsistent fields",DIR("?")=" NO - To discontinue this process."  D ^DIR K DIR  Q (Y=1)  ; SCREENS ;  N IBH  D ^IBCSCU,^IBCSC1  I $G(IBV) K IBPOPOUT  Q  ; DISP24(IBIFN,IBCORR,IBQUIT) ;  W @IOF D BL24^IBCSCH(IBIFN,0)  S DIR("A",1)=" ",DIR("A")="Are the above charges correct for this bill? ",DIR("B")="YES",DIR(0)="YA" D ^DIR K DIR  I Y'=1 D  . I Y=0,$$ASKEDIT($G(IBAC)) S IBCORR=1 Q  . S IBQUIT=1  Q  ; IICM(IBIFN) ; Ingenix ClaimsManager: Claim Scrubber  ; Send the bill to ClaimsManager, the IBCISTAT variable returned from ClaimsManager indicates  ; 3 - Passed CM with no errors  ; 5 - User overriding the CM errors  ; 7 - the CM interface isn't working  ; 11 - User overriding the CM errors (CM not updated)  ;   ; Returns False (0) if the bill fails the ClaimsManager Scrubber/errors found  ; Returns True (1) if the bill passed the ClaimsManager Scrubber/no errors found or ClaimsManager not On at site  ;  N IBOK S IBOK=1  I +$G(IBIFN),$$CM^IBCIUT1(IBIFN) S IBCISNT=1 D ST2^IBCIST I '$F(".3.5.7.11.","."\_IBCISTAT\_".") S IBOK=0  Q IBOK  ; IIQMED(IBIFN) ; DSS QuadraMed Interface: QuadraMed Claim Scrubber  ; Send the bill to the QuadraMed Claim Scrubber  ; Returns False (0) if the bill fails the QuadraMed Scrubber/errors found  ; Returns True (1) if the bill passed the QuadraMed Scrubber/no errors found or QuadraMed not On at site  ;  ; QuadraMed Scrubber EN^VEJDIBSC returns IBQMED = 1 if no error found, returns 0 if error found  ;  N IBQMED S IBQMED=1  I +$G(IBIFN),$$QMED^IBCU1("EN^VEJDIBSC",IBIFN) D EN^VEJDIBSC(IBIFN)  Q IBQMED | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCBB | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** |  | | | | | | | | |
| **Related Options** | None | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  |  | | | | |  | | | |
| **Data Dictionary (DD) References** | CLAIMS TRACKING File [#356] | | | | | | | | |
| **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** | | | | | | | | | |
| IBCBB ;ALB/AAS - EDIT CHECK ROUTINE TO BE INVOKED BEFORE ALL BILL APPROVAL ACTIONS ;2-NOV-89  ;;2.0;INTEGRATED BILLING;\*\*80,51,137,288,327,361,371,377,400,432,461,547\*\*;21-MAR-94;Build 119  ;;Per VA Directive 6402, this routine should not be modified.  ;  ;MAP TO DGCRBB  ;  ;IBNDn = IBND(n) = ^ib(399,n)  ;RETURNS:  ;IBER=fields with errors separated by semi-colons  ;PRCASV("OKAY")=1 if iber="" and $D(prcasv("array")) compete  ; GVAR ;set up variables for mccr  Q:'$D(IBIFN) F I=0,"M","U","U1","S","MP","TX","UF3","UF31","U2" S @("IBND"\_I)=$G(^DGCR(399,IBIFN,I))  S IBBNO=$P(IBND0,"^"),DFN=$P(IBND0,"^",2),IBEVDT=$P(IBND0,"^",3)  S IBLOC=$P(IBND0,"^",4),IBCL=$P(IBND0,"^",5),IBTF=$P(IBND0,"^",6)  S IBAT=$P(IBND0,"^",7),IBWHO=$P(IBND0,"^",11),IBST=$P(IBND0,"^",13),IBFT=$P(IBND0,"^",19)  S IBFDT=$P(IBNDU,"^",1),IBTDT=$P(IBNDU,"^",2)  S IBTC=$P(IBNDU1,"^",1),IBFY=$P(IBNDU1,"^",9),IBFYC=$P(IBNDU1,"^",10)  S IBEU=$P(IBNDS,"^",2),IBRU=$P(IBNDS,"^",5),IBAU=$P(IBNDS,"^",8)  S IBTOB=$$TOB(IBND0),IBTOB12=$E(IBTOB,1,2)  K ^TMP($J,"BILL-WARN")  Q  ; EN ;Entry to check for errors  N IBQ,IBXERR,IBXDATA,IBXSAVE,IBZPRC92,IBQUIT,IBISEQ,IDDATA,IBFOR,IBC  I $D(IBFL) N IBFL  K ^TMP($J)  W !  S IBER="" D GVAR I '$D(IBND0) S IBER=-1 Q  ;  ;patient in patient file  I DFN="" S IBER=IBER\_"IB057;"  I DFN]"",'$D(^DPT(DFN)) S IBER=IBER\_"IB057;"  ;  ;Event date in correct format  I IBEVDT="" S IBER=IBER\_"IB049;"  I IBEVDT]"",IBEVDT'?7N&(IBEVDT'?7N1".".N) S IBER=IBER\_"IB049;"  ;  ;Rate Type  I IBAT="" S IBER=IBER\_"IB059;"  I IBAT]"",'$D(^DGCR(399.3,IBAT,0)) S IBER=IBER\_"IB059;"  I IBAT]"",$D(^DGCR(399.3,IBAT,0)),'$P(^(0),"^",6) S IBER=IBER\_"IB059;",IBAT=""  I IBAT]"",$P($G(^DGCR(399.3,IBAT,0)),"^",6) S IBARTP=$P($$CATN^PRCAFN($P(^DGCR(399.3,IBAT,0),"^",6)),"^",3)  ;Check that AR category expects same debtor as defined in who's responsible.  I $D(IBARTP),IBWHO="i"&(IBARTP'="T")!(IBWHO="p"&("PC"'[IBARTP))!(IBWHO="o"&(IBARTP'="N")) S IBER=IBER\_"IB058;"  ;  ;Who's Responsible  I IBWHO=""!($L(IBWHO)>1)!("iop"'[IBWHO) S IBER=IBER\_"IB065;"  S IBMRA=$S($$MCRWNR^IBEFUNC(+$$CURR^IBCEF2(IBIFN)):$$TXMT^IBCEF4(IBIFN)>0,1:0)  ; MCR will not reimburse is only valid if there is subsequent insurance  ; that will reimburse  I IBWHO="i" D  . I IBMRA D  Q  .. N Z,IBZ  .. S IBZ=0  .. F Z=$$COBN^IBCEF(IBIFN):1:3 I $D(^DGCR(399,IBIFN,"I"\_(Z+1))),$P($G(^DIC(36,+$G(^DGCR(399,IBIFN,"I"\_(Z+1))),0)),U,2)'="N" S IBZ=1 Q  .. I 'IBZ S IBER=IBER\_"IB054;" D WARN^IBCBB11("A valid claim for MEDICARE WNR needs subsequent ins. that will reimburse")  ..  . I $$COB^IBCEF(IBIFN)="S",$$MCRWNR^IBEFUNC(+$$CURR^IBCEF2(IBIFN))=1,$D(^DGCR(399,IBIFN,"I3")) Q  . I $S('IBNDMP:1,1:$P(IBNDMP,U,2)'=$$BPP^IBCNS2(IBIFN,1)) S IBER=IBER\_"IB054;"  I IBWHO="o",'$P(IBNDM,"^",11) S IBER=IBER\_"IB053;"  ;  ; Outpatient Statement dates can not span the ICD-10 activation date  I IBCL>2,$$ICD10S^IBCU4(IBFDT,IBTDT) S IBER=IBER\_"IB354;"  ;  ; All bill ICD codes must match Code Version on Statement To Date IB356  D ICD10V^IBCBB0(IBIFN)  ;  ; Billing Provider check - IB\*2\*400  D BP^IBCBB0(IBIFN)  ;  ; Pay-to Provider check - IB\*2\*400  D PAYTO^IBCBB0(IBIFN)  ;  ; All insurance subscribers must have a birth date on file  ; - 11/10/04 - IB\*2.0\*288  ; - 12/14/06 - IB\*2.0\*361 - must have INSURED'S SEX too  ; IB error codes  ; IB221 - Primary insurance subscriber missing date of birth  ; IB222 - Secondary insurance subscriber missing date of birth  ; IB223 - Tertiary insurance subscriber missing date of birth  ; IB261 - Primary insurance subscriber is missing INSURED'S SEX  ; IB262 - Secondary insurance subscriber is missing INSURED'S SEX  ; IB263 - Tertiary insurance subscriber is missing INSURED'S SEX  ;  F IBISEQ=1:1:3 D  . I '$P($G(^DGCR(399,IBIFN,"I"\_IBISEQ)),U,1) Q   ; no insurance here  . K ^UTILITY("VADM",$J),^UTILITY("VAPA",$J)  . S IDDATA=$$INSDEM^IBCEF(IBIFN,IBISEQ)  . K ^UTILITY("VADM",$J),^UTILITY("VAPA",$J)  . ;  . I '$P(IDDATA,U,1) D ERR(221) ; birth date missing  . ;  . I "^M^F^"'[(U\_$P(IDDATA,U,2)\_U) D ERR(261) ; sex missing  . ;  . ; IB\*2\*371 - esg - check for other missing insurance pieces  . ; check insured's name, primary ID#, pt. relationship to insured,  . ; and subscriber address data   . N INNAME,SUBID,PTREL,SFA,CAS,LN,FN  . ;  . ; IB273 - Primary Insurance name of insured missing  . ; IB274 - Secondary Insurance name of insured missing  . ; IB275 - Tertiary Insurance name of insured missing  . S INNAME=$$POLICY^IBCEF(IBIFN,17,IBISEQ)  . S LN=$P(INNAME,",",1),FN=$P(INNAME,",",2) ; last name,first name  . S LN=$$NOPUNCT^IBCEF(LN,1)  . S FN=$$NOPUNCT^IBCEF(FN,1)  . ; ib\*2.0\*547 - subscriber only needs last name  . ;I LN=""!(FN="") D ERR(273) ; name of insured missing or invalid  . I LN="" D ERR(273) ; name of insured missing or invalid  . S LN=$$NAME^IBCEFG1(INNAME) ; additional name checks  . S FN=$P(LN,U,2)  . S LN=$P(LN,U,1)  . ;I LN=""!(FN="") D ERR(273) ; name of insured missing or invalid  . I LN="" D ERR(273) ; name of insured missing or invalid  . ;  . ; IB276 - Primary Insurance subscriber ID missing  . ; IB277 - Secondary Insurance subscriber ID missing  . ; IB278 - Tertiary Insurance subscriber ID missing  . S SUBID=$$NOPUNCT^IBCEF($$POLICY^IBCEF(IBIFN,2,IBISEQ),1)  . I SUBID="" D ERR(276) ; subscriber ID# missing  . ;  . ; IB279 - Primary Insurance missing pt relationship  . ; IB280 - Secondary Insurance missing pt relationship  . ; IB281 - Tertiary Insurance missing pt relationship  . S PTREL=$$POLICY^IBCEF(IBIFN,16,IBISEQ)  . I PTREL="" D ERR(279) ; missing patient relationship to insured  . ;  . ; subscriber address section  . S SFA=$$INSADDR^IBCEF(IBIFN,IBISEQ) ; full address all pieces  . S CAS=$$NOPUNCT^IBCEF($P(SFA,U,2,5),1) ; string city,st,zip,addr1  . ;  . ; IB282 - Primary Insurance address line 1 missing  . ; IB283 - Secondary Insurance address line 1 missing  . ; IB284 - Tertiary Insurance address line 1 missing  . I $$NOPUNCT^IBCEF($P(SFA,U,5),1)="" D   ; address line 1 is blank  .. ; pat=subscriber and current insurance - address is required  .. I +PTREL=1,IBISEQ=$$COBN^IBCEF(IBIFN) D ERR(282) Q  .. ; if any part of the address is there, then all fields are required  .. I CAS'="" D ERR(282) Q  .. Q  . ;  . ; IB285 - Primary Insurance city missing  . ; IB286 - Secondary Insurance city missing  . ; IB287 - Tertiary Insurance city missing  . I $$NOPUNCT^IBCEF($P(SFA,U,2),1)="" D   ; city is blank  .. ; pat=subscriber and current insurance - address is required  .. I +PTREL=1,IBISEQ=$$COBN^IBCEF(IBIFN) D ERR(285) Q  .. ; if any part of the address is there, then all fields are required  .. I CAS'="" D ERR(285) Q  .. Q  . ;  . ; IB288 - Primary Insurance state missing  . ; IB289 - Secondary Insurance state missing  . ; IB290 - Tertiary Insurance state missing  . I $$NOPUNCT^IBCEF($P(SFA,U,3),1)="" D   ; state is blank  .. ; pat=subscriber and current insurance - address is required  .. I +PTREL=1,IBISEQ=$$COBN^IBCEF(IBIFN) D ERR(288) Q  .. ; if any part of the address is there, then all fields are required  .. I CAS'="" D ERR(288) Q  .. Q  . ;  . ; IB291 - Primary Insurance zipcode missing  . ; IB292 - Secondary Insurance zipcode missing  . ; IB293 - Tertiary Insurance zipcode missing  . I $$NOPUNCT^IBCEF($P(SFA,U,4),1)="" D   ; zipcode is blank  .. ; pat=subscriber and current insurance - address is required  .. I +PTREL=1,IBISEQ=$$COBN^IBCEF(IBIFN) D ERR(291) Q  .. ; if any part of the address is there, then all fields are required  .. I CAS'="" D ERR(291) Q  .. Q  . ;  . Q  ;  ; esg - IB\*2\*371 - check patient address fields  K ^UTILITY("VAPA",$J)  ;  S IBFOR=0 ; foreign address flag  S IBC=+$$PTADDR^IBCEF(IBIFN,25) ; country code ien  I IBC D  . N CODE  . S CODE=$$GET1^DIQ(779.004,IBC,.01) ; .01 code field file 779.004  . I CODE'="",CODE'="USA" S IBFOR=1 ; foreign country exists  . Q  ;  I $$NOPUNCT^IBCEF($$PTADDR^IBCEF(IBIFN,1),1)="" S IBER=IBER\_"IB269;"  I $$NOPUNCT^IBCEF($$PTADDR^IBCEF(IBIFN,4),1)="" S IBER=IBER\_"IB270;"  I $$NOPUNCT^IBCEF($$PTADDR^IBCEF(IBIFN,5),1)="",'IBFOR S IBER=IBER\_"IB271;"  I $$NOPUNCT^IBCEF($$PTADDR^IBCEF(IBIFN,11),1)="",'IBFOR S IBER=IBER\_"IB272;"  K ^UTILITY("VAPA",$J)  ;  D PAYERADD^IBCBB0(IBIFN) ; check the payer addresses  D ^IBCBB1  Q  ; The remaining code below is being removed with Patch IB\*2.0\*432.  ;  ; esg - 9/20/07 - IB patch 371 - prevent EDI transmission for 3 payer  ; claims for all but the first payer. To be removed when Emdeon  ; and FSC are able to deal with these.  ;  I +$G(^DGCR(399,IBIFN,"I2")),+$G(^DGCR(399,IBIFN,"I3")),$$TXMT^IBCEF4(IBIFN) D  . ; for MRA request claims, make sure the MRA secondary claim is forced to print  . I $$REQMRA^IBEFUNC(IBIFN) D  Q  .. I '$P($G(^DGCR(399,IBIFN,"TX")),U,9) S IBER=IBER\_"IB146;"  .. Q  . ;  . I $$COBN^IBCEF(IBIFN)=1 Q   ; primary payer sequence claims are OK  . ;  . ; But claims with a payer sequence of 2 or 3 need to print locally  . S IBER=IBER\_"IB147;"  . Q  ;  Q  ; EDIT(IBIFN) ; Run edits from within the billing edit screens  N IBVIEW,IBDISP,IBNOFIX,DIR,X,Y  S (IBNOFIX,IBVIEW,IBDISP)=1  D EDITS^IBCB2  W ! S DIR("A")="Press RETURN to continue",DIR(0)="E" D ^DIR K DIR  Q  ; TOB(IBND0) ;  ; IBND0 = the 0-node of the bill (file 399)  Q ($P(IBND0,U,24)\_$P($G(^DGCR(399.1,+$P(IBND0,U,25),0)),U,2)\_$P(IBND0,U,26))  ; ERR(Z) ; update IBER variable from the above insurance checks  ; Z is the IB error code# for the primary insurance error  N IBERRNO  S IBERRNO="IB"\_(Z+IBISEQ-1)  I IBER[IBERRNO Q  S IBER=IBER\_IBERRNO\_";"  Q | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCBB ;ALB/AAS - EDIT CHECK ROUTINE TO BE INVOKED BEFORE ALL BILL APPROVAL ACTIONS ;2-NOV-89  ;;2.0;INTEGRATED BILLING;\*\*80,51,137,288,327,361,371,377,400,432,461,547,**592**\*\*;21-MAR-94;Build 119  ;;Per VA Directive 6402, this routine should not be modified.  ;  ;MAP TO DGCRBB  ;  ;IBNDn = IBND(n) = ^ib(399,n)  ;RETURNS:  ;IBER=fields with errors separated by semi-colons  ;PRCASV("OKAY")=1 if iber="" and $D(prcasv("array")) compete  ; GVAR ;set up variables for mccr  Q:'$D(IBIFN) F I=0,"M","U","U1","S","MP","TX","UF3","UF31","U2" S @("IBND"\_I)=$G(^DGCR(399,IBIFN,I))  S IBBNO=$P(IBND0,"^"),DFN=$P(IBND0,"^",2),IBEVDT=$P(IBND0,"^",3)  S IBLOC=$P(IBND0,"^",4),IBCL=$P(IBND0,"^",5),IBTF=$P(IBND0,"^",6)  S IBAT=$P(IBND0,"^",7),IBWHO=$P(IBND0,"^",11),IBST=$P(IBND0,"^",13),IBFT=$P(IBND0,"^",19)  S IBFDT=$P(IBNDU,"^",1),IBTDT=$P(IBNDU,"^",2)  S IBTC=$P(IBNDU1,"^",1),IBFY=$P(IBNDU1,"^",9),IBFYC=$P(IBNDU1,"^",10)  S IBEU=$P(IBNDS,"^",2),IBRU=$P(IBNDS,"^",5),IBAU=$P(IBNDS,"^",8)  S IBTOB=$$TOB(IBND0),IBTOB12=$E(IBTOB,1,2)  K ^TMP($J,"BILL-WARN")  Q  ; EN ;Entry to check for errors  N IBQ,IBXERR,IBXDATA,IBXSAVE,IBZPRC92,IBQUIT,IBISEQ,IDDATA,IBFOR,IBC  I $D(IBFL) N IBFL  K ^TMP($J)  W !  S IBER="" D GVAR I '$D(IBND0) S IBER=-1 Q  ;  ;patient in patient file  I DFN="" S IBER=IBER\_"IB057;"  I DFN]"",'$D(^DPT(DFN)) S IBER=IBER\_"IB057;"  ;  ;Event date in correct format  I IBEVDT="" S IBER=IBER\_"IB049;"  I IBEVDT]"",IBEVDT'?7N&(IBEVDT'?7N1".".N) S IBER=IBER\_"IB049;"  ;  ;Rate Type  I IBAT="" S IBER=IBER\_"IB059;"  I IBAT]"",'$D(^DGCR(399.3,IBAT,0)) S IBER=IBER\_"IB059;"  I IBAT]"",$D(^DGCR(399.3,IBAT,0)),'$P(^(0),"^",6) S IBER=IBER\_"IB059;",IBAT=""  I IBAT]"",$P($G(^DGCR(399.3,IBAT,0)),"^",6) S IBARTP=$P($$CATN^PRCAFN($P(^DGCR(399.3,IBAT,0),"^",6)),"^",3)  ;Check that AR category expects same debtor as defined in who's responsible.  I $D(IBARTP),IBWHO="i"&(IBARTP'="T")!(IBWHO="p"&("PC"'[IBARTP))!(IBWHO="o"&(IBARTP'="N")) S IBER=IBER\_"IB058;"  ;  ;Who's Responsible  I IBWHO=""!($L(IBWHO)>1)!("iop"'[IBWHO) S IBER=IBER\_"IB065;"  S IBMRA=$S($$MCRWNR^IBEFUNC(+$$CURR^IBCEF2(IBIFN)):$$TXMT^IBCEF4(IBIFN)>0,1:0)  ; MCR will not reimburse is only valid if there is subsequent insurance  ; that will reimburse  I IBWHO="i" D  **. ;JWS;IB\*2.0\*592;US1109; If Dental and Plan Coverage Limitation is NO skip  . I $$FT^IBCEF(IBIFN)=7,'$$PTCOV^IBCNSU3(DFN,$P($G(^DGCR(399,IBIFN,0)),"^",3),"DENTAL")** S IBER=IBER\_"IB362"  . I IBMRA D  Q  **.. ;JWS;IB\*2.0\*592;Do not allow to bill Dental to Medicare WNR  .. I $$FT^IBCEF(IBIFN)=7,'$F(IBER,"IB359;") S IBER=IBER\_"IB359;"**  .. N Z,IBZ  .. S IBZ=0  .. F Z=$$COBN^IBCEF(IBIFN):1:3 I $D(^DGCR(399,IBIFN,"I"\_(Z+1))),$P($G(^DIC(36,+$G(^DGCR(399,IBIFN,"I"\_(Z+1))),0)),U,2)'="N" S IBZ=1 Q  .. I 'IBZ S IBER=IBER\_"IB054;" D WARN^IBCBB11("A valid claim for MEDICARE WNR needs subsequent ins. that will reimburse")  . I $$COB^IBCEF(IBIFN)="S",$$MCRWNR^IBEFUNC(+$$CURR^IBCEF2(IBIFN))=1,$D(^DGCR(399,IBIFN,"I3")) Q  . I $S('IBNDMP:1,1:$P(IBNDMP,U,2)'=$$BPP^IBCNS2(IBIFN,1)) S IBER=IBER\_"IB054;"  I IBWHO="o",'$P(IBNDM,"^",11) S IBER=IBER\_"IB053;"  ;  ; Outpatient Statement dates can not span the ICD-10 activation date  I IBCL>2,$$ICD10S^IBCU4(IBFDT,IBTDT) S IBER=IBER\_"IB354;"  ;  ; All bill ICD codes must match Code Version on Statement To Date IB356  D ICD10V^IBCBB0(IBIFN)  ;  ; Billing Provider check - IB\*2\*400  D BP^IBCBB0(IBIFN)  ;  ; Pay-to Provider check - IB\*2\*400  D PAYTO^IBCBB0(IBIFN)  ;  ; All insurance subscribers must have a birth date on file  ; - 11/10/04 - IB\*2.0\*288  ; - 12/14/06 - IB\*2.0\*361 - must have INSURED'S SEX too  ; IB error codes  ; IB221 - Primary insurance subscriber missing date of birth  ; IB222 - Secondary insurance subscriber missing date of birth  ; IB223 - Tertiary insurance subscriber missing date of birth  ; IB261 - Primary insurance subscriber is missing INSURED'S SEX  ; IB262 - Secondary insurance subscriber is missing INSURED'S SEX  ; IB263 - Tertiary insurance subscriber is missing INSURED'S SEX  ;  F IBISEQ=1:1:3 D  . I '$P($G(^DGCR(399,IBIFN,"I"\_IBISEQ)),U,1) Q   ; no insurance here  . K ^UTILITY("VADM",$J),^UTILITY("VAPA",$J)  . S IDDATA=$$INSDEM^IBCEF(IBIFN,IBISEQ)  . K ^UTILITY("VADM",$J),^UTILITY("VAPA",$J)  . ;  . I '$P(IDDATA,U,1) D ERR(221) ; birth date missing  . ;  . I "^M^F^"'[(U\_$P(IDDATA,U,2)\_U) D ERR(261) ; sex missing  . ;  . ; IB\*2\*371 - esg - check for other missing insurance pieces  . ; check insured's name, primary ID#, pt. relationship to insured,  . ; and subscriber address data   . N INNAME,SUBID,PTREL,SFA,CAS,LN,FN  . ;  . ; IB273 - Primary Insurance name of insured missing  . ; IB274 - Secondary Insurance name of insured missing  . ; IB275 - Tertiary Insurance name of insured missing  . S INNAME=$$POLICY^IBCEF(IBIFN,17,IBISEQ)  . S LN=$P(INNAME,",",1),FN=$P(INNAME,",",2) ; last name,first name  . S LN=$$NOPUNCT^IBCEF(LN,1)  . S FN=$$NOPUNCT^IBCEF(FN,1)  . ; ib\*2.0\*547 - subscriber only needs last name  . ;I LN=""!(FN="") D ERR(273) ; name of insured missing or invalid  . I LN="" D ERR(273) ; name of insured missing or invalid  . S LN=$$NAME^IBCEFG1(INNAME) ; additional name checks  . S FN=$P(LN,U,2)  . S LN=$P(LN,U,1)  . ;I LN=""!(FN="") D ERR(273) ; name of insured missing or invalid  . I LN="" D ERR(273) ; name of insured missing or invalid  . ;  . ; IB276 - Primary Insurance subscriber ID missing  . ; IB277 - Secondary Insurance subscriber ID missing  . ; IB278 - Tertiary Insurance subscriber ID missing  . S SUBID=$$NOPUNCT^IBCEF($$POLICY^IBCEF(IBIFN,2,IBISEQ),1)  . I SUBID="" D ERR(276) ; subscriber ID# missing  . ;  . ; IB279 - Primary Insurance missing pt relationship  . ; IB280 - Secondary Insurance missing pt relationship  . ; IB281 - Tertiary Insurance missing pt relationship  . S PTREL=$$POLICY^IBCEF(IBIFN,16,IBISEQ)  . I PTREL="" D ERR(279) ; missing patient relationship to insured  . ;  . ; subscriber address section  . S SFA=$$INSADDR^IBCEF(IBIFN,IBISEQ) ; full address all pieces  . S CAS=$$NOPUNCT^IBCEF($P(SFA,U,2,5),1) ; string city,st,zip,addr1  . ;  . ; IB282 - Primary Insurance address line 1 missing  . ; IB283 - Secondary Insurance address line 1 missing  . ; IB284 - Tertiary Insurance address line 1 missing  . I $$NOPUNCT^IBCEF($P(SFA,U,5),1)="" D   ; address line 1 is blank  .. ; pat=subscriber and current insurance - address is required  .. I +PTREL=1,IBISEQ=$$COBN^IBCEF(IBIFN) D ERR(282) Q  .. ; if any part of the address is there, then all fields are required  .. I CAS'="" D ERR(282) Q  .. Q  . ;  . ; IB285 - Primary Insurance city missing  . ; IB286 - Secondary Insurance city missing  . ; IB287 - Tertiary Insurance city missing  . I $$NOPUNCT^IBCEF($P(SFA,U,2),1)="" D   ; city is blank  .. ; pat=subscriber and current insurance - address is required  .. I +PTREL=1,IBISEQ=$$COBN^IBCEF(IBIFN) D ERR(285) Q  .. ; if any part of the address is there, then all fields are required  .. I CAS'="" D ERR(285) Q  .. Q  . ;  . ; IB288 - Primary Insurance state missing  . ; IB289 - Secondary Insurance state missing  . ; IB290 - Tertiary Insurance state missing  . I $$NOPUNCT^IBCEF($P(SFA,U,3),1)="" D   ; state is blank  .. ; pat=subscriber and current insurance - address is required  .. I +PTREL=1,IBISEQ=$$COBN^IBCEF(IBIFN) D ERR(288) Q  .. ; if any part of the address is there, then all fields are required  .. I CAS'="" D ERR(288) Q  .. Q  . ;  . ; IB291 - Primary Insurance zipcode missing  . ; IB292 - Secondary Insurance zipcode missing  . ; IB293 - Tertiary Insurance zipcode missing  . I $$NOPUNCT^IBCEF($P(SFA,U,4),1)="" D   ; zipcode is blank  .. ; pat=subscriber and current insurance - address is required  .. I +PTREL=1,IBISEQ=$$COBN^IBCEF(IBIFN) D ERR(291) Q  .. ; if any part of the address is there, then all fields are required  .. I CAS'="" D ERR(291) Q  .. Q  . ;  . Q  ;  ; esg - IB\*2\*371 - check patient address fields  K ^UTILITY("VAPA",$J)  ;  S IBFOR=0 ; foreign address flag  S IBC=+$$PTADDR^IBCEF(IBIFN,25) ; country code ien  I IBC D  . N CODE  . S CODE=$$GET1^DIQ(779.004,IBC,.01) ; .01 code field file 779.004  . I CODE'="",CODE'="USA" S IBFOR=1 ; foreign country exists  . Q  ;  I $$NOPUNCT^IBCEF($$PTADDR^IBCEF(IBIFN,1),1)="" S IBER=IBER\_"IB269;"  I $$NOPUNCT^IBCEF($$PTADDR^IBCEF(IBIFN,4),1)="" S IBER=IBER\_"IB270;"  I $$NOPUNCT^IBCEF($$PTADDR^IBCEF(IBIFN,5),1)="",'IBFOR S IBER=IBER\_"IB271;"  I $$NOPUNCT^IBCEF($$PTADDR^IBCEF(IBIFN,11),1)="",'IBFOR S IBER=IBER\_"IB272;"  K ^UTILITY("VAPA",$J)  ;  D PAYERADD^IBCBB0(IBIFN) ; check the payer addresses  D ^IBCBB1  Q  ; The remaining code below is being removed with Patch IB\*2.0\*432.  ;  ; esg - 9/20/07 - IB patch 371 - prevent EDI transmission for 3 payer  ; claims for all but the first payer. To be removed when Emdeon  ; and FSC are able to deal with these.  ;  I +$G(^DGCR(399,IBIFN,"I2")),+$G(^DGCR(399,IBIFN,"I3")),$$TXMT^IBCEF4(IBIFN) D  . ; for MRA request claims, make sure the MRA secondary claim is forced to print  . I $$REQMRA^IBEFUNC(IBIFN) D  Q  .. I '$P($G(^DGCR(399,IBIFN,"TX")),U,9) S IBER=IBER\_"IB146;"  .. Q  . ;  . I $$COBN^IBCEF(IBIFN)=1 Q   ; primary payer sequence claims are OK  . ;  . ; But claims with a payer sequence of 2 or 3 need to print locally  . S IBER=IBER\_"IB147;"  . Q  ;  Q  ; EDIT(IBIFN) ; Run edits from within the billing edit screens  N IBVIEW,IBDISP,IBNOFIX,DIR,X,Y  S (IBNOFIX,IBVIEW,IBDISP)=1  D EDITS^IBCB2  W ! S DIR("A")="Press RETURN to continue",DIR(0)="E" D ^DIR K DIR  Q  ; TOB(IBND0) ;  ; IBND0 = the 0-node of the bill (file 399)  Q ($P(IBND0,U,24)\_$P($G(^DGCR(399.1,+$P(IBND0,U,25),0)),U,2)\_$P(IBND0,U,26))  ; ERR(Z) ; update IBER variable from the above insurance checks  ; Z is the IB error code# for the primary insurance error  N IBERRNO  S IBERRNO="IB"\_(Z+IBISEQ-1)  I IBER[IBERRNO Q  S IBER=IBER\_IBERRNO\_";"  Q  ; | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCBB1 | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** |  | | | | | | | | |
| **Related Options** | None | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  |  | | | | |  | | | |
| **Data Dictionary (DD) References** | CLAIMS TRACKING File [#356] | | | | | | | | |
| **Related Protocols** | None | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Current Logic** | | | | | | | | | |
| IBCBB1 ;ALB/AAS - CONTINUATION OF EDIT CHECK ROUTINE ;2-NOV-89  ;;2.0;INTEGRATED BILLING;\*\*27,52,80,93,106,51,151,148,153,137,232,280,155,320,343,349,363,371,395,384,432,447,488\*\*;21-MAR-94;Build 184  ;;Per VHA Directive 2004-038, this routine should not be modified.  ;  ; \*\*\* Begin IB\*2.0\*488 VD (Issue 46 RBN)  N I  S I=""  S X=+$G(^DGCR(399,IBIFN,"MP"))  I 'X,$$MCRWNR^IBEFUNC(+$$CURR^IBCEF2(IBIFN)) S X=+$$CURR^IBCEF2(IBIFN)  I X,+$G(^DIC(36,X,3)) S I=$P(^(3),U,$S($$FT^IBCEF(IBIFN)=2:2,1:4))  S I=$$UP^XLFSTR(I)  I (I'=""&(I["PRNT")&($G(IBER)'["IB488")) D   . S IBER=$G(IBER)\_"IB488;"  ;  ; Cause an error if FORCED TO PRINT TO CLEARINGHOUSE  I $P($G(^DGCR(399,IBIFN,"TX")),U,8)=2 D  . S IBER=$G(IBER)\_"IB489;"  ;  ; Cause a fatal error if the claim has no procedures & is NOT a UB-04 Inpatient claim.  I +$O(^DGCR(399,IBIFN,"CP",0))=0 D  .I $$INPAT^IBCEF(IBIFN,1),$$INSPRF^IBCEF(IBIFN) Q   ; inpatient UB-04 check  .I '$$INPAT^IBCEF(IBIFN,1),$$INSPRF^IBCEF(IBIFN) D  Q      ; Outpatient Institutional Claim.  ..I IBER["IB352" Q  ..S IBER=IBER\_"IB352;"  .;  .; Professional claim  .I IBER["IB353" Q  .S IBER=IBER\_"IB353;"  .Q  ; \*\*\* End IB\*2.0\*488 -- VD  ;  ;MAP TO DGCRBB1  ; % ;Bill Status  N Z,Z0,Z1,IBFT  I $S(+IBST=0:1,1:"^1^2^3^4^7^"'[(U\_IBST\_U)) S IBER=IBER\_"IB045;"  ;  ;Statement Covers From  I IBFDT="" S IBER=IBER\_"IB061;"  I IBFDT]"",IBFDT'?7N&(IBFDT'?7N1".".N) S IBER=IBER\_"IB061;"  I IBFDT>IBTDT S IBER=IBER\_"IB061;" ; from must be on or before the to date   S IBFFY=$$FY^IBOUTL(IBFDT)  ; if inpat - from date must not be prior to admit date.  I $$INPAT^IBCEF(IBIFN,1),(IBFDT<($P($G(^DGPT(+$P(IBND0,U,8),0)),U,2)\1)) S IBER=IBER\_"IB061;"  ;  ;Statement Covers To  I IBTDT="" S IBER=IBER\_"IB062;"  I IBTDT]"",IBTDT'?7N&(IBTDT'?7N1".".N) S IBER=IBER\_"IB062;"  I IBTDT>DT!(IBTDT<IBFDT) S IBER=IBER\_"IB062;"  ; to date must not be >than today's date  S IBTFY=$$FY^IBOUTL(IBTDT)  ;  ;Total Charges  ; IB\*2.0\*447/TAZ Removed this error so that zero dollar revenue codes can process on the 837  ;I +IBTC'>0!(+IBTC'=IBTC) S IBER=IBER\_"IB064;"  ;  ;Billable charges for secondary claim  I $$MCRONBIL^IBEFUNC(IBIFN)&(($P(IBNDU1,U,1)-$P(IBNDU1,U,2))'>0) S IBER=IBER\_"IB094;"  ;Fiscal Year 1  S IBFFY=$$FY^IBOUTL(IBFDT)  ;  ;Check provider link for current user, enterer, reviewer and Authorizor  I '$D(^VA(200,DUZ,0)) S IBER=IBER\_"IB048;"  I IBEU]"",'$D(^VA(200,IBEU,0)) S IBER=IBER\_"IB048;"  I IBRU]"",'$D(^VA(200,IBRU,0)) S IBER=IBER\_"IB060;"  I IBAU]"",'$D(^VA(200,IBAU,0)) S IBER=IBER\_"IB041;"  ;  I IBER="",+$$STA^PRCAFN(IBIFN)=104 S IBER=IBER\_"IB040;"  ; If ins bill, must have valid COB sequence  I $P(IBND0,U,11)="i",$S($P(IBND0,U,21)="":1,1:"PST"'[$P(IBND0,U,21)) S IBER=IBER\_"IB324;"  ;  ; Check for valid sec provider id for current ins  S Z=0 F  S Z=$O(^DGCR(399,IBIFN,"PRV",Z)) Q:'Z  S Z0=$G(^(Z,0)),Z1=+$$COBN^IBCEF(IBIFN) I $P(Z0,U,4+Z1)'="",$P(Z0,U,11+Z1)'="" D  . I '$$SECIDCK^IBCEF74(IBIFN,Z1,$P(Z0,U,11+Z1),Z) D WARN^IBCBB11("Prov secondary id type for the "\_$P("PRIMARY^SECONDARY^TERTIARY",U,Z1)\_" "\_$$EXTERNAL^DILFD(399.0222,.01,,+Z0)\_" is invalid/won't transmit")  ; Check NPIs  D NPICHK^IBCBB11  ;  ; Check multiple rx NPIs  D RXNPI^IBCBB11(IBIFN)  ;  ; Check taxonomies  D TAXCHK^IBCBB11  ;  ; Check for Physician Name  K IBXDATA D F^IBCEF("N-ATT/REND PHYSICIAN NAME",,,IBIFN)  ; IB\*2.0\*432 - CMS1500 no longer needs a claim level rendering  S IBFT=$$FT^IBCEF(IBIFN)  I IBFT'=2,$P($G(IBXDATA),U)="" S IBER=IBER\_"IB303;"  ;  N FUNCTION,IBINS  ; IB\*2.0\*432 - CMS1500 no longer needs a claim level rendering  ;S FUNCTION=$S($$FT^IBCEF(IBIFN)=3:4,1:3)  S FUNCTION=$S(IBFT=3:4,1:3)  I IBFT'=2,IBER'["IB303;" D  . F IBINS=1:1:3 D  .. S Z=$$GETTYP^IBCEP2A(IBIFN,IBINS)  .. I Z,$P(Z,U,2) D  ; Rendering/attending prov secondary id required  ... N IBID,IBOK,Q0  ... D PROVINF^IBCEF74(IBIFN,IBINS,.IBID,1,"C") ; check all as though they were current  ... S IBOK=0  ... S Q0=0 F  S Q0=$O(IBID(1,FUNCTION,Q0)) Q:'Q0  I $P(IBID(1,FUNCTION,Q0),U,9)=+Z S IBOK=1 Q  ... I 'IBOK S IBER=IBER\_$S(IBINS=1:"IB236;",IBINS=2:"IB237;",IBINS=3:"IB238;",1:"")  ;  ; Patch 432 enh5:The IB system shall no longer prevent users from authorizing(fatal error message)a claim because the system cannot find the providersSSNorEIN  ; D PRIIDCHK^IBCBB11  ;  N IBM,IBM1  S IBM=$G(^DGCR(399,IBIFN,"M"))  S IBM1=$G(^DGCR(399,IBIFN,"M1"))  I $P(IBM,U),$P($G(^DIC(36,$P(IBM,U),4)),U,6),$P(IBM1,U,2)="" S IBER=IBER\_"IB244;"  I $P(IBM,U,2),$P($G(^DIC(36,$P(IBM,U,2),4)),U,6),$P(IBM1,U,3)="" S IBER=IBER\_"IB245;"  I $P(IBM,U,3),$P($G(^DIC(36,$P(IBM,U,3),4)),U,6),$P(IBM1,U,4)="" S IBER=IBER\_"IB246;"  ;  ; If outside facility, check for ID and qualifier in 355.93  ; 5/15/06 - esg - hard error IB243 turned into warning message instead  S Z=$P($G(^DGCR(399,IBIFN,"U2")),U,10)  I Z D  . I $P($G(^IBA(355.93,Z,0)),U,9)=""!($P($G(^IBA(355.93,Z,0)),U,13)="") D  .. N Z1,Z2  .. S Z1="Missing Lab or Facility Primary ID for non-VA facility, "  .. S Z2=$$EXTERNAL^DILFD(399,232,,Z)  .. I $L(Z2)'>19 D WARN^IBCBB11(Z1\_Z2) Q  .. D WARN^IBCBB11(Z1),WARN^IBCBB11(" "\_Z2)  .. Q  . Q  ;  ; Must be one and only one division on bill  S IBZ=$$MULTDIV^IBCBB11(IBIFN,IBND0)  ; I IBZ S IBER=IBER\_$S(IBZ=1:"IB095;",IBZ=2:"IB104;",1:"IB105;")  ; Allow multi-divisional for OP instutional claims  I IBZ,$$INPAT^IBCEF(IBIFN)!'($$INSPRF^IBCEF(IBIFN)) S IBER=IBER\_$S(IBZ=1:"IB095;",IBZ=2:"IB104;",1:"IB105;")  ; Still need error msg on OP Institutional if No Default division  I IBZ=3,'$$INPAT^IBCEF(IBIFN),$$INSPRF^IBCEF(IBIFN) S IBER=IBER\_"IB105;"  ; Division address must be defined in institution file  I $P(IBND0,U,22) D  . N Z,Z0,Z1  . S Z0=$G(^DIC(4,+$P($G(^DG(40.8,+$P(IBND0,U,22),0)),U,7),0))  . S Z1=$G(^DIC(4,+$P($G(^DG(40.8,+$P(IBND0,U,22),0)),U,7),1))  . I $P(Z0,U,2)="" S IBER=IBER\_"IB097;" Q  . F Z=1,3,4 I $P(Z1,U,Z)="" S IBER=IBER\_"IB097;" Q  ;  ; IB\*2.0\*432 Check ambulance addresses, COB Non-covered amt. & Attachment Control  I $$AMBCK^IBCBB11(IBIFN)=1 S IBER=IBER\_"IB329;"  I $$COBAMT^IBCBB11(IBIFN)=1 S IBER=IBER\_"IB330;"  I $$TMCK^IBCBB11(IBIFN)=1 S IBER=IBER\_"IB331;"  I $$ACCK^IBCBB11(IBIFN)=1 S IBER=IBER\_"IB332;"  I $$COBMRA^IBCBB11(IBIFN)=1 S IBER=IBER\_"IB342;"  I $$COBSEC^IBCBB11(IBIFN)=1 S IBER=IBER\_"IB343;"  ;  ;CHAMPVA Rate Type and Primary Insurance Carriers Type of Coverage must match  S (IBRTCHV,IBPICHV)=0  I $P($G(^DGCR(399.3,+IBAT,0)),U,1)="CHAMPVA" S IBRTCHV=1  I $P($G(^IBE(355.2,+$P($G(^DIC(36,+IBNDMP,0)),U,13),0)),U,1)="CHAMPVA" S IBPICHV=1  I (+IBRTCHV!+IBPICHV)&('IBRTCHV!'IBPICHV) S IBER=IBER\_"IB085;"  ;  N IBZPRC,IBZPRCUB  D F^IBCEF("N-ALL PROCEDURES","IBZPRC",,IBIFN)  ; Procedure Clinic is required for Surgical Procedures Outpt Facility Charges  I +$P(IBND0,U,27)'=2,$$BILLRATE^IBCRU3(IBAT,IBCL,IBEVDT,"RC OUTPATIENT") D  . N Z,Z0,Z1,ZE S (ZE,Z)=0 F  S Z=$O(^DGCR(399,IBIFN,"CP",Z)) Q:'Z  D  I +ZE S IBER=IBER\_"IB320;" Q  .. S Z0=$G(^DGCR(399,IBIFN,"CP",Z,0)),Z1=+Z0 I Z0'[";ICPT(" Q  .. I '((Z1'<10000)&(Z1'>69999))&'((Z1'<93501)&(Z1'>93533)) Q  .. I '$P(Z0,U,7) S ZE=1  ;  ; Extract procedures for UB-04  D F^IBCEF("N-UB-04 PROCEDURES","IBZPRCUB",,IBIFN)  ; Does this bill have ANY prescriptions associated with it?  ; Must bill prescriptions separately from other charges  ;  ; DEM;432 - Call line level provider edit checks.  D LNPROV^IBCBB12(IBIFN) ; DEM;432 - If there are line provider edits, then routine LNPROV^IBCBB12(IBIFN) updates IBER string.  ; DEM;432 - Call to Other Operating/Operating Provider edit checks.  I $$OPPROVCK^IBCBB12(IBIFN)=1 S IBER=IBER\_"IB337;"  ; DEM;432  ; DEM;432 - Line level Attachment Control edits.  I $$LNTMCK^IBCBB11(IBIFN)=1 S IBER=IBER\_"IB331;"  ; DEM;432  I $$LNACCK^IBCBB11(IBIFN)=1 S IBER=IBER\_"IB332;"  ; DEM;432  ;  I $$ISRX^IBCEF1(IBIFN) D  . N IBZ,IBRXDEF  . S IBRXDEF=$P($G(^IBE(350.9,1,1)),U,30),IBZ=0  . F  S IBZ=$O(IBZPRCUB(IBZ)) Q:'IBZ  I IBZPRCUB(IBZ),+$P(IBZPRCUB(IBZ),U)'=IBRXDEF S IBER=IBER\_"IB102;" Q  . K IBZ  ;  ; Check that COB sequences are not skipped  K Z  F Z=1:1:3 S:+$G(^DGCR(399,IBIFN,"I"\_Z)) Z(Z)=""  F Z=0:1:2 S Z0=$O(Z(Z)) Q:'Z0  I Z0'=(Z+1) S IBER=IBER\_"IB322;" Q  K Z  ; HD64676 IB\*2\*371 - OK for payer sequence to be blank when the Rate  ; Type is either Interagency or Sharing Agreement  I $P($G(^DGCR(399,IBIFN,0)),U,21)="",$P($G(^DGCR(399,IBIFN,0)),U,7)'=4,$P($G(^DGCR(399,IBIFN,0)),U,7)'=9 S IBER=IBER\_"IB323;"  K IBXDATA D F^IBCEF("N-PROCEDURE CODING METHD",,,IBIFN)  ; Coding method should agree with types of procedure codes  S IBOK=$S('$O(IBZPRC(0))!(IBXDATA=""):1,1:0)  I 'IBOK S IBOK=1,IBZ=0 F  S IBZ=$O(IBZPRC(IBZ)) Q:'IBZ  I IBZPRC(IBZ),$P(IBZPRC(IBZ),U)'[$S(IBXDATA=9:"ICD",1:"ICP") S IBOK=0 Q  I 'IBOK D WARN^IBCBB11("Coding Method does not agree with all procedure codes found on bill")  D EDITMRA^IBCBB3(.IBQUIT,.IBER,IBIFN,IBFT)  Q:$G(IBQUIT)  ;  ;Other things that could be added: Rev Code - calculating charges  ; Diagnosis Coding, if MT copay - check for other co-payments  ;  I $P(IBNDTX,U,8),$$REQMRA^IBEFUNC(IBIFN) S IBER=IBER\_"IB121;"   ; can't force MRAs to print  I $P(IBNDTX,U,8)!$P(IBNDTX,U,9) D  . Q:$P(IBNDTX,U,8)=2 ; Don't want to do this for option 2 any more.  . D WARN^IBCBB11($S($$REQMRA^IBEFUNC(IBIFN)&($P(IBNDTX,U,9)):"MRA Secondary ",1:"")\_"Bill has been forced to print "\_$S($P(IBNDTX,U,8)=1!($P(IBNDTX,U,9)=1):"locally",1:"at clearinghouse"))  N IBXZ,IBIZ F IBIZ=12,13,14 S IBXZ=$P(IBNDM,U,IBIZ) I +IBXZ S IBXZ=$P($G(^DPT(DFN,.312,IBXZ,0)),U,18) I +IBXZ S IBXZ=$G(^IBA(355.3,+IBXZ,0)) I +$P(IBXZ,U,12) D  . D WARN^IBCBB11($P($G(^DIC(36,+IBXZ,0)),U,1)\_" requires Amb Care Certification")  ;  D VALNDC^IBCBB11(IBIFN,DFN) ;validate NDC#  ;  ;Build AR array if no errors and MRA not needed or already rec'd  I IBER="",$S($$NEEDMRA^IBEFUNC(IBIFN)!($$REQMRA^IBEFUNC(IBIFN)):0,1:1) D ARRAY  ;  ;Check ROI  N ROIERR  S ROIERR=0 I $P($G(^DGCR(399,IBIFN,"U")),U,5)=1,+$P($G(^DGCR(399,IBIFN,"U")),U,7)=0 S ROIERR=1 ; screen 7 sensitive record and no ROI  I $$ROICHK^IBCBB11(IBIFN,DFN,+IBNDMP) S ROIERR=1 ; check file for sensitive Rx and missing ROI  I ROIERR S IBER=IBER\_"IB328;"  ;  ;Verify Line Charges Match Claim Total Charge. IB\*2.0\*447 BI  I +$$GET1^DIQ(399,IBIFN\_",",201)'=+$$IBLNTOT^IBCBB13(IBIFN) S IBER=IBER\_"IB344;"  ;  ;Test for valid EIN/SY ID Values. IB\*2.0\*447 BI  I $$IBSYEI^IBCBB13(IBIFN) S IBER=IBER\_"IB345;"  ;  ;Test for a missing ICN. IB\*2.0\*447 BI  I $$IBMICN^IBCBB13(IBIFN) S IBER=IBER\_"IB346;"  ;  ;Test for a ZERO charge amounts. IB\*2.0\*447 BI  I $$IBRCCHK^IBCBB13(IBIFN) D WARN^IBCBB11("Claim contains revenue codes with no associated charges.")  ;  ;Test for missing "Patient reason for visit". IB\*2.0\*447 BI  I $$FT^IBCEF(IBIFN)=3,'$$INPAT^IBCEF(IBIFN),$$IBPRV3^IBCBB13(IBIFN) S IBER=IBER\_"IB347;"  ;  ;Test for missing Payer ID. IB\*2.0\*447 BI  ;I $$IBMPID^IBCBB13(IBIFN) S IBER=IBER\_"IB348;"  ;Changed Error to Warning. IB\*2.0\*447 TAZ  I $$IBMPID^IBCBB13(IBIFN) D WARN^IBCBB11("Not all payers have Payer IDs.")  ;  ;Test for missing "Priority (Type) of Admission" for UB-04. IB\*2.0\*447 BI  I $$FT^IBCEF(IBIFN)=3,$$GET1^DIQ(399,IBIFN\_",",158)="" S IBER=IBER\_"IB349;"  ; END ;Don't kill IBIFN, IBER, DFN  I $O(^TMP($J,"BILL-WARN",0)),$G(IBER)="" S IBER="WARN" ;Warnings only  K IBBNO,IBEVDT,IBLOC,IBCL,IBTF,IBAT,IBWHO,IBST,IBFDT,IBTDT,IBTC,IBFY,IBFY1,IBAU,IBRU,IBEU,IBARTP,IBFYC,IBMRA,IBTOB,IBTOB12,IBNDU2,IBNDUF3,IBNDUF31,IBNDTX  K IBNDS,IBND0,IBNDU,IBNDM,IBNDMP,IBNDU1,IBFFY,IBTFY,IBFT,IBRTCHV,IBPICHV,IBXDATA,IBOK  I $D(IBER),IBER="" W !,"No Errors found for National edits"  Q  ; ARRAY ;Build PRCASV(array)  N IBCOBN,X  K PRCASV  Q:$$MCRWNR^IBEFUNC(+$$CURR^IBCEF2(IBIFN))  S IBCOBN=$$COBN^IBCEF(IBIFN)  S X=IBIFN  S PRCASV("BDT")=DT,PRCASV("ARREC")=IBIFN  S PRCASV("APR")=DUZ  S PRCASV("PAT")=DFN,PRCASV("CAT")=$P(^DGCR(399.3,IBAT,0),"^",6)  I IBWHO="i" S PRCASV("DEBTOR")=+IBNDMP\_";DIC(36,"  S PRCASV("DEBTOR")=$S(IBWHO="p":DFN\_";DPT(",IBWHO="o":$P(IBNDM,"^",11)\_";DIC(4,",IBWHO="i":PRCASV("DEBTOR"),1:"")  S PRCASV("CARE")=$E($$TOB^IBCEF1(IBIFN),1,2)  S PRCASV("FY")=$$FY^IBOUTL(DT)\_U\_($P(IBNDU1,U)-$P(IBNDU1,U,2))  ;S PRCASV("FY")=$P(IBNDU1,U,9)\_U\_$S($P(IBNDU1,U,2)]"":($P(IBNDU1,U,10)-$P(IBNDU1,U,2)),1:$P(IBNDU1,U,10))\_$S($P(IBNDU1,U,11)]"":U\_$P(IBNDU1,U,11)\_U\_$P(IBNDU1,U,12),1:"") PLUS I IBWHO="i",$P(IBNDM,"^",2),$D(^DIC(36,$P(IBNDM,"^",2),0)) S PRCASV("2NDINS")=$P(IBNDM,"^",2)  I IBWHO="i",$P(IBNDM,"^",3),$D(^DIC(36,$P(IBNDM,"^",3),0)) S PRCASV("3RDINS")=$P(IBNDM,"^",3)  ;  N IBX S IBX=$P(IBND0,U,21),IBX=$S(IBX="P":"I1",IBX="S":"I2",IBX="T":"I3",1:"") Q:IBX=""  N IBNDI1  Q:'$D(^DGCR(399,IBIFN,IBX)) S IBNDI1=^(IBX)  S:$P(IBNDI1,"^",3)]"" PRCASV("GPNO")=$P(IBNDI1,"^",3)  S:$P(IBNDI1,"^",15)]"" PRCASV("GPNM")=$P(IBNDI1,"^",15)  S:$P(IBNDI1,"^",17)]"" PRCASV("INPA")=$P(IBNDI1,"^",17)  S:$P(IBNDI1,"^",2)]"" PRCASV("IDNO")=$P(IBNDI1,"^",2),PRCASV("INID")=PRCASV("IDNO")  ; Check that this is a secondary or tertiary bill and insurance for previous  ; COB sequence is Medicare WNR and MRA is active --> send data elements to AR  I IBCOBN>1,$$WNRBILL^IBEFUNC(IBIFN,IBCOBN-1),$$EDIACTV^IBCEF4(2) D MRA  Q  ; MRA N IBEOB S IBEOB=0  ;  K PRCASV("MEDURE"),PRCASV("MEDCA")  ; Get EOB data  F  S IBEOB=$O(^IBM(361.1,"B",IBIFN,IBEOB)) Q:'IBEOB  D  . D MRACALC^IBCEMU2(IBEOB,IBIFN,1,.PRCASV)  Q  ;MRA  ;  ;; PREGNANCY DX CODES: V22\*\*-V24\*\*, V27\*\*-V28\*\*, 630\*\*-677\*\*  ;; FLU SHOTS PROCEDURE CODES: 90724, G0008, 90732, G0009 | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCBB1 ;ALB/AAS - CONTINUATION OF EDIT CHECK ROUTINE ;2-NOV-89  ;;2.0;INTEGRATED BILLING;\*\*27,52,80,93,106,51,151,148,153,137,232,280,155,320,343,349,363,371,395,384,432,447,488,554,577,592\*\*;21-MAR-94;Build 1  ;Per VA Directive 6402, this routine should not be modified.  ;  ; \*\*\* Begin IB\*2.0\*488 VD (Issue 46 RBN)  N I  S I=""  S X=+$G(^DGCR(399,IBIFN,"MP"))  I 'X,$$MCRWNR^IBEFUNC(+$$CURR^IBCEF2(IBIFN)) S X=+$$CURR^IBCEF2(IBIFN)  ;JWS;IB\*2.0\*592:US1108 - Dental form check  I X,+$G(^DIC(36,X,3)) S I=$P(^(3),U,$S($$FT^IBCEF(IBIFN)=2:2,$$FT^IBCEF(IBIFN)=7:2,1:4))  S I=$$UP^XLFSTR(I)  I (I'=""&(I["PRNT")&($G(IBER)'["IB488")) D   . S IBER=$G(IBER)\_"IB488;"  ;  ; Cause an error if FORCED TO PRINT TO CLEARINGHOUSE  I $P($G(^DGCR(399,IBIFN,"TX")),U,8)=2 D  . S IBER=$G(IBER)\_"IB489;"  ;  ; Cause a fatal error if the claim has no procedures & is NOT a UB-04 Inpatient claim.  I +$O(^DGCR(399,IBIFN,"CP",0))=0 D  .I $$INPAT^IBCEF(IBIFN,1),$$INSPRF^IBCEF(IBIFN) Q   ; inpatient UB-04 check  .I '$$INPAT^IBCEF(IBIFN,1),$$INSPRF^IBCEF(IBIFN) D  Q      ; Outpatient Institutional Claim.  ..I IBER["IB352" Q  ..S IBER=IBER\_"IB352;"  .;  .; Professional claim  .I IBER["IB353" Q  .S IBER=IBER\_"IB353;"  .Q  ; \*\*\* End IB\*2.0\*488 -- VD  ;  ;MAP TO DGCRBB1  ; % ;Bill Status  N Z,Z0,Z1,IBFT  I $S(+IBST=0:1,1:"^1^2^3^4^7^"'[(U\_IBST\_U)) S IBER=IBER\_"IB045;"  ;  ;Statement Covers From  I IBFDT="" S IBER=IBER\_"IB061;"  I IBFDT]"",IBFDT'?7N&(IBFDT'?7N1".".N) S IBER=IBER\_"IB061;"  I IBFDT>IBTDT S IBER=IBER\_"IB061;" ; from must be on or before the to date   S IBFFY=$$FY^IBOUTL(IBFDT)  ; if inpat - from date must not be prior to admit date.  I $$INPAT^IBCEF(IBIFN,1),(IBFDT<($P($G(^DGPT(+$P(IBND0,U,8),0)),U,2)\1)) S IBER=IBER\_"IB061;"  ;  ;Statement Covers To  I IBTDT="" S IBER=IBER\_"IB062;"  I IBTDT]"",IBTDT'?7N&(IBTDT'?7N1".".N) S IBER=IBER\_"IB062;"  I IBTDT>DT!(IBTDT<IBFDT) S IBER=IBER\_"IB062;"  ; to date must not be >than today's date  S IBTFY=$$FY^IBOUTL(IBTDT)  ;  ;Total Charges  ; IB\*2.0\*447/TAZ Removed this error so that zero dollar revenue codes can process on the 837  ;I +IBTC'>0!(+IBTC'=IBTC) S IBER=IBER\_"IB064;"  ;  ;Billable charges for secondary claim  I $$MCRONBIL^IBEFUNC(IBIFN)&(($P(IBNDU1,U,1)-$P(IBNDU1,U,2))'>0) S IBER=IBER\_"IB094;"  ;Fiscal Year 1  S IBFFY=$$FY^IBOUTL(IBFDT)  ;  ;Check provider link for current user, enterer, reviewer and Authorizor  I '$D(^VA(200,DUZ,0)) S IBER=IBER\_"IB048;"  I IBEU]"",'$D(^VA(200,IBEU,0)) S IBER=IBER\_"IB048;"  I IBRU]"",'$D(^VA(200,IBRU,0)) S IBER=IBER\_"IB060;"  I IBAU]"",'$D(^VA(200,IBAU,0)) S IBER=IBER\_"IB041;"  ;  I IBER="",+$$STA^PRCAFN(IBIFN)=104 S IBER=IBER\_"IB040;"  ; If ins bill, must have valid COB sequence  I $P(IBND0,U,11)="i",$S($P(IBND0,U,21)="":1,1:"PST"'[$P(IBND0,U,21)) S IBER=IBER\_"IB324;"  ;  ; Check for valid sec provider id for current ins  S Z=0 F  S Z=$O(^DGCR(399,IBIFN,"PRV",Z)) Q:'Z  S Z0=$G(^(Z,0)),Z1=+$$COBN^IBCEF(IBIFN) I $P(Z0,U,4+Z1)'="",$P(Z0,U,11+Z1)'="" D  . I '$$SECIDCK^IBCEF74(IBIFN,Z1,$P(Z0,U,11+Z1),Z) D WARN^IBCBB11("Prov secondary id type for the "\_$P("PRIMARY^SECONDARY^TERTIARY",U,Z1)\_" "\_$$EXTERNAL^DILFD(399.0222,.01,,+Z0)\_" is invalid/won't transmit")  ; Check NPIs  D NPICHK^IBCBB11  ;  ; Check multiple rx NPIs  D RXNPI^IBCBB11(IBIFN)  ;  ; Check taxonomies  D TAXCHK^IBCBB11  ;  ; Check for Physician Name  K IBXDATA D F^IBCEF("N-ATT/REND PHYSICIAN NAME",,,IBIFN)  ; IB\*2.0\*432 - CMS1500 no longer needs a claim level rendering  S IBFT=$$FT^IBCEF(IBIFN)  ;JWS;IB\*2.0\*592:US1108 - Dental form check  I IBFT'=2,IBFT'=7,$P($G(IBXDATA),U)="" S IBER=IBER\_"IB303;"  ;  N FUNCTION,IBINS  ; IB\*2.0\*432 - CMS1500 no longer needs a claim level rendering  ;S FUNCTION=$S($$FT^IBCEF(IBIFN)=3:4,1:3)  S FUNCTION=$S(IBFT=3:4,1:3)  ;JWS;IB\*2.0\*592:US1108 - Dental form check  I IBFT'=2,IBFT'=7,IBER'["IB303;" D  . F IBINS=1:1:3 D  .. S Z=$$GETTYP^IBCEP2A(IBIFN,IBINS)  .. I Z,$P(Z,U,2) D  ; Rendering/attending prov secondary id required  ... N IBID,IBOK,Q0  ... D PROVINF^IBCEF74(IBIFN,IBINS,.IBID,1,"C") ; check all as though they were current  ... S IBOK=0  ... S Q0=0 F  S Q0=$O(IBID(1,FUNCTION,Q0)) Q:'Q0  I $P(IBID(1,FUNCTION,Q0),U,9)=+Z S IBOK=1 Q  ... I 'IBOK S IBER=IBER\_$S(IBINS=1:"IB236;",IBINS=2:"IB237;",IBINS=3:"IB238;",1:"")  ;  ; Patch 432 enh5:The IB system shall no longer prevent users from authorizing(fatal error message)a claim because the system cannot find the providersSSNorEIN  ; D PRIIDCHK^IBCBB11  ;  N IBM,IBM1  S IBM=$G(^DGCR(399,IBIFN,"M"))  S IBM1=$G(^DGCR(399,IBIFN,"M1"))  I $P(IBM,U),$P($G(^DIC(36,$P(IBM,U),4)),U,6),$P(IBM1,U,2)="" S IBER=IBER\_"IB244;"  I $P(IBM,U,2),$P($G(^DIC(36,$P(IBM,U,2),4)),U,6),$P(IBM1,U,3)="" S IBER=IBER\_"IB245;"  I $P(IBM,U,3),$P($G(^DIC(36,$P(IBM,U,3),4)),U,6),$P(IBM1,U,4)="" S IBER=IBER\_"IB246;"  ;  ; If outside facility, check for ID and qualifier in 355.93  ; 5/15/06 - esg - hard error IB243 turned into warning message instead  S Z=$P($G(^DGCR(399,IBIFN,"U2")),U,10)  I Z D  . I $P($G(^IBA(355.93,Z,0)),U,9)=""!($P($G(^IBA(355.93,Z,0)),U,13)="") D  .. N Z1,Z2  .. S Z1="Missing Lab or Facility Primary ID for non-VA facility, "  .. S Z2=$$EXTERNAL^DILFD(399,232,,Z)  .. I $L(Z2)'>19 D WARN^IBCBB11(Z1\_Z2) Q  .. D WARN^IBCBB11(Z1),WARN^IBCBB11(" "\_Z2)  .. Q  . Q  ;  ; Must be one and only one division on bill  S IBZ=$$MULTDIV^IBCBB11(IBIFN,IBND0)  ; I IBZ S IBER=IBER\_$S(IBZ=1:"IB095;",IBZ=2:"IB104;",1:"IB105;")  ; Allow multi-divisional for OP instutional claims  I IBZ,$$INPAT^IBCEF(IBIFN)!'($$INSPRF^IBCEF(IBIFN)) S IBER=IBER\_$S(IBZ=1:"IB095;",IBZ=2:"IB104;",1:"IB105;")  ; Still need error msg on OP Institutional if No Default division  I IBZ=3,'$$INPAT^IBCEF(IBIFN),$$INSPRF^IBCEF(IBIFN) S IBER=IBER\_"IB105;"  ; Division address must be defined in institution file  I $P(IBND0,U,22) D  . N Z,Z0,Z1  . S Z0=$G(^DIC(4,+$P($G(^DG(40.8,+$P(IBND0,U,22),0)),U,7),0))  . S Z1=$G(^DIC(4,+$P($G(^DG(40.8,+$P(IBND0,U,22),0)),U,7),1))  . I $P(Z0,U,2)="" S IBER=IBER\_"IB097;" Q  . F Z=1,3,4 I $P(Z1,U,Z)="" S IBER=IBER\_"IB097;" Q  ;  ; IB\*2.0\*432 Check ambulance addresses, COB Non-covered amt. & Attachment Control  I $$AMBCK^IBCBB11(IBIFN)=1 S IBER=IBER\_"IB329;"  I $$COBAMT^IBCBB11(IBIFN)=1 S IBER=IBER\_"IB330;"  I $$TMCK^IBCBB11(IBIFN)=1 S IBER=IBER\_"IB331;"  I $$ACCK^IBCBB11(IBIFN)=1 S IBER=IBER\_"IB332;"  I $$COBMRA^IBCBB11(IBIFN)=1 S IBER=IBER\_"IB342;"  I $$COBSEC^IBCBB11(IBIFN)=1 S IBER=IBER\_"IB343;"  ;  ;CHAMPVA Rate Type and Primary Insurance Carriers Type of Coverage must match  S (IBRTCHV,IBPICHV)=0  I $P($G(^DGCR(399.3,+IBAT,0)),U,1)="CHAMPVA" S IBRTCHV=1  I $P($G(^IBE(355.2,+$P($G(^DIC(36,+IBNDMP,0)),U,13),0)),U,1)="CHAMPVA" S IBPICHV=1  I (+IBRTCHV!+IBPICHV)&('IBRTCHV!'IBPICHV) S IBER=IBER\_"IB085;"  ;  ;Non-VA bill must use FEE REIMB INS rate type; FEE REIMB INS rate type can only be used for Non-VA bill  ;IB\*2.0\*554/DRF 10/9/2015  ;N IBNVART,IBNVAST  ;S (IBNVART,IBNVAST)=0  ;I $P($G(^DGCR(399.3,+IBAT,0)),U,1)="FEE REIMB INS" S IBNVART=1  ;S IBNVAST=$$NONVAFLG(IBIFN)  ;I IBNVART,'IBNVAST S IBER=IBER\_"IB360;" ;Non-VA rate type used for bill that is not Non-VA  ;I 'IBNVART,IBNVAST S IBER=IBER\_"IB361;" ;Non-VA rate type not used for bill that is Non-VA  ;  N IBZPRC,IBZPRCUB  D F^IBCEF("N-ALL PROCEDURES","IBZPRC",,IBIFN)  ; Procedure Clinic is required for Surgical Procedures Outpt Facility Charges  I +$P(IBND0,U,27)'=2,$$BILLRATE^IBCRU3(IBAT,IBCL,IBEVDT,"RC OUTPATIENT") D  . N Z,Z0,Z1,ZE S (ZE,Z)=0 F  S Z=$O(^DGCR(399,IBIFN,"CP",Z)) Q:'Z  D  I +ZE S IBER=IBER\_"IB320;" Q  .. S Z0=$G(^DGCR(399,IBIFN,"CP",Z,0)),Z1=+Z0 I Z0'[";ICPT(" Q  .. I '((Z1'<10000)&(Z1'>69999))&'((Z1'<93501)&(Z1'>93533)) Q  .. I '$P(Z0,U,7) S ZE=1  ;  ; Extract procedures for UB-04  D F^IBCEF("N-UB-04 PROCEDURES","IBZPRCUB",,IBIFN)  ; Does this bill have ANY prescriptions associated with it?  ; Must bill prescriptions separately from other charges  ;  ; DEM;432 - Call line level provider edit checks.  D LNPROV^IBCBB12(IBIFN) ; DEM;432 - If there are line provider edits, then routine LNPROV^IBCBB12(IBIFN) updates IBER string.  ; DEM;432 - Call to Other Operating/Operating Provider edit checks.  I $$OPPROVCK^IBCBB12(IBIFN)=1 S IBER=IBER\_"IB337;"  ; DEM;432  ; DEM;432 - Line level Attachment Control edits.  I $$LNTMCK^IBCBB11(IBIFN)=1 S IBER=IBER\_"IB331;"  ; DEM;432  I $$LNACCK^IBCBB11(IBIFN)=1 S IBER=IBER\_"IB332;"  ; DEM;432  ;  ; vd/Beginning of IB\*2\*577 - Validate Line Level NDC edits.  I $$LNNDCCK^IBCBB11(IBIFN)=1 S IBER=IBER\_"IB360;"  ;IB\*2\*577  ; vd/End of IB\*2\*577  I $$ISRX^IBCEF1(IBIFN) D  . N IBZ,IBRXDEF  . S IBRXDEF=$P($G(^IBE(350.9,1,1)),U,30),IBZ=0  . F  S IBZ=$O(IBZPRCUB(IBZ)) Q:'IBZ  I IBZPRCUB(IBZ),+$P(IBZPRCUB(IBZ),U)'=IBRXDEF S IBER=IBER\_"IB102;" Q  . K IBZ  ;  ; Check that COB sequences are not skipped  K Z  F Z=1:1:3 S:+$G(^DGCR(399,IBIFN,"I"\_Z)) Z(Z)=""  F Z=0:1:2 S Z0=$O(Z(Z)) Q:'Z0  I Z0'=(Z+1) S IBER=IBER\_"IB322;" Q  K Z  ; HD64676 IB\*2\*371 - OK for payer sequence to be blank when the Rate  ; Type is either Interagency or Sharing Agreement  I $P($G(^DGCR(399,IBIFN,0)),U,21)="",$P($G(^DGCR(399,IBIFN,0)),U,7)'=4,$P($G(^DGCR(399,IBIFN,0)),U,7)'=9 S IBER=IBER\_"IB323;"  K IBXDATA D F^IBCEF("N-PROCEDURE CODING METHD",,,IBIFN)  ; Coding method should agree with types of procedure codes  S IBOK=$S('$O(IBZPRC(0))!(IBXDATA=""):1,1:0)  I 'IBOK S IBOK=1,IBZ=0 F  S IBZ=$O(IBZPRC(IBZ)) Q:'IBZ  I IBZPRC(IBZ),$P(IBZPRC(IBZ),U)'[$S(IBXDATA=9:"ICD",1:"ICP") S IBOK=0 Q  I 'IBOK D WARN^IBCBB11("Coding Method does not agree with all procedure codes found on bill")  D EDITMRA^IBCBB3(.IBQUIT,.IBER,IBIFN,IBFT)  Q:$G(IBQUIT)  ;  ;Other things that could be added: Rev Code - calculating charges  ; Diagnosis Coding, if MT copay - check for other co-payments  ;  I $P(IBNDTX,U,8),$$REQMRA^IBEFUNC(IBIFN) S IBER=IBER\_"IB121;"   ; can't force MRAs to print  I $P(IBNDTX,U,8)!$P(IBNDTX,U,9) D  . Q:$P(IBNDTX,U,8)=2 ; Don't want to do this for option 2 any more.  . D WARN^IBCBB11($S($$REQMRA^IBEFUNC(IBIFN)&($P(IBNDTX,U,9)):"MRA Secondary ",1:"")\_"Bill has been forced to print "\_$S($P(IBNDTX,U,8)=1!($P(IBNDTX,U,9)=1):"locally",1:"at clearinghouse"))  N IBXZ,IBIZ F IBIZ=12,13,14 S IBXZ=$P(IBNDM,U,IBIZ) I +IBXZ S IBXZ=$P($G(^DPT(DFN,.312,IBXZ,0)),U,18) I +IBXZ S IBXZ=$G(^IBA(355.3,+IBXZ,0)) I +$P(IBXZ,U,12) D  . D WARN^IBCBB11($P($G(^DIC(36,+IBXZ,0)),U,1)\_" requires Amb Care Certification")  ;  D VALNDC^IBCBB11(IBIFN,DFN) ;validate NDC#  ;  ;Build AR array if no errors and MRA not needed or already rec'd  I IBER="",$S($$NEEDMRA^IBEFUNC(IBIFN)!($$REQMRA^IBEFUNC(IBIFN)):0,1:1) D ARRAY  ;  ;Check ROI  N ROIERR  S ROIERR=0 I $P($G(^DGCR(399,IBIFN,"U")),U,5)=1,+$P($G(^DGCR(399,IBIFN,"U")),U,7)=0 S ROIERR=1 ; screen 7 sensitive record and no ROI  I $$ROICHK^IBCBB11(IBIFN,DFN,+IBNDMP) S ROIERR=1 ; check file for sensitive Rx and missing ROI  I ROIERR S IBER=IBER\_"IB328;"  ;  ;Verify Line Charges Match Claim Total Charge. IB\*2.0\*447 BI  I +$$GET1^DIQ(399,IBIFN\_",",201)'=+$$IBLNTOT^IBCBB13(IBIFN) S IBER=IBER\_"IB344;"  ;  ;Test for valid EIN/SY ID Values. IB\*2.0\*447 BI  I $$IBSYEI^IBCBB13(IBIFN) S IBER=IBER\_"IB345;"  ;  ;Test for a missing ICN. IB\*2.0\*447 BI  I $$IBMICN^IBCBB13(IBIFN) S IBER=IBER\_"IB346;"  ;  ;Test for a ZERO charge amounts. IB\*2.0\*447 BI  I $$IBRCCHK^IBCBB13(IBIFN) D WARN^IBCBB11("Claim contains revenue codes with no associated charges.")  ;  ;Test for missing "Patient reason for visit". IB\*2.0\*447 BI  I $$FT^IBCEF(IBIFN)=3,'$$INPAT^IBCEF(IBIFN),$$IBPRV3^IBCBB13(IBIFN) S IBER=IBER\_"IB347;"  ;  ;Test for missing Payer ID. IB\*2.0\*447 BI  ;I $$IBMPID^IBCBB13(IBIFN) S IBER=IBER\_"IB348;"  ;Changed Error to Warning. IB\*2.0\*447 TAZ  I $$IBMPID^IBCBB13(IBIFN) D WARN^IBCBB11("Not all payers have Payer IDs.")  ;  ;Test for missing "Priority (Type) of Admission" for UB-04. IB\*2.0\*447 BI  I $$FT^IBCEF(IBIFN)=3,$$GET1^DIQ(399,IBIFN\_",",158)="" S IBER=IBER\_"IB349;"  ; END ;Don't kill IBIFN, IBER, DFN  I $O(^TMP($J,"BILL-WARN",0)),$G(IBER)="" S IBER="WARN" ;Warnings only  K IBBNO,IBEVDT,IBLOC,IBCL,IBTF,IBAT,IBWHO,IBST,IBFDT,IBTDT,IBTC,IBFY,IBFY1,IBAU,IBRU,IBEU,IBARTP,IBFYC,IBMRA,IBTOB,IBTOB12,IBNDU2,IBNDUF3,IBNDUF31,IBNDTX  K IBNDS,IBND0,IBNDU,IBNDM,IBNDMP,IBNDU1,IBFFY,IBTFY,IBFT,IBRTCHV,IBPICHV,IBXDATA,IBOK  I $D(IBER),IBER="" W !,"No Errors found for National edits"  Q  ; ARRAY ;Build PRCASV(array)  N IBCOBN,X  K PRCASV  Q:$$MCRWNR^IBEFUNC(+$$CURR^IBCEF2(IBIFN))  S IBCOBN=$$COBN^IBCEF(IBIFN)  S X=IBIFN  S PRCASV("BDT")=DT,PRCASV("ARREC")=IBIFN  S PRCASV("APR")=DUZ  S PRCASV("PAT")=DFN,PRCASV("CAT")=$P(^DGCR(399.3,IBAT,0),"^",6)  I IBWHO="i" S PRCASV("DEBTOR")=+IBNDMP\_";DIC(36,"  S PRCASV("DEBTOR")=$S(IBWHO="p":DFN\_";DPT(",IBWHO="o":$P(IBNDM,"^",11)\_";DIC(4,",IBWHO="i":PRCASV("DEBTOR"),1:"")  S PRCASV("CARE")=$E($$TOB^IBCEF1(IBIFN),1,2)  S PRCASV("FY")=$$FY^IBOUTL(DT)\_U\_($P(IBNDU1,U)-$P(IBNDU1,U,2))  ;S PRCASV("FY")=$P(IBNDU1,U,9)\_U\_$S($P(IBNDU1,U,2)]"":($P(IBNDU1,U,10)-$P(IBNDU1,U,2)),1:$P(IBNDU1,U,10))\_$S($P(IBNDU1,U,11)]"":U\_$P(IBNDU1,U,11)\_U\_$P(IBNDU1,U,12),1:"") PLUS I IBWHO="i",$P(IBNDM,"^",2),$D(^DIC(36,$P(IBNDM,"^",2),0)) S PRCASV("2NDINS")=$P(IBNDM,"^",2)  I IBWHO="i",$P(IBNDM,"^",3),$D(^DIC(36,$P(IBNDM,"^",3),0)) S PRCASV("3RDINS")=$P(IBNDM,"^",3)  ;  N IBX S IBX=$P(IBND0,U,21),IBX=$S(IBX="P":"I1",IBX="S":"I2",IBX="T":"I3",1:"") Q:IBX=""  N IBNDI1  Q:'$D(^DGCR(399,IBIFN,IBX)) S IBNDI1=^(IBX)  S:$P(IBNDI1,"^",3)]"" PRCASV("GPNO")=$P(IBNDI1,"^",3)  S:$P(IBNDI1,"^",15)]"" PRCASV("GPNM")=$P(IBNDI1,"^",15)  S:$P(IBNDI1,"^",17)]"" PRCASV("INPA")=$P(IBNDI1,"^",17)  S:$P(IBNDI1,"^",2)]"" PRCASV("IDNO")=$P(IBNDI1,"^",2),PRCASV("INID")=PRCASV("IDNO")  ; Check that this is a secondary or tertiary bill and insurance for previous  ; COB sequence is Medicare WNR and MRA is active --> send data elements to AR  I IBCOBN>1,$$WNRBILL^IBEFUNC(IBIFN,IBCOBN-1),$$EDIACTV^IBCEF4(2) D MRA  Q  ; MRA N IBEOB S IBEOB=0  ;  K PRCASV("MEDURE"),PRCASV("MEDCA")  ; Get EOB data  F  S IBEOB=$O(^IBM(361.1,"B",IBIFN,IBEOB)) Q:'IBEOB  D  . D MRACALC^IBCEMU2(IBEOB,IBIFN,1,.PRCASV)  Q  ;MRA  ;  ;; PREGNANCY DX CODES: V22\*\*-V24\*\*, V27\*\*-V28\*\*, 630\*\*-677\*\*  ;; FLU SHOTS PROCEDURE CODES: 90724, G0008, 90732, G0009  ; NONVAFLG(IBIFN) ; Check if Non-VA bill  ; Function returns 1 if Non-VA bill  ; IB\*2.0\*554/DRF 10/9/2015  N FLAG,PTF  S FLAG=0  I $P($G(^DGCR(399,IBIFN,"U2")),U,10)]"" S FLAG=1 ;Non-VA provider defined  S PTF=$P($G(^DGCR(399,IBIFN,0)),U,8)  I PTF,$P($G(^DGPT(PTF,0)),U,4)=1 S FLAG=1 ;PTF entry indicates Non-VA  Q FLAG | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCBB11 | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** |  | | | | | | | | |
| **Related Options** | None | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  |  | | | | |  | | | |
| **Data Dictionary (DD) References** | CLAIMS TRACKING File [#356] | | | | | | | | |
| **Related Protocols** | None | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Current Logic** | | | | | | | | | |
| IBCBB11 ;ALB/AAS/OIFO-BP/PIJ - CONTINUATION OF EDIT CHECK ROUTINE ;12 Jun 2006 3:45 PM  ;;2.0;INTEGRATED BILLING;\*\*51,343,363,371,395,392,401,384,400,436,432,516,550,577,592\*\*;21-MAR-94;Build 1  ;;Per VA Directive 6402, this routine should not be modified.  ; WARN(IBDISP) ; Set warning in global  ; DISP = warning text to display  ;  N Z  S Z=+$O(^TMP($J,"BILL-WARN",""),-1)  I Z=0 S ^TMP($J,"BILL-WARN",1)=$J("",5)\_"\*\*Warnings\*\*:",Z=1  S Z=Z+1,^TMP($J,"BILL-WARN",Z)=$J("",5)\_IBDISP  Q  ; MULTDIV(IBIFN,IBND0) ; Check for multiple divisions on a bill ien IBIFN  ; IBND0 = 0-node of bill  ;  ; Function returns 1 if more than 1 division found on bill  N Z,Z0,Z1,MULT  S MULT=0,Z1=$P(IBND0,U,22)  I Z1 D  . S Z=0 F  S Z=$O(^DGCR(399,IBIFN,"RC",Z)) Q:'Z  S Z0=$P(^(Z,0),U,7) I Z0,Z0'=Z1 S MULT=1 Q  . S Z=0 F  S Z=$O(^DGCR(399,IBIFN,"CP",Z)) Q:'Z  S Z0=$P(^(Z,0),U,6) I Z0,Z0'=Z1 S MULT=2 Q  I 'Z1 S MULT=3  Q MULT  ;  ;; PREGNANCY DX CODES: V22\*\*-V24\*\*, V27\*\*-V28\*\*, 630\*\*-677\*\*  ;; FLU SHOTS PROCEDURE CODES: 90724, G0008, 90732, G0009  ; NPICHK ; Check for required NPIs  N IBNPIS,IBNONPI,IBNPIREQ,Z,IBNFI,IBTF,IBWC,IBXSAVE,IBPRV,IBLINE  ;\*\*\* pij start IB\*20\*436 \*\*\*  N IBRATYPE,IBLEGAL  S (IBRATYPE,IBLEGAL)=""  S IBRATYPE=$P($G(^DGCR(399,IBIFN,0)),U,7)  ; Legal types for this use.  ; 7=NO FAULT INS.  ; 10=TORT FEASOR  ; 11=WORKERS' COMP.  S IBNFI=$O(^DGCR(399.3,"B","NO FAULT INS.",0)) S:'IBNFI IBNFI=7  S IBTF=$O(^DGCR(399.3,"B","TORT FEASOR",0)) S:'IBTF IBTF=10  S IBWC=$O(^DGCR(399.3,"B","WORKERS' COMP.",0)) S:'IBWC IBWC=11  ;  I IBRATYPE=IBNFI!(IBRATYPE=IBTF)!(IBRATYPE=IBWC) D  . ; One of the legal types - force local print  . S IBLEGAL=1  ;\*\*\* pij end \*\*\*  S IBNPIREQ=$$NPIREQ^IBCEP81(DT) ; Check if NPI is required  ; Check providers  ; IB\*2.0\*432 changed the NPI check to the new Provider Array  ;S IBNPIS=$$PROVNPI^IBCEF73A(IBIFN,.IBNONPI)  D ALLIDS^IBCEFP(IBIFN,.IBXSAVE,1)  S IBPRV=""  F  S IBPRV=$O(IBXSAVE("PROVINF",IBIFN,"C",1,IBPRV)) Q:'IBPRV  D  . I $P($G(IBXSAVE("PROVINF",IBIFN,"C",1,IBPRV,0)),U,4)="" S IBNONPI(IBPRV)=""  S IBLINE=""  F  S IBLINE=$O(IBXSAVE("L-PROV",IBIFN,IBLINE)) Q:'IBLINE  D  . S IBPRV=""  . F  S IBPRV=$O(IBXSAVE("L-PROV",IBIFN,IBLINE,"C",1,IBPRV)) Q:IBPRV=""  D  .. I $P($G(IBXSAVE("L-PROV",IBIFN,IBLINE,"C",1,IBPRV,0)),U,4)="" S IBNONPI(IBPRV)=""  I $D(IBNONPI) S IBPRV="" F  S IBPRV=$O(IBNONPI(IBPRV)) Q:'IBPRV  D  . ;JWS;IB\*2.0\*592;Assistant Surgeon for dental  . I IBPRV=6 S IBER=IBER\_"IB358;" Q  . S IBER=IBER\_"IB"\_(140+IBPRV)\_";" Q  ; If required, set error IB\*2\*516  ; Check organizations  S IBNONPI=""  S IBNPIS=$$ORGNPI^IBCEF73A(IBIFN,.IBNONPI)  I $L(IBNONPI) F Z=1:1:$L(IBNONPI,U) D  . S IBER=IBER\_$P("IB339;^IB340;^IB341;",U,$P(IBNONPI,U,Z)) ; DEM;432 Added NPI errors.  Q  ; TAXCHK ; Check for required taxonomies  N IBDT,IBLINE,IBNOTAX,IBPRV,IBTAXS,IBXSAVE,Z  ;  ; MRD;IB\*2.0\*516 - This check is now moot; 'today' is always on or  ; after May 23, 2008, so taxonomy codes are always required  ; for certain providers.  ;S IBTAXREQ=$$TAXREQ^IBCEP81(DT) ; Check if taxonomy is required  ;  ; Check providers  ; IB\*2.0\*432 changed the Taxonomy check to the new Provider Array  ;S IBTAXS=$$PROVTAX^IBCEF73A(IBIFN,.IBNOTAX)  D ALLIDS^IBCEFP(IBIFN,.IBXSAVE,1)  S IBPRV=""  F  S IBPRV=$O(IBXSAVE("PROVINF",IBIFN,"C",1,IBPRV)) Q:'IBPRV  D  . I $G(IBXSAVE("PROVINF",IBIFN,"C",1,IBPRV,"TAXONOMY"))="" S IBNOTAX(IBPRV)=""  . Q  ;  S IBLINE=""  F  S IBLINE=$O(IBXSAVE("L-PROV",IBIFN,IBLINE)) Q:'IBLINE  D  . S IBPRV=""  . F  S IBPRV=$O(IBXSAVE("L-PROV",IBIFN,IBLINE,"C",1,IBPRV)) Q:IBPRV=""  D  . . I $G(IBXSAVE("L-PROV",IBIFN,IBLINE,"C",1,IBPRV,"TAXONOMY"))="" S IBNOTAX(IBPRV)=""  . . Q  . Q  ;  ; IB251 = Referring provider taxonomy missing.  ; IB253 = Rendering provider taxonomy missing.  ; IB254 = Attending provider taxonomy missing.  ; IB256 = Assistant Surgeon taxonomy missing. ;JWS;IB\*2.0\*592  ;JWS;IB\*2.0\*592;dental start  I $D(IBNOTAX) S IBPRV="" F  S IBPRV=$O(IBNOTAX(IBPRV)) Q:'IBPRV  D  . ; Only Referring, Rendering and Attending are currently sent to the payer  . ;I IBTAXREQ,"134"[IBPRV S IBER=IBER\_"IB"\_(250+IBPRV)\_";" Q ; MRD;IB\*2.0\*516 - Always required.  . I "1346"[IBPRV S IBER=IBER\_"IB"\_(250+IBPRV)\_";" Q  ; If required, set error and quit  . D WARN("Taxonomy for the "\_$P("referring^operating^rendering^attending^supervising^assistant surgeon^^^other",U,IBPRV)\_$S(IBPRV=6:"",1:" provider")\_" has no value") ; Else, set warning  . Q  ;JWS;IB\*2.0\*592;end  ;  ; Check organizations. The function ORGTAX will set IBNOTAX to be a  ; list of entities missing taxonomy codes, if any (n, n^m, n^m^p,  ; where each 1 is service facility, 2 is non-VA service facility and  ; 3 is billing provider.  ;  S IBNOTAX=""  S IBTAXS=$$ORGTAX^IBCEF73A(IBIFN,.IBNOTAX)  I $L(IBNOTAX) F Z=1:1:$L(IBNOTAX,U) D  . ; IB167 = Billing Provider taxonomy missing.  . ;I IBTAXREQ,$P(IBNOTAX,U,Z)=3 S IBER=IBER\_"IB167;" Q ; MRD;IB\*2.0\*516 - Always required.  . I $P(IBNOTAX,U,Z)=3 S IBER=IBER\_"IB167;" Q  . ; MRD;IB\*2.0\*516 - Remove warning message for missing taxonomy code for lab or facility.  . ; D WARN("Taxonomy for the "\_$P("Service Facility^Non-VA Service Facility^Billing Provider",U,$P(IBNOTAX,U,Z))\_" has no value") ; Else, set warning  . Q  ;  Q  ; VALNDC(IBIFN,IBDFN) ; IB\*2\*363 - validate NDC# between PRESCRIPTION file (#52)  ; and IB BILL/CLAIMS PRESCRIPTION REFILL file (#362.4)  ; input - IBIFN = internal entry number of the billing record in the BILL/CLAIMS file (#399)  ; IBDFN = internal entry number of patient record in the PATIENT file (#2)  N IBX,IBRXCOL  ; call program that determines if NDC differences exist  D VALNDC^IBEFUNC3(IBIFN,IBDFN,.IBRXCOL)  Q:'$D(IBRXCOL)  ; at least one RX on the IB record has an NDC discrepancy   S IBX=0 F  S IBX=$O(IBRXCOL(IBX)) Q:'IBX  D WARN("NDC# on Bill does not equal the NDC# on Rx "\_IBRXCOL(IBX))  Q  ; PRIIDCHK ; Check for required Pimarary ID (SSN/EIN)  ; If the provider is on the claim, he must have one  ;   N IBI,IBZ  I $$TXMT^IBCEF4(IBIFN) D  . D F^IBCEF("N-ALL ATT/REND PROV SSN/EI","IBZ",,IBIFN)  . S IBI="" F  S IBI=$O(^DGCR(399,IBIFN,"PRV","B",IBI)) Q:IBI=""  D  .. I $P(IBZ,U,IBI)="" S IBER=IBER\_$S(IBI=1:"IB151;",IBI=2:"IB152;",IBI=3!(IBI=4):"IB321;",IBI=5:"IB153;",IBI=9:"IB154;",1:"")  Q  ; RXNPI(IBIFN) ; check for multiple pharmacy npi's on the same bill  N IBORG,IBRXNPI,IBX,IBY  S IBORG=$$RXSITE^IBCEF73A(IBIFN,.IBORG)  S IBX=0 F  S IBX=$O(IBORG(IBX)) Q:'IBX  S IBY=0 F  S IBY=$O(IBORG(IBX,IBY)) Q:'IBY  S IBRXNPI(+IBORG(IBX,IBY))=""  S (IBX,IBY)=0 F  S IBX=$O(IBRXNPI(IBX)) Q:'IBX  S IBY=IBY+1  I IBY>1 D WARN("Bill has prescriptions resulting from "\_IBY\_" different NPI locations")  Q  ; ROICHK(IBIFN,IBDFN,IBINS) ; IB\*2.0\*384 - check prescriptions that contain the  ; SENSITIVE DIAGNOSIS DRUG field #87 in the DRUG File #50 set to 1 against  ; the Claims Tracking ROI file (#356.25) to see if an ROI is on file  ; input - IBIFN = IEN of the Bill/Claims file (#399)  ; IBDFN = IEN of the patient  ; IBINS = IEN of the payer insurance company (#36)  ; OUTPUT - 0 = no error   ; 1 = a prescription is sensitive and there is no ROI on file  ;  N IBX,IBY0,IBRXIEN,IBDT,IBDRUG,ROIQ  S ROIQ=0  S IBX=0 F  S IBX=$O(^IBA(362.4,"C",IBIFN,IBX)) Q:'IBX  D  .S IBY0=^IBA(362.4,IBX,0),IBRXIEN=$P(IBY0,U,5) I 'IBRXIEN Q  .S IBDT=$P(IBY0,U,3),IBDRUG=$P(IBY0,U,4)  .D ZERO^IBRXUTL(IBDRUG)  .I $$SENS^IBNCPDR(IBDRUG) D  ; Sensitive Diagnosis Drug - check for ROI  .. I $$ROI^IBNCPDR4(IBDFN,IBDRUG,IBINS,IBDT) Q  ;ROI is on file  .. D WARN("ROI not on file for prescription "\_$$RXAPI1^IBNCPUT1(IBRXIEN,.01,"E"))  .. S ROIQ=1 ROICHKQ ;  K ^TMP($J,"IBDRUG")  Q ROIQ  ; AMBCK(IBIFN) ; IB\*2.0\*432 - if ambulance location defined, address must be defined  ; if there is anything entered in any of the address fields (either p/up or drop/off fields), than there needs to be:   ; Address 1, State and ZIP unless the State is not a US state or possession, then zip code is not needed (CMS1500 only)  ; input - IBIFN = IEN of the Bill/Claims file (#399)  ; OUTPUT - 0 = no error   ; 1 = Error  ;  N IBPAMB,IBDAMB,IBAMBR,IBCK  S IBAMBR=0  Q:$$INSPRF^IBCEF(IBIFN)'=0 IBAMBR  S IBPAMB=$G(^DGCR(399,IBIFN,"U5")),IBDAMB=$G(^DGCR(399,IBIFN,"U6"))  S IBCK(5)=$$NOPUNCT^IBCEF($P(IBPAMB,U,2,6),1),IBCK(6)=$$NOPUNCT^IBCEF($P(IBDAMB,U,1,6),1)  I IBCK(5)="",IBCK(6)="" Q IBAMBR  ; at this point we know that at least one ambulance field has data, so check to see if all have data  I IBCK(5)'="" F I=2,4,5 I $P(IBPAMB,U,I)="" S IBAMBR=1  I IBCK(6)'="" F I=1,2,4,5 I $P(IBDAMB,U,I)="" S IBAMBR=1  Q:IBAMBR=1 IBAMBR  ; now check zip code. OK to be null if state is not a US Posession  F I="IBPAMB","IBDAMB" I $P(I,U,5)'="",$P($G(^DIC(5,$P(I,U,5),0)),U,6)=1,$P(I,U,6)="" S IBAMBR=1  Q IBAMBR  ; COBAMT(IBIFN) ; IB\*2.0\*432 - IF there is a COB amt. it must equal the Total Claim Charge Amount  ; input - IBIFN = IEN of the Bill/Claims file (#399)  ; OUTPUT - 0 = no error   ; 1 = Error  ;  Q:IBIFN="" 0  Q:$P($G(^DGCR(399,IBIFN,"U4")),U)="" 0  Q:+$P($G(^DGCR(399,IBIFN,"U1")),U)'=+$P($G(^DGCR(399,IBIFN,"U4")),U) 1  Q 0  ; COBMRA(IBIFN) ; IB\*2.0\*432 - If there is a 'COB total non-covered amount' (File#399, Field#260),   ; Primary Insurance must be Medicare that never went to Medicare, and this must be a 2ndary or tertiary claim  ; input - IBIFN = IEN of the Bill/Claims file (#399)  ; OUTPUT - 0 = no error   ; 1 = Error  ;  N IBP  Q:IBIFN="" 0  Q:$P($G(^DGCR(399,IBIFN,"U4")),U)="" 0  S IBP=$P($G(^DGCR(399,IBIFN,"M1")),U,5) S:IBP="" IBP=IBIFN  I $$WNRBILL^IBEFUNC(IBIFN,1),$P($G(^DGCR(399,IBP,"S")),U,7)="",$$COBN^IBCEF(IBIFN)>1 Q 0  Q 1  ; COBSEC(IBIFN) ; IB\*2.0\*432 - If there is NOT a 'COB total non-covered amount' (File#399, Field#260),   ; and Primary Insurance is Medicare that never went to Medicare, 2ndary or tertiary claim cannot be set to transmit  ; input - IBIFN = IEN of the Bill/Claims file (#399)  ; OUTPUT - 0 = no error   ; 1 = Error  ;  N IBP  Q:IBIFN="" 0  Q:$P($G(^DGCR(399,IBIFN,"U4")),U)'="" 0  Q:$$COBN^IBCEF(IBIFN)<2 0  S IBP=$P($G(^DGCR(399,IBIFN,"M1")),U,5) S:IBP="" IBP=IBIFN  I $$WNRBILL^IBEFUNC(IBIFN,1),$P($G(^DGCR(399,IBP,"S")),U,7)="",$P($G(^DGCR(399,IBIFN,"TX")),U,8)'=1 Q 1  Q 0  ; TMCK(IBIFN) ; IB\*2.0\*432 - Attachment Control Number - REQUIRED when Transmission Method = BM, EL, EM, or FT  ; input - IBIFN = IEN of the Bill/Claims file (#399)  ; OUTPUT - 0 = no error   ; 1 = Error  ;  N IBAC  Q:IBIFN="" 0  F I=1,3 S IBAC(I)=$P($G(^DGCR(399,IBIFN,"U8")),U,I)  Q:IBAC(3)="" 0  Q:IBAC(1)'="" 0  Q:IBAC(3)="AA" 0  Q 1  ; ACCK(IBIFN) ; IB\*2.0\*432 If any of the loop info is present, then Report Type & Transmission Method req'd  ; input - IBIFN = IEN of the Bill/Claims file (#399)  ; OUTPUT - 0 = no error   ; 1 = Error  ;  N IBAC  Q:IBIFN="" 0  F I=1:1:3 S IBAC(I)=$P($G(^DGCR(399,IBIFN,"U8")),U,I)  ; All fields null, no error  I IBAC(1)="",IBAC(2)="",IBAC(3)="" Q 0  ; Both required fields complete, no error  I IBAC(2)'="",IBAC(3)'="" Q 0  ; At this point, one of the 2 required fields has data and one does not, so error  Q 1  ; LNTMCK(IBIFN) ; DEM;IB\*2.0\*432 - (Line Level) Attachment Control Number - REQUIRED when Transmission Method = BM, EL, EM, or FT  ; input - IBIFN = IEN of the Bill/Claims file (#399)  ; OUTPUT - IBLNERR = 0 = no error   ; IBLNERR = 1 = Error  ;  N IBAC,IBPROCP,I,IBLNERR  S IBLNERR=0 ; DEM;432 - Initialize error flag IBLNERR to '0' for no errors.  Q:IBIFN="" IBLNERR  S IBPROCP=0 F  S IBPROCP=$O(^DGCR(399,IBIFN,"CP",IBPROCP)) Q:'IBPROCP  D  Q:IBLNERR  . Q:'($D(^DGCR(399,IBIFN,"CP",IBPROCP,0))#10) ; DEM;432 - Node '0' is procedure node.  . Q:'($D(^DGCR(399,IBIFN,"CP",IBPROCP,1))#10) ; DEM;432 - Node '1' is line level Attachment Control fields.  . F I=1,3 S IBAC(I)=$P(^DGCR(399,IBIFN,"CP",IBPROCP,1),U,I)  . I IBAC(3)="" S IBLNERR=0 Q  . I IBAC(1)'="" S IBLNERR=0 Q  . I (IBAC(3)="AA") S IBLNERR=0 Q  . S IBLNERR=1  . Q  ;  Q IBLNERR  ; LNACCK(IBIFN) ; DEM;IB\*2.0\*432 (Line Level) If any of the loop info is present, then Report Type & Transmission Method req'd  ; input - IBIFN = IEN of the Bill/Claims file (#399)  ; OUTPUT - IBLNERR = 0 = no error   ; IBLNERR = 1 = Error  ;  N IBAC,IBPROCP,I,IBLNERR  S IBLNERR=0 ; DEM;432 - Initialize error flag IBLNERR to '0' for no errors.  Q:IBIFN="" IBLNERR  S IBPROCP=0 F  S IBPROCP=$O(^DGCR(399,IBIFN,"CP",IBPROCP)) Q:'IBPROCP  D  Q:IBLNERR  . Q:'($D(^DGCR(399,IBIFN,"CP",IBPROCP,0))#10) ; DEM;432 - Node '0' is procedure node.  . Q:'($D(^DGCR(399,IBIFN,"CP",IBPROCP,1))#10) ; DEM;432 - Node '1' is line level Attachment Control fields.  . F I=1:1:3 S IBAC(I)=$P(^DGCR(399,IBIFN,"CP",IBPROCP,1),U,I)  . ; All fields null, no error  . I IBAC(1)="",IBAC(2)="",IBAC(3)="" S IBLNERR=0 Q  . ; Both required fields complete, no error  . I IBAC(2)'="",IBAC(3)'="" S IBLNERR=0 Q  . ; At this point, one of the 2 required fields has data and one does not, so error  . S IBLNERR=1  . Q  ;  Q IBLNERR  ;  ;vd/Beginning of IB\*2\*577 - Validate Line Level for NDC LNNDCCK(IBIFN) ;IB\*2\*577 (Line Level) The Units and Units/Basis of Measurement fields are required if the NDC field is populated.  ; INPUT - IBIFN = IEN of the Bill/Claims file (#399)  ; OUTPUT - IBLNERR = 0 = no error  ; IBLNERR = 1 = Error  ;  N IBAC,IBPROCP,I,IBLNERR  S IBLNERR=0 ; IB\*2\*577 - Initialize error flag IBLNERR to '0' for no errors.  Q:IBIFN="" IBLNERR  S IBPROCP=0 F  S IBPROCP=$O(^DGCR(399,IBIFN,"CP",IBPROCP)) Q:'IBPROCP  D  Q:IBLNERR  . Q:($$GET1^DIQ(399.0304,IBPROCP\_","\_IBIFN\_",","NDC","I")="") ; IB\*2\*577 - No NDC Code  . ; If there is an NDC Code, then the UNITS and UNITS/BASIS OF MEASUREMENT are Required.  . I $$GET1^DIQ(399.0304,IBPROCP\_","\_IBIFN\_",","UNITS/BASIS OF MEASUREMENT","I")="" S IBLNERR=1 Q  . I $$GET1^DIQ(399.0304,IBPROCP\_","\_IBIFN\_",","UNITS","I")="" S IBLNERR=1 Q  ;Units (Quantity) is required if there is an NDC Code.  . Q  ;  Q IBLNERR  ;vd/End of IB\*2\*577 | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCBB11 ;ALB/AAS/OIFO-BP/PIJ - CONTINUATION OF EDIT CHECK ROUTINE ;12 Jun 2006 3:45 PM  ;;2.0;INTEGRATED BILLING;\*\*51,343,363,371,395,392,401,384,400,436,432,516,550,577,592\*\*;21-MAR-94;Build 1  ;;Per VA Directive 6402, this routine should not be modified.  ; WARN(IBDISP) ; Set warning in global  ; DISP = warning text to display  ;  N Z  S Z=+$O(^TMP($J,"BILL-WARN",""),-1)  I Z=0 S ^TMP($J,"BILL-WARN",1)=$J("",5)\_"\*\*Warnings\*\*:",Z=1  S Z=Z+1,^TMP($J,"BILL-WARN",Z)=$J("",5)\_IBDISP  Q  ; MULTDIV(IBIFN,IBND0) ; Check for multiple divisions on a bill ien IBIFN  ; IBND0 = 0-node of bill  ;  ; Function returns 1 if more than 1 division found on bill  N Z,Z0,Z1,MULT  S MULT=0,Z1=$P(IBND0,U,22)  I Z1 D  . S Z=0 F  S Z=$O(^DGCR(399,IBIFN,"RC",Z)) Q:'Z  S Z0=$P(^(Z,0),U,7) I Z0,Z0'=Z1 S MULT=1 Q  . S Z=0 F  S Z=$O(^DGCR(399,IBIFN,"CP",Z)) Q:'Z  S Z0=$P(^(Z,0),U,6) I Z0,Z0'=Z1 S MULT=2 Q  I 'Z1 S MULT=3  Q MULT  ;  ;; PREGNANCY DX CODES: V22\*\*-V24\*\*, V27\*\*-V28\*\*, 630\*\*-677\*\*  ;; FLU SHOTS PROCEDURE CODES: 90724, G0008, 90732, G0009  ; NPICHK ; Check for required NPIs  N IBNPIS,IBNONPI,IBNPIREQ,Z,IBNFI,IBTF,IBWC,IBXSAVE,IBPRV,IBLINE  ;\*\*\* pij start IB\*20\*436 \*\*\*  N IBRATYPE,IBLEGAL  S (IBRATYPE,IBLEGAL)=""  S IBRATYPE=$P($G(^DGCR(399,IBIFN,0)),U,7)  ; Legal types for this use.  ; 7=NO FAULT INS.  ; 10=TORT FEASOR  ; 11=WORKERS' COMP.  S IBNFI=$O(^DGCR(399.3,"B","NO FAULT INS.",0)) S:'IBNFI IBNFI=7  S IBTF=$O(^DGCR(399.3,"B","TORT FEASOR",0)) S:'IBTF IBTF=10  S IBWC=$O(^DGCR(399.3,"B","WORKERS' COMP.",0)) S:'IBWC IBWC=11  ;  I IBRATYPE=IBNFI!(IBRATYPE=IBTF)!(IBRATYPE=IBWC) D  . ; One of the legal types - force local print  . S IBLEGAL=1  ;\*\*\* pij end \*\*\*  S IBNPIREQ=$$NPIREQ^IBCEP81(DT) ; Check if NPI is required  ; Check providers  ; IB\*2.0\*432 changed the NPI check to the new Provider Array  ;S IBNPIS=$$PROVNPI^IBCEF73A(IBIFN,.IBNONPI)  D ALLIDS^IBCEFP(IBIFN,.IBXSAVE,1)  S IBPRV=""  F  S IBPRV=$O(IBXSAVE("PROVINF",IBIFN,"C",1,IBPRV)) Q:'IBPRV  D  . I $P($G(IBXSAVE("PROVINF",IBIFN,"C",1,IBPRV,0)),U,4)="" S IBNONPI(IBPRV)=""  S IBLINE=""  F  S IBLINE=$O(IBXSAVE("L-PROV",IBIFN,IBLINE)) Q:'IBLINE  D  . S IBPRV=""  . F  S IBPRV=$O(IBXSAVE("L-PROV",IBIFN,IBLINE,"C",1,IBPRV)) Q:IBPRV=""  D  .. I $P($G(IBXSAVE("L-PROV",IBIFN,IBLINE,"C",1,IBPRV,0)),U,4)="" S IBNONPI(IBPRV)=""  I $D(IBNONPI) S IBPRV="" F  S IBPRV=$O(IBNONPI(IBPRV)) Q:'IBPRV  D  . ;JWS;IB\*2.0\*592;Assistant Surgeon for dental  . I IBPRV=6 S IBER=IBER\_"IB358;" Q  . S IBER=IBER\_"IB"\_(140+IBPRV)\_";" Q  ; If required, set error IB\*2\*516  ; Check organizations  S IBNONPI=""  S IBNPIS=$$ORGNPI^IBCEF73A(IBIFN,.IBNONPI)  I $L(IBNONPI) F Z=1:1:$L(IBNONPI,U) D  . S IBER=IBER\_$P("IB339;^IB340;^IB341;",U,$P(IBNONPI,U,Z)) ; DEM;432 Added NPI errors.  Q  ; TAXCHK ; Check for required taxonomies  N IBDT,IBLINE,IBNOTAX,IBPRV,IBTAXS,IBXSAVE,Z  ;  ; MRD;IB\*2.0\*516 - This check is now moot; 'today' is always on or  ; after May 23, 2008, so taxonomy codes are always required  ; for certain providers.  ;S IBTAXREQ=$$TAXREQ^IBCEP81(DT) ; Check if taxonomy is required  ;  ; Check providers  ; IB\*2.0\*432 changed the Taxonomy check to the new Provider Array  ;S IBTAXS=$$PROVTAX^IBCEF73A(IBIFN,.IBNOTAX)  D ALLIDS^IBCEFP(IBIFN,.IBXSAVE,1)  S IBPRV=""  F  S IBPRV=$O(IBXSAVE("PROVINF",IBIFN,"C",1,IBPRV)) Q:'IBPRV  D  . I $G(IBXSAVE("PROVINF",IBIFN,"C",1,IBPRV,"TAXONOMY"))="" S IBNOTAX(IBPRV)=""  . Q  ;  S IBLINE=""  F  S IBLINE=$O(IBXSAVE("L-PROV",IBIFN,IBLINE)) Q:'IBLINE  D  . S IBPRV=""  . F  S IBPRV=$O(IBXSAVE("L-PROV",IBIFN,IBLINE,"C",1,IBPRV)) Q:IBPRV=""  D  . . I $G(IBXSAVE("L-PROV",IBIFN,IBLINE,"C",1,IBPRV,"TAXONOMY"))="" S IBNOTAX(IBPRV)=""  . . Q  . Q  ;  ; IB251 = Referring provider taxonomy missing.  ; IB253 = Rendering provider taxonomy missing.  ; IB254 = Attending provider taxonomy missing.  ; IB256 = Assistant Surgeon taxonomy missing. ;JWS;IB\*2.0\*592  ;JWS;IB\*2.0\*592;dental start  I $D(IBNOTAX) S IBPRV="" F  S IBPRV=$O(IBNOTAX(IBPRV)) Q:'IBPRV  D  . ; Only Referring, Rendering and Attending are currently sent to the payer  . ;I IBTAXREQ,"134"[IBPRV S IBER=IBER\_"IB"\_(250+IBPRV)\_";" Q ; MRD;IB\*2.0\*516 - Always required.  . I "1346"[IBPRV S IBER=IBER\_"IB"\_(250+IBPRV)\_";" Q  ; If required, set error and quit  . D WARN("Taxonomy for the "\_$P("referring^operating^rendering^attending^supervising^assistant surgeon^^^other",U,IBPRV)\_$S(IBPRV=6:"",1:" provider")\_" has no value") ; Else, set warning  . Q  ;JWS;IB\*2.0\*592;end  ;  ; Check organizations. The function ORGTAX will set IBNOTAX to be a  ; list of entities missing taxonomy codes, if any (n, n^m, n^m^p,  ; where each 1 is service facility, 2 is non-VA service facility and  ; 3 is billing provider.  ;  S IBNOTAX=""  S IBTAXS=$$ORGTAX^IBCEF73A(IBIFN,.IBNOTAX)  I $L(IBNOTAX) F Z=1:1:$L(IBNOTAX,U) D  . ; IB167 = Billing Provider taxonomy missing.  . ;I IBTAXREQ,$P(IBNOTAX,U,Z)=3 S IBER=IBER\_"IB167;" Q ; MRD;IB\*2.0\*516 - Always required.  . I $P(IBNOTAX,U,Z)=3 S IBER=IBER\_"IB167;" Q  . ; MRD;IB\*2.0\*516 - Remove warning message for missing taxonomy code for lab or facility.  . ; D WARN("Taxonomy for the "\_$P("Service Facility^Non-VA Service Facility^Billing Provider",U,$P(IBNOTAX,U,Z))\_" has no value") ; Else, set warning  . Q  ;  Q  ; VALNDC(IBIFN,IBDFN) ; IB\*2\*363 - validate NDC# between PRESCRIPTION file (#52)  ; and IB BILL/CLAIMS PRESCRIPTION REFILL file (#362.4)  ; input - IBIFN = internal entry number of the billing record in the BILL/CLAIMS file (#399)  ; IBDFN = internal entry number of patient record in the PATIENT file (#2)  N IBX,IBRXCOL  ; call program that determines if NDC differences exist  D VALNDC^IBEFUNC3(IBIFN,IBDFN,.IBRXCOL)  Q:'$D(IBRXCOL)  ; at least one RX on the IB record has an NDC discrepancy   S IBX=0 F  S IBX=$O(IBRXCOL(IBX)) Q:'IBX  D WARN("NDC# on Bill does not equal the NDC# on Rx "\_IBRXCOL(IBX))  Q  ; PRIIDCHK ; Check for required Pimarary ID (SSN/EIN)  ; If the provider is on the claim, he must have one  ;   N IBI,IBZ  I $$TXMT^IBCEF4(IBIFN) D  . D F^IBCEF("N-ALL ATT/REND PROV SSN/EI","IBZ",,IBIFN)  . S IBI="" F  S IBI=$O(^DGCR(399,IBIFN,"PRV","B",IBI)) Q:IBI=""  D  .. I $P(IBZ,U,IBI)="" S IBER=IBER\_$S(IBI=1:"IB151;",IBI=2:"IB152;",IBI=3!(IBI=4):"IB321;",IBI=5:"IB153;",IBI=9:"IB154;",1:"")  Q  ; RXNPI(IBIFN) ; check for multiple pharmacy npi's on the same bill  N IBORG,IBRXNPI,IBX,IBY  S IBORG=$$RXSITE^IBCEF73A(IBIFN,.IBORG)  S IBX=0 F  S IBX=$O(IBORG(IBX)) Q:'IBX  S IBY=0 F  S IBY=$O(IBORG(IBX,IBY)) Q:'IBY  S IBRXNPI(+IBORG(IBX,IBY))=""  S (IBX,IBY)=0 F  S IBX=$O(IBRXNPI(IBX)) Q:'IBX  S IBY=IBY+1  I IBY>1 D WARN("Bill has prescriptions resulting from "\_IBY\_" different NPI locations")  Q  ; ROICHK(IBIFN,IBDFN,IBINS) ; IB\*2.0\*384 - check prescriptions that contain the  ; SENSITIVE DIAGNOSIS DRUG field #87 in the DRUG File #50 set to 1 against  ; the Claims Tracking ROI file (#356.25) to see if an ROI is on file  ; input - IBIFN = IEN of the Bill/Claims file (#399)  ; IBDFN = IEN of the patient  ; IBINS = IEN of the payer insurance company (#36)  ; OUTPUT - 0 = no error   ; 1 = a prescription is sensitive and there is no ROI on file  ;  N IBX,IBY0,IBRXIEN,IBDT,IBDRUG,ROIQ  S ROIQ=0  S IBX=0 F  S IBX=$O(^IBA(362.4,"C",IBIFN,IBX)) Q:'IBX  D  .S IBY0=^IBA(362.4,IBX,0),IBRXIEN=$P(IBY0,U,5) I 'IBRXIEN Q  .S IBDT=$P(IBY0,U,3),IBDRUG=$P(IBY0,U,4)  .D ZERO^IBRXUTL(IBDRUG)  .I $$SENS^IBNCPDR(IBDRUG) D  ; Sensitive Diagnosis Drug - check for ROI  .. I $$ROI^IBNCPDR4(IBDFN,IBDRUG,IBINS,IBDT) Q  ;ROI is on file  .. D WARN("ROI not on file for prescription "\_$$RXAPI1^IBNCPUT1(IBRXIEN,.01,"E"))  .. S ROIQ=1 ROICHKQ ;  K ^TMP($J,"IBDRUG")  Q ROIQ  ; AMBCK(IBIFN) ; IB\*2.0\*432 - if ambulance location defined, address must be defined  ; if there is anything entered in any of the address fields (either p/up or drop/off fields), than there needs to be:   ; Address 1, State and ZIP unless the State is not a US state or possession, then zip code is not needed (CMS1500 only)  ; input - IBIFN = IEN of the Bill/Claims file (#399)  ; OUTPUT - 0 = no error   ; 1 = Error  ;  N IBPAMB,IBDAMB,IBAMBR,IBCK  S IBAMBR=0  Q:$$INSPRF^IBCEF(IBIFN)'=0 IBAMBR  S IBPAMB=$G(^DGCR(399,IBIFN,"U5")),IBDAMB=$G(^DGCR(399,IBIFN,"U6"))  S IBCK(5)=$$NOPUNCT^IBCEF($P(IBPAMB,U,2,6),1),IBCK(6)=$$NOPUNCT^IBCEF($P(IBDAMB,U,1,6),1)  I IBCK(5)="",IBCK(6)="" Q IBAMBR  ; at this point we know that at least one ambulance field has data, so check to see if all have data  I IBCK(5)'="" F I=2,4,5 I $P(IBPAMB,U,I)="" S IBAMBR=1  I IBCK(6)'="" F I=1,2,4,5 I $P(IBDAMB,U,I)="" S IBAMBR=1  Q:IBAMBR=1 IBAMBR  ; now check zip code. OK to be null if state is not a US Posession  F I="IBPAMB","IBDAMB" I $P(I,U,5)'="",$P($G(^DIC(5,$P(I,U,5),0)),U,6)=1,$P(I,U,6)="" S IBAMBR=1  Q IBAMBR  ; COBAMT(IBIFN) ; IB\*2.0\*432 - IF there is a COB amt. it must equal the Total Claim Charge Amount  ; input - IBIFN = IEN of the Bill/Claims file (#399)  ; OUTPUT - 0 = no error   ; 1 = Error  ;  Q:IBIFN="" 0  Q:$P($G(^DGCR(399,IBIFN,"U4")),U)="" 0  Q:+$P($G(^DGCR(399,IBIFN,"U1")),U)'=+$P($G(^DGCR(399,IBIFN,"U4")),U) 1  Q 0  ; COBMRA(IBIFN) ; IB\*2.0\*432 - If there is a 'COB total non-covered amount' (File#399, Field#260),   ; Primary Insurance must be Medicare that never went to Medicare, and this must be a 2ndary or tertiary claim  ; input - IBIFN = IEN of the Bill/Claims file (#399)  ; OUTPUT - 0 = no error   ; 1 = Error  ;  N IBP  Q:IBIFN="" 0  Q:$P($G(^DGCR(399,IBIFN,"U4")),U)="" 0  S IBP=$P($G(^DGCR(399,IBIFN,"M1")),U,5) S:IBP="" IBP=IBIFN  I $$WNRBILL^IBEFUNC(IBIFN,1),$P($G(^DGCR(399,IBP,"S")),U,7)="",$$COBN^IBCEF(IBIFN)>1 Q 0  Q 1  ; COBSEC(IBIFN) ; IB\*2.0\*432 - If there is NOT a 'COB total non-covered amount' (File#399, Field#260),   ; and Primary Insurance is Medicare that never went to Medicare, 2ndary or tertiary claim cannot be set to transmit  ; input - IBIFN = IEN of the Bill/Claims file (#399)  ; OUTPUT - 0 = no error   ; 1 = Error  ;  N IBP  Q:IBIFN="" 0  Q:$P($G(^DGCR(399,IBIFN,"U4")),U)'="" 0  Q:$$COBN^IBCEF(IBIFN)<2 0  S IBP=$P($G(^DGCR(399,IBIFN,"M1")),U,5) S:IBP="" IBP=IBIFN  I $$WNRBILL^IBEFUNC(IBIFN,1),$P($G(^DGCR(399,IBP,"S")),U,7)="",$P($G(^DGCR(399,IBIFN,"TX")),U,8)'=1 Q 1  Q 0  ; TMCK(IBIFN) ; IB\*2.0\*432 - Attachment Control Number - REQUIRED when Transmission Method = BM, EL, EM, or FT  ; input - IBIFN = IEN of the Bill/Claims file (#399)  ; OUTPUT - 0 = no error   ; 1 = Error  ;  N IBAC  Q:IBIFN="" 0  F I=1,3 S IBAC(I)=$P($G(^DGCR(399,IBIFN,"U8")),U,I)  Q:IBAC(3)="" 0  Q:IBAC(1)'="" 0  Q:IBAC(3)="AA" 0  Q 1  ; ACCK(IBIFN) ; IB\*2.0\*432 If any of the loop info is present, then Report Type & Transmission Method req'd  ; input - IBIFN = IEN of the Bill/Claims file (#399)  ; OUTPUT - 0 = no error   ; 1 = Error  ;  N IBAC  Q:IBIFN="" 0  F I=1:1:3 S IBAC(I)=$P($G(^DGCR(399,IBIFN,"U8")),U,I)  ; All fields null, no error  I IBAC(1)="",IBAC(2)="",IBAC(3)="" Q 0  ; Both required fields complete, no error  I IBAC(2)'="",IBAC(3)'="" Q 0  ; At this point, one of the 2 required fields has data and one does not, so error  Q 1  ; LNTMCK(IBIFN) ; DEM;IB\*2.0\*432 - (Line Level) Attachment Control Number - REQUIRED when Transmission Method = BM, EL, EM, or FT  ; input - IBIFN = IEN of the Bill/Claims file (#399)  ; OUTPUT - IBLNERR = 0 = no error   ; IBLNERR = 1 = Error  ;  N IBAC,IBPROCP,I,IBLNERR  S IBLNERR=0 ; DEM;432 - Initialize error flag IBLNERR to '0' for no errors.  Q:IBIFN="" IBLNERR  S IBPROCP=0 F  S IBPROCP=$O(^DGCR(399,IBIFN,"CP",IBPROCP)) Q:'IBPROCP  D  Q:IBLNERR  . Q:'($D(^DGCR(399,IBIFN,"CP",IBPROCP,0))#10) ; DEM;432 - Node '0' is procedure node.  . Q:'($D(^DGCR(399,IBIFN,"CP",IBPROCP,1))#10) ; DEM;432 - Node '1' is line level Attachment Control fields.  . F I=1,3 S IBAC(I)=$P(^DGCR(399,IBIFN,"CP",IBPROCP,1),U,I)  . I IBAC(3)="" S IBLNERR=0 Q  . I IBAC(1)'="" S IBLNERR=0 Q  . I (IBAC(3)="AA") S IBLNERR=0 Q  . S IBLNERR=1  . Q  ;  Q IBLNERR  ; LNACCK(IBIFN) ; DEM;IB\*2.0\*432 (Line Level) If any of the loop info is present, then Report Type & Transmission Method req'd  ; input - IBIFN = IEN of the Bill/Claims file (#399)  ; OUTPUT - IBLNERR = 0 = no error   ; IBLNERR = 1 = Error  ;  N IBAC,IBPROCP,I,IBLNERR  S IBLNERR=0 ; DEM;432 - Initialize error flag IBLNERR to '0' for no errors.  Q:IBIFN="" IBLNERR  S IBPROCP=0 F  S IBPROCP=$O(^DGCR(399,IBIFN,"CP",IBPROCP)) Q:'IBPROCP  D  Q:IBLNERR  . Q:'($D(^DGCR(399,IBIFN,"CP",IBPROCP,0))#10) ; DEM;432 - Node '0' is procedure node.  . Q:'($D(^DGCR(399,IBIFN,"CP",IBPROCP,1))#10) ; DEM;432 - Node '1' is line level Attachment Control fields.  . F I=1:1:3 S IBAC(I)=$P(^DGCR(399,IBIFN,"CP",IBPROCP,1),U,I)  . ; All fields null, no error  . I IBAC(1)="",IBAC(2)="",IBAC(3)="" S IBLNERR=0 Q  . ; Both required fields complete, no error  . I IBAC(2)'="",IBAC(3)'="" S IBLNERR=0 Q  . ; At this point, one of the 2 required fields has data and one does not, so error  . S IBLNERR=1  . Q  ;  Q IBLNERR  ;  ;vd/Beginning of IB\*2\*577 - Validate Line Level for NDC LNNDCCK(IBIFN) ;IB\*2\*577 (Line Level) The Units and Units/Basis of Measurement fields are required if the NDC field is populated.  ; INPUT - IBIFN = IEN of the Bill/Claims file (#399)  ; OUTPUT - IBLNERR = 0 = no error  ; IBLNERR = 1 = Error  ;  N IBAC,IBPROCP,I,IBLNERR  S IBLNERR=0 ; IB\*2\*577 - Initialize error flag IBLNERR to '0' for no errors.  Q:IBIFN="" IBLNERR  S IBPROCP=0 F  S IBPROCP=$O(^DGCR(399,IBIFN,"CP",IBPROCP)) Q:'IBPROCP  D  Q:IBLNERR  . Q:($$GET1^DIQ(399.0304,IBPROCP\_","\_IBIFN\_",","NDC","I")="") ; IB\*2\*577 - No NDC Code  . ; If there is an NDC Code, then the UNITS and UNITS/BASIS OF MEASUREMENT are Required.  . I $$GET1^DIQ(399.0304,IBPROCP\_","\_IBIFN\_",","UNITS/BASIS OF MEASUREMENT","I")="" S IBLNERR=1 Q  . I $$GET1^DIQ(399.0304,IBPROCP\_","\_IBIFN\_",","UNITS","I")="" S IBLNERR=1 Q  ;Units (Quantity) is required if there is an NDC Code.  . Q  ;  Q IBLNERR  ;vd/End of IB\*2\*577 | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCBB12 | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** |  | | | | | | | | |
| **Related Options** | None | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  |  | | | | |  | | | |
| **Data Dictionary (DD) References** | CLAIMS TRACKING File [#356] | | | | | | | | |
| **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** | | | | | | | | | |
| IBCBB12 ;ALB/DEM - PROCEDURE AND LINE LEVEL PROVIDER EDITS ;17-OCT-2010  ;;2.0;INTEGRATED BILLING;\*\*432\*\*;21-MAR-94;Build 192  ;;Per VHA Directive 2004-038, this routine should not be modified.  Q  ; LNPROV(IBIFN) ; DEM;432 - Edits for line level providers.  ;  ; Input:  ; IBIFN - Claim number IEN.  ;  ; Output:  ; OK - '1' Edits  ; '0' No Edits.  ; \*Note: OK returned if called as function.  ; Can be called as routine as well.  ; IBER - Edit error string. Only updated if errors.  ;  ; Patch 432 EDITS:  ;  ; (1) Not all procedures have a Line Level Rendering Provider,  ; and no Claim Level Rendering Provider.  ; Error Message in Billing for Prof Rendering.  ; \*Note: Only applies to Rendering Provider Type.  ;  ; (2) All procedures have a Line Level Rendering Provider,  ; and a Claim Level Rendering Provider who is different  ; from any of the Line Level Rendering Providers.  ; Error in Billing.  ; \*Note: Apply to all provider types (Rendering, Referring, Supervising, Attending, Operating, and Other Operating).  ;  N OK  S OK=0 ; Initialize OK=0 for FALSE.  Q:'$G(IBIFN) OK  ; Need claim number IEN to continue.  N IBPRVFUN,IBCLPRV,IBLNPRV,PRVFUN  S:'$G(IBFT) IBFT=$$FT^IBCEF(IBIFN) ; Form Type for claim.  Q:(IBFT'=2)&(IBFT'=3) OK  ; Must be CMS-1500 (2) or UB-04 (3) Form Type.  S:IBFT=2 PRVFUN(2)="RENDERING,REFERRING,SUPERVISING"  ; Allowable line provider functions for CMS-1500.  S:IBFT=3 PRVFUN(3)="RENDERING,REFERRING,OPERATING,OTHER OPERATING"  ; Allowable line provider functions for UB-04.  F PRVFUN("CNT")=1:1:$L(PRVFUN(IBFT),",") S IBPRVFUN=$P(PRVFUN(IBFT),",",PRVFUN("CNT")) D  . I IBFT=2,IBPRVFUN="RENDERING",'$$LNPRV2(IBPRVFUN),'$D(^DGCR(399,IBIFN,"PRV","C",IBPRVFUN)) D  Q  ; Edit Check (1).  . . S OK=1 ; OK=1 indicates we have at least one error.  . . S IBER=IBER\_"IB333;"  . . Q  . ;  . Q:'$$LNPRV2(IBPRVFUN,.IBLNPRV) ; Quit if not all the procedures have a line level provider of the same provider type.  . Q:'$D(^DGCR(399,IBIFN,"PRV","C",IBPRVFUN)) ; No claim level provider for this provider type.  . ;  . Q:'$$CLPRV2(IBPRVFUN,.IBCLPRV) ; Must have provider for provider type IBPRVFUN to continue (Edit (2)).  . ;  . S IBCLPRV=0 F  S IBCLPRV=$O(IBCLPRV(IBPRVFUN,IBCLPRV)) Q:'IBCLPRV  D  ; Edit Check (2).  . . Q:$D(IBLNPRV(IBPRVFUN,IBCLPRV)) ; Check against line provider array IBLNPRV.  . . S OK=1  . . S IBER=IBER\_"IB334;"  . . Q  . Q  ;  Q OK  ; LNPRV2(IBPRVFUN,IBLNPRV) ; Function - Edit Check (2) for line level provider.  ; See Edit Check (2) at top of routine for details.  ;  ; Input:  ; IBPRVFUN - Provider Type (FUNCTION). Example: RENDERING.  ; IBLNPRV(Array) - Passed by reference. Initially undefined.  ;  ; Output:  ; OK - If Edit Check (2) line level provider condition has  ; been met, then OK will return '1' for TRUE, ELSE, '0'  ; for FALSE.  ; \*See Edit Check (2) at top of routine for details.  ; IBLNPRV(Array) - If Edit Check (2) condition has been met,  ; then IBLNPRV will contain the provider type,  ; and provider variable pointer as array  ; subscripts, and array element is SET to  ; NULL. => IBLNPRV(IBPRVFUN,IBLNPROV)="".  ;  N OK,IBPROCP,IBLPIEN,IBLNPROV  S IBPROCP=0 F  S IBPROCP=$O(^DGCR(399,IBIFN,"CP",IBPROCP)) Q:'IBPROCP  D  I $D(OK),'OK Q  . Q:'($D(^DGCR(399,IBIFN,"CP",IBPROCP,0))#10)  . I '$D(^DGCR(399,IBIFN,"CP",IBPROCP,"LNPRV","C",IBPRVFUN)) S OK=0 Q  ; No line provider function for this procedure.  . S IBLPIEN=$O(^DGCR(399,IBIFN,"CP",IBPROCP,"LNPRV","C",IBPRVFUN,0))  . I 'IBLPIEN S OK=0 Q  ; No line provider IEN for this line provider function.  . I '($D(^DGCR(399,IBIFN,"CP",IBPROCP,"LNPRV",IBLPIEN,0))#10) S OK=0 Q  ; No zero node for line level provider.  . S IBLNPROV=$P(^DGCR(399,IBIFN,"CP",IBPROCP,"LNPRV",IBLPIEN,0),"^",2)  . I 'IBLNPROV S OK=0 Q  ; No line provider for this line provider function.  . S IBLNPRV(IBPRVFUN,IBLNPROV)=""  . Q  ;  Q:$D(OK) OK  ; OK will never equal '1' for TRUE at this point.  I '$D(OK),'$D(IBLNPRV(IBPRVFUN)) S OK=0 Q OK  ; No line provider array for this line provider function.  S OK=1 ; Edit Check (2) line provider condition has been met.  Q OK  ; CLPRV2(IBPRVFUN,IBCLPRV) ; Function - Edit Check (2) for claim level provider.  ; See Edit Check (2) at top of routine for details.  ;  ; Input:  ; IBPRVFUN - Provider Type (FUNCTION). Example: RENDERING.  ; IBCLPRV(Array) - Passed by reference. Intially undefined.  ;  ; Output:  ; OK - If Edit Check (2) claim level provider condition has  ; been met, then OK will return '1' for TRUE, ELSE, '0'  ; for FALSE.  ; \*See Edit Check (2) at top of routine for details.  ; IBCLPRV(Array) - If Edit Check (2) condition has been met,  ; then IBCLPRV will contain the provider type,  ; and provider variable pointer as array  ; subscripts, and array element is SET to  ; NULL. => IBCLPRV(IBPRVFUN,IBCLPROV)="".  ;  N IBCLPIEN,IBCLPROV,OK  S OK=0 ; Initialize OK=0 for FALSE.  S IBCLPIEN=0 F  S IBCLPIEN=$O(^DGCR(399,IBIFN,"PRV","C",IBPRVFUN,IBCLPIEN)) Q:'IBCLPIEN  D  Q:OK  . Q:'($D(^DGCR(399,IBIFN,"PRV",IBCLPIEN,0))#10)  . S IBCLPROV=$P(^DGCR(399,IBIFN,"PRV",IBCLPIEN,0),"^",2)  . Q:'IBCLPROV  . S IBCLPRV(IBPRVFUN,IBCLPROV)=""  ; Set array for Edit Check (2) to compare claim level provider with line level provider.  . S OK=1 ; At this point we have our claim level provider of provider type IBPRVFUN. Set OK=1 for TRUE.  . Q  ;  Q:'OK OK  S OK=1  Q OK  ; OPPROVCK(IBIFN) ; DEM;432 - Other Operating Provider edit checks.  ;  ; Input:  ; IBIFN - Claim number IEN.  ;  ; Output:  ; OK - '1' Edits  ; '0' No Edits.  ; \*Note: OK returned if called as function ($$).  ; Can be called as routine as well.  ;  ; Patch 432 line level Other Operating Provider Edit checks:  ;  ; (1) If claim level Other Operating Provider, then  ; (1.1) claim must have claim level Operating Provider.  ; OR  ; (1.2) every line must have Operating Provider.  ;  ; If (1) Passes, then do edit check (2) below.  ;  ; (2) If any claim line has Other Operating Provider, then  ; (2.1) must have Operating Provider on same claim line,  ; OR  ; (2.2) must have claim level Operating Provider.  ;  N OK  S OK=0 ; Initialize OK=0 for FALSE.  Q:'$G(IBIFN) OK  ; Need claim number IEN to continue.  S:'$G(IBFT) IBFT=$$FT^IBCEF(IBIFN) ; Form Type for claim.  Q:(IBFT'=2)&(IBFT'=3) OK  ; Must be CMS-1500 (2) or UB-04 (3) Form Type.  ;  N IBPRVFUN,IBLNFLAG,IBLNPRV,CLOK,LNOK  ;  ; Note: Claim level provider - OTHER and OTHER OPERATING are the same.  ; Check if condition (1) has been met.  F IBPRVFUN="OTHER","OTHER OPERATING" S CLOK=$$CLOPPRV1(IBPRVFUN) Q:CLOK  Q:'CLOK OK  ; No claim level OTHER OPERATING PROVIDER, then QUIT, no further checks.  S OK=0 ; Initialize OK=0 for FALSE.   ; Condition (1) has been met, check condition (1.1).  S CLOK=0 ; Initialize CLOK=0 for FALSE.  I $D(^DGCR(399,IBIFN,"PRV","C","OPERATING")) S IBPRVFUN="OPERATING",CLOK=$$CLOPPRV1(IBPRVFUN) ; Check condition (1.1).  ; If CLOK at this point, then skip condition check (1.2) and continue to condition (2).  S LNOK=0 ; Initialize LNOK=0 for FALSE.  I 'CLOK S IBPRVFUN="OPERATING",LNOK=$$LNOPPRV1(IBPRVFUN) I 'LNOK S OK=1 Q OK  ; Check condition (1.2). If 'LNOK, then we have an error and QUIT.  ; If LNOK, then continue to condition check (2).  S LNOK=0 ; Initialize LNOK=0 for FALSE.  K IBLNPRV  ; KILL IBLNPRV array before call to $$LNOPPRV1(IBPRVFUN,1,.IBLNPRV).  S IBPRVFUN="OTHER OPERATING",LNOK=$$LNOPPRV1(IBPRVFUN,1,.IBLNPRV) ; Condition check (2) start.  I '$D(IBLNPRV("PRVFUN")) S OK=0 Q OK  ; If no data in IBLNPRV("PRVFUN") array, then skip rest of checks, no error.  ; If data in IBLNPRV("PRVFUN") array, then check condition (2.1).  S IBPRVFUN="OPERATING",LNOK=$$LNOPPRV1(IBPRVFUN,1,.IBLNPRV) ; Condition check (2.1) start.  S LNOK=0 ; Initialize LNOK=0 for FALSE.  D:$D(IBLNPRV("PRVFUN")) ; If data in IBLNPRV("PRVFUN") array, then continue condition check (2.1).  . N IBPROCP  . S IBPROCP=0 F  S IBPROCP=$O(IBLNPRV("PROC",IBPROCP)) Q:'IBPROCP  D  Q:'LNOK  . . I $D(IBLNPRV("PROC",IBPROCP,"OTHER OPERATING")),'$D(IBLNPRV("PROC",IBPROCP,"OPERATING")) S LNOK=0 Q  . . S LNOK=1 ; At this point, we have at least one match. If there wasn't a match, then LNOK=0 and we would have QUIT.  . . Q  . Q  I LNOK S OK=0 Q OK  ; Conditions (2) and (2.1) are met (no error). SET OK=0 and QUIT.  ; If 'LNOK, then continue to condition check (2.2).  S CLOK=0 ; Initialize CLOK=0 for FALSE.  S IBPRVFUN="OPERATING",CLOK=$$CLOPPRV1(IBPRVFUN) ; Condition check (2.2).  I CLOK S OK=0 Q OK  ; Conditions (2) and (2.2) are met (no error). SET OK=0 and QUIT.  ; At this point, we have an error. SET OK=1, and QUIT.  S OK=1  Q OK  ; CLOPPRV1(IBPRVFUN) ; Claim level provider/provider function check.  ;  ; Check if there is a claim level provider with provider function IBPRVFUN.  ;  ; Input:  ; IBPRVFUN - PROVIDER FUNCTION.  ;  ; Output:  ; OK - '1' Claim level provider exist for provider function IBPRVFUN.  ; '0' No Claim level provider exist for provider function IBPRVFUN.  ;  N OK,IBCLPIEN,IBCLPROV  S OK=0 ; Initialize OK=0 for FALSE.  ;  I $D(^DGCR(399,IBIFN,"PRV","C",IBPRVFUN)) D  . S IBCLPIEN=0 F  S IBCLPIEN=$O(^DGCR(399,IBIFN,"PRV","C",IBPRVFUN,IBCLPIEN)) Q:'IBCLPIEN  D  Q:OK  . . Q:'($D(^DGCR(399,IBIFN,"PRV",IBCLPIEN,0))#10)  . . S IBCLPROV=$P(^DGCR(399,IBIFN,"PRV",IBCLPIEN,0),U,2)  . . Q:'IBCLPROV  . . S OK=1 ; At this point we have claim level provider with provider function IBPRVFUN and can QUIT function/subroutine.  . . Q  . Q  ;  Q OK  ; LNOPPRV1(IBPRVFUN,IBLNFLAG,IBLNPRV,IBPROCHK) ; Check every claim line for provider function IBPRVFUN.  ;  ;  ; Input:  ; IBPRVFUN - PROVIDER FUNCTION.  ; IBLNFLAG(Optional) = 1 or 0. 1 indicates return IBLNPRV array passed by reference, otherwise '0' for NO.  ; IBLNPRV(Optional) - Array passed by reference => IF SET OK=1, then  ; I $G(IBLNFLAG) S IBLNPRV("PROC",IBPROCP,IBPRVFUN)="",IBLNPRV("PRVFUN",IBPRVFUN,IBPROCP)=""  ; IBPROCHK - Condition on PROCEDURE (ICD, CPT, or HCFA procedure codes).  ;  ; Output:  ; OK - '1' Every line level provider exist for provider function IBPRVFUN.  ; '0' Not every line level provider exist for provider function IBPRVFUN.  ;  N OK  S OK=0 ; Initialize OK=0 for FALSE.  ;  N IBLPIEN,IBLNPROV,IBPROCP  S IBPROCP=0 F  S IBPROCP=$O(^DGCR(399,IBIFN,"CP",IBPROCP)) Q:'IBPROCP  D  Q:($D(^DGCR(399,IBIFN,"CP",IBPROCP,0))#10)&('OK)  . Q:'($D(^DGCR(399,IBIFN,"CP",IBPROCP,0))#10) ; No procedure '0' node.  . I $G(IBPROCHK)'="" Q:$P(^DGCR(399,IBIFN,"CP",IBPROCP,0),U,1)'[IBPROCHK  . I '$D(^DGCR(399,IBIFN,"CP",IBPROCP,"LNPRV","C",IBPRVFUN)) S OK=0 Q  ; No line provider function IBPRVFUN for this procedure.  . S IBLPIEN=$O(^DGCR(399,IBIFN,"CP",IBPROCP,"LNPRV","C",IBPRVFUN,0))  . I 'IBLPIEN S OK=0 Q  ; No line provider IEN for this line provider function.  . I '($D(^DGCR(399,IBIFN,"CP",IBPROCP,"LNPRV",IBLPIEN,0))#10) S OK=0 Q  ; No '0' node for line level provider.  . S IBLNPROV=$P(^DGCR(399,IBIFN,"CP",IBPROCP,"LNPRV",IBLPIEN,0),U,2)  . I 'IBLNPROV S OK=0 Q  ; No line provider for this line provider function.  . ; At this point we have line level provider of type IBPRVFUN.  . ; S OK=1 for this claim line. OK can be changed back to '0', for FALSE, if claim line fails condition.  . ; We would not get to this point if any line level provider with provider function IBPRVFUN didn't exist.  . S OK=1  . I $G(IBLNFLAG) S IBLNPRV("PROC",IBPROCP,IBPRVFUN)="",IBLNPRV("PRVFUN",IBPRVFUN,IBPROCP)=""  . Q  ;  Q OK  ; UBPRVCK(IBIFN) ; DEM;432 - Check if claim requires operating provider.  ;  ; Description: This function checks if claim requires an operating provider.  ;  ; Checks:  ;  ; (1) If claim has a claim level operating provider,  ; then no further checks (OK=1=TRUE).  ; (2) If claim doesn't have a claim level operating provider,  ; then check:  ; (2.1) Is this a UB-04 claim? NO = QUIT (OK=1), YES = Continue to next check.  ; (2.2) Check every claim line that includes HCPCS procs - operating provider.  ; If every claim line that includes HCPCS procs has an operating provider,  ; then we are OK and QUIT (OK=1).  ; If any claim line that includes HCPCS procs doesn't have an operating  ; provider, then we have an ERROR (OK=0).   ;  ; Input:  ; IBIFN = Claim number IEN.  ;  ; Output:  ; OK = 0 = claim doesn't have an operating provider  ; when operating provider or rendering provider required.  ; OK = 1 = claim has an operating provider, or,  ; claim doesn't require operating provider.  ;  N OK  ; If claim doesn't have any procedure codes, then no checks required.  I '$O(^DGCR(399,IBIFN,"CP",0)) S OK=1 Q OK  ;  S OK=$$CLOPPRV1("OPERATING") ; Do we have a claim level OPERATING PROVIDER (OK=1=TRUE)?  Q:OK OK  ; QUIT, we have a claim level OPERATING PROVIDER (OK=1=TRUE).  ;  N IBFT  S IBFT=($$FT^IBCEF(IBIFN)=3) ; UB-04 claim (1 = TRUE, 0 = FALSE)?  S OK=1 ; Initialize OK=1.  Q:'IBFT OK  ; QUIT OK=1, not a UB-04 claim.  ;  ; Claim level check did not pass, check claim lines.  ; No claim level OPERATING PROVIDER, so check every PROCEDURE for OPERATING PROVIDER.  S OK=$$UBPRVCK1("") ; Does every procedure have an OPERATING PROVIDER(1=TRUE,0=FALSE)?  ;  Q OK  ; UBPRVCK1(IBPROCHK,IBONE) ; DEM;432 - Continuation of UBPRVCK function.  ;  ; Input:  ; IBPROCHK(Optional) - Optional condition on PROCEDURE CODE (ICD, CPT, or HCFA procedure codes).  ; IBONE(Optional) - Quit if at least one line has an OPERATING  ;  ; Output:  ; OK - '1' Every procedure code that contains IBPROCHK (optional check) has an OPERATING PROVIDER.  ; or if IBONE, then at least one procedure code that contains IBPROCHK (optional check) has an OPERATING PROVIDER.  ; '0' Not every procedure code that contains IBPROCHK (optional check) has an OPERATING PROVIDER.  ; or if IBONE, then NO procedure codes that contain IBPROCHK (optional check) has an OPERATING PROVIDER.  ;  N OK  S OK=0 ; Initialize OK=0 for FALSE.  ;  N IBLPIEN,IBLNPROV,IBPROCP  S IBPROCP=0 F  S IBPROCP=$O(^DGCR(399,IBIFN,"CP",IBPROCP)) Q:'IBPROCP  D  Q:($D(^DGCR(399,IBIFN,"CP",IBPROCP,0))#10)&('OK)&('$G(IBONE)) I $G(IBONE),$G(OK) Q  . Q:'($D(^DGCR(399,IBIFN,"CP",IBPROCP,0))#10) ; No procedure '0' node.  . I $G(IBPROCHK)'="" Q:$P(^DGCR(399,IBIFN,"CP",IBPROCP,0),U,1)'[IBPROCHK  . I '$D(^DGCR(399,IBIFN,"CP",IBPROCP,"LNPRV","C","OPERATING")) S OK=0 Q  ; No line OPERATING PROVIDER for this procedure.  . S IBLPIEN=$O(^DGCR(399,IBIFN,"CP",IBPROCP,"LNPRV","C","OPERATING",0))  . I 'IBLPIEN S OK=0 Q  ; No line provider IEN for this line provider function.  . I '($D(^DGCR(399,IBIFN,"CP",IBPROCP,"LNPRV",IBLPIEN,0))#10) S OK=0 Q  ; No '0' node for line level provider.  . S IBLNPROV=$P(^DGCR(399,IBIFN,"CP",IBPROCP,"LNPRV",IBLPIEN,0),U,2)  . I 'IBLNPROV S OK=0 Q  ; No line provider for this line provider function.  . ; At this point we have line level provider of type OPERATING.  . ; S OK=1 for this claim line. OK can be changed back to '0', for FALSE, if claim line fails condition.  . ; We would not get to this point if any line level provider with provider function OPERATING didn't exist.  . S OK=1  . Q  ;  Q OK | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCBB12 ;ALB/DEM - PROCEDURE AND LINE LEVEL PROVIDER EDITS ;17-OCT-2010  ;;2.0;INTEGRATED BILLING;\*\*432,592\*\*;21-MAR-94;Build 192  ;;Per VHA Directive 2004-038, this routine should not be modified.  Q  ; LNPROV(IBIFN) ; DEM;432 - Edits for line level providers.  ;  ; Input:  ; IBIFN - Claim number IEN.  ;  ; Output:  ; OK - '1' Edits  ; '0' No Edits.  ; \*Note: OK returned if called as function.  ; Can be called as routine as well.  ; IBER - Edit error string. Only updated if errors.  ;  ; Patch 432 EDITS:  ;  ; (1) Not all procedures have a Line Level Rendering Provider,  ; and no Claim Level Rendering Provider.  ; Error Message in Billing for Prof Rendering.  ; \*Note: Only applies to Rendering Provider Type.  ;  ; (2) All procedures have a Line Level Rendering Provider,  ; and a Claim Level Rendering Provider who is different  ; from any of the Line Level Rendering Providers.  ; Error in Billing.  ; \*Note: Apply to all provider types (Rendering, Referring, Supervising, Attending, Operating, and Other Operating).  ;  N OK  S OK=0 ; Initialize OK=0 for FALSE.  Q:'$G(IBIFN) OK  ; Need claim number IEN to continue.  N IBPRVFUN,IBCLPRV,IBLNPRV,PRVFUN  S:'$G(IBFT) IBFT=$$FT^IBCEF(IBIFN) ; Form Type for claim.  ; JWS;IB\*2.0\*592 US1108 - Dental form check  I IBFT'=2,IBFT'=3,IBFT'=7 Q OK  ; Must be CMS-1500 (2) or UB-04 (3) or (7) Dental J430D Form Type.  S:IBFT=2 PRVFUN(2)="RENDERING,REFERRING,SUPERVISING"  ; Allowable line provider functions for CMS-1500.  S:IBFT=3 PRVFUN(3)="RENDERING,REFERRING,OPERATING,OTHER OPERATING"  ; Allowable line provider functions for UB-04.  S:IBFT=7 PRVFUN(7)="RENDERING,REFERRING,SUPERVISING,ASSISTANT SURGEON"  ; Allowable line provider functions for Dental form J430D.  ; JWS;IB\*2.0\*592 US1108 - end  F PRVFUN("CNT")=1:1:$L(PRVFUN(IBFT),",") S IBPRVFUN=$P(PRVFUN(IBFT),",",PRVFUN("CNT")) D  . I IBFT=2,IBPRVFUN="RENDERING",'$$LNPRV2(IBPRVFUN),'$D(^DGCR(399,IBIFN,"PRV","C",IBPRVFUN)) D  Q  ; Edit Check (1).  .. S OK=1 ; OK=1 indicates we have at least one error.  .. S IBER=IBER\_"IB333;"  .. Q  . ;JWS;IB\*2.0\*592 - US1108 start  . I IBFT=7,IBPRVFUN="RENDERING",'$$LNPRV2(IBPRVFUN),'$D(^DGCR(399,IBIFN,"PRV","C",IBPRVFUN)) D  Q:OK  ;Edit check for dental  .. I $D(^DGCR(399,IBIFN,"PRV","C","ASSISTANT SURGEON")) Q  .. I $$LNPRV2("ASSISTANT SURGEON") Q  .. S OK=1,IBER=IBER\_"IB357;"  .. Q  . I IBFT=7,IBPRVFUN="ASSISTANT SURGEON",'$$LNPRV2(IBPRVFUN),'$D(^DGCR(399,IBIFN,"PRV","C",IBPRVFUN)) D  Q:OK  ;Edit check for dental  .. I $D(^DGCR(399,IBIFN,"PRV","C","RENDERING")) Q  .. I $$LNPRV2("RENDERING") Q  .. S OK=1,IBER=IBER\_"IB357;"  .. Q  . ;JWS;IB\*2.0\*592 - US1108 end  . Q:'$$LNPRV2(IBPRVFUN,.IBLNPRV) ; Quit if not all the procedures have a line level provider of the same provider type.  . Q:'$D(^DGCR(399,IBIFN,"PRV","C",IBPRVFUN)) ; No claim level provider for this provider type.  . ;  . Q:'$$CLPRV2(IBPRVFUN,.IBCLPRV) ; Must have provider for provider type IBPRVFUN to continue (Edit (2)).  . ;  . S IBCLPRV=0 F  S IBCLPRV=$O(IBCLPRV(IBPRVFUN,IBCLPRV)) Q:'IBCLPRV  D  ; Edit Check (2).  .. Q:$D(IBLNPRV(IBPRVFUN,IBCLPRV)) ; Check against line provider array IBLNPRV.  .. S OK=1  .. S IBER=IBER\_$S(IBPRVFUN="ASSISTANT SURGEON":"IB335;",1:"IB334;")  .. Q  . Q  ;  Q OK  ; LNPRV2(IBPRVFUN,IBLNPRV) ; Function - Edit Check (2) for line level provider.  ; See Edit Check (2) at top of routine for details.  ;  ; Input:  ; IBPRVFUN - Provider Type (FUNCTION). Example: RENDERING.  ; IBLNPRV(Array) - Passed by reference. Intially undefined.  ;  ; Output:  ; OK - If Edit Check (2) line level provider condition has  ; been met, then OK will return '1' for TRUE, ELSE, '0'  ; for FALSE.  ; \*See Edit Check (2) at top of routine for details.  ; IBLNPRV(Array) - If Edit Check (2) condition has been met,  ; then IBLNPRV will contain the provider type,  ; and provider variable pointer as array  ; subscripts, and array element is SET to  ; NULL. => IBLNPRV(IBPRVFUN,IBLNPROV)="".  ;  N OK,IBPROCP,IBLPIEN,IBLNPROV  S IBPROCP=0 F  S IBPROCP=$O(^DGCR(399,IBIFN,"CP",IBPROCP)) Q:'IBPROCP  D  I $D(OK),'OK Q  . Q:'($D(^DGCR(399,IBIFN,"CP",IBPROCP,0))#10)  . I '$D(^DGCR(399,IBIFN,"CP",IBPROCP,"LNPRV","C",IBPRVFUN)) S OK=0 Q  ; No line provider function for this procedure.  . S IBLPIEN=$O(^DGCR(399,IBIFN,"CP",IBPROCP,"LNPRV","C",IBPRVFUN,0))  . I 'IBLPIEN S OK=0 Q  ; No line provider IEN for this line provider function.  . I '($D(^DGCR(399,IBIFN,"CP",IBPROCP,"LNPRV",IBLPIEN,0))#10) S OK=0 Q  ; No zero node for line level provider.  . S IBLNPROV=$P(^DGCR(399,IBIFN,"CP",IBPROCP,"LNPRV",IBLPIEN,0),"^",2)  . I 'IBLNPROV S OK=0 Q  ; No line provider for this line provider function.  . S IBLNPRV(IBPRVFUN,IBLNPROV)=""  . Q  ;  Q:$D(OK) OK  ; OK will never equal '1' for TRUE at this point.  I '$D(OK),'$D(IBLNPRV(IBPRVFUN)) S OK=0 Q OK  ; No line provider array for this line provider function.  S OK=1 ; Edit Check (2) line provider condition has been met.  Q OK  ; CLPRV2(IBPRVFUN,IBCLPRV) ; Function - Edit Check (2) for claim level provider.  ; See Edit Check (2) at top of routine for details.  ;  ; Input:  ; IBPRVFUN - Provider Type (FUNCTION). Example: RENDERING.  ; IBCLPRV(Array) - Passed by reference. Intially undefined.  ;  ; Output:  ; OK - If Edit Check (2) claim level provider condition has  ; been met, then OK will return '1' for TRUE, ELSE, '0'  ; for FALSE.  ; \*See Edit Check (2) at top of routine for details.  ; IBCLPRV(Array) - If Edit Check (2) condition has been met,  ; then IBCLPRV will contain the provider type,  ; and provider variable pointer as array  ; subscripts, and array element is SET to  ; NULL. => IBCLPRV(IBPRVFUN,IBCLPROV)="".  ;  N IBCLPIEN,IBCLPROV,OK  S OK=0 ; Initialize OK=0 for FALSE.  S IBCLPIEN=0 F  S IBCLPIEN=$O(^DGCR(399,IBIFN,"PRV","C",IBPRVFUN,IBCLPIEN)) Q:'IBCLPIEN  D  Q:OK  . Q:'($D(^DGCR(399,IBIFN,"PRV",IBCLPIEN,0))#10)  . S IBCLPROV=$P(^DGCR(399,IBIFN,"PRV",IBCLPIEN,0),"^",2)  . Q:'IBCLPROV  . S IBCLPRV(IBPRVFUN,IBCLPROV)=""  ; Set array for Edit Check (2) to compare claim level provider with line level provider.  . S OK=1 ; At this point we have our claim level provider of provider type IBPRVFUN. Set OK=1 for TRUE.  . Q  ;  Q:'OK OK  S OK=1  Q OK  ; OPPROVCK(IBIFN) ; DEM;432 - Other Operating Provider edit checks.  ;  ; Input:  ; IBIFN - Claim number IEN.  ;  ; Output:  ; OK - '1' Edits  ; '0' No Edits.  ; \*Note: OK returned if called as function ($$).  ; Can be called as routine as well.  ;  ; Patch 432 line level Other Operating Provider Edit checks:  ;  ; (1) If claim level Other Operating Provider, then  ; (1.1) claim must have claim level Operating Provider.  ; OR  ; (1.2) every line must have Operating Provider.  ;  ; If (1) Passes, then do edit check (2) below.  ;  ; (2) If any claim line has Other Operating Provider, then  ; (2.1) must have Operating Provider on same claim line,  ; OR  ; (2.2) must have claim level Operating Provider.  ;  N OK  S OK=0 ; Initialize OK=0 for FALSE.  Q:'$G(IBIFN) OK  ; Need claim number IEN to continue.  S:'$G(IBFT) IBFT=$$FT^IBCEF(IBIFN) ; Form Type for claim.  ; JWS;IB\*2.0\*592 US1108 - Dental form check   I IBFT'=2,IBFT'=3,IBFT'=7 Q OK  ; Must be CMS-1500 (2) or UB-04 (3) Form Type or (7) Dental J430D  ;  N IBPRVFUN,IBLNFLAG,IBLNPRV,CLOK,LNOK  ;  ; Note: Claim level provider - OTHER and OTHER OPERATING are the same.  ; Check if condition (1) has been met.  F IBPRVFUN="OTHER","OTHER OPERATING" S CLOK=$$CLOPPRV1(IBPRVFUN) Q:CLOK  Q:'CLOK OK  ; No claim level OTHER OPERATING PROVIDER, then QUIT, no further checks.  S OK=0 ; Initialize OK=0 for FALSE.   ; Condition (1) has been met, check condtion (1.1).  S CLOK=0 ; Initialize CLOK=0 for FALSE.  I $D(^DGCR(399,IBIFN,"PRV","C","OPERATING")) S IBPRVFUN="OPERATING",CLOK=$$CLOPPRV1(IBPRVFUN) ; Check condition (1.1).  ; If CLOK at this point, then skip condition check (1.2) and continue to condition (2).  S LNOK=0 ; Initialize LNOK=0 for FALSE.  I 'CLOK S IBPRVFUN="OPERATING",LNOK=$$LNOPPRV1(IBPRVFUN) I 'LNOK S OK=1 Q OK  ; Check condition (1.2). If 'LNOK, then we have an error and QUIT.  ; If LNOK, then continue to condition check (2).  S LNOK=0 ; Initialize LNOK=0 for FALSE.  K IBLNPRV  ; KILL IBLNPRV array before call to $$LNOPPRV1(IBPRVFUN,1,.IBLNPRV).  S IBPRVFUN="OTHER OPERATING",LNOK=$$LNOPPRV1(IBPRVFUN,1,.IBLNPRV) ; Condition check (2) start.  I '$D(IBLNPRV("PRVFUN")) S OK=0 Q OK  ; If no data in IBLNPRV("PRVFUN") array, then skip rest of checks, no error.  ; If data in IBLNPRV("PRVFUN") array, then check condition (2.1).  S IBPRVFUN="OPERATING",LNOK=$$LNOPPRV1(IBPRVFUN,1,.IBLNPRV) ; Condition check (2.1) start.  S LNOK=0 ; Initialize LNOK=0 for FALSE.  D:$D(IBLNPRV("PRVFUN")) ; If data in IBLNPRV("PRVFUN") array, then continue condition check (2.1).  . N IBPROCP  . S IBPROCP=0 F  S IBPROCP=$O(IBLNPRV("PROC",IBPROCP)) Q:'IBPROCP  D  Q:'LNOK  . . I $D(IBLNPRV("PROC",IBPROCP,"OTHER OPERATING")),'$D(IBLNPRV("PROC",IBPROCP,"OPERATING")) S LNOK=0 Q  . . S LNOK=1 ; At this point, we have at least one match. If there wasn't a match, then LNOK=0 and we would have QUIT.  . . Q  . Q  I LNOK S OK=0 Q OK  ; Conditions (2) and (2.1) are met (no error). SET OK=0 and QUIT.  ; If 'LNOK, then continue to condition check (2.2).  S CLOK=0 ; Initialize CLOK=0 for FALSE.  S IBPRVFUN="OPERATING",CLOK=$$CLOPPRV1(IBPRVFUN) ; Condition check (2.2).  I CLOK S OK=0 Q OK  ; Conditions (2) and (2.2) are met (no error). SET OK=0 and QUIT.  ; At this point, we have an error. SET OK=1, and QUIT.  S OK=1  Q OK  ; CLOPPRV1(IBPRVFUN) ; Claim level provider/provider function check.  ;  ; Check if there is a claim level provider with provider function IBPRVFUN.  ;  ; Input:  ; IBPRVFUN - PROVIDER FUNCTION.  ;  ; Output:  ; OK - '1' Claim level provider exist for provider function IBPRVFUN.  ; '0' No Claim level provider exist for provider function IBPRVFUN.  ;  N OK,IBCLPIEN,IBCLPROV  S OK=0 ; Initialize OK=0 for FALSE.  ;  I $D(^DGCR(399,IBIFN,"PRV","C",IBPRVFUN)) D  . S IBCLPIEN=0 F  S IBCLPIEN=$O(^DGCR(399,IBIFN,"PRV","C",IBPRVFUN,IBCLPIEN)) Q:'IBCLPIEN  D  Q:OK  . . Q:'($D(^DGCR(399,IBIFN,"PRV",IBCLPIEN,0))#10)  . . S IBCLPROV=$P(^DGCR(399,IBIFN,"PRV",IBCLPIEN,0),U,2)  . . Q:'IBCLPROV  . . S OK=1 ; At this point we have claim level provider with provider function IBPRVFUN and can QUIT function/subroutine.  . . Q  . Q  ;  Q OK  ; LNOPPRV1(IBPRVFUN,IBLNFLAG,IBLNPRV,IBPROCHK) ; Check every claim line for provider function IBPRVFUN.  ;  ;  ; Input:  ; IBPRVFUN - PROVIDER FUNCTION.  ; IBLNFLAG(Optional) = 1 or 0. 1 indicates return IBLNPRV array passed by reference, otherwise '0' for NO.  ; IBLNPRV(Optional) - Array passed by reference => IF SET OK=1, then  ; I $G(IBLNFLAG) S IBLNPRV("PROC",IBPROCP,IBPRVFUN)="",IBLNPRV("PRVFUN",IBPRVFUN,IBPROCP)=""  ; IBPROCHK - Condition on PROCEDURE (ICD, CPT, or HCFA procedure codes).  ;  ; Output:  ; OK - '1' Every line level provider exist for provider function IBPRVFUN.  ; '0' Not every line level provider exist for provider function IBPRVFUN.  ;  N OK  S OK=0 ; Initialize OK=0 for FALSE.  ;  N IBLPIEN,IBLNPROV,IBPROCP  S IBPROCP=0 F  S IBPROCP=$O(^DGCR(399,IBIFN,"CP",IBPROCP)) Q:'IBPROCP  D  Q:($D(^DGCR(399,IBIFN,"CP",IBPROCP,0))#10)&('OK)  . Q:'($D(^DGCR(399,IBIFN,"CP",IBPROCP,0))#10) ; No procedure '0' node.  . I $G(IBPROCHK)'="" Q:$P(^DGCR(399,IBIFN,"CP",IBPROCP,0),U,1)'[IBPROCHK  . I '$D(^DGCR(399,IBIFN,"CP",IBPROCP,"LNPRV","C",IBPRVFUN)) S OK=0 Q  ; No line provider function IBPRVFUN for this procedure.  . S IBLPIEN=$O(^DGCR(399,IBIFN,"CP",IBPROCP,"LNPRV","C",IBPRVFUN,0))  . I 'IBLPIEN S OK=0 Q  ; No line provider IEN for this line provider function.  . I '($D(^DGCR(399,IBIFN,"CP",IBPROCP,"LNPRV",IBLPIEN,0))#10) S OK=0 Q  ; No '0' node for line level provider.  . S IBLNPROV=$P(^DGCR(399,IBIFN,"CP",IBPROCP,"LNPRV",IBLPIEN,0),U,2)  . I 'IBLNPROV S OK=0 Q  ; No line provider for this line provider function.  . ; At this point we have line level provider of type IBPRVFUN.  . ; S OK=1 for this claim line. OK can be changed back to '0', for FALSE, if claim line fails condition.  . ; We would not get to this point if any line level provider with provider function IBPRVFUN didn't exist.  . S OK=1  . I $G(IBLNFLAG) S IBLNPRV("PROC",IBPROCP,IBPRVFUN)="",IBLNPRV("PRVFUN",IBPRVFUN,IBPROCP)=""  . Q  ;  Q OK  ; UBPRVCK(IBIFN) ; DEM;432 - Check if claim requires operating provider.  ;  ; Description: This function checks if claim requires an operating provider.  ;  ; Checks:  ;  ; (1) If claim has a claim level operating provider,  ; then no further checks (OK=1=TRUE).  ; (2) If claim doesn't have a claim level operating provider,  ; then check:  ; (2.1) Is this a UB-04 claim? NO = QUIT (OK=1), YES = Continue to next check.  ; (2.2) Check every claim line that includes HCPCS procs - operating provider.  ; If every claim line that includes HCPCS procs has an operating provider,  ; then we are OK and QUIT (OK=1).  ; If any claim line that includes HCPCS procs doesn't have an operating  ; provider, then we have an ERROR (OK=0).   ;  ; Input:  ; IBIFN = Claim number IEN.  ;  ; Output:  ; OK = 0 = claim doesn't have an operating provider  ; when operating provider or rendering provider required.  ; OK = 1 = claim has an operating provider, or,  ; claim doesn't require operating provider.  ;  N OK  ; If claim doesn't have any procedure codes, then no checks required.  I '$O(^DGCR(399,IBIFN,"CP",0)) S OK=1 Q OK  ;  S OK=$$CLOPPRV1("OPERATING") ; Do we have a claim level OPERATING PROVIDER (OK=1=TRUE)?  Q:OK OK  ; QUIT, we have a claim level OPERATING PROVIDER (OK=1=TRUE).  ;  N IBFT  S IBFT=($$FT^IBCEF(IBIFN)=3) ; UB-04 claim (1 = TRUE, 0 = FALSE)?  S OK=1 ; Initialize OK=1.  Q:'IBFT OK  ; QUIT OK=1, not a UB-04 claim.  ;  ; Claim level check did not pass, check claim lines.  ; No claim level OPERATING PROVIDER, so check every PROCEDURE for OPERATING PROVIDER.  S OK=$$UBPRVCK1("") ; Does every procedure have an OPERATING PROVIDER(1=TRUE,0=FALSE)?  ;  Q OK  ; UBPRVCK1(IBPROCHK,IBONE) ; DEM;432 - Continuation of UBPRVCK function.  ;  ; Input:  ; IBPROCHK(Optional) - Optional condition on PROCEDURE CODE (ICD, CPT, or HCFA procedure codes).  ; IBONE(Optional) - Quit if at least one line has an OPERATING  ;  ; Output:  ; OK - '1' Every procedure code that contains IBPROCHK (optional check) has an OPERATING PROVIDER.  ; or if IBONE, then at least one procedure code that contains IBPROCHK (optional check) has an OPERATING PROVIDER.  ; '0' Not every procedure code that contains IBPROCHK (optional check) has an OPERATING PROVIDER.  ; or if IBONE, then NO procedure codes that contain IBPROCHK (optional check) has an OPERATING PROVIDER.  ;  N OK  S OK=0 ; Initialize OK=0 for FALSE.  ;  N IBLPIEN,IBLNPROV,IBPROCP  S IBPROCP=0 F  S IBPROCP=$O(^DGCR(399,IBIFN,"CP",IBPROCP)) Q:'IBPROCP  D  Q:($D(^DGCR(399,IBIFN,"CP",IBPROCP,0))#10)&('OK)&('$G(IBONE)) I $G(IBONE),$G(OK) Q  . Q:'($D(^DGCR(399,IBIFN,"CP",IBPROCP,0))#10) ; No procedure '0' node.  . I $G(IBPROCHK)'="" Q:$P(^DGCR(399,IBIFN,"CP",IBPROCP,0),U,1)'[IBPROCHK  . I '$D(^DGCR(399,IBIFN,"CP",IBPROCP,"LNPRV","C","OPERATING")) S OK=0 Q  ; No line OPERATING PROVIDER for this procedure.  . S IBLPIEN=$O(^DGCR(399,IBIFN,"CP",IBPROCP,"LNPRV","C","OPERATING",0))  . I 'IBLPIEN S OK=0 Q  ; No line provider IEN for this line provider function.  . I '($D(^DGCR(399,IBIFN,"CP",IBPROCP,"LNPRV",IBLPIEN,0))#10) S OK=0 Q  ; No '0' node for line level provider.  . S IBLNPROV=$P(^DGCR(399,IBIFN,"CP",IBPROCP,"LNPRV",IBLPIEN,0),U,2)  . I 'IBLNPROV S OK=0 Q  ; No line provider for this line provider function.  . ; At this point we have line level provider of type OPERATING.  . ; S OK=1 for this claim line. OK can be changed back to '0', for FALSE, if claim line fails condition.  . ; We would not get to this point if any line level provider with provider function OPERATING didn't exist.  . S OK=1  . Q  ;  Q OK | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCBB3 | | | | | | | | |
| **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** | | | | | | | | | |
| IBCBB3 ;ALB/TMP - CONTINUATION OF EDIT CHECKS ROUTINE (MEDICARE) ;06/23/98  ;;2.0;INTEGRATED BILLING;\*\*51,137,155,349,371,377,432\*\*;21-MAR-94;Build 192  ;;Per VHA Directive 2004-038, this routine should not be modified.  ; EDITMRA(IBQUIT,IBER,IBIFN,IBFT) ;  ; Requires execution of GVAR^IBCBB, IBIFN defined  ; File IB ERROR (350.8) contains error codes/text  ;  N IBMRATYP,Z,IBZP,IBZP1,IBOK  S IBQUIT=0 ;Flag to say we have too many errors - quit edits  ;  S IBMRATYP=$$MRATYPE^IBEFUNC(IBIFN,"C")  ;  I IBFT=3 D  . D PARTA  ;  I IBFT=2 D PARTB^IBCBB9  ;  K IBXDATA D F^IBCEF("N-ADMITTING DIAGNOSIS",,,IBIFN)  ; Req. for UB-04 type of bills 11x!18x  I $G(IBXDATA)="",IBFT=3 D  Q:IBQUIT  . N Z  . I "^11^18^"[(U\_IBTOB12\_U) S IBQUIT=$$IBER(.IBER,231) Q  . I $$INPAT^IBCEF(IBIFN,1) S Z="Admitting Diagnosis may be required by payer, please verify" D WARN^IBCBB11(Z)  ;  D GETPRV^IBCEU(IBIFN,"2,3,4",.Z)  S IBOK=1,Z=0,IBZP=U F  S Z=$O(Z(Z)) Q:'Z  S:$S($P($G(Z(Z,1)),U,3)["VA(200":1,1:0) IBZP=IBZP\_+$P(Z(Z,1),U,3)\_U  D ALLPROC^IBCVA1(IBIFN,.IBZP1)  ;patch 432, enh5: The IB system shall no longer provide users with a warning message when authorizing a claim when line level and claim level providers are not the same.  ;S Z=0 F S Z=$O(IBZP1(Z)) Q:'Z I $P(IBZP1(Z),U,18),IBZP'[(U\_$P(IBZP1(Z),U,18)\_U) S IBOK=0 Q  ;I 'IBOK D WARN^IBCBB11("At least one provider on a procedure does not match your "\_$S(IBFT=2:"render",1:"attend")\_"ing or operating provider")  I IBFT=2 D EN^IBCBB2  ; edit checks for UB-04 (institutional) forms  I IBFT=3 D EN^IBCBB21(.IBZPRC92)  ;  Q  ; PARTA ; MEDICARE specific edit checks for PART A claims (UB-04 formats)  ;  N IBI,IBJ,IBX,IBCTYP,VADM,VAPA,IBSTOP,IBDXC,IBDXARY,IBPR,IBLABS,REQMRA  N IBS,IBTUNIT,IBCAGE,IBREV1,IBOCCS,IBOCSDT,IBVALCD,IBOCCD,IBNOPR  N IBCCARY1,IBPATST,IBZADMIT,IBZDISCH,IBXIEN,IBXERR,IBXDATA,IBOCSP  N IBCOV,IBNCOV,IBREVC,IBREVDUP,IBBCPT,IBREVC12,IBREVTOT,IBECAT,IBINC  ;  ; Medicare is the current payer, but no diagnosis codes  I $$WNRBILL^IBEFUNC(IBIFN) D SET^IBCSC4D(IBIFN,.IBDX,.IBDXO) I '$P(IBDX,U,2) S IBQUIT=$$IBER(.IBER,120) Q:IBQUIT  ;  ; Type of Bill must be three digits  I IBTOB'?3N S X=$$IBER(.IBER,103) Q  ;  ; Covered Days  S IBCTYP=0  S IBCOV=$P(IBNDU2,U,2),IBNCOV=$P(IBNDU2,U,3)  ;  ; If interim bill, covered days must not be greater than 60  ; remove for IB\*2.0\*432  ; I "23"[$E(IBTOB,3),IBCOV>60 S IBQUIT=$$IBER(.IBER,"096") Q:IBQUIT  ;  ; I bill type is 11x or 18x or 21x then we need covered days  ; remove for IB\*2.0\*432  ; I "^11^18^21^"[(U\_IBTOB12\_U) S IBCTYP=1 I IBCOV="" S IBQUIT=$$IBER(.IBER,106) Q:IBQUIT  ;  S (IBI,IBJ)=0  K IBXDATA D F^IBCEF("N-CONDITION CODES",,,IBIFN)  ; Re-sort the condition codes by code  S IBI=0 F  S IBI=$O(IBXDATA(IBI)) Q:'IBI  S IBCCARY1($P(IBXDATA(IBI),U))=""  ;  ; for condition code 40, covered days must be 0  ; remove for IB\*2.0\*432  ; I $D(IBCCARY1(40)),IBCOV'=0 S IBQUIT=$$IBER(.IBER,107) Q:IBQUIT  ;  ; cov days+non=to date -from date unless the patient status = 30 (still  ; pt) or outpatient or if the to date and from date are same then add 1  S IBPATST="",IBX=$P(IBNDU,U,12),IBPATST=$P($G(^DGCR(399.1,+IBX,0)),U,2)  S IBINC=$S(IBPATST=30!(IBFDT=IBTDT):1,1:0)  ; remove for IB\*2.0\*432  ;I $$INPAT^IBCEF(IBIFN,1),(IBCOV+IBNCOV)'=($$FMDIFF^XLFDT(IBTDT,IBFDT)+IBINC) S IBQUIT=$$IBER(.IBER,108) Q:IBQUIT  ;  ; if covered days >100 and type of bill is 21x or 18x error  ; remove for IB\*2.0\*432  ; I IBCOV>100,(IBTOB12=18!(IBTOB12=21)) S IBQUIT=$$IBER(.IBER,109) Q:IBQUIT  ;  S (IBJ,IBTUNIT,IBS,IBREVTOT("AC"),IBREVTOT("AI"),IBREVTOT("AO"),IBREVTOT)=0  ;  K IBXDATA D F^IBCEF("N-UB-04 SERVICE LINE (EDI)",,,IBIFN) ;Get rev codes  ;  ; Re-sort the revenue codes by code  ;>> IBREV1(rev code,x)=Rev code^ptr cpt^unit chg^units^total^tot unc  ; IBREV1(rev code) = revenue code edit category  ;  ; IBNOPR = flag that determines if there are revenue codes with  ; charges that do not have a procedure - no need to check  ; for billable MCR procedures if at least one RC is billable  ; 1 = there is at least one billable revenue code without a  ; procedure  ;  S REQMRA=$$REQMRA^IBEFUNC(IBIFN)  S (IBNOPR,IBI)=0  F  S IBI=$O(IBXDATA(IBI)) Q:'IBI  D  . I REQMRA D GYMODCHK(IBXDATA(IBI)) ; IB\*2\*377 GY modifier check  . S IBJ=$P(IBXDATA(IBI),U),IBECAT=""  . I 'IBNOPR D  .. I $P(IBXDATA(IBI),U,2)'="" S IBPR($P(IBXDATA(IBI),U,2))=IBI Q  .. S IBNOPR=1 K IBPR  . S:$D(IBREV1(IBJ)) IBECAT=$G(IBREV1(IBJ))  . I '$D(IBREV1(IBJ))!(IBECAT="") D  S IBREV1(IBJ)=IBECAT  . . ;  . . ; Accomodations (AC)  . . I (IBJ'<100&(IBJ'>219))!(IBJ=224) S IBECAT="AC" Q  . . ;  . . ; Ancillary Outpatient (AO)  . . I '$$INPAT^IBCEF(IBIFN,1) S IBECAT="AO" Q  . . ;  . . ; Ancillary Inpatient (AI)  . . S IBECAT="AI"  . ;  . S IBREV1(IBJ,+$O(IBREV1(IBJ,""),-1)+1)=IBXDATA(IBI)  . S IBREVTOT(IBECAT)=IBREVTOT(IBECAT)+$P(IBXDATA(IBI),U,6)  . I IBECAT="AC" S IBTUNIT=IBTUNIT+$P(IBXDATA(IBI),U,4)  ;  I $$NEEDMRA^IBEFUNC(IBIFN),$O(IBPR(""))'="" D  Q:IBQUIT  . ; Don't allow a bill containing only billable procedures for:  . ; Oxygen, labs, or influenza shots  . ; OR a bill with prosthetics on it  . ; to be sent to MEDICARE for an MRA  . D NONMCR(.IBPR,.IBLABS) ; Remove Oxygen, labs, influenza shots  . I $G(IBLABS) D WARN^IBCBB11("There are Lab procedures on this claim."),WARN^IBCBB11("Please verify that MEDICARE does not reimburse these labs at 100%") Q  . I $O(IBPR(""))="" D  .. S IBQUIT=$$IBER(.IBER,"098")  ;  ; covered days+non covered = units of accom rev codes  ; Check room and board  ; remove for IB\*2.0\*432  ;I IBTUNIT,IBTUNIT'=(IBCOV+IBNCOV) S IBQUIT=$$IBER(.IBER,114) Q:IBQUIT  ;  ; Non Covered Days  ; required when the type of bill is 11x,18x,21x or covered days=0  ; remove for IB\*2.0\*432  ; I IBNCOV="",(IBCTYP!(IBCOV=0)) S IBQUIT=$$IBER(.IBER,115) Q:IBQUIT  ;  ; if cc code=40 then non-covered days must be 1  ; remove for IB\*2.0\*432  ; I $D(IBCCARY1(40)),IBNCOV'=1 S IBQUIT=$$IBER(.IBER,116) Q:IBQUIT  ;  ; Patient Sex  ; must be "M" or "F"  D DEM^VADPT  I $P(VADM(5),U)'="M",$P(VADM(5),U)'="F" S IBQUIT=$$IBER(.IBER,124) Q:IBQUIT  ;  ; esg - 10/17/07 - patch 371  ; For Part A replacement MRA request claims, make sure  ; the Medicare ICN/DCN number is present and also text in FL-80.  I $$REQMRA^IBEFUNC(IBIFN),$F(".137.138.117.118.","."\_IBTOB\_".") D  Q:IBQUIT  . N IBZ,FL80TXT  . D F^IBCEF("N-CURR INS FORM LOC 64","IBZ",,IBIFN) ; see CI3-11  . I IBZ="" S IBQUIT=$$IBER(.IBER,205) Q:IBQUIT      ; missing ICN/DCN  . S FL80TXT=$P($G(^DGCR(399,IBIFN,"UF2")),U,3)  . I FL80TXT="" S IBQUIT=$$IBER(.IBER,206) Q:IBQUIT  ; missing FL80 text  . Q  ;  D ^IBCBB4  Q  ; IBER(IBER,ERRNO) ; Sets error list  ; NOTE: add code to check error list > 20 ... If so, display message and  ; quit so we don't get too many errors at once to handle  ; Print all if printing list  ;  I '$G(IBQUIT) D  . I ERRNO?1N.N S:$L(ERRNO)<3 ERRNO=$E("00",1,3-$L(ERRNO))\_ERRNO  . I $L(IBER,";")>19,'$G(IBPRT("PRT")) S IBER=IBER\_"IB999;",IBQUIT=1  . I $G(IBER)'[("IB"\_ERRNO\_";") S IBER=IBER\_"IB"\_ERRNO\_";"  Q IBQUIT  ; NONMCR(IBPR,IBLABS) ; Delete all oxygen and lab, flu shot CPT entries from IBPR  ; IBPR = array subscripted by CPT codes from bill  ; IBLABS = flag returned =1 if labs found on bill  N Z S IBLABS=0  ; Oxygen  F Z="A0422","A4575","A4616","A4619","A4620","A4621","E0455","E1353","E1355" K IBPR(Z)  F Z=77:1:85 S Z0="E13"\_Z K IBPR(Z0)  ; Labs  S Z="80000" F  S Z=$O(IBPR(Z)) Q:Z'?1"8"4N  S IBLABS=1  ; Flu shots  F Z="90724","G0008","90732","G0009","90657","90658","90659","90660" K IBPR(Z)  Q  ; MCRANUM(IBIFN) ; Determine MEDICARE A provider ID # from bedsection for  ; bill ien IBIFN  N IBX  ; PART A MRA (only) needed - determine if psych/non-psych claim  N IBX,IBI  S IBI=$P($G(^DGCR(399,IBIFN,"U")),U,11)  S IBX=$S($TR($P($G(^DGCR(399.1,+IBI,0)),U),"psych","PSYCH")'["PSYCH":670899,1:674499)  Q IBX  ; MCRACK(IBIFN,X,IBFLD) ; Check for MEDICARE A for bill IBIFN  ; Called from CLAIM STATUS MRA field (#24) xrefs in file 399  ; X = current value of field 399;24  ; IBFLD = 1 for primary ins co, 2 for secondary, 3 for tertiary  N IB  S IB=0  I +X,$$COBN^IBCEF(IBIFN)=IBFLD,$$WNRBILL^IBEFUNC(IBIFN,IBFLD),$$MRATYPE^IBEFUNC(IBIFN,"C")="A" S IB=1  Q IB  ; GYMODCHK(Z) ; GY modifier check procedure. IB\*2\*377 - 2/4/08  ; Z is the IBXDATA(IBI) service line EDI  N MODS  I IBER["IB123" Q     ; error already found  S MODS=$P(Z,U,9) ; list of modifiers separated by commas  I MODS'["GY" Q       ; GY modifier not here on this line item  I $P(Z,U,6) Q        ; non-covered charges exist on this line item  S IBQUIT=$$IBER(.IBER,123) GYMODX ;  Q  ; | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCBB3 ;ALB/TMP - CONTINUATION OF EDIT CHECKS ROUTINE (MEDICARE) ;06/23/98  ;;2.0;INTEGRATED BILLING;\*\*51,137,155,349,371,377,432,592\*\*;21-MAR-94;Build 192  ;;Per VHA Directive 2004-038, this routine should not be modified.  ; EDITMRA(IBQUIT,IBER,IBIFN,IBFT) ;  ; Requires execution of GVAR^IBCBB, IBIFN defined  ; File IB ERROR (350.8) contains error codes/text  ;  N IBMRATYP,Z,IBZP,IBZP1,IBOK  S IBQUIT=0 ;Flag to say we have too many errors - quit edits  ;  S IBMRATYP=$$MRATYPE^IBEFUNC(IBIFN,"C")  ;  I IBFT=3 D  . D PARTA  ;JWS;IB\*2.0\*592 US1108 - Dental form check  I IBFT=2!(IBFT=7) D PARTB^IBCBB9  ;  K IBXDATA D F^IBCEF("N-ADMITTING DIAGNOSIS",,,IBIFN)  ; Req. for UB-04 type of bills 11x!18x  I $G(IBXDATA)="",IBFT=3 D  Q:IBQUIT  . N Z  . I "^11^18^"[(U\_IBTOB12\_U) S IBQUIT=$$IBER(.IBER,231) Q  . I $$INPAT^IBCEF(IBIFN,1) S Z="Admitting Diagnosis may be required by payer, please verify" D WARN^IBCBB11(Z)  ;  D GETPRV^IBCEU(IBIFN,"2,3,4",.Z)  S IBOK=1,Z=0,IBZP=U F  S Z=$O(Z(Z)) Q:'Z  S:$S($P($G(Z(Z,1)),U,3)["VA(200":1,1:0) IBZP=IBZP\_+$P(Z(Z,1),U,3)\_U  D ALLPROC^IBCVA1(IBIFN,.IBZP1)  ;patch 432, enh5: The IB system shall no longer provide users with a warning message when authorizing a claim when line level and claim level providers are not the same.  ;S Z=0 F S Z=$O(IBZP1(Z)) Q:'Z I $P(IBZP1(Z),U,18),IBZP'[(U\_$P(IBZP1(Z),U,18)\_U) S IBOK=0 Q  ;I 'IBOK D WARN^IBCBB11("At least one provider on a procedure does not match your "\_$S(IBFT=2:"render",1:"attend")\_"ing or operating provider")  ; JWS;IB\*2.0\*592 US1108 - Dental form check   I IBFT=2!(IBFT=7) D EN^IBCBB2  ; edit checks for UB-04 (institutional) forms  I IBFT=3 D EN^IBCBB21(.IBZPRC92)  ;  Q  ; PARTA ; MEDICARE specific edit checks for PART A claims (UB-04 formats)  ;  N IBI,IBJ,IBX,IBCTYP,VADM,VAPA,IBSTOP,IBDXC,IBDXARY,IBPR,IBLABS,REQMRA  N IBS,IBTUNIT,IBCAGE,IBREV1,IBOCCS,IBOCSDT,IBVALCD,IBOCCD,IBNOPR  N IBCCARY1,IBPATST,IBZADMIT,IBZDISCH,IBXIEN,IBXERR,IBXDATA,IBOCSP  N IBCOV,IBNCOV,IBREVC,IBREVDUP,IBBCPT,IBREVC12,IBREVTOT,IBECAT,IBINC  ;  ; Medicare is the current payer, but no diagnosis codes  I $$WNRBILL^IBEFUNC(IBIFN) D SET^IBCSC4D(IBIFN,.IBDX,.IBDXO) I '$P(IBDX,U,2) S IBQUIT=$$IBER(.IBER,120) Q:IBQUIT  ;  ; Type of Bill must be three digits  I IBTOB'?3N S X=$$IBER(.IBER,103) Q  ;  ; Covered Days  S IBCTYP=0  S IBCOV=$P(IBNDU2,U,2),IBNCOV=$P(IBNDU2,U,3)  ;  ; If interim bill, covered days must not be greater than 60  ; remove for IB\*2.0\*432  ; I "23"[$E(IBTOB,3),IBCOV>60 S IBQUIT=$$IBER(.IBER,"096") Q:IBQUIT  ;  ; I bill type is 11x or 18x or 21x then we need covered days  ; remove for IB\*2.0\*432  ; I "^11^18^21^"[(U\_IBTOB12\_U) S IBCTYP=1 I IBCOV="" S IBQUIT=$$IBER(.IBER,106) Q:IBQUIT  ;  S (IBI,IBJ)=0  K IBXDATA D F^IBCEF("N-CONDITION CODES",,,IBIFN)  ; Re-sort the condition codes by code  S IBI=0 F  S IBI=$O(IBXDATA(IBI)) Q:'IBI  S IBCCARY1($P(IBXDATA(IBI),U))=""  ;  ; for condition code 40, covered days must be 0  ; remove for IB\*2.0\*432  ; I $D(IBCCARY1(40)),IBCOV'=0 S IBQUIT=$$IBER(.IBER,107) Q:IBQUIT  ;  ; cov days+non=to date -from date unless the patient status = 30 (still  ; pt) or outpatient or if the to date and from date are same then add 1  S IBPATST="",IBX=$P(IBNDU,U,12),IBPATST=$P($G(^DGCR(399.1,+IBX,0)),U,2)  S IBINC=$S(IBPATST=30!(IBFDT=IBTDT):1,1:0)  ; remove for IB\*2.0\*432  ;I $$INPAT^IBCEF(IBIFN,1),(IBCOV+IBNCOV)'=($$FMDIFF^XLFDT(IBTDT,IBFDT)+IBINC) S IBQUIT=$$IBER(.IBER,108) Q:IBQUIT  ;  ; if covered days >100 and type of bill is 21x or 18x error  ; remove for IB\*2.0\*432  ; I IBCOV>100,(IBTOB12=18!(IBTOB12=21)) S IBQUIT=$$IBER(.IBER,109) Q:IBQUIT  ;  S (IBJ,IBTUNIT,IBS,IBREVTOT("AC"),IBREVTOT("AI"),IBREVTOT("AO"),IBREVTOT)=0  ;  K IBXDATA D F^IBCEF("N-UB-04 SERVICE LINE (EDI)",,,IBIFN) ;Get rev codes  ;  ; Re-sort the revenue codes by code  ;>> IBREV1(rev code,x)=Rev code^ptr cpt^unit chg^units^total^tot unc  ; IBREV1(rev code) = revenue code edit category  ;  ; IBNOPR = flag that determines if there are revenue codes with  ; charges that do not have a procedure - no need to check  ; for billable MCR procedures if at least one RC is billable  ; 1 = there is at least one billable revenue code without a  ; procedure  ;  S REQMRA=$$REQMRA^IBEFUNC(IBIFN)  S (IBNOPR,IBI)=0  F  S IBI=$O(IBXDATA(IBI)) Q:'IBI  D  . I REQMRA D GYMODCHK(IBXDATA(IBI)) ; IB\*2\*377 GY modifier check  . S IBJ=$P(IBXDATA(IBI),U),IBECAT=""  . I 'IBNOPR D  .. I $P(IBXDATA(IBI),U,2)'="" S IBPR($P(IBXDATA(IBI),U,2))=IBI Q  .. S IBNOPR=1 K IBPR  . S:$D(IBREV1(IBJ)) IBECAT=$G(IBREV1(IBJ))  . I '$D(IBREV1(IBJ))!(IBECAT="") D  S IBREV1(IBJ)=IBECAT  . . ;  . . ; Accomodations (AC)  . . I (IBJ'<100&(IBJ'>219))!(IBJ=224) S IBECAT="AC" Q  . . ;  . . ; Ancillary Outpatient (AO)  . . I '$$INPAT^IBCEF(IBIFN,1) S IBECAT="AO" Q  . . ;  . . ; Ancillary Inpatient (AI)  . . S IBECAT="AI"  . ;  . S IBREV1(IBJ,+$O(IBREV1(IBJ,""),-1)+1)=IBXDATA(IBI)  . S IBREVTOT(IBECAT)=IBREVTOT(IBECAT)+$P(IBXDATA(IBI),U,6)  . I IBECAT="AC" S IBTUNIT=IBTUNIT+$P(IBXDATA(IBI),U,4)  ;  I $$NEEDMRA^IBEFUNC(IBIFN),$O(IBPR(""))'="" D  Q:IBQUIT  . ; Don't allow a bill containing only billable procedures for:  . ; Oxygen, labs, or influenza shots  . ; OR a bill with prosthetics on it  . ; to be sent to MEDICARE for an MRA  . D NONMCR(.IBPR,.IBLABS) ; Remove Oxygen, labs, influenza shots  . I $G(IBLABS) D WARN^IBCBB11("There are Lab procedures on this claim."),WARN^IBCBB11("Please verify that MEDICARE does not reimburse these labs at 100%") Q  . I $O(IBPR(""))="" D  .. S IBQUIT=$$IBER(.IBER,"098")  ;  ; covered days+non covered = units of accom rev codes  ; Check room and board  ; remove for IB\*2.0\*432  ;I IBTUNIT,IBTUNIT'=(IBCOV+IBNCOV) S IBQUIT=$$IBER(.IBER,114) Q:IBQUIT  ;  ; Non Covered Days  ; required when the type of bill is 11x,18x,21x or covered days=0  ; remove for IB\*2.0\*432  ; I IBNCOV="",(IBCTYP!(IBCOV=0)) S IBQUIT=$$IBER(.IBER,115) Q:IBQUIT  ;  ; if cc code=40 then non-covered days must be 1  ; remove for IB\*2.0\*432  ; I $D(IBCCARY1(40)),IBNCOV'=1 S IBQUIT=$$IBER(.IBER,116) Q:IBQUIT  ;  ; Patient Sex  ; must be "M" or "F"  D DEM^VADPT  I $P(VADM(5),U)'="M",$P(VADM(5),U)'="F" S IBQUIT=$$IBER(.IBER,124) Q:IBQUIT  ;  ; esg - 10/17/07 - patch 371  ; For Part A replacement MRA request claims, make sure  ; the Medicare ICN/DCN number is present and also text in FL-80.  I $$REQMRA^IBEFUNC(IBIFN),$F(".137.138.117.118.","."\_IBTOB\_".") D  Q:IBQUIT  . N IBZ,FL80TXT  . D F^IBCEF("N-CURR INS FORM LOC 64","IBZ",,IBIFN) ; see CI3-11  . I IBZ="" S IBQUIT=$$IBER(.IBER,205) Q:IBQUIT      ; missing ICN/DCN  . S FL80TXT=$P($G(^DGCR(399,IBIFN,"UF2")),U,3)  . I FL80TXT="" S IBQUIT=$$IBER(.IBER,206) Q:IBQUIT  ; missing FL80 text  . Q  ;  D ^IBCBB4  Q  ; IBER(IBER,ERRNO) ; Sets error list  ; NOTE: add code to check error list > 20 ... If so, display message and  ; quit so we don't get too many errors at once to handle  ; Print all if printing list  ;  I '$G(IBQUIT) D  . I ERRNO?1N.N S:$L(ERRNO)<3 ERRNO=$E("00",1,3-$L(ERRNO))\_ERRNO  . I $L(IBER,";")>19,'$G(IBPRT("PRT")) S IBER=IBER\_"IB999;",IBQUIT=1  . I $G(IBER)'[("IB"\_ERRNO\_";") S IBER=IBER\_"IB"\_ERRNO\_";"  Q IBQUIT  ; NONMCR(IBPR,IBLABS) ; Delete all oxygen and lab, flu shot CPT entries from IBPR  ; IBPR = array subscripted by CPT codes from bill  ; IBLABS = flag returned =1 if labs found on bill  N Z S IBLABS=0  ; Oxygen  F Z="A0422","A4575","A4616","A4619","A4620","A4621","E0455","E1353","E1355" K IBPR(Z)  F Z=77:1:85 S Z0="E13"\_Z K IBPR(Z0)  ; Labs  S Z="80000" F  S Z=$O(IBPR(Z)) Q:Z'?1"8"4N  S IBLABS=1  ; Flu shots  F Z="90724","G0008","90732","G0009","90657","90658","90659","90660" K IBPR(Z)  Q  ; MCRANUM(IBIFN) ; Determine MEDICARE A provider ID # from bedsection for  ; bill ien IBIFN  N IBX  ; PART A MRA (only) needed - determine if psych/non-psych claim  N IBX,IBI  S IBI=$P($G(^DGCR(399,IBIFN,"U")),U,11)  S IBX=$S($TR($P($G(^DGCR(399.1,+IBI,0)),U),"psych","PSYCH")'["PSYCH":670899,1:674499)  Q IBX  ; MCRACK(IBIFN,X,IBFLD) ; Check for MEDICARE A for bill IBIFN  ; Called from CLAIM STATUS MRA field (#24) xrefs in file 399  ; X = current value of field 399;24  ; IBFLD = 1 for primary ins co, 2 for secondary, 3 for tertiary  N IB  S IB=0  I +X,$$COBN^IBCEF(IBIFN)=IBFLD,$$WNRBILL^IBEFUNC(IBIFN,IBFLD),$$MRATYPE^IBEFUNC(IBIFN,"C")="A" S IB=1  Q IB  ; GYMODCHK(Z) ; GY modifier check procedure. IB\*2\*377 - 2/4/08  ; Z is the IBXDATA(IBI) service line EDI  N MODS  I IBER["IB123" Q     ; error already found  S MODS=$P(Z,U,9) ; list of modifiers separated by commas  I MODS'["GY" Q       ; GY modifier not here on this line item  I $P(Z,U,6) Q        ; non-covered charges exist on this line item  S IBQUIT=$$IBER(.IBER,123) GYMODX ;  Q  ; | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCCC2 | | | | | | | | |
| **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** | | | | | | | | | |
| IBCCC2 ;ALB/AAS - CANCEL AND CLONE A BILL - CONTINUED ;6/6/03 9:56am  ;;2.0;INTEGRATED BILLING;\*\*80,106,124,138,51,151,137,161,182,211,245,155,296,320,348,349,371,400,433,432,447,516\*\*;21-MAR-94;Build 123  ;;Per VA Directive 6402, this routine should not be modified.  ;  ;MAP TO DGCRCC2  ;  ;STEP 5 - get remainder of data to move and store in MCCR then x-ref  ;STEP 6 - go to screens, come out to IBB1 or something like that  ; STEP5 S IBIFN1=$P(^DGCR(399,IBIFN,0),"^",15) G END:$S(IBIFN1="":1,'$D(^DGCR(399,IBIFN1,0)):1,1:0)  ; NOTE: any new or changed data nodes may also need to be updated in IBNCPDP5  ;move pure data nodes  ; MRD;IB\*2.0\*516 - Added "In7" nodes.  F I="I1","I17","I2","I27","I3","I37","M1" I $D(^DGCR(399,IBIFN1,I)) S ^DGCR(399,IBIFN,I)=^DGCR(399,IBIFN1,I)  ;  ;move top level data node. ;Do not move 'TX' node EXCEPT piece 8 (added with IB\*2.0\*432)  ;F I="U","U1","U2","U3","UF2","UF3","UF31","C","M" I $D(^DGCR(399,IBIFN1,I)) S IBND(I)=^(I) D @I  ; add new data nodes introduced with IB\*2.0\*432  F I="TX","U","U1","U2","U3","U4","U5","U6","U7","U8","UF2","UF3","UF31","UF32","C","M" I $D(^DGCR(399,IBIFN1,I)) S IBND(I)=^(I) D @I  ;  ;move multiple level data  ;F I="CC","OC","OP","OT","RC","CP","CV","PRV" I $D(^DGCR(399,IBIFN1,I,0)) D @I  ; add new data nodes introduced with IB\*2.0\*447 BI  F I="CC","OC","OP","OT","RC","CP","CV","PRV","U9" I $D(^DGCR(399,IBIFN1,I,0)) D @I  ;  ; IB\*2.0\*432 ADDED IBSILENT flag so that this can be processed in background  D FTPRV^IBCEU5(IBIFN,$G(IBSILENT)) ; Ask change prov type if form type not the same  D COBCHG(IBIFN,,.IBCOB)  ;  D ^IBCCC3 ; copy table files (362.3)  ;  S I=$G(^DGCR(399,IBIFN1,0)) I $P(I,U,13)=7,$P(I,U,20)=1 D COPYB^IBCDC(IBIFN1,IBIFN) ; update auto bill files  D PRIOR(IBIFN) ; add new bill to previous bills in series, primary/secondary  ;  I +$G(IBCTCOPY) N IBAUTO S IBAUTO=1 D PROC^IBCU7A(IBIFN),BILL^IBCRBC(IBIFN),CPTMOD26^IBCU73(IBIFN) D RECALL^DILFD(399,IBIFN\_",",DUZ) G END  ; STEP6 N IBGOEND  ; need to kill CRD flag prior to entering billing screens in case a copy for corresponding claim is needed  K IBCNCRD  ; don't call IB bill edit screens if this is non-MRA background processing  I $G(IBSTSM)=1 G END  I '$G(IBCE("EDI"))!$G(IBCE("EDI","NEW")),'$G(IBCEAUTO) D IBSCEDT G END:$G(IBGOEND)  ;  ; END K DFN,IB,IBA,IBA2,IBAD,IBADD1,IBBNO,IBCAN,IBCCC,IBDA,IBDPT,IBDR,IBDT,IBI,IBI1,IBIDS,IBIFN,IBIFN1,IBND,IBQUIT,IBU,IBUN,IBARST,IBCOB,IBCNCOPY,IBCBCOPY,IBCNCRD,IBKEY  K IBV,IBV1,IBW,IBWW,IBYN,IBZZ,PRCASV,PRCAERCD,PRCAERR,PRCASVC,PRCAT,IBBT,IBCH,IBNDS,IBOA,IBREV,IBX,DGXRF1,VAEL,VAERR,IBAC,IBCCC,IBDD1,IBIN,DGREV,DGREV00,DGREVHDR,IBCHK  K IBBS,IBLS,DGPCM,IBIP,IBND0,IBNDU,IBO,IBPTF,IBST,IBUC,IBDD,D,%,%DT,DIC,VA,VADM,X,X1,X2,X3,X4,Y,I,J,K,DGRVRCAL,DDH,DGACTDT,DGAMNT,DGBR,DGBRN,DGBSI,DGBSLOS,IBA1,IBOD,IBINS,IBN,IBPROC,DGFUNC,DGIFN  Q  ;  ; IBSCEDT ; call the IB bill edit screens and validate the data  N IBV,IBPAR,IBAC,IBHV,IBH,IBCIREDT  ; if the user came from CBW->PC and this is a non-MRA claim w/a paper EOB, set force print flag IB\*2.0\*432  ; also, if the user came from CBW->PC and this is a non-MRA claim and the only EEOB we have has filing errors, set force print flag  I $G(IBMRANOT)=1,$$COBN^IBCEF(IBIFN)>1,$G(IBFROM)=2 D   .I $G(IBDA)="" D FORCEPRT^IBCAPP($G(IBIFN)) Q  .I $D(^IBM(361.1,IBDA,"ERR")) D FORCEPRT^IBCAPP($G(IBIFN)) Q  D RECALL^DILFD(399,IBIFN\_",",DUZ) ST1 S IBV=0 D ^IBCSCU,^IBCSC1 I $G(IBPOPOUT) S IBGOEND=1 G IBSCX  S IBAC=1  D ^IBCB1  I $G(IBCIREDT) G ST1 IBSCX ;  Q  ;  ; TX F J=8 I $P(IBND("TX"),"^",J)]"" S $P(^DGCR(399,IBIFN,"TX"),"^",J)=$P(IBND("TX"),"^",J)  Q U F J=3,4,6:1:17,20 I $P(IBND("U"),"^",J)]"" S $P(^DGCR(399,IBIFN,"U"),"^",J)=$P(IBND("U"),"^",J)  Q U1 F J=1:1:3,15 I $P(IBND("U1"),"^",J)]"" S $P(^DGCR(399,IBIFN,"U1"),"^",J)=$P(IBND("U1"),"^",J)  Q U2 F J=1:1:19 I $P(IBND("U2"),"^",J)]"" S $P(^DGCR(399,IBIFN,"U2"),"^",J)=$P(IBND("U2"),"^",J)  Q U3 F J=1:1:11 I $P(IBND("U3"),"^",J)]"" S $P(^DGCR(399,IBIFN,"U3"),"^",J)=$P(IBND("U3"),"^",J)  Q UF2 F J=1,3 I $P(IBND("UF2"),"^",J)]"" S $P(^DGCR(399,IBIFN,"UF2"),"^",J)=$P(IBND("UF2"),"^",J)  Q UF3 F J=4:1:6 I $P(IBND("UF3"),"^",J)]"" S $P(^DGCR(399,IBIFN,"UF3"),"^",J)=$P(IBND("UF3"),"^",J)  Q U4 F J=1:1:14 I $P(IBND("U4"),"^",J)]"" S $P(^DGCR(399,IBIFN,"U4"),"^",J)=$P(IBND("U4"),"^",J)  Q U5 F J=1:1:6 I $P(IBND("U5"),"^",J)]"" S $P(^DGCR(399,IBIFN,"U5"),"^",J)=$P(IBND("U5"),"^",J)  Q U6 F J=1:1:6 I $P(IBND("U6"),"^",J)]"" S $P(^DGCR(399,IBIFN,"U6"),"^",J)=$P(IBND("U6"),"^",J)  Q U7 F J=1:1:5 I $P(IBND("U7"),"^",J)]"" S $P(^DGCR(399,IBIFN,"U7"),"^",J)=$P(IBND("U7"),"^",J)  Q U8 F J=1:1:3 I $P(IBND("U8"),"^",J)]"" S $P(^DGCR(399,IBIFN,"U8"),"^",J)=$P(IBND("U8"),"^",J)  Q UF31 F J=3 I $P(IBND("UF31"),"^",J)]"" S $P(^DGCR(399,IBIFN,"UF31"),"^",J)=$P(IBND("UF31"),"^",J)  Q UF32 F J=1:1:3 I $P(IBND("UF32"),"^",J)]"" S $P(^DGCR(399,IBIFN,"UF32"),"^",J)=$P(IBND("UF32"),"^",J)  Q C F J=10 I $P(IBND("C"),"^",J)]"" S $P(^DGCR(399,IBIFN,"C"),"^",J)=$P(IBND("C"),"^",J)  I '$D(^DGCR(399,IBIFN1,"CP")) D CP1  Q M F J=1:1:9,11:1:14 I $P(IBND("M"),"^",J)]"" S $P(^DGCR(399,IBIFN,"M"),"^",J)=$P(IBND("M"),"^",J)  Q CC S ^DGCR(399,IBIFN,I,0)=^DGCR(399,IBIFN1,I,0)  S IBDD=399.04 F J=0:0 S J=$O(^DGCR(399,IBIFN1,I,J)) Q:'J  I $D(^(J,0)) S ^DGCR(399,IBIFN,I,J,0)=^DGCR(399,IBIFN1,I,J,0),X=$P(^(0),"^") OP S ^DGCR(399,IBIFN,I,0)=^DGCR(399,IBIFN1,I,0)  S IBDD=399.043 F J=0:0 S J=$O(^DGCR(399,IBIFN1,I,J)) Q:'J  I $D(^(J,0)) S ^DGCR(399,IBIFN,I,J,0)=^DGCR(399,IBIFN1,I,J,0),X=$P(^(0),"^")  Q OC S ^DGCR(399,IBIFN,I,0)=^DGCR(399,IBIFN1,I,0)  S IBDD=399.041 F J=0:0 S J=$O(^DGCR(399,IBIFN1,I,J)) Q:'J  I $D(^(J,0)) S ^DGCR(399,IBIFN,I,J,0)=^DGCR(399,IBIFN1,I,J,0),X=$P(^(0),"^")  Q OT S ^DGCR(399,IBIFN,I,0)=^DGCR(399,IBIFN1,I,0)  S IBDD=399.048 F J=0:0 S J=$O(^DGCR(399,IBIFN1,I,J)) Q:'J  I $D(^(J,0)) S ^DGCR(399,IBIFN,I,J,0)=^DGCR(399,IBIFN1,I,J,0),X=$P(^(0),"^")  Q CV ; Don't copy value codes from inpatient inst to inpatient prof bills  I $$FT^IBCEF(IBIFN1)'=2,$$FT^IBCEF(IBIFN)=2 Q  S ^DGCR(399,IBIFN,I,0)=^DGCR(399,IBIFN1,I,0)  S IBDD=399.047 F J=0:0 S J=$O(^DGCR(399,IBIFN1,I,J)) Q:'J  I $D(^(J,0)) S ^DGCR(399,IBIFN,I,J,0)=^DGCR(399,IBIFN1,I,J,0),X=$P(^(0),"^")  Q RC S ^DGCR(399,IBIFN,I,0)=^DGCR(399,IBIFN1,I,0)  S IBDD=399.042 F J=0:0 S J=$O(^DGCR(399,IBIFN1,I,J)) Q:'J  I $D(^(J,0)) S IBND("RC")=^(0) F K=1:1:16 S $P(^DGCR(399,IBIFN,I,J,0),"^",K)=$P(IBND("RC"),"^",K),X=$P(IBND("RC"),"^",K)  Q CP S ^DGCR(399,IBIFN,I,0)=^DGCR(399,IBIFN1,I,0)  I +$G(IBNOCPT) Q  S IBDD=399.0304 F J=0:0 S J=$O(^DGCR(399,IBIFN1,I,J)) Q:'J  I $D(^(J,0)) S IBND("CP")=^(0),IBND("CP1")=$G(^(1)),IBND("CP-AUX")=$G(^("AUX")) D  . F K=1:1:7,9:1:14,16:1:22 S $P(^DGCR(399,IBIFN,I,J,0),"^",K)=$P(IBND("CP"),"^",K)  . ; IB\*2.0\*432 add new 1 node  . ; MRD;IB\*2.0\*516 - Added pieces 7 & 8 (NDC, Units) to 1-node.  . F K=1:1:8 S $P(^DGCR(399,IBIFN,I,J,1),"^",K)=$P(IBND("CP1"),"^",K)  . ; esg - 11/2/06 - IB\*2\*348 - 50.09 field was added - AUX piece [9]  . I IBND("CP-AUX")'="" F K=1:1:9 S $P(^DGCR(399,IBIFN,I,J,"AUX"),"^",K)=$P(IBND("CP-AUX"),"^",K)  . ; IB\*2.0\*432 add new LNPRV multiple  . I $D(^DGCR(399,IBIFN1,I,J,"LNPRV",0)) S ^DGCR(399,IBIFN,I,J,"LNPRV",0)=^DGCR(399,IBIFN1,I,J,"LNPRV",0) D  .. S K=0 F  S K=$O(^DGCR(399,IBIFN1,I,J,"LNPRV",K)) Q:'K  D  ... S ^DGCR(399,IBIFN,I,J,"LNPRV",K,0)=^DGCR(399,IBIFN1,I,J,"LNPRV",K,0)  . I $D(^DGCR(399,IBIFN1,I,J,"MOD",0)) S ^DGCR(399,IBIFN,I,J,"MOD",0)=^DGCR(399,IBIFN1,I,J,"MOD",0) D  .. S K=0 F  S K=$O(^DGCR(399,IBIFN1,I,J,"MOD",K)) Q:'K  D  ... I $G(IBNOTC),$P($$MOD^ICPTMOD(+$P($G(^DGCR(399,IBIFN1,I,J,"MOD",K,0)),U,2),"I"),U,2)="TC" Q  ; Don't copy TC modifier from inst to prof bill  ... S ^DGCR(399,IBIFN,I,J,"MOD",K,0)=^DGCR(399,IBIFN1,I,J,"MOD",K,0) CP1 S IBCOD=$P($G(^DGCR(399,IBIFN,0)),"^",9) Q:IBCOD=""!('$D(^DGCR(399,IBIFN1,"C")))  I IBCOD=9 F DGI=4,5,6 I $P(^DGCR(399,IBIFN1,"C"),"^",DGI) S X=$P(^("C"),"^",DGI)\_";ICD0(",DGPROCDT=$P(^("C"),"^",DGI+7) D FILE  I IBCOD=4 F DGI=1,2,3 I $P(^DGCR(399,IBIFN1,"C"),"^",DGI) S X=$P(^("C"),"^",DGI)\_";ICPT(",DGPROCDT=$P(^("C"),"^",DGI+10) D FILE  I IBCOD=5 F DGI=7,8,9 I $P(^DGCR(399,IBIFN1,"C"),"^",DGI) S X=$P(^("C"),"^",DGI)\_";ICPT(",DGPROCDT=$P(^("C"),"^",DGI+4) D FILE  Q  ; PRV ; Copy providers for cloned claim  N Z,Z0,CNT  S Z=$P($G(^DGCR(399,IBIFN,0)),U,19),Z0=$P($G(^DGCR(399,IBIFN1,0)),U,19),CNT=0  S IBDD=399.0222 F J=0:0 S J=$O(^DGCR(399,IBIFN1,I,J)) Q:'J  I $D(^(J,0)) D  . I $$GETNPI^IBCEF73A($P(^DGCR(399,IBIFN1,I,J,0),U,2))="" Q  ;Don't file provider if no NPI - IB\*2\*516  . S CNT=CNT+1,^DGCR(399,IBIFN,I,CNT,0)=^DGCR(399,IBIFN1,I,J,0),X=$P(^(0),"^")  . I Z'=Z0,$S(X=3:Z0=3,X=4:Z0=2,1:0) S $P(^DGCR(399,IBIFN,I,CNT,0),U)=(Z0+1)  I CNT S ^DGCR(399,IBIFN,I,0)=^DGCR(399,IBIFN1,I,0),$P(^DGCR(399,IBIFN,I,0),U,3)=CNT,$P(^DGCR(399,IBIFN,I,0),U,4)=CNT  Q  ; U9 ; Added for new data elements in IB\*2.0\*447 BI  M ^DGCR(399,IBIFN,I)=^DGCR(399,IBIFN1,I)  Q  ; COB S J=0 F  S J=$O(IBCOB(I,J)) Q:'J  S $P(^DGCR(399,IBIFN,I),U,J)=IBCOB(I,J)  Q  ; FILE N DIC,DIE,DR,DA,X,Y,DLAYGO,DD,DO  I '$D(^DGCR(399,IBIFN,"CP",0)) S DIC("P")=$$GETSPEC^IBEFUNC(399,304)  S DIC(0)="L",DLAYGO=399,DA(1)=IBIFN,DIC="^DGCR(399,"\_DA(1)\_",""CP""," Q:X=""  D FILE^DICN K DO,DD Q:+Y<1 S DA=+Y  S DIE="^DGCR(399,"\_DA(1)\_",""CP"",",DR="1///"\_DGPROCDT D ^DIE  K DGPROCDT  Q  ; INDEX ;index entire file (set logic)  N IBMAED D SAVERC(IBIFN,.IBMAED) ; IB\*2.0\*447 BI - Save the value of piece 16 of each RC node before re-indexing.  S DIK="^DGCR(399,",DA=IBIFN D IX1^DIK K DA,DIK  D RESTRC(IBIFN,.IBMAED) ; IB\*2.0\*447 BI - Restore the value of piece 16 of each RC node before re-indexing.  Q  ; PRIOR(IBIFN) ; set Secondary/Tertiary Bill #s on prior bills, if the bill is cancelled remove it from prior bills  N IBSEQ,IBSEQN,IBM1,I,IBIFN1  S IBSEQ=$$COB^IBCEF(IBIFN)  S IBSEQN=$S(IBSEQ="S":6,IBSEQ="T":7,1:"") Q:'IBSEQN  ;  S IBM1=$G(^DGCR(399,IBIFN,"M1")) I +$P(^DGCR(399,IBIFN,0),U,13)=7 S IBIFN=""  F I=5,6 I I<IBSEQN  S IBIFN1=+$P(IBM1,U,I) I +IBIFN1,$D(^DGCR(399,+IBIFN1,0)) S $P(^DGCR(399,IBIFN1,"M1"),U,IBSEQN)=IBIFN  Q  ; COBCHG(IBIFN,IBINS,IBCOB) ; Make changes for a new COB payer for bill  ; IBIFN = ien of bill in file 399  ; IBINS = ien of bill's current insurance (optional)  ; IBCOB = array subscripted by node,piece of COB data field change  ;  N I,IBFRMTYP,IBTAXLST  ; Subtract the Prior Payments from the bill's Offset (these are re-added by triggers)  F I=4,5,6 S $P(^DGCR(399,IBIFN,"U1"),U,2)=$P($G(^DGCR(399,IBIFN,"U1")),U,2)-$P($G(^DGCR(399,IBIFN,"U2")),U,I)  ;  I $G(IBINS),$$MCRWNR^IBEFUNC(IBINS) D  . ;MCRWNR is current insurance ... move payer only  . N IBCOBN,IBX  . S IBCOBN=$$COBN^IBCEF(IBIFN)  . S IBCOB(0,21)=$P("S^T^",U,IBCOBN)  . S IBCOB("M1",IBCOBN+4)=IBIFN  . S IBCOB("TX",1)="",IBCOB("TX",2)=""  . S IBX=$$REQMRA^IBEFUNC(IBIFN)  . I IBX=0 S IBCOB("TX",5)=0 ; MRA not needed  . I IBX["R" S IBCOB("TX",5)="A"                     ; MRA skipped  . I IBX=1,$$CHK^IBCEMU1(IBIFN) S IBCOB("TX",5)="C"  ; MRA on file  . I $G(IBPRCOB) S IBCOB("TX",5)="C"                 ; MRA being proc'd  . D PRIOR(IBIFN)  . Q  ;  ;reset fields for next Sequence Payer  F I=0,"M1","U2","TX" I $D(IBCOB(I)) D COB  ;  ; IB\*2.0\*211  ; save off Form Type  S IBFRMTYP=$P($G(^DGCR(399,IBIFN,0)),U,19)  ; Save off Taxonomies for providers.  S I=0 F  S I=$O(^DGCR(399,IBIFN,"PRV",I)) Q:'I  S IBTAXLST(I)=$P($G(^DGCR(399,IBIFN,"PRV",I,0)),U,15)  ;  ; fire xrefs set logic  D INDEX  ;  ; Restore Form Type if changed, but don't restore Form Type if  ; creating CMS-1500 claim from CTCOPY1^IBCCCB  I $G(IBCTCOPY)'=1,IBFRMTYP'=$P($G(^DGCR(399,IBIFN,0)),U,19) N DA,DIE,DR S DA=IBIFN,DIE="^DGCR(399,",DR=".19////"\_IBFRMTYP D ^DIE  ;  ; Restore Claim MRA Status field since triggers in fields 101 & 102  ; will overwrite the correct value when processing the MRA/EOB.  ; If we're processing the MRA/EOB, then a valid MRA has been received.  I $G(IBPRCOB) N DA,DIE,DR S DA=IBIFN,DIE="^DGCR(399,",DR="24////C" D ^DIE  ;  ; Only if cloning, then restore Taxonomies in fields 243 and 244 and 252.  I '$G(IBINS),'$G(IBPRCOB) D  . S I=$P($G(IBND("U3")),U,2)  . I I'=$P($G(^DGCR(399,IBIFN,"U3")),U,2) D  .. N DA,DIE,DR S DA=IBIFN,DIE="^DGCR(399,",DR="243////"\_$S(I'="":I,1:"@") D ^DIE  . ;  . S I=$P($G(IBND("U3")),U,3)  . I I'=$P($G(^DGCR(399,IBIFN,"U3")),U,3) D  .. N DA,DIE,DR S DA=IBIFN,DIE="^DGCR(399,",DR="244////"\_$S(I'="":I,1:"@") D ^DIE  . ;  . S I=$P($G(IBND("U3")),U,11)  . I I'=$P($G(^DGCR(399,IBIFN,"U3")),U,11) D  .. N DA,DIE,DR S DA=IBIFN,DIE="^DGCR(399,",DR="252////"\_$S(I'="":I,1:"@") D ^DIE  . Q  ;  ; Restore Taxonomies in field .15 in sub-file 399.0222.  S IBTAXLST=0 F  S IBTAXLST=$O(IBTAXLST(IBTAXLST)) Q:'IBTAXLST  D  . S I=IBTAXLST(IBTAXLST)  . I I=$P($G(^DGCR(399,IBIFN,"PRV",IBTAXLST,0)),U,15) Q  ; No change  . N DA,DIE,DR  . S DA(1)=IBIFN,DA=IBTAXLST  . S DIE="^DGCR(399,"\_DA(1)\_",""PRV"",",DR=".15////"\_$S(I'="":I,1:"@")  . D ^DIE  . Q  ;  K IBCOB("TX")  Q  ; SAVERC(IBIFN,IBMAED) ; IB\*2.0\*447 BI - Save the value of piece 16 of each RC node before re-indexing.  Q:$G(IBCTCOPY)=1 Q:$G(IBCTCOPY)=2  N IBCNT S IBCNT=0  Q:'$G(IBIFN) Q:'$D(^DGCR(399,IBIFN,"RC"))  F  S IBCNT=$O(^DGCR(399,IBIFN,"RC",IBCNT)) Q:+IBCNT=0 D  . S IBMAED(IBCNT)=$P($G(^DGCR(399,IBIFN,"RC",IBCNT,0)),U,16)  Q  ; RESTRC(IBIFN,IBMAED) ; IB\*2.0\*447 BI - Restore the value of piece 16 of each RC node after re-indexing.  Q:$G(IBCTCOPY)=1 Q:$G(IBCTCOPY)=2  N IBCNT S IBCNT=0  Q:'$G(IBIFN) Q:'$D(^DGCR(399,IBIFN,"RC"))  F  S IBCNT=$O(IBMAED(IBCNT)) Q:+IBCNT=0 D  . S $P(^DGCR(399,IBIFN,"RC",IBCNT,0),U,16)=IBMAED(IBCNT)  Q | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCCC2 ;ALB/AAS - CANCEL AND CLONE A BILL - CONTINUED ;6/6/03 9:56am  ;;2.0;INTEGRATED BILLING;\*\*80,106,124,138,51,151,137,161,182,211,245,155,296,320,348,349,371,400,433,432,447,516,**592**\*\*;21-MAR-94;Build 123  ;;Per VA Directive 6402, this routine should not be modified.  ;  ;MAP TO DGCRCC2  ;  ;STEP 5 - get remainder of data to move and store in MCCR then x-ref  ;STEP 6 - go to screens, come out to IBB1 or something like that  ; STEP5 S IBIFN1=$P(^DGCR(399,IBIFN,0),"^",15) G END:$S(IBIFN1="":1,'$D(^DGCR(399,IBIFN1,0)):1,1:0)  ; NOTE: any new or changed data nodes may also need to be updated in IBNCPDP5  ;move pure data nodes  ; MRD;IB\*2.0\*516 - Added "In7" nodes.  F I="I1","I17","I2","I27","I3","I37","M1" I $D(^DGCR(399,IBIFN1,I)) S ^DGCR(399,IBIFN,I)=^DGCR(399,IBIFN1,I)  ;  ;move top level data node. ;Do not move 'TX' node EXCEPT piece 8 (added with IB\*2.0\*432)  ;F I="U","U1","U2","U3","UF2","UF3","UF31","C","M" I $D(^DGCR(399,IBIFN1,I)) S IBND(I)=^(I) D @I  ; add new data nodes introduced with IB\*2.0\*432  F I="TX","U","U1","U2","U3","U4","U5","U6","U7","U8","UF2","UF3","UF31","UF32","C","M" I $D(^DGCR(399,IBIFN1,I)) S IBND(I)=^(I) D @I  ;  ;move multiple level data  ;F I="CC","OC","OP","OT","RC","CP","CV","PRV" I $D(^DGCR(399,IBIFN1,I,0)) D @I  ; add new data nodes introduced with IB\*2.0\*447 BI  F I="CC","OC","OP","OT","RC","CP","CV","PRV","U9" I $D(^DGCR(399,IBIFN1,I,0)) D @I  ;  **;JWS;IB\*2.0\*592;add new Dental Claim fields  I $D(^DGCR(399,IBIFN1,"DEN")) S ^DGCR(399,IBIFN,"DEN")=^DGCR(399,IBIFN1,"DEN")  I $D(^DGCR(399,IBIFN1,"DEN1",0)) S ^DGCR(399,IBIFN,"DEN1",0)=^DGCR(399,IBIFN1,"DEN1",0) D  . S K=0 F  S K=$O(^DGCR(399,IBIFN1,"DEN1",K)) Q:'K  S ^DGCR(399,IBIFN,"DEN1",K,0)=^DGCR(399,IBIFN1,"DEN1",K,0)  I $D(^DGCR(399,IBIFN1,"DEN2")) S ^DGCR(399,IBIFN,"DEN2")=^DGCR(399,IBIFN1,"DEN2")**  ;  ; IB\*2.0\*432 ADDED IBSILENT flag so that this can be processed in background  D FTPRV^IBCEU5(IBIFN,$G(IBSILENT)) ; Ask change prov type if form type not the same  D COBCHG(IBIFN,,.IBCOB)  ;  D ^IBCCC3 ; copy table files (362.3)  ;  S I=$G(^DGCR(399,IBIFN1,0)) I $P(I,U,13)=7,$P(I,U,20)=1 D COPYB^IBCDC(IBIFN1,IBIFN) ; update auto bill files  D PRIOR(IBIFN) ; add new bill to previous bills in series, primary/secondary  ;  I +$G(IBCTCOPY) N IBAUTO S IBAUTO=1 D PROC^IBCU7A(IBIFN),BILL^IBCRBC(IBIFN),CPTMOD26^IBCU73(IBIFN) D RECALL^DILFD(399,IBIFN\_",",DUZ) G END  ; STEP6 N IBGOEND  ; need to kill CRD flag prior to entering billing screens in case a copy for corresponding claim is needed  K IBCNCRD  ; don't call IB bill edit screens if this is non-MRA background processing  I $G(IBSTSM)=1 G END  I '$G(IBCE("EDI"))!$G(IBCE("EDI","NEW")),'$G(IBCEAUTO) D IBSCEDT G END:$G(IBGOEND)  ;  ; END K DFN,IB,IBA,IBA2,IBAD,IBADD1,IBBNO,IBCAN,IBCCC,IBDA,IBDPT,IBDR,IBDT,IBI,IBI1,IBIDS,IBIFN,IBIFN1,IBND,IBQUIT,IBU,IBUN,IBARST,IBCOB,IBCNCOPY,IBCBCOPY,IBCNCRD,IBKEY  K IBV,IBV1,IBW,IBWW,IBYN,IBZZ,PRCASV,PRCAERCD,PRCAERR,PRCASVC,PRCAT,IBBT,IBCH,IBNDS,IBOA,IBREV,IBX,DGXRF1,VAEL,VAERR,IBAC,IBCCC,IBDD1,IBIN,DGREV,DGREV00,DGREVHDR,IBCHK  K IBBS,IBLS,DGPCM,IBIP,IBND0,IBNDU,IBO,IBPTF,IBST,IBUC,IBDD,D,%,%DT,DIC,VA,VADM,X,X1,X2,X3,X4,Y,I,J,K,DGRVRCAL,DDH,DGACTDT,DGAMNT,DGBR,DGBRN,DGBSI,DGBSLOS,IBA1,IBOD,IBINS,IBN,IBPROC,DGFUNC,DGIFN  Q  ;  ; IBSCEDT ; call the IB bill edit screens and validate the data  N IBV,IBPAR,IBAC,IBHV,IBH,IBCIREDT  ; if the user came from CBW->PC and this is a non-MRA claim w/a paper EOB, set force print flag IB\*2.0\*432  ; also, if the user came from CBW->PC and this is a non-MRA claim and the only EEOB we have has filing errors, set force print flag  I $G(IBMRANOT)=1,$$COBN^IBCEF(IBIFN)>1,$G(IBFROM)=2 D   .I $G(IBDA)="" D FORCEPRT^IBCAPP($G(IBIFN)) Q  .I $D(^IBM(361.1,IBDA,"ERR")) D FORCEPRT^IBCAPP($G(IBIFN)) Q  D RECALL^DILFD(399,IBIFN\_",",DUZ) ST1 S IBV=0 D ^IBCSCU,^IBCSC1 I $G(IBPOPOUT) S IBGOEND=1 G IBSCX  S IBAC=1  D ^IBCB1  I $G(IBCIREDT) G ST1 IBSCX ;  Q  ;  ; TX F J=8 I $P(IBND("TX"),"^",J)]"" S $P(^DGCR(399,IBIFN,"TX"),"^",J)=$P(IBND("TX"),"^",J)  Q U F J=3,4,6:1:17,20 I $P(IBND("U"),"^",J)]"" S $P(^DGCR(399,IBIFN,"U"),"^",J)=$P(IBND("U"),"^",J)  Q U1 F J=1:1:3,15 I $P(IBND("U1"),"^",J)]"" S $P(^DGCR(399,IBIFN,"U1"),"^",J)=$P(IBND("U1"),"^",J)  Q U2 F J=1:1:19 I $P(IBND("U2"),"^",J)]"" S $P(^DGCR(399,IBIFN,"U2"),"^",J)=$P(IBND("U2"),"^",J)  Q U3 F J=1:1:11 I $P(IBND("U3"),"^",J)]"" S $P(^DGCR(399,IBIFN,"U3"),"^",J)=$P(IBND("U3"),"^",J)  Q UF2 F J=1,3 I $P(IBND("UF2"),"^",J)]"" S $P(^DGCR(399,IBIFN,"UF2"),"^",J)=$P(IBND("UF2"),"^",J)  Q UF3 F J=4:1:6 I $P(IBND("UF3"),"^",J)]"" S $P(^DGCR(399,IBIFN,"UF3"),"^",J)=$P(IBND("UF3"),"^",J)  Q U4 F J=1:1:14 I $P(IBND("U4"),"^",J)]"" S $P(^DGCR(399,IBIFN,"U4"),"^",J)=$P(IBND("U4"),"^",J)  Q U5 F J=1:1:6 I $P(IBND("U5"),"^",J)]"" S $P(^DGCR(399,IBIFN,"U5"),"^",J)=$P(IBND("U5"),"^",J)  Q U6 F J=1:1:6 I $P(IBND("U6"),"^",J)]"" S $P(^DGCR(399,IBIFN,"U6"),"^",J)=$P(IBND("U6"),"^",J)  Q U7 F J=1:1:5 I $P(IBND("U7"),"^",J)]"" S $P(^DGCR(399,IBIFN,"U7"),"^",J)=$P(IBND("U7"),"^",J)  Q U8 F J=1:1:3 I $P(IBND("U8"),"^",J)]"" S $P(^DGCR(399,IBIFN,"U8"),"^",J)=$P(IBND("U8"),"^",J)  Q UF31 F J=3 I $P(IBND("UF31"),"^",J)]"" S $P(^DGCR(399,IBIFN,"UF31"),"^",J)=$P(IBND("UF31"),"^",J)  Q UF32 F J=1:1:3 I $P(IBND("UF32"),"^",J)]"" S $P(^DGCR(399,IBIFN,"UF32"),"^",J)=$P(IBND("UF32"),"^",J)  Q C F J=10 I $P(IBND("C"),"^",J)]"" S $P(^DGCR(399,IBIFN,"C"),"^",J)=$P(IBND("C"),"^",J)  I '$D(^DGCR(399,IBIFN1,"CP")) D CP1  Q M F J=1:1:9,11:1:14 I $P(IBND("M"),"^",J)]"" S $P(^DGCR(399,IBIFN,"M"),"^",J)=$P(IBND("M"),"^",J)  Q CC S ^DGCR(399,IBIFN,I,0)=^DGCR(399,IBIFN1,I,0)  S IBDD=399.04 F J=0:0 S J=$O(^DGCR(399,IBIFN1,I,J)) Q:'J  I $D(^(J,0)) S ^DGCR(399,IBIFN,I,J,0)=^DGCR(399,IBIFN1,I,J,0),X=$P(^(0),"^") OP S ^DGCR(399,IBIFN,I,0)=^DGCR(399,IBIFN1,I,0)  S IBDD=399.043 F J=0:0 S J=$O(^DGCR(399,IBIFN1,I,J)) Q:'J  I $D(^(J,0)) S ^DGCR(399,IBIFN,I,J,0)=^DGCR(399,IBIFN1,I,J,0),X=$P(^(0),"^")  Q OC S ^DGCR(399,IBIFN,I,0)=^DGCR(399,IBIFN1,I,0)  S IBDD=399.041 F J=0:0 S J=$O(^DGCR(399,IBIFN1,I,J)) Q:'J  I $D(^(J,0)) S ^DGCR(399,IBIFN,I,J,0)=^DGCR(399,IBIFN1,I,J,0),X=$P(^(0),"^")  Q OT S ^DGCR(399,IBIFN,I,0)=^DGCR(399,IBIFN1,I,0)  S IBDD=399.048 F J=0:0 S J=$O(^DGCR(399,IBIFN1,I,J)) Q:'J  I $D(^(J,0)) S ^DGCR(399,IBIFN,I,J,0)=^DGCR(399,IBIFN1,I,J,0),X=$P(^(0),"^")  Q CV ; Don't copy value codes from inpatient inst to inpatient prof bills  I $$FT^IBCEF(IBIFN1)'=2,$$FT^IBCEF(IBIFN)=2 Q  S ^DGCR(399,IBIFN,I,0)=^DGCR(399,IBIFN1,I,0)  S IBDD=399.047 F J=0:0 S J=$O(^DGCR(399,IBIFN1,I,J)) Q:'J  I $D(^(J,0)) S ^DGCR(399,IBIFN,I,J,0)=^DGCR(399,IBIFN1,I,J,0),X=$P(^(0),"^")  Q RC S ^DGCR(399,IBIFN,I,0)=^DGCR(399,IBIFN1,I,0)  S IBDD=399.042 F J=0:0 S J=$O(^DGCR(399,IBIFN1,I,J)) Q:'J  I $D(^(J,0)) S IBND("RC")=^(0) F K=1:1:16 S $P(^DGCR(399,IBIFN,I,J,0),"^",K)=$P(IBND("RC"),"^",K),X=$P(IBND("RC"),"^",K)  Q CP S ^DGCR(399,IBIFN,I,0)=^DGCR(399,IBIFN1,I,0)  I +$G(IBNOCPT) Q  S IBDD=399.0304 F J=0:0 S J=$O(^DGCR(399,IBIFN1,I,J)) Q:'J  I $D(^(J,0)) S IBND("CP")=^(0),IBND("CP1")=$G(^(1)),IBND("CP-AUX")=$G(^("AUX")) D  . F K=1:1:7,9:1:14,16:1:22 S $P(^DGCR(399,IBIFN,I,J,0),"^",K)=$P(IBND("CP"),"^",K)  . ; IB\*2.0\*432 add new 1 node  . ; MRD;IB\*2.0\*516 - Added pieces 7 & 8 (NDC, Units) to 1-node.  . F K=1:1:8 S $P(^DGCR(399,IBIFN,I,J,1),"^",K)=$P(IBND("CP1"),"^",K)  . ; esg - 11/2/06 - IB\*2\*348 - 50.09 field was added - AUX piece [9]  . I IBND("CP-AUX")'="" F K=1:1:9 S $P(^DGCR(399,IBIFN,I,J,"AUX"),"^",K)=$P(IBND("CP-AUX"),"^",K)  . ; IB\*2.0\*432 add new LNPRV multiple  . I $D(^DGCR(399,IBIFN1,I,J,"LNPRV",0)) S ^DGCR(399,IBIFN,I,J,"LNPRV",0)=^DGCR(399,IBIFN1,I,J,"LNPRV",0) D  .. S K=0 F  S K=$O(^DGCR(399,IBIFN1,I,J,"LNPRV",K)) Q:'K  D  ... S ^DGCR(399,IBIFN,I,J,"LNPRV",K,0)=^DGCR(399,IBIFN1,I,J,"LNPRV",K,0)  . I $D(^DGCR(399,IBIFN1,I,J,"MOD",0)) S ^DGCR(399,IBIFN,I,J,"MOD",0)=^DGCR(399,IBIFN1,I,J,"MOD",0) D  .. S K=0 F  S K=$O(^DGCR(399,IBIFN1,I,J,"MOD",K)) Q:'K  D  ... I $G(IBNOTC),$P($$MOD^ICPTMOD(+$P($G(^DGCR(399,IBIFN1,I,J,"MOD",K,0)),U,2),"I"),U,2)="TC" Q  ; Don't copy TC modifier from inst to prof bill  ... S ^DGCR(399,IBIFN,I,J,"MOD",K,0)=^DGCR(399,IBIFN1,I,J,"MOD",K,0)  **. ;JWS;IB\*2.0\*592;add new Dental claim form fields  . I $D(^DGCR(399,IBIFN1,I,J,"DEN")) S ^DGCR(399,IBIFN,I,J,"DEN")=^DGCR(399,IBIFN1,I,J,"DEN")  . I $D(^DGCR(399,IBIFN1,I,J,"DEN1",0)) S ^DGCR(399,IBIFN,I,J,"DEN1",0)=^DGCR(399,IBIFN1,I,J,"DEN1",0) D  .. S K=0 F  S K=$O(^DGCR(399,IBIFN1,I,J,"DEN1",K)) Q:'K  D  ... S ^DGCR(399,IBIFN,I,J,"DEN1",K,0)=^DGCR(399,IBIFN1,I,J,"DEN1",K,0)** CP1 S IBCOD=$P($G(^DGCR(399,IBIFN,0)),"^",9) Q:IBCOD=""!('$D(^DGCR(399,IBIFN1,"C")))  I IBCOD=9 F DGI=4,5,6 I $P(^DGCR(399,IBIFN1,"C"),"^",DGI) S X=$P(^("C"),"^",DGI)\_";ICD0(",DGPROCDT=$P(^("C"),"^",DGI+7) D FILE  I IBCOD=4 F DGI=1,2,3 I $P(^DGCR(399,IBIFN1,"C"),"^",DGI) S X=$P(^("C"),"^",DGI)\_";ICPT(",DGPROCDT=$P(^("C"),"^",DGI+10) D FILE  I IBCOD=5 F DGI=7,8,9 I $P(^DGCR(399,IBIFN1,"C"),"^",DGI) S X=$P(^("C"),"^",DGI)\_";ICPT(",DGPROCDT=$P(^("C"),"^",DGI+4) D FILE  Q  ; PRV ; Copy providers for cloned claim  N Z,Z0,CNT  S Z=$P($G(^DGCR(399,IBIFN,0)),U,19),Z0=$P($G(^DGCR(399,IBIFN1,0)),U,19),CNT=0  S IBDD=399.0222 F J=0:0 S J=$O(^DGCR(399,IBIFN1,I,J)) Q:'J  I $D(^(J,0)) D  . I $$GETNPI^IBCEF73A($P(^DGCR(399,IBIFN1,I,J,0),U,2))="" Q  ;Don't file provider if no NPI - IB\*2\*516  . S CNT=CNT+1,^DGCR(399,IBIFN,I,CNT,0)=^DGCR(399,IBIFN1,I,J,0),X=$P(^(0),"^")  . I Z'=Z0,$S(X=3:Z0=3,X=4:Z0=2,1:0) S $P(^DGCR(399,IBIFN,I,CNT,0),U)=(Z0+1)  I CNT S ^DGCR(399,IBIFN,I,0)=^DGCR(399,IBIFN1,I,0),$P(^DGCR(399,IBIFN,I,0),U,3)=CNT,$P(^DGCR(399,IBIFN,I,0),U,4)=CNT  Q  ; U9 ; Added for new data elements in IB\*2.0\*447 BI  M ^DGCR(399,IBIFN,I)=^DGCR(399,IBIFN1,I)  Q  ; COB S J=0 F  S J=$O(IBCOB(I,J)) Q:'J  S $P(^DGCR(399,IBIFN,I),U,J)=IBCOB(I,J)  Q  ; FILE N DIC,DIE,DR,DA,X,Y,DLAYGO,DD,DO  I '$D(^DGCR(399,IBIFN,"CP",0)) S DIC("P")=$$GETSPEC^IBEFUNC(399,304)  S DIC(0)="L",DLAYGO=399,DA(1)=IBIFN,DIC="^DGCR(399,"\_DA(1)\_",""CP""," Q:X=""  D FILE^DICN K DO,DD Q:+Y<1 S DA=+Y  S DIE="^DGCR(399,"\_DA(1)\_",""CP"",",DR="1///"\_DGPROCDT D ^DIE  K DGPROCDT  Q  ; INDEX ;index entire file (set logic)  N IBMAED D SAVERC(IBIFN,.IBMAED) ; IB\*2.0\*447 BI - Save the value of piece 16 of each RC node before re-indexing.  S DIK="^DGCR(399,",DA=IBIFN D IX1^DIK K DA,DIK  D RESTRC(IBIFN,.IBMAED) ; IB\*2.0\*447 BI - Restore the value of piece 16 of each RC node before re-indexing.  Q  ; PRIOR(IBIFN) ; set Secondary/Tertiary Bill #s on prior bills, if the bill is cancelled remove it from prior bills  N IBSEQ,IBSEQN,IBM1,I,IBIFN1  S IBSEQ=$$COB^IBCEF(IBIFN)  S IBSEQN=$S(IBSEQ="S":6,IBSEQ="T":7,1:"") Q:'IBSEQN  ;  S IBM1=$G(^DGCR(399,IBIFN,"M1")) I +$P(^DGCR(399,IBIFN,0),U,13)=7 S IBIFN=""  F I=5,6 I I<IBSEQN  S IBIFN1=+$P(IBM1,U,I) I +IBIFN1,$D(^DGCR(399,+IBIFN1,0)) S $P(^DGCR(399,IBIFN1,"M1"),U,IBSEQN)=IBIFN  Q  ; COBCHG(IBIFN,IBINS,IBCOB) ; Make changes for a new COB payer for bill  ; IBIFN = ien of bill in file 399  ; IBINS = ien of bill's current insurance (optional)  ; IBCOB = array subscripted by node,piece of COB data field change  ;  N I,IBFRMTYP,IBTAXLST  ; Subtract the Prior Payments from the bill's Offset (these are re-added by triggers)  F I=4,5,6 S $P(^DGCR(399,IBIFN,"U1"),U,2)=$P($G(^DGCR(399,IBIFN,"U1")),U,2)-$P($G(^DGCR(399,IBIFN,"U2")),U,I)  ;  I $G(IBINS),$$MCRWNR^IBEFUNC(IBINS) D  . ;MCRWNR is current insurance ... move payer only  . N IBCOBN,IBX  . S IBCOBN=$$COBN^IBCEF(IBIFN)  . S IBCOB(0,21)=$P("S^T^",U,IBCOBN)  . S IBCOB("M1",IBCOBN+4)=IBIFN  . S IBCOB("TX",1)="",IBCOB("TX",2)=""  . S IBX=$$REQMRA^IBEFUNC(IBIFN)  . I IBX=0 S IBCOB("TX",5)=0 ; MRA not needed  . I IBX["R" S IBCOB("TX",5)="A"                     ; MRA skipped  . I IBX=1,$$CHK^IBCEMU1(IBIFN) S IBCOB("TX",5)="C"  ; MRA on file  . I $G(IBPRCOB) S IBCOB("TX",5)="C"                 ; MRA being proc'd  . D PRIOR(IBIFN)  . Q  ;  ;reset fields for next Sequence Payer  F I=0,"M1","U2","TX" I $D(IBCOB(I)) D COB  ;  ; IB\*2.0\*211  ; save off Form Type  S IBFRMTYP=$P($G(^DGCR(399,IBIFN,0)),U,19)  ; Save off Taxonomies for providers.  S I=0 F  S I=$O(^DGCR(399,IBIFN,"PRV",I)) Q:'I  S IBTAXLST(I)=$P($G(^DGCR(399,IBIFN,"PRV",I,0)),U,15)  ;  ; fire xrefs set logic  D INDEX  ;  ; Restore Form Type if changed, but don't restore Form Type if  ; creating CMS-1500 claim from CTCOPY1^IBCCCB  I $G(IBCTCOPY)'=1,IBFRMTYP'=$P($G(^DGCR(399,IBIFN,0)),U,19) N DA,DIE,DR S DA=IBIFN,DIE="^DGCR(399,",DR=".19////"\_IBFRMTYP D ^DIE  ;  ; Restore Claim MRA Status field since triggers in fields 101 & 102  ; will overwrite the correct value when processing the MRA/EOB.  ; If we're processing the MRA/EOB, then a valid MRA has been received.  I $G(IBPRCOB) N DA,DIE,DR S DA=IBIFN,DIE="^DGCR(399,",DR="24////C" D ^DIE  ;  ; Only if cloning, then restore Taxonomies in fields 243 and 244 and 252.  I '$G(IBINS),'$G(IBPRCOB) D  . S I=$P($G(IBND("U3")),U,2)  . I I'=$P($G(^DGCR(399,IBIFN,"U3")),U,2) D  .. N DA,DIE,DR S DA=IBIFN,DIE="^DGCR(399,",DR="243////"\_$S(I'="":I,1:"@") D ^DIE  . ;  . S I=$P($G(IBND("U3")),U,3)  . I I'=$P($G(^DGCR(399,IBIFN,"U3")),U,3) D  .. N DA,DIE,DR S DA=IBIFN,DIE="^DGCR(399,",DR="244////"\_$S(I'="":I,1:"@") D ^DIE  . ;  . S I=$P($G(IBND("U3")),U,11)  . I I'=$P($G(^DGCR(399,IBIFN,"U3")),U,11) D  .. N DA,DIE,DR S DA=IBIFN,DIE="^DGCR(399,",DR="252////"\_$S(I'="":I,1:"@") D ^DIE  . Q  ;  ; Restore Taxonomies in field .15 in sub-file 399.0222.  S IBTAXLST=0 F  S IBTAXLST=$O(IBTAXLST(IBTAXLST)) Q:'IBTAXLST  D  . S I=IBTAXLST(IBTAXLST)  . I I=$P($G(^DGCR(399,IBIFN,"PRV",IBTAXLST,0)),U,15) Q  ; No change  . N DA,DIE,DR  . S DA(1)=IBIFN,DA=IBTAXLST  . S DIE="^DGCR(399,"\_DA(1)\_",""PRV"",",DR=".15////"\_$S(I'="":I,1:"@")  . D ^DIE  . Q  ;  K IBCOB("TX")  Q  ; SAVERC(IBIFN,IBMAED) ; IB\*2.0\*447 BI - Save the value of piece 16 of each RC node before re-indexing.  Q:$G(IBCTCOPY)=1 Q:$G(IBCTCOPY)=2  N IBCNT S IBCNT=0  Q:'$G(IBIFN) Q:'$D(^DGCR(399,IBIFN,"RC"))  F  S IBCNT=$O(^DGCR(399,IBIFN,"RC",IBCNT)) Q:+IBCNT=0 D  . S IBMAED(IBCNT)=$P($G(^DGCR(399,IBIFN,"RC",IBCNT,0)),U,16)  Q  ; RESTRC(IBIFN,IBMAED) ; IB\*2.0\*447 BI - Restore the value of piece 16 of each RC node after re-indexing.  Q:$G(IBCTCOPY)=1 Q:$G(IBCTCOPY)=2  N IBCNT S IBCNT=0  Q:'$G(IBIFN) Q:'$D(^DGCR(399,IBIFN,"RC"))  F  S IBCNT=$O(IBMAED(IBCNT)) Q:+IBCNT=0 D  . S $P(^DGCR(399,IBIFN,"RC",IBCNT,0),U,16)=IBMAED(IBCNT)  Q | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCECOB4 | | | | | | | | |
| **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** | | | | | | | | | |
| IBCECOB4 ;ALB/CXW - IB EM MANAGEMENT - REVIEW STATUS SCREEN ;16-MAY-2000  ;;2.0;INTEGRATED BILLING;\*\*137,181,348,349\*\*;21-MAR-1994;Build 46  ;;Per VHA Directive 2004-038, this routine should not be modified.  ; EN ; -- main entry point for claims status awaiting resolution detail  S VALMCNT=0,VALMBG=1  D EN^VALM("IBCEM EOB REVIEW")  Q  ; HDR ; -- header code  ;IBDA - ien EOB selection screen  N IBST  S IBST=$P($G(^IBM(361.1,IBDA,0)),U,16)  S VALMHDR(2)="Review Status= "\_$S(IBST=1:"REVIEW IN PROCESS",IBST=2:"ACCEPTED-INTERIM EOB",IBST=3:"ACCEPTED-COMPLETE EOB",IBST=4:"REJECTED",IBST=9:"CLAIM CANCELLED",1:"NOT REVIEWED")  Q  ; INIT ; -- init variables and list array  N I,X,Y,Z,IBZ,IBFST,IBPAT  K ^TMP("IBCECOC",$J) SCR S VALMCNT=0  ; IBCMT = the data extracted into ^TMP("IBCECOB1",$J)  ; IBIFN = the ien of the bill  ; IBDA = the ien of the entry in 361.1  S Z=$G(^DPT(+$P($G(^DGCR(399,IBIFN,0)),U,2),0))  S IBPAT=$E($P(Z,U),1,25)\_"/"\_$E($P(Z,U,9),6,9)  S X=""  S X=$$SETFLD^VALM1($$BN1^PRCAFN(IBIFN),X,"BILL")  S X=$$SETFLD^VALM1($$DAT1^IBOUTL($P(IBCMT,U)),X,"SERVICE")  S X=$$SETFLD^VALM1(IBPAT,X,"PATNM")  S X=$$SETFLD^VALM1(" "\_$P("PRI^SEC^TER",U,+$P(IBCMT,U,16)),X,"SEQ")  S X=$$SETFLD^VALM1(" "\_$$TYPE^IBJTLA1($P(IBCMT,U,5))\_"/"\_$S(+$P(IBCMT,U,6)=2:"CMS-1500",1:"UB-04"),X,"BTYPE")  D SET(X)  S Z=0 F  S Z=$O(^IBM(361.1,IBDA,21,Z)) Q:'Z  S I=$G(^(Z,0)) D  . S X=$$SETSTR^VALM1("Review Date/Time: "\_$$EXPAND^IBTRE(361.121,.01,+I),"",2,40)  . D SET(X)  . I $P($G(^VA(200,+$P(I,U,2),0)),U)'="" S X=$$SETSTR^VALM1("Reviewed By: "\_$P($G(^VA(200,+$P(I,U,2),0)),U),"",2,50) D SET(X)  . S (IBFST,Y)=0 F  S Y=$O(^IBM(361.1,IBDA,21,Z,1,Y)) Q:'Y  D  .. S X=$$SETSTR^VALM1($S('IBFST:"Comments: ",1:"")\_$G(^IBM(361.1,IBDA,21,Z,1,Y,0)),"",2,$S('IBFST:140,1:150))  .. D SET(X)  .. S IBFST=1  . D SET("") INITQ Q  ; HELP ; -- help code  S X="?" D DISP^XQORM1 W !!  Q  ; EXIT ; -- exit code  K ^TMP("IBCECOC",$J)  D CLEAN^VALM10  Q  ; SET(X) ;  S VALMCNT=VALMCNT+1  S ^TMP("IBCECOC",$J,VALMCNT,0)=X  S ^TMP("IBCECOC",$J,"IDX",VALMCNT,1)=""  S ^TMP("IBCECOC",$J,1)=VALMCNT  Q  ; STATUS ; Edit review status  ;IBDA - EOB ien  N DA,DIE,DR,IBOLD,DIC,DO,DD,DLAYGO,IBFINAL,IBO,IBNEW,IBFACT  D FULL^VALM1  S DIE="^IBM(361.1,"  S DA=IBDA  G:'DA STATUSQ  S IBOLD=$P($G(^IBM(361.1,DA,0)),U,16),IBFINAL=0,IBO=$S(IBOLD'="":"/"\_IBOLD,1:"@")  S DR="@1;.16;I +X<3 S IBFINAL=0,Y=""@99"";S IBFINAL=1;.2;I X="""" W !,""For a final status, this field is required"" S Y=""@98"";S Y=""@99"";@98;.16///"\_IBO\_";S Y=""@1"";@99"  L +^IBM(361.1,IBDA):3 I '$T D  G STATUSQ  . W !,"Sorry, another user currently editing this entry (#"\_IBDA\_")."  D ^DIE  ;  I $G(IBFINAL) D  ;Final status selected - let remarks be entered  . N Z  . S Z=IBDA  . N IBDA,Q,DIE,DR,DA,X,Y  . S IBDA(1)=Z,IBDA=""  . D ADDCOM(.IBDA,.DUZ,.IBCOM)  . I $P($G(^IBM(361.1,IBDA(1),0)),U,20)="F",'$O(^IBM(361.1,IBDA(1),21,+IBDA,0)) D   ; Require remarks for 'OTHER ACTION' final status  .. W !,"Since FILED - NO ACTION final status was selected, you must enter a",!," comment explaining the FILED - NO ACTION" D ADDCOM(.IBDA,.DUZ,.IBCOM,1)  .. I IBDA D  ... ; Delete entry if just entered without a comment  ... D KILLREV(.IBDA)  .. I '$O(^IBM(361.1,IBDA(1),21,+IBDA,0)) S DIE="^IBM(361.1,",DA=IBDA(1),DR=".20///@;.16///"\_IBO D ^DIE W !,"The review status was not changed because no comment was entered",! Q  S IBNEW=$P($G(^IBM(361.1,DA,0)),U,16)  ;if time out-no change in review status  S IBFACT=$P($G(^IBM(361.1,DA,0)),U,20)  I $G(IBFINAL),IBFACT="",IBNEW>1 D  G STATUSQ  . W !,"The review status was not changed because no final status was selected"  . S DR=".16////"\_IBOLD,DIE="^IBM(361.1," D ^DIE  I IBNEW>1,$P(^IBM(361.1,DA,0),U,19) D  . I "CR"'[IBFACT D  .. N DIR,X,Y  .. S DIR("?",1)="IF THIS BILL HAS RECEIVED ITS FINAL ELECTRONIC MESSAGE AND NO FURTHER ACTION",DIR("?",2)="WILL BE TAKEN ON IT, ANSWER YES"  .. S DIR("A")="DO YOU WANT TO CLOSE THE TRANSMISSION RECORD FOR THIS CLAIM?: ",DIR("B")="NO",DIR(0)="YA" D ^DIR  .. I Y>0 S IBFACT="N"  . I "NCR"[IBFACT D UPDEDI^IBCEM(+$P(^IBM(361.1,DA,0),U,19),IBFACT) Q  I IBOLD'=IBNEW D  ;Note the change and who made it  . N IBIEN,IBTEXT,DA  . S DA(1)=IBDA,DIC="^IBM(361.1,"\_DA(1)\_",21,",DIC(0)="L",DLAYGO=361.121  . S X=$$NOW^XLFDT  . S DIC("P")=$$GETSPEC^IBEFUNC(361.1,21)  . D FILE^DICN K DIC,DD,DO,DLAYGO  . Q:Y'>0  . S DA(2)=DA(1),DA(1)=+Y,IBIEN=DA(1)\_","\_DA(2)\_",",IBTEXT(1)="REVIEW STATUS CHANGED TO '"\_$$EXPAND^IBTRE(361.1,.16,$P(^IBM(361.1,DA(2),0),U,16))\_"' BY: "\_$$EXPAND^IBTRE(361.121,.02,+$G(DUZ))  . D WP^DIE(361.121,IBIEN,1,,"IBTEXT") K ^TMP("DIERR",$J)  . D HDR,INIT  L -^IBM(361.1,DA) STATUSQ ;  D PAUSE^VALM1  S VALMBCK="R"  Q  ; ADDCOM(IBDA,DUZ,IBCOM,ADD) ; Add review comment to file 361.1  ; IBDA = array containing the DA references for the file add -  ; pass by reference  ; DUZ = ien of the user  ; ADD = flag when set to 1 says the review date exists,  ; just allow comment entry  ; Returns IBDA = the entry # of the comment  ; and IBCOM array referencing any comments added by the user  ;  N DA,DIC,DD,DO,DLAYGO,X,Y  S DR=$S($G(DUZ):".02////"\_DUZ\_";",1:"")\_"1"  I '$G(ADD) D  . K DO,DD  . S DIC="^IBM(361.1,"\_IBDA(1)\_",21,",DA(1)=IBDA(1),X=$$NOW^XLFDT  . W !,"New Review Date: "\_$$FMTE^XLFDT(X,2)  . S DIC("DR")=DR,DLAYGO=361.121  . S DIC(0)="L",DIC("P")=$$GETSPEC^IBEFUNC(361.1,21)  . D FILE^DICN K DIC,DD,DO,DLAYGO  . S IBDA=+Y  I IBDA>0 D  . I $G(ADD) S DIE="^IBM(361.1,"\_IBDA(1)\_",21,",DA(1)=IBDA(1),DA=IBDA D ^DIE  . I '$O(^IBM(361.1,IBDA(1),21,IBDA,0)) D KILLREV(.IBDA) Q  . S IBCOM(DUZ,IBDA)=""  Q  ; KILLREV(IBDA) ; Deletes a review date if no comments entered  N DA,DIK  S DA=IBDA,DA(1)=IBDA(1),DIK="^IBM(361.1,"\_IBDA(1)\_",21,"  K IBCOM(DUZ,IBDA)  D ^DIK  Q  ; | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCECOB4 ;ALB/CXW - IB EM MANAGEMENT - REVIEW STATUS SCREEN ;16-MAY-2000  ;;2.0;INTEGRATED BILLING;\*\*137,181,348,349,592\*\*;21-MAR-1994;Build 46  ;;Per VHA Directive 2004-038, this routine should not be modified.  ; EN ; -- main entry point for claims status awaiting resolution detail  S VALMCNT=0,VALMBG=1  D EN^VALM("IBCEM EOB REVIEW")  Q  ; HDR ; -- header code  ;IBDA - ien EOB selection screen  N IBST  S IBST=$P($G(^IBM(361.1,IBDA,0)),U,16)  S VALMHDR(2)="Review Status= "\_$S(IBST=1:"REVIEW IN PROCESS",IBST=2:"ACCEPTED-INTERIM EOB",IBST=3:"ACCEPTED-COMPLETE EOB",IBST=4:"REJECTED",IBST=9:"CLAIM CANCELLED",1:"NOT REVIEWED")  Q  ; INIT ; -- init variables and list array  N I,X,Y,Z,IBZ,IBFST,IBPAT  K ^TMP("IBCECOC",$J) SCR S VALMCNT=0  ; IBCMT = the data extracted into ^TMP("IBCECOB1",$J)  ; IBIFN = the ien of the bill  ; IBDA = the ien of the entry in 361.1  S Z=$G(^DPT(+$P($G(^DGCR(399,IBIFN,0)),U,2),0))  S IBPAT=$E($P(Z,U),1,25)\_"/"\_$E($P(Z,U,9),6,9)  S X=""  S X=$$SETFLD^VALM1($$BN1^PRCAFN(IBIFN),X,"BILL")  S X=$$SETFLD^VALM1($$DAT1^IBOUTL($P(IBCMT,U)),X,"SERVICE")  S X=$$SETFLD^VALM1(IBPAT,X,"PATNM")  S X=$$SETFLD^VALM1(" "\_$P("PRI^SEC^TER",U,+$P(IBCMT,U,16)),X,"SEQ")  ;JWS;IB\*2.0\*592:Dental form #7 J430D  S X=$$SETFLD^VALM1(" "\_$$TYPE^IBJTLA1($P(IBCMT,U,5))\_"/"\_$S(+$P(IBCMT,U,6)=2:"CMS-1500",$P(IBCMT,U,6)=7:"J430D",1:"UB-04"),X,"BTYPE")  D SET(X)  S Z=0 F  S Z=$O(^IBM(361.1,IBDA,21,Z)) Q:'Z  S I=$G(^(Z,0)) D  . S X=$$SETSTR^VALM1("Review Date/Time: "\_$$EXPAND^IBTRE(361.121,.01,+I),"",2,40)  . D SET(X)  . I $P($G(^VA(200,+$P(I,U,2),0)),U)'="" S X=$$SETSTR^VALM1("Reviewed By: "\_$P($G(^VA(200,+$P(I,U,2),0)),U),"",2,50) D SET(X)  . S (IBFST,Y)=0 F  S Y=$O(^IBM(361.1,IBDA,21,Z,1,Y)) Q:'Y  D  .. S X=$$SETSTR^VALM1($S('IBFST:"Comments: ",1:"")\_$G(^IBM(361.1,IBDA,21,Z,1,Y,0)),"",2,$S('IBFST:140,1:150))  .. D SET(X)  .. S IBFST=1  . D SET("") INITQ Q  ; HELP ; -- help code  S X="?" D DISP^XQORM1 W !!  Q  ; EXIT ; -- exit code  K ^TMP("IBCECOC",$J)  D CLEAN^VALM10  Q  ; SET(X) ;  S VALMCNT=VALMCNT+1  S ^TMP("IBCECOC",$J,VALMCNT,0)=X  S ^TMP("IBCECOC",$J,"IDX",VALMCNT,1)=""  S ^TMP("IBCECOC",$J,1)=VALMCNT  Q  ; STATUS ; Edit review status  ;IBDA - EOB ien  N DA,DIE,DR,IBOLD,DIC,DO,DD,DLAYGO,IBFINAL,IBO,IBNEW,IBFACT  D FULL^VALM1  S DIE="^IBM(361.1,"  S DA=IBDA  G:'DA STATUSQ  S IBOLD=$P($G(^IBM(361.1,DA,0)),U,16),IBFINAL=0,IBO=$S(IBOLD'="":"/"\_IBOLD,1:"@")  S DR="@1;.16;I +X<3 S IBFINAL=0,Y=""@99"";S IBFINAL=1;.2;I X="""" W !,""For a final status, this field is required"" S Y=""@98"";S Y=""@99"";@98;.16///"\_IBO\_";S Y=""@1"";@99"  L +^IBM(361.1,IBDA):3 I '$T D  G STATUSQ  . W !,"Sorry, another user currently editing this entry (#"\_IBDA\_")."  D ^DIE  ;  I $G(IBFINAL) D  ;Final status selected - let remarks be entered  . N Z  . S Z=IBDA  . N IBDA,Q,DIE,DR,DA,X,Y  . S IBDA(1)=Z,IBDA=""  . D ADDCOM(.IBDA,.DUZ,.IBCOM)  . I $P($G(^IBM(361.1,IBDA(1),0)),U,20)="F",'$O(^IBM(361.1,IBDA(1),21,+IBDA,0)) D   ; Require remarks for 'OTHER ACTION' final status  .. W !,"Since FILED - NO ACTION final status was selected, you must enter a",!," comment explaining the FILED - NO ACTION" D ADDCOM(.IBDA,.DUZ,.IBCOM,1)  .. I IBDA D  ... ; Delete entry if just entered without a comment  ... D KILLREV(.IBDA)  .. I '$O(^IBM(361.1,IBDA(1),21,+IBDA,0)) S DIE="^IBM(361.1,",DA=IBDA(1),DR=".20///@;.16///"\_IBO D ^DIE W !,"The review status was not changed because no comment was entered",! Q  S IBNEW=$P($G(^IBM(361.1,DA,0)),U,16)  ;if time out-no change in review status  S IBFACT=$P($G(^IBM(361.1,DA,0)),U,20)  I $G(IBFINAL),IBFACT="",IBNEW>1 D  G STATUSQ  . W !,"The review status was not changed because no final status was selected"  . S DR=".16////"\_IBOLD,DIE="^IBM(361.1," D ^DIE  I IBNEW>1,$P(^IBM(361.1,DA,0),U,19) D  . I "CR"'[IBFACT D  .. N DIR,X,Y  .. S DIR("?",1)="IF THIS BILL HAS RECEIVED ITS FINAL ELECTRONIC MESSAGE AND NO FURTHER ACTION",DIR("?",2)="WILL BE TAKEN ON IT, ANSWER YES"  .. S DIR("A")="DO YOU WANT TO CLOSE THE TRANSMISSION RECORD FOR THIS CLAIM?: ",DIR("B")="NO",DIR(0)="YA" D ^DIR  .. I Y>0 S IBFACT="N"  . I "NCR"[IBFACT D UPDEDI^IBCEM(+$P(^IBM(361.1,DA,0),U,19),IBFACT) Q  I IBOLD'=IBNEW D  ;Note the change and who made it  . N IBIEN,IBTEXT,DA  . S DA(1)=IBDA,DIC="^IBM(361.1,"\_DA(1)\_",21,",DIC(0)="L",DLAYGO=361.121  . S X=$$NOW^XLFDT  . S DIC("P")=$$GETSPEC^IBEFUNC(361.1,21)  . D FILE^DICN K DIC,DD,DO,DLAYGO  . Q:Y'>0  . S DA(2)=DA(1),DA(1)=+Y,IBIEN=DA(1)\_","\_DA(2)\_",",IBTEXT(1)="REVIEW STATUS CHANGED TO '"\_$$EXPAND^IBTRE(361.1,.16,$P(^IBM(361.1,DA(2),0),U,16))\_"' BY: "\_$$EXPAND^IBTRE(361.121,.02,+$G(DUZ))  . D WP^DIE(361.121,IBIEN,1,,"IBTEXT") K ^TMP("DIERR",$J)  . D HDR,INIT  L -^IBM(361.1,DA) STATUSQ ;  D PAUSE^VALM1  S VALMBCK="R"  Q  ; ADDCOM(IBDA,DUZ,IBCOM,ADD) ; Add review comment to file 361.1  ; IBDA = array containing the DA references for the file add -  ; pass by reference  ; DUZ = ien of the user  ; ADD = flag when set to 1 says the review date exists,  ; just allow comment entry  ; Returns IBDA = the entry # of the comment  ; and IBCOM array referencing any comments added by the user  ;  N DA,DIC,DD,DO,DLAYGO,X,Y  S DR=$S($G(DUZ):".02////"\_DUZ\_";",1:"")\_"1"  I '$G(ADD) D  . K DO,DD  . S DIC="^IBM(361.1,"\_IBDA(1)\_",21,",DA(1)=IBDA(1),X=$$NOW^XLFDT  . W !,"New Review Date: "\_$$FMTE^XLFDT(X,2)  . S DIC("DR")=DR,DLAYGO=361.121  . S DIC(0)="L",DIC("P")=$$GETSPEC^IBEFUNC(361.1,21)  . D FILE^DICN K DIC,DD,DO,DLAYGO  . S IBDA=+Y  I IBDA>0 D  . I $G(ADD) S DIE="^IBM(361.1,"\_IBDA(1)\_",21,",DA(1)=IBDA(1),DA=IBDA D ^DIE  . I '$O(^IBM(361.1,IBDA(1),21,IBDA,0)) D KILLREV(.IBDA) Q  . S IBCOM(DUZ,IBDA)=""  Q  ; KILLREV(IBDA) ; Deletes a review date if no comments entered  N DA,DIK  S DA=IBDA,DA(1)=IBDA(1),DIK="^IBM(361.1,"\_IBDA(1)\_",21,"  K IBCOM(DUZ,IBDA)  D ^DIK  Q  ; | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCECSA5 | | | | | | | | |
| **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** | | | | | | | | | |
| IBCECSA5 ;ALB/CXW - VIEW EOB SCREEN ;01-OCT-1999  ;;2.0;INTEGRATED BILLING;\*\*137,135,263,280,155,349,489,488,547\*\*;21-MAR-1994;Build 119  ;;Per VA Directive 6402, this routine should not be modified.  ;  ; reference to $$VFILE^DILFD allowed with IA#2055 (IB\*2.0\*547)  ; EN ; -- main entry point for VIEW EOB  N VALMCNT,VALMBG,VALMHDR  S VALMCNT=0,VALMBG=1  D EN^VALM("IBCEM VIEW EOB")  Q  ; INIT ; -- init variables and list array  I '$G(IBIFN) S VALMQUIT="" G INITQ    ; bill# is required  D HDR^IBCEOB2 ; build the VALMHDR array  K IBCNT,IBONE,^TMP("IBCECSD",$J) ; kill vars and scratch global  ;  ; 8/13/03 - If variable IBEOBIFN is set, then this is the 361.1 ien  ; that the user selected from a list. Build the detail.  I $G(IBEOBIFN) S IBCNT=IBEOBIFN,IBONE=1 D BLD^IBCECSA6,EOBERR G INITQ  ;  D BLD^IBCEOB2 ; build ^TMP("IBCEOB",$J) containing MRA/EOB lister  S IBONE=0  M ^TMP("IBCECSD",$J)=^TMP("IBCEOB",$J)  ;  ; 4/7/03 - If only 1 EOB record found for this bill, then set the  ; IBCNT variable, the IBONE one-time flag, and build the  ; detail sections of this list.  I $G(VALMCNT)=1 S IBCNT=$P($G(^TMP("IBCECSD",$J,1)),U,2),IBONE=1 I IBCNT D BLD^IBCECSA6  D EOBERR   ; IB\*2.0\*488 (vd)  ; INITQ Q  ; HELP ; -- help code  S X="?" D DISP^XQORM1 W !!  Q  ; EXIT ; -- exit code  K ^TMP("IBCECSD",$J)  D CLEAR^VALM1,CLEAN^VALM10  Q MIN ;  N IBREC1,IBRM1,IBRM2,IBRM3,IBRM4,IBRM5,IBRL,IBTYPE,IBT,IBTX,IBD  ; flag for inpatient mra  S IBTYPE=$S($G(IBSRC):1,$$INPAT^IBCEF(+IBREC):1,1:0)  ;  S IB=$$SETSTR^VALM1("MEDICARE INFORMATION:","",1,50)  D SET(IB)  I '$G(IBSRC) D  . D CNTRL^VALM10(VALMCNT,1,21,IORVON,IORVOFF)  . S ^TMP("IBCECSD",$J,"X",5)=VALMCNT  I $G(IBSRC),'$D(^IBM(361.1,IBCNT,4)) Q  I '$G(IBSRC),'$$INPAT^IBCEF(+IBREC) Q  D SET(" INPATIENT:")  S IBREC1=$G(^IBM(361.1,IBCNT,4)),(IB,IBRL)=""  ;  F IBT=2:1 S IBTX=$P($T(MINDAT+IBT),";",3) Q:IBTX=""  D  . S IBD=$P(IBREC1,"^",+IBTX)  . I $L($P(IBTX,"^",4)) X $P(IBTX,"^",4) E  N IBFULL S IBFULL=1  . I $S(IBFULL:1,1:IBD) D  .. I $L($P(IBTX,"^",4)) X $P(IBTX,"^",4) I  Q  .. X "S IBD="\_$S($L($P(IBTX,"^",3)):$P(IBTX,"^",3),1:"$$A10(IBD)")  .. S IB=$$SETSTR^VALM1($P(IBTX,"^",2)\_IBD,IB,$S('IBRL:4,1:37),$S('IBRL:41,1:38))  .. S IBRL=$S(IBRL:0,1:1)  .. I 'IBRL D SET(IB,IBRL) S IB=""  ;  D:IBRL'="" SET(IB)  D REMARK  Q  ; MINDAT ; data for MIN tag  ; format: piece^label^special format code^special decision for disp  ;;1^Cov Days/Visit Ct : ^$$RJ(+IBD)^I $G(IBSRC)  ;;3^Claim DRG Amt :   ;;2^Lifetm Psych Dy Ct : ^$$RJ(IBD)  ;;5^Disprop Share Amt : ^^I IBTYPE  ;;4^Cap Exception Amt :   ;;7^PPS Capital Amt : ^^I IBTYPE  ;;6^MSP Pass Thru Amt :   ;;9^PPS Cap HSP-DRG Amt: ^^I IBTYPE  ;;8^PPS Cap FSP-DRG Amt: ^^I IBTYPE  ;;11^Old Capital Amt : ^^I IBTYPE  ;;10^PPS Cap DSH-DRG Amt: ^^I IBTYPE  ;;13^PPS Op Hos DRG Amt :   ;;12^PPS Capital IME Amt: ^^I IBTYPE  ;;15^PPS Op Fed DRG Amt : ^^I IBTYPE  ;;14^Cost Report Day Ct : ^$$RJ(IBD)^I IBTYPE  ;;17^Indirect Teach Amt : ^^I IBTYPE  ;;16^PPS Cap Outlier Amt: ^^I IBTYPE  ;;18^Non-Pay Prof Comp : ^$$RJ(IBD)  ;;19^Non-Covered Days Ct: ^$$RJ(+IBD)^I IBTYPE  ;;  ; REMARK ; set up remarks and line level details  N IBREC1,IBP,IBT,IBX,RCODE,RDESC,REXIST  Q:$G(IBREM) S IBREM=1  D SET(" ")  D SET(" Claim Level Remark Information")  D SET(" Code Description")  I '$G(IBSRC) D  . D CNTRL^VALM10(VALMCNT,4,4,IOUON,IOUOFF)  . D CNTRL^VALM10(VALMCNT,13,11,IOUON,IOUOFF)  . Q  ;  S IBREC1=$P($G(^IBM(361.1,IBCNT,3)),U,3,7)  I $P(IBREC1,U,1)="" S IBREC1=$P($G(^IBM(361.1,IBCNT,5)),U,1,5)  S REXIST=0  ;  F IBP=1:1:5 D  . S RCODE=$P(IBREC1,U,IBP)  . S RDESC=$G(^IBM(361.1,IBCNT,"RM"\_IBP))  . ; IB\*2.0\*547 - get RARC desription from new AR file 346 when available  . I '$$VFILE^DILFD(346),RCODE="",RDESC="" Q  . K IBT  . Q:RCODE=""  . I '$$VFILE^DILFD(346) S REXIST=1,IBT(IBP)=RDESC  . I $$VFILE^DILFD(346) S REXIST=$$CARC(RCODE,346,60,"IBT") Q:REXIST<1  . D TXT1(.IBT,0,60)  . D SET(" "\_$$LJ^XLFSTR(RCODE,6)\_"- "\_$G(IBT(1)))  . S IBX=1  . F  S IBX=$O(IBT(IBX)) Q:'IBX  D SET($J("",12)\_IBT(IBX))  . Q  ;  I 'REXIST D SET(" No claim level remarks on file")  D SET(" ")  Q:$G(IBSRC) ; MRA Only  ; MRALLA S IB=$$SETSTR^VALM1("LINE LEVEL ADJUSTMENTS:","",1,50)  D SET(IB)  I '$G(IBSRC) D  . D CNTRL^VALM10(VALMCNT,1,23,IORVON,IORVOFF)  . S ^TMP("IBCECSD",$J,"X",7)=VALMCNT  I '$D(^IBM(361.1,IBCNT,15,0)) D SET(" NONE") Q  ; only if there is info  ;  ; look up all billed data  N IBZDATA,IBFORM,IBX2,IBX3,IBREC2,IBREC3,IBTX,IBT,IBRC,IBZ,IBTXL  S IBFORM=0 ; cms-1500  I $$FT^IBCEF(+IBREC)=3 S IBFORM=1 ; UB-04  D F^IBCEF("N-"\_$S(IBFORM:"UB-04",1:"HCFA 1500")\_" SERVICE LINE (EDI)","IBZDATA",,+IBREC)  ;  S IBX=0 F  S IBX=$O(^IBM(361.1,IBCNT,15,IBX)) Q:IBX<1 S IBREC1=^IBM(361.1,IBCNT,15,IBX,0) D  . NEW RVL  . D SET(" # SV DT REVCD PROC MOD UNITS BILLED DEDUCT COINS ALLOW PYMT")  . S RVL=+$P(IBREC1,U,12) ; referenced Vista line#  . I 'RVL S RVL=IBX             ; use the EOB line# if not there  . S IBT=$$RJ($P(IBREC1,"^"),3) ; line number  . S IBT=IBT\_" "\_$$RJ($$DAT1^IBOUTL($P($P(IBREC1,"^",16),".")),8) ; service date  . S IBT=IBT\_" "\_$$RJ($$EXTERNAL^DILFD(361.115,.1,"",$P(IBREC1,"^",10)),6) ; revcd  . S IBT=IBT\_" "\_$$RJ($P(IBREC1,"^",4),5) ; procedure  . S IBT=IBT\_" "\_$$RJ($P($G(^IBM(361.1,IBCNT,15,IBX,2,1,0)),"^"),3)\_$S($D(^IBM(361.1,IBCNT,15,IBX,2,2,0)):"+",1:" ") ; modifiers  . S IBT=IBT\_" "\_$$RJ($FN($P(IBREC1,"^",11),"",0),5) ; units  . S IBT=IBT\_" "\_$$RJ($FN($S(IBFORM:$P($G(IBZDATA(RVL)),"^",5),1:$P($G(IBZDATA(RVL)),"^",8)\*$P($G(IBZDATA(RVL)),"^",9)),"",2),8) ; billed  . S IBT=IBT\_" "\_$$RJ($FN($P($G(^IBM(361.1,IBCNT,15,IBX,1,+$O(^IBM(361.1,IBCNT,15,IBX,1,"B","PR",0)),1,+$O(^IBM(361.1,IBCNT,15,IBX,1,+$O(^IBM(361.1,IBCNT,15,IBX,1,"B","PR",0)),1,"B",1,0)),0)),"^",2),"",2),7) ; deduct  . S IBT=IBT\_" "\_$$RJ($FN($P($G(^IBM(361.1,IBCNT,15,IBX,1,+$O(^IBM(361.1,IBCNT,15,IBX,1,"B","PR",0)),1,+$O(^IBM(361.1,IBCNT,15,IBX,1,+$O(^IBM(361.1,IBCNT,15,IBX,1,"B","PR",0)),1,"B",2,0)),0)),"^",2),"",2),6) ; coins  . S IBT=IBT\_" "\_$$RJ($FN($P(IBREC1,"^",13),"",2),8) ; allow  . S IBT=IBT\_" "\_$$RJ($FN($P(IBREC1,"^",3),"",2),8) ; payment  . D SET(IBT)  . S IBX2=0 F  S IBX2=$O(^IBM(361.1,IBCNT,15,IBX,1,IBX2)) Q:IBX2<1 D  .. S IBREC2=^IBM(361.1,IBCNT,15,IBX,1,IBX2,0),IBX3=0  .. F  S IBX3=$O(^IBM(361.1,IBCNT,15,IBX,1,IBX2,1,IBX3)) Q:IBX3<1 D  ... S IBREC3=^IBM(361.1,IBCNT,15,IBX,1,IBX2,1,IBX3,0)  ... ; line level adjustments; don't display kludges (esg 10/23/03)  ... I $P(IBREC2,U,1)="PR",$P(IBREC3,U,1)="AAA" Q  ... I $P(IBREC2,U,1)="OA",$P(IBREC3,U,1)="AB3" Q  ... I $P(IBREC2,U,1)="LQ" Q  ... ; IB\*2.0\*547 - get CARC description from AR file 345, when ready  ... I '$$VFILE^DILFD(345) S IBTX(1)="ADJ: "\_$P(IBREC2,"^")\_" "\_$P(IBREC3,"^")\_" "\_$P(IBREC3,"^",4) D TXT1(.IBTX,0,79) S IBT=0 F  S IBT=$O(IBTX(IBT)) Q:IBT<1 D SET(IBTX(IBT))  ... I $$VFILE^DILFD(345) S IBT=$$CARC($P(IBREC3,"^"),345,79,"IBTX"),IBTX(1)="ADJ: "\_$P(IBREC2,"^")\_" "\_$P(IBREC3,"^")\_": "\_$G(IBTX(1)) D TXT1(.IBTX,0,79) S IBT=0 F  S IBT=$O(IBTX(IBT)) Q:IBT<1 D SET(IBTX(IBT))  ... K IBTX  ... D SET("ADJ AMT: "\_$FN($P(IBREC3,"^",2),"",2))  . S IBRC=0  . F  S IBRC=$O(^IBM(361.1,IBCNT,15,IBX,4,IBRC)) Q:'IBRC  S IBREC2=$G(^(IBRC,0)) I IBREC2 K IBTX,IBZ S IBTX(1)=" -REMARK CODE("\_+IBREC2\_"): ",IBTXL=$L(IBTX(1)) D  .. ; IB\*2.0\*547 - get RARC description from AR file 346, when ready  .. I '$$VFILE^DILFD(346) S IBTX(1)=IBTX(1)\_$P(IBREC2,U,2)\_" "\_$P(IBREC2,U,3)  .. I $$VFILE^DILFD(346) S IBT=$$CARC($P(IBREC2,U,2),346,79,"IBTX"),IBTX(1)=IBTX(1)\_$P(IBREC2,U,2)\_" "\_$G(IBT(1))  .. I $L(IBTX(1))>79 D  ... D TXT1(.IBTX,0,79) D SET(IBTX(1)) M IBZ=IBTX K IBTX S IBTX(1)="",IBT=1 F  S IBT=$O(IBZ(IBT)) Q:'IBT  S IBTX(1)=IBTX(1)\_IBZ(IBT)\_" "  .. E  D  ... S IBTXL=0  .. D TXT1(.IBTX,IBTXL,79) S IBT=0 F  S IBT=$O(IBTX(IBT)) Q:IBT<1 D SET(IBTX(IBT))  . D SET(" ")  D SET(" ")  Q  ;  ;/Beginning IB\*2.0\*488 (vd) EOBERR ; Display information about any 361.1 message storage or filing errors  N ERRTXT,DASHES,Z  S DASHES="---------------------------------------------------------------------"  I '$O(^IBM(361.1,IBCNT,"ERR",0)) Q  D SET("VistA could not match all of the Line Level data received in the EEOB")  D SET("(835 Record 40) to the claim in VistA.")  D SET(" ")  S Z=0 F  S Z=$O(^IBM(361.1,IBCNT,"ERR",Z)) Q:'Z  D  .S ERRTXT=$G(^IBM(361.1,IBCNT,"ERR",Z,0))  .I ERRTXT["##RAW DATA" S ERRTXT=DASHES  .D SET(ERRTXT)  Q  ;/End of IB\*2.0\*488 (vd)  ; TXT(IBRM,IBLN,IBXY) ;display text over 79 chars  ;IBRM - text, IBLN - length, IBXY - position  S IBRM=$E(IBRM,IBLN+1,999) REP I $E(IBRM,1,IBLN)'="" S IB=$$SETSTR^VALM1($E(IBRM,1,IBLN),"",IBXY,IBLN) D SET(IB) S IBRM=$E(IBRM,IBLN+1,999) G REP  Q  ; SET(IB,IBSAV) ;  I '$G(IBSAV) D SET^IBCECSA6($G(IBSRC),IB,+$G(CNT),IBCNT)  Q  ; A10(X) ;  Q $$A10^IBCECSA6(X)  ; A7(X) ; returns a dollar amount right justified to 7 characters  Q $$RJ($FN(X,"",2),7)  ; TXT1(IBT,DIWL,DIWR) ; sets up text for over 79 chars  ; IBT - pass by ref, array of text to be formatted back in array  ; DIWL - left margin, DIWR = right margin  N IBX,X,DIWF,IBS K ^UTILITY($J,"W")  S DIWF="|I"\_DIWL  S IBX=0 F  S IBX=$O(IBT(IBX)) Q:IBX<1 S X=IBT(IBX) D ^DIWP  K IBT F  S IBX=$O(^UTILITY($J,"W",DIWL,IBX)) Q:IBX<1 S IBT(IBX)=^UTILITY($J,"W",DIWL,IBX,0)  K ^UTILITY($J,"W")  Q  ; RJ(X,Y) ; right just, default is 10  Q $$RJ^XLFSTR(X,$G(Y,10)," ")  ; CARC(IBCDE,IBF,IBML,IBARY) ;new CARC/RACR API for IB\*2.0\*547  ; IBCDE = reason code from EOB to lookup in carc/rarc file (REQUIRED)  ; IBF = file# to do lookup (either 345-CARC or 346-RARC) \*REQUIRED\*  ; IBML = max length for each line (default is 79)  ; IBARY = (required) subscripted array to return description data in:  ; array(1)=first line of word-processed description  ; array(2)= 2nd line of wp description, and so on  ;  ; Returns total # of lines in description  ;  N IBY,IBX,IBC,IBI,IBN,IBALN,IBSTP,IBDSC  S IBC=0  Q:$G(IBARY)="" IBC  Q:$G(IBCDE)="" IBC  Q:$G(IBF)="" IBC  S:$G(IBML)="" IBML=79  S IBY=$$FIND1^DIC(IBF,,"BX",IBCDE) Q:IBY<1 IBC  S IBX=$$GET1^DIQ(IBF,IBY\_",",4,"","IBDSC")  S IBI=0 F  S IBI=$O(IBDSC(IBI)) Q:'IBI  D  .S IBC=IBC+1,IBSTP=0,IBALN=$L(IBDSC(IBI))  .S @IBARY@(IBI)=$E(IBDSC(IBI),1,IBML) Q:IBML>IBALN  .S IBDSC(IBI+1)=($E(IBDSC(IBI),(IBML+1),IBALN)\_" "\_$G(IBDSC(IBI+1)))  .; make sure we don't break words in 2  .Q:$E(@IBARY@(IBI),IBML)=" "  .F IBN=IBML:-1:1 Q:$G(IBSTP)=1 D  ..Q:$E(IBDSC(IBI),IBN)'=" "   ..S @IBARY@(IBI)=$E(IBDSC(IBI),1,IBN),IBDSC(IBI+1)=($E(IBDSC(IBI),(IBN+1),IBML)\_$G(IBDSC(IBI+1))),IBSTP=1 Q  Q IBC  ; | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCECSA5 ;ALB/CXW - VIEW EOB SCREEN ;01-OCT-1999  ;;2.0;INTEGRATED BILLING;\*\*137,135,263,280,155,349,489,488,547,592\*\*;21-MAR-1994;Build 119  ;;Per VA Directive 6402, this routine should not be modified.  ;  ; reference to $$VFILE^DILFD allowed with IA#2055 (IB\*2.0\*547)  ; EN ; -- main entry point for VIEW EOB  N VALMCNT,VALMBG,VALMHDR  S VALMCNT=0,VALMBG=1  D EN^VALM("IBCEM VIEW EOB")  Q  ; INIT ; -- init variables and list array  I '$G(IBIFN) S VALMQUIT="" G INITQ    ; bill# is required  D HDR^IBCEOB2 ; build the VALMHDR array  K IBCNT,IBONE,^TMP("IBCECSD",$J) ; kill vars and scratch global  ;  ; 8/13/03 - If variable IBEOBIFN is set, then this is the 361.1 ien  ; that the user selected from a list. Build the detail.  I $G(IBEOBIFN) S IBCNT=IBEOBIFN,IBONE=1 D BLD^IBCECSA6,EOBERR G INITQ  ;  D BLD^IBCEOB2 ; build ^TMP("IBCEOB",$J) containing MRA/EOB lister  S IBONE=0  M ^TMP("IBCECSD",$J)=^TMP("IBCEOB",$J)  ;  ; 4/7/03 - If only 1 EOB record found for this bill, then set the  ; IBCNT variable, the IBONE one-time flag, and build the  ; detail sections of this list.  I $G(VALMCNT)=1 S IBCNT=$P($G(^TMP("IBCECSD",$J,1)),U,2),IBONE=1 I IBCNT D BLD^IBCECSA6  D EOBERR   ; IB\*2.0\*488 (vd)  ; INITQ Q  ; HELP ; -- help code  S X="?" D DISP^XQORM1 W !!  Q  ; EXIT ; -- exit code  K ^TMP("IBCECSD",$J)  D CLEAR^VALM1,CLEAN^VALM10  Q MIN ;  N IBREC1,IBRM1,IBRM2,IBRM3,IBRM4,IBRM5,IBRL,IBTYPE,IBT,IBTX,IBD  ; flag for inpatient mra  S IBTYPE=$S($G(IBSRC):1,$$INPAT^IBCEF(+IBREC):1,1:0)  ;  S IB=$$SETSTR^VALM1("MEDICARE INFORMATION:","",1,50)  D SET(IB)  I '$G(IBSRC) D  . D CNTRL^VALM10(VALMCNT,1,21,IORVON,IORVOFF)  . S ^TMP("IBCECSD",$J,"X",5)=VALMCNT  I $G(IBSRC),'$D(^IBM(361.1,IBCNT,4)) Q  I '$G(IBSRC),'$$INPAT^IBCEF(+IBREC) Q  D SET(" INPATIENT:")  S IBREC1=$G(^IBM(361.1,IBCNT,4)),(IB,IBRL)=""  ;  F IBT=2:1 S IBTX=$P($T(MINDAT+IBT),";",3) Q:IBTX=""  D  . S IBD=$P(IBREC1,"^",+IBTX)  . I $L($P(IBTX,"^",4)) X $P(IBTX,"^",4) E  N IBFULL S IBFULL=1  . I $S(IBFULL:1,1:IBD) D  .. I $L($P(IBTX,"^",4)) X $P(IBTX,"^",4) I  Q  .. X "S IBD="\_$S($L($P(IBTX,"^",3)):$P(IBTX,"^",3),1:"$$A10(IBD)")  .. S IB=$$SETSTR^VALM1($P(IBTX,"^",2)\_IBD,IB,$S('IBRL:4,1:37),$S('IBRL:41,1:38))  .. S IBRL=$S(IBRL:0,1:1)  .. I 'IBRL D SET(IB,IBRL) S IB=""  ;  D:IBRL'="" SET(IB)  D REMARK  Q  ; MINDAT ; data for MIN tag  ; format: piece^label^special format code^special decision for disp  ;;1^Cov Days/Visit Ct : ^$$RJ(+IBD)^I $G(IBSRC)  ;;3^Claim DRG Amt :   ;;2^Lifetm Psych Dy Ct : ^$$RJ(IBD)  ;;5^Disprop Share Amt : ^^I IBTYPE  ;;4^Cap Exception Amt :   ;;7^PPS Capital Amt : ^^I IBTYPE  ;;6^MSP Pass Thru Amt :   ;;9^PPS Cap HSP-DRG Amt: ^^I IBTYPE  ;;8^PPS Cap FSP-DRG Amt: ^^I IBTYPE  ;;11^Old Capital Amt : ^^I IBTYPE  ;;10^PPS Cap DSH-DRG Amt: ^^I IBTYPE  ;;13^PPS Op Hos DRG Amt :   ;;12^PPS Capital IME Amt: ^^I IBTYPE  ;;15^PPS Op Fed DRG Amt : ^^I IBTYPE  ;;14^Cost Report Day Ct : ^$$RJ(IBD)^I IBTYPE  ;;17^Indirect Teach Amt : ^^I IBTYPE  ;;16^PPS Cap Outlier Amt: ^^I IBTYPE  ;;18^Non-Pay Prof Comp : ^$$RJ(IBD)  ;;19^Non-Covered Days Ct: ^$$RJ(+IBD)^I IBTYPE  ;;  ; REMARK ; set up remarks and line level details  N IBREC1,IBP,IBT,IBX,RCODE,RDESC,REXIST  Q:$G(IBREM) S IBREM=1  D SET(" ")  D SET(" Claim Level Remark Information")  D SET(" Code Description")  I '$G(IBSRC) D  . D CNTRL^VALM10(VALMCNT,4,4,IOUON,IOUOFF)  . D CNTRL^VALM10(VALMCNT,13,11,IOUON,IOUOFF)  . Q  ;  S IBREC1=$P($G(^IBM(361.1,IBCNT,3)),U,3,7)  I $P(IBREC1,U,1)="" S IBREC1=$P($G(^IBM(361.1,IBCNT,5)),U,1,5)  S REXIST=0  ;  F IBP=1:1:5 D  . S RCODE=$P(IBREC1,U,IBP)  . S RDESC=$G(^IBM(361.1,IBCNT,"RM"\_IBP))  . ; IB\*2.0\*547 - get RARC desription from new AR file 346 when available  . I '$$VFILE^DILFD(346),RCODE="",RDESC="" Q  . K IBT  . Q:RCODE=""  . I '$$VFILE^DILFD(346) S REXIST=1,IBT(IBP)=RDESC  . I $$VFILE^DILFD(346) S REXIST=$$CARC(RCODE,346,60,"IBT") Q:REXIST<1  . D TXT1(.IBT,0,60)  . D SET(" "\_$$LJ^XLFSTR(RCODE,6)\_"- "\_$G(IBT(1)))  . S IBX=1  . F  S IBX=$O(IBT(IBX)) Q:'IBX  D SET($J("",12)\_IBT(IBX))  . Q  ;  I 'REXIST D SET(" No claim level remarks on file")  D SET(" ")  Q:$G(IBSRC) ; MRA Only  ; MRALLA S IB=$$SETSTR^VALM1("LINE LEVEL ADJUSTMENTS:","",1,50)  D SET(IB)  I '$G(IBSRC) D  . D CNTRL^VALM10(VALMCNT,1,23,IORVON,IORVOFF)  . S ^TMP("IBCECSD",$J,"X",7)=VALMCNT  I '$D(^IBM(361.1,IBCNT,15,0)) D SET(" NONE") Q  ; only if there is info  ;  ; look up all billed data  N IBZDATA,IBFORM,IBX2,IBX3,IBREC2,IBREC3,IBTX,IBT,IBRC,IBZ,IBTXL  ;JWS;IB\*2.0\*592:Dental form #7 do same as CMS-1500  S IBFORM=0 ; cms-1500 & J430D  I $$FT^IBCEF(+IBREC)=3 S IBFORM=1 ; UB-04  ;JWS;IB\*2.0\*592:Dental form #7  D F^IBCEF("N-"\_$S(IBFORM=1:"UB-04",$$FT^IBCEF(+IBREC)=7:"J430D",1:"HCFA 1500")\_" SERVICE LINE (EDI)","IBZDATA",,+IBREC)  S IBX=0 F  S IBX=$O(^IBM(361.1,IBCNT,15,IBX)) Q:IBX<1 S IBREC1=^IBM(361.1,IBCNT,15,IBX,0) D  . NEW RVL  . D SET(" # SV DT REVCD PROC MOD UNITS BILLED DEDUCT COINS ALLOW PYMT")  . S RVL=+$P(IBREC1,U,12) ; referenced Vista line#  . I 'RVL S RVL=IBX             ; use the EOB line# if not there  . S IBT=$$RJ($P(IBREC1,"^"),3) ; line number  . S IBT=IBT\_" "\_$$RJ($$DAT1^IBOUTL($P($P(IBREC1,"^",16),".")),8) ; service date  . S IBT=IBT\_" "\_$$RJ($$EXTERNAL^DILFD(361.115,.1,"",$P(IBREC1,"^",10)),6) ; revcd  . S IBT=IBT\_" "\_$$RJ($P(IBREC1,"^",4),5) ; procedure  . S IBT=IBT\_" "\_$$RJ($P($G(^IBM(361.1,IBCNT,15,IBX,2,1,0)),"^"),3)\_$S($D(^IBM(361.1,IBCNT,15,IBX,2,2,0)):"+",1:" ") ; modifiers  . S IBT=IBT\_" "\_$$RJ($FN($P(IBREC1,"^",11),"",0),5) ; units  . ;JWS;IB\*2.0\*592:Dental form #7 do same as CMS-1500 no change, just comment  . S IBT=IBT\_" "\_$$RJ($FN($S(IBFORM:$P($G(IBZDATA(RVL)),"^",5),1:$P($G(IBZDATA(RVL)),"^",8)\*$P($G(IBZDATA(RVL)),"^",9)),"",2),8) ; billed  . S IBT=IBT\_" "\_$$RJ($FN($P($G(^IBM(361.1,IBCNT,15,IBX,1,+$O(^IBM(361.1,IBCNT,15,IBX,1,"B","PR",0)),1,+$O(^IBM(361.1,IBCNT,15,IBX,1,+$O(^IBM(361.1,IBCNT,15,IBX,1,"B","PR",0)),1,"B",1,0)),0)),"^",2),"",2),7) ; deduct  . S IBT=IBT\_" "\_$$RJ($FN($P($G(^IBM(361.1,IBCNT,15,IBX,1,+$O(^IBM(361.1,IBCNT,15,IBX,1,"B","PR",0)),1,+$O(^IBM(361.1,IBCNT,15,IBX,1,+$O(^IBM(361.1,IBCNT,15,IBX,1,"B","PR",0)),1,"B",2,0)),0)),"^",2),"",2),6) ; coins  . S IBT=IBT\_" "\_$$RJ($FN($P(IBREC1,"^",13),"",2),8) ; allow  . S IBT=IBT\_" "\_$$RJ($FN($P(IBREC1,"^",3),"",2),8) ; payment  . D SET(IBT)  . S IBX2=0 F  S IBX2=$O(^IBM(361.1,IBCNT,15,IBX,1,IBX2)) Q:IBX2<1 D  .. S IBREC2=^IBM(361.1,IBCNT,15,IBX,1,IBX2,0),IBX3=0  .. F  S IBX3=$O(^IBM(361.1,IBCNT,15,IBX,1,IBX2,1,IBX3)) Q:IBX3<1 D  ... S IBREC3=^IBM(361.1,IBCNT,15,IBX,1,IBX2,1,IBX3,0)  ... ; line level adjustments; don't display kludges (esg 10/23/03)  ... I $P(IBREC2,U,1)="PR",$P(IBREC3,U,1)="AAA" Q  ... I $P(IBREC2,U,1)="OA",$P(IBREC3,U,1)="AB3" Q  ... I $P(IBREC2,U,1)="LQ" Q  ... ; IB\*2.0\*547 - get CARC description from AR file 345, when ready  ... I '$$VFILE^DILFD(345) S IBTX(1)="ADJ: "\_$P(IBREC2,"^")\_" "\_$P(IBREC3,"^")\_" "\_$P(IBREC3,"^",4) D TXT1(.IBTX,0,79) S IBT=0 F  S IBT=$O(IBTX(IBT)) Q:IBT<1 D SET(IBTX(IBT))  ... I $$VFILE^DILFD(345) S IBT=$$CARC($P(IBREC3,"^"),345,79,"IBTX"),IBTX(1)="ADJ: "\_$P(IBREC2,"^")\_" "\_$P(IBREC3,"^")\_": "\_$G(IBTX(1)) D TXT1(.IBTX,0,79) S IBT=0 F  S IBT=$O(IBTX(IBT)) Q:IBT<1 D SET(IBTX(IBT))  ... K IBTX  ... D SET("ADJ AMT: "\_$FN($P(IBREC3,"^",2),"",2))  . S IBRC=0  . F  S IBRC=$O(^IBM(361.1,IBCNT,15,IBX,4,IBRC)) Q:'IBRC  S IBREC2=$G(^(IBRC,0)) I IBREC2 K IBTX,IBZ S IBTX(1)=" -REMARK CODE("\_+IBREC2\_"): ",IBTXL=$L(IBTX(1)) D  .. ; IB\*2.0\*547 - get RARC description from AR file 346, when ready  .. I '$$VFILE^DILFD(346) S IBTX(1)=IBTX(1)\_$P(IBREC2,U,2)\_" "\_$P(IBREC2,U,3)  .. I $$VFILE^DILFD(346) S IBT=$$CARC($P(IBREC2,U,2),346,79,"IBTX"),IBTX(1)=IBTX(1)\_$P(IBREC2,U,2)\_" "\_$G(IBT(1))  .. I $L(IBTX(1))>79 D  ... D TXT1(.IBTX,0,79) D SET(IBTX(1)) M IBZ=IBTX K IBTX S IBTX(1)="",IBT=1 F  S IBT=$O(IBZ(IBT)) Q:'IBT  S IBTX(1)=IBTX(1)\_IBZ(IBT)\_" "  .. E  D  ... S IBTXL=0  .. D TXT1(.IBTX,IBTXL,79) S IBT=0 F  S IBT=$O(IBTX(IBT)) Q:IBT<1 D SET(IBTX(IBT))  . D SET(" ")  D SET(" ")  Q  ;  ;/Beginning IB\*2.0\*488 (vd) EOBERR ; Display information about any 361.1 message storage or filing errors  N ERRTXT,DASHES,Z  S DASHES="---------------------------------------------------------------------"  I '$O(^IBM(361.1,IBCNT,"ERR",0)) Q  D SET("VistA could not match all of the Line Level data received in the EEOB")  D SET("(835 Record 40) to the claim in VistA.")  D SET(" ")  S Z=0 F  S Z=$O(^IBM(361.1,IBCNT,"ERR",Z)) Q:'Z  D  .S ERRTXT=$G(^IBM(361.1,IBCNT,"ERR",Z,0))  .I ERRTXT["##RAW DATA" S ERRTXT=DASHES  .D SET(ERRTXT)  Q  ;/End of IB\*2.0\*488 (vd)  ; TXT(IBRM,IBLN,IBXY) ;display text over 79 chars  ;IBRM - text, IBLN - length, IBXY - position  S IBRM=$E(IBRM,IBLN+1,999) REP I $E(IBRM,1,IBLN)'="" S IB=$$SETSTR^VALM1($E(IBRM,1,IBLN),"",IBXY,IBLN) D SET(IB) S IBRM=$E(IBRM,IBLN+1,999) G REP  Q  ; SET(IB,IBSAV) ;  I '$G(IBSAV) D SET^IBCECSA6($G(IBSRC),IB,+$G(CNT),IBCNT)  Q  ; A10(X) ;  Q $$A10^IBCECSA6(X)  ; A7(X) ; returns a dollar amount right justified to 7 characters  Q $$RJ($FN(X,"",2),7)  ; TXT1(IBT,DIWL,DIWR) ; sets up text for over 79 chars  ; IBT - pass by ref, array of text to be formatted back in array  ; DIWL - left margin, DIWR = right margin  N IBX,X,DIWF,IBS K ^UTILITY($J,"W")  S DIWF="|I"\_DIWL  S IBX=0 F  S IBX=$O(IBT(IBX)) Q:IBX<1 S X=IBT(IBX) D ^DIWP  K IBT F  S IBX=$O(^UTILITY($J,"W",DIWL,IBX)) Q:IBX<1 S IBT(IBX)=^UTILITY($J,"W",DIWL,IBX,0)  K ^UTILITY($J,"W")  Q  ; RJ(X,Y) ; right just, default is 10  Q $$RJ^XLFSTR(X,$G(Y,10)," ")  ; CARC(IBCDE,IBF,IBML,IBARY) ;new CARC/RACR API for IB\*2.0\*547  ; IBCDE = reason code from EOB to lookup in carc/rarc file (REQUIRED)  ; IBF = file# to do lookup (either 345-CARC or 346-RARC) \*REQUIRED\*  ; IBML = max length for each line (default is 79)  ; IBARY = (required) subscripted array to return description data in:  ; array(1)=first line of word-processed description  ; array(2)= 2nd line of wp description, and so on  ;  ; Returns total # of lines in description  ;  N IBY,IBX,IBC,IBI,IBN,IBALN,IBSTP,IBDSC  S IBC=0  Q:$G(IBARY)="" IBC  Q:$G(IBCDE)="" IBC  Q:$G(IBF)="" IBC  S:$G(IBML)="" IBML=79  S IBY=$$FIND1^DIC(IBF,,"BX",IBCDE) Q:IBY<1 IBC  S IBX=$$GET1^DIQ(IBF,IBY\_",",4,"","IBDSC")  S IBI=0 F  S IBI=$O(IBDSC(IBI)) Q:'IBI  D  .S IBC=IBC+1,IBSTP=0,IBALN=$L(IBDSC(IBI))  .S @IBARY@(IBI)=$E(IBDSC(IBI),1,IBML) Q:IBML>IBALN  .S IBDSC(IBI+1)=($E(IBDSC(IBI),(IBML+1),IBALN)\_" "\_$G(IBDSC(IBI+1)))  .; make sure we don't break words in 2  .Q:$E(@IBARY@(IBI),IBML)=" "  .F IBN=IBML:-1:1 Q:$G(IBSTP)=1 D  ..Q:$E(IBDSC(IBI),IBN)'=" "   ..S @IBARY@(IBI)=$E(IBDSC(IBI),1,IBN),IBDSC(IBI+1)=($E(IBDSC(IBI),(IBN+1),IBML)\_$G(IBDSC(IBI+1))),IBSTP=1 Q  Q IBC  ; | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCEDP | | | | | | | | |
| **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** | | | | | | | | | |
| IBCEDP ;ALB/ESG - EDI CLAIM STATUS REPORT PRINT ;13-DEC-2007  ;;2.0;INTEGRATED BILLING;\*\*377\*\*;21-MAR-94;Build 23  ;;Per VHA Directive 2004-038, this routine should not be modified.  ;  Q  ; PRINT ; entry point to print the report  NEW CRT,IBPAGE,IBSTOP,IBCT,SV1,SV2,SV3,IEN,DATA,NEWHDR  NEW DIR,X,Y,DTOUT,DUOUT,DIROUT,DIRUT  I IOST["C-" S CRT=1  E  S CRT=0  ;  S IBPAGE=0,IBSTOP=0,IBCT=0,NEWHDR=0  ;  I '$D(^TMP($J,"IBCEDC")) D HDR W !!?5,"No data found for this report." G PX  I $G(ZTSTOP) D HDR W !!?5,"This report was halted during compilation by TaskManager Request." G PX  ;  D HDR   ; initial header display  S SV1=""  F  S SV1=$O(^TMP($J,"IBCEDC",SV1)) Q:SV1=""!IBSTOP  D SD(SV1) D  Q:IBSTOP  . S SV2=""  . F  S SV2=$O(^TMP($J,"IBCEDC",SV1,SV2)) Q:SV2=""!IBSTOP  D  Q:IBSTOP  .. S SV3=""  .. F  S SV3=$O(^TMP($J,"IBCEDC",SV1,SV2,SV3)) Q:SV3=""!IBSTOP  D  Q:IBSTOP  ... S IEN=0  ... F  S IEN=$O(^TMP($J,"IBCEDC",SV1,SV2,SV3,IEN)) Q:'IEN!IBSTOP  D  Q:IBSTOP  .... S DATA=$G(^TMP($J,"IBCEDC",SV1,SV2,SV3,IEN))  .... D PRT(DATA)  .... Q  ... Q  .. Q  . Q  ;  I IBSTOP G PRINTX  D:$Y>(IOSL-4) HDR G:IBSTOP PRINTX  W !!?5,"Total number of EDI Claims: ",IBCT  D:$Y>(IOSL-4) HDR G:IBSTOP PRINTX  W !!,"\*\*\* End of Report \*\*\*"  ; PX ;  I CRT,'$D(ZTQUEUED) S DIR(0)="E" D ^DIR K DIR PRINTX ;  Q  ; PRT(Z) ; print a line on the report  ; Z - data from the scratch global node  N DIV,PAY,ADDR1  D:$Y>(IOSL-3) HDR G:IBSTOP PRTX  S IBCT=IBCT+1  S DIV=$P($G(^DG(40.8,+$P(Z,U,10),0)),U,2) ; division abbr  S PAY=$P($G(^DIC(36,+$P(Z,U,12),0)),U,1) ; payer name  S ADDR1=$P($G(^DIC(36,+$P(Z,U,12),.11)),U,1) ; payer address line 1  ;  W !,$P(Z,U,1) ; claim#  W ?9,$S($P(Z,U,2)=2:1500,1:"UB04") ; form type  W ?14,$S($P(Z,U,3):"INPT",1:"OUTPT") ; inpat/outpat  W ?21,$P(Z,U,4) ; payer sequence  W ?25,$P(Z,U,5) ; EDI status code  W ?29,$E($P(Z,U,13),1,9) ; IB status abbr  W ?39,$E($P(Z,U,11),1,2) ; ar status abbr  W ?44,$$FMTE^XLFDT($P(Z,U,6)\1,"2Z") ; last transmit date  W ?55,$J($P(Z,U,7),4) ; age in days  W ?62,$P(Z,U,8) ; batch#  W ?69,$J($FN($P(Z,U,9),"",2),9) ; balance due  W ?81,DIV                                                ; division  W ?89,$E(PAY,1,23) ; payer name  W ?114,$E(ADDR1,1,18) ; payer address line 1  ;  S NEWHDR=0 ; toggle new header flag PRTX ;  Q  ; HDR ; report header  ;  ; if screen output and page# already exists, do a page break at the bottom of the screen  I IBPAGE,CRT D  I IBSTOP G HDRX  . S DIR(0)="E" D ^DIR K DIR  . I 'Y S IBSTOP=1  . Q  ;  ; if screen output OR page# already exists, do a form feed  I IBPAGE!CRT W @IOF  I 'IBPAGE,'CRT W $C(13) ; first printer page - left margin set  ;  S IBPAGE=IBPAGE+1  ;  W "EDI Claim Status Report",?96,$$FMTE^XLFDT($$NOW^XLFDT)," Page: ",IBPAGE  W !,"\*\* A claim may appear multiple times if transmitted more than once. \*\*"  W !?3,"Sorted by ",$$SD^IBCEDS1($G(IBSORT1))  I $G(IBSORT2)'="" W ", then by ",$$SD^IBCEDS1(IBSORT2)  I $G(IBSORT3)'="" W ", then by ",$$SD^IBCEDS1(IBSORT3)  ;  ; display column headers  W !?25,"\*-- Statuses --\*"  W !,"Claim",?9,"Form",?14,"Type",?20,"Seq",?25,"EDI",?31,"IB",?39,"AR",?44,"Trans Dt",?56,"Age",?62,"Batch#",?71,"Bal Due"  W ?81,"Div",?89,"Payer"  ;  N Z S Z="",$P(Z,"-",133)="" W !,Z  ;  S NEWHDR=1 ; flag indicating a new page header was just printed  ;  ; check for a TaskManager stop request  I $D(ZTQUEUED),$$S^%ZTLOAD() D  G HDRX  . S (ZTSTOP,IBSTOP)=1  . W !!!?5,"\*\*\* Report Halted by TaskManager Request \*\*\*"  . Q  ; HDRX ;  Q  ; SD(SV) ; primary sort value display break. This procedure is to display a break whenever the primary sort value changes  ; SV - subscript value of the primary sort  I IBSORT1=4!(IBSORT1=6) G SDX  ; don't display a break for current balance or for claim# primary sorts  ;  D:$Y>(IOSL-4) HDR G:IBSTOP SDX  I 'NEWHDR W ! ; an extra line break if a page header was not just printed  I $E(SV)="-",$D(IBSORTOR(IBSORT1)) S SV=$E(SV,2,999) ; remove leading "-" on descending numerical sorts  ;  I IBSORT1=1 S SV=$$FMTE^XLFDT(SV,"5Z") ; last transmitted date/time  I IBSORT1=2 D                            ; payer name and address  . N INS,ADDR  . S INS=+$P(SV,U,2) ; ins co ien 2nd piece of subscript  . S ADDR=$$INSADD^IBCNSC02(INS) ; address fields  . S SV=$P(SV,U,1)\_" "\_$P(ADDR,U,2)\_" "\_$P(ADDR,U,6)\_" "\_$P(ADDR,U,5)  . Q  I IBSORT1=3 S SV=SV\_" - "\_$$EXTERNAL^DILFD(364,.03,,SV) ; edi claim status and description  I IBSORT1=5 D                                             ; division  . N DZ,DIVNM  . S DZ=+$O(^DG(40.8,"C",SV,"")) ; division ien  . S DIVNM=$P($G(^DG(40.8,DZ,0)),U,1) ; division name  . S SV=SV\_" - "\_DIVNM  . Q  I IBSORT1=7 D                                             ; AR status  . N AZ,ANM  . S AZ=+$O(^PRCA(430.3,"C",SV,"")) ; AR status ien  . S ANM=$P($G(^PRCA(430.3,AZ,0)),U,1) ; AR status description  . S SV=SV\_" - "\_ANM  . Q  I IBSORT1=8 S SV=SV\_" Days"  ;  S SV=$$SD^IBCEDS1(IBSORT1)\_": "\_SV  W !,SV SDX ;  Q  ; | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCEDP ;ALB/ESG - EDI CLAIM STATUS REPORT PRINT ;13-DEC-2007  ;;2.0;INTEGRATED BILLING;\*\*377,592\*\*;21-MAR-94;Build 23  ;;Per VHA Directive 2004-038, this routine should not be modified.  ;  Q  ; PRINT ; entry point to print the report  NEW CRT,IBPAGE,IBSTOP,IBCT,SV1,SV2,SV3,IEN,DATA,NEWHDR  NEW DIR,X,Y,DTOUT,DUOUT,DIROUT,DIRUT  I IOST["C-" S CRT=1  E  S CRT=0  ;  S IBPAGE=0,IBSTOP=0,IBCT=0,NEWHDR=0  ;  I '$D(^TMP($J,"IBCEDC")) D HDR W !!?5,"No data found for this report." G PX  I $G(ZTSTOP) D HDR W !!?5,"This report was halted during compilation by TaskManager Request." G PX  ;  D HDR   ; initial header display  S SV1=""  F  S SV1=$O(^TMP($J,"IBCEDC",SV1)) Q:SV1=""!IBSTOP  D SD(SV1) D  Q:IBSTOP  . S SV2=""  . F  S SV2=$O(^TMP($J,"IBCEDC",SV1,SV2)) Q:SV2=""!IBSTOP  D  Q:IBSTOP  .. S SV3=""  .. F  S SV3=$O(^TMP($J,"IBCEDC",SV1,SV2,SV3)) Q:SV3=""!IBSTOP  D  Q:IBSTOP  ... S IEN=0  ... F  S IEN=$O(^TMP($J,"IBCEDC",SV1,SV2,SV3,IEN)) Q:'IEN!IBSTOP  D  Q:IBSTOP  .... S DATA=$G(^TMP($J,"IBCEDC",SV1,SV2,SV3,IEN))  .... D PRT(DATA)  .... Q  ... Q  .. Q  . Q  ;  I IBSTOP G PRINTX  D:$Y>(IOSL-4) HDR G:IBSTOP PRINTX  W !!?5,"Total number of EDI Claims: ",IBCT  D:$Y>(IOSL-4) HDR G:IBSTOP PRINTX  W !!,"\*\*\* End of Report \*\*\*"  ; PX ;  I CRT,'$D(ZTQUEUED) S DIR(0)="E" D ^DIR K DIR PRINTX ;  Q  ; PRT(Z) ; print a line on the report  ; Z - data from the scratch global node  N DIV,PAY,ADDR1,TAB  ;JRA IB\*2.0\*592 Added TAB  D:$Y>(IOSL-3) HDR G:IBSTOP PRTX  S IBCT=IBCT+1  S DIV=$P($G(^DG(40.8,+$P(Z,U,10),0)),U,2) ; division abbr  S PAY=$P($G(^DIC(36,+$P(Z,U,12),0)),U,1) ; payer name  S ADDR1=$P($G(^DIC(36,+$P(Z,U,12),.11)),U,1) ; payer address line 1  ;  W !,$P(Z,U,1) ; claim#  ;JRA IB\*2\*592 Add Condition for Dental Form Type 7  ;W ?9,$S($P(Z,U,2)=2:1500,1:"UB04") ; form type ;JRA IB\*2.0\*592 ';'  ;JRA IB\*2.0\*592 Dental Form Type is 5 chars vs. 4, so set TAB accordingly  S TAB=$S($P(Z,U,2)=7:8,1:9) ; Set tab per form type ;JRA IB\*2.0\*592  W ?TAB,$S($P(Z,U,2)=2:"1500",$P(Z,U,2)=7:"J430D",1:"UB04") ; form type ;JRA IB\*2.0\*592  W ?14,$S($P(Z,U,3):"INPT",1:"OUTPT") ; inpat/outpat  W ?21,$P(Z,U,4) ; payer sequence  W ?25,$P(Z,U,5) ; EDI status code  W ?29,$E($P(Z,U,13),1,9) ; IB status abbr  W ?39,$E($P(Z,U,11),1,2) ; ar status abbr  W ?44,$$FMTE^XLFDT($P(Z,U,6)\1,"2Z") ; last transmit date  W ?55,$J($P(Z,U,7),4) ; age in days  W ?62,$P(Z,U,8) ; batch#  W ?69,$J($FN($P(Z,U,9),"",2),9) ; balance due  W ?81,DIV                                                ; division  W ?89,$E(PAY,1,23) ; payer name  W ?114,$E(ADDR1,1,18) ; payer address line 1  ;  S NEWHDR=0 ; toggle new header flag PRTX ;  Q  ; HDR ; report header  ;  ; if screen output and page# already exists, do a page break at the bottom of the screen  I IBPAGE,CRT D  I IBSTOP G HDRX  . S DIR(0)="E" D ^DIR K DIR  . I 'Y S IBSTOP=1  . Q  ;  ; if screen output OR page# already exists, do a form feed  I IBPAGE!CRT W @IOF  I 'IBPAGE,'CRT W $C(13) ; first printer page - left margin set  ;  S IBPAGE=IBPAGE+1  ;  W "EDI Claim Status Report",?96,$$FMTE^XLFDT($$NOW^XLFDT)," Page: ",IBPAGE  W !,"\*\* A claim may appear multiple times if transmitted more than once. \*\*"  W !?3,"Sorted by ",$$SD^IBCEDS1($G(IBSORT1))  I $G(IBSORT2)'="" W ", then by ",$$SD^IBCEDS1(IBSORT2)  I $G(IBSORT3)'="" W ", then by ",$$SD^IBCEDS1(IBSORT3)  ;  ; display column headers  W !?25,"\*-- Statuses --\*"  W !,"Claim",?9,"Form",?14,"Type",?20,"Seq",?25,"EDI",?31,"IB",?39,"AR",?44,"Trans Dt",?56,"Age",?62,"Batch#",?71,"Bal Due"  W ?81,"Div",?89,"Payer"  ;  N Z S Z="",$P(Z,"-",133)="" W !,Z  ;  S NEWHDR=1 ; flag indicating a new page header was just printed  ;  ; check for a TaskManager stop request  I $D(ZTQUEUED),$$S^%ZTLOAD() D  G HDRX  . S (ZTSTOP,IBSTOP)=1  . W !!!?5,"\*\*\* Report Halted by TaskManager Request \*\*\*"  . Q  ; HDRX ;  Q  ; SD(SV) ; primary sort value display break. This procedure is to display a break whenever the primary sort value changes  ; SV - subscript value of the primary sort  I IBSORT1=4!(IBSORT1=6) G SDX  ; don't display a break for current balance or for claim# primary sorts  ;  D:$Y>(IOSL-4) HDR G:IBSTOP SDX  I 'NEWHDR W ! ; an extra line break if a page header was not just printed  I $E(SV)="-",$D(IBSORTOR(IBSORT1)) S SV=$E(SV,2,999) ; remove leading "-" on descending numerical sorts  ;  I IBSORT1=1 S SV=$$FMTE^XLFDT(SV,"5Z") ; last transmitted date/time  I IBSORT1=2 D                            ; payer name and address  . N INS,ADDR  . S INS=+$P(SV,U,2) ; ins co ien 2nd piece of subscript  . S ADDR=$$INSADD^IBCNSC02(INS) ; address fields  . S SV=$P(SV,U,1)\_" "\_$P(ADDR,U,2)\_" "\_$P(ADDR,U,6)\_" "\_$P(ADDR,U,5)  . Q  I IBSORT1=3 S SV=SV\_" - "\_$$EXTERNAL^DILFD(364,.03,,SV) ; edi claim status and description  I IBSORT1=5 D                                             ; division  . N DZ,DIVNM  . S DZ=+$O(^DG(40.8,"C",SV,"")) ; division ien  . S DIVNM=$P($G(^DG(40.8,DZ,0)),U,1) ; division name  . S SV=SV\_" - "\_DIVNM  . Q  I IBSORT1=7 D                                             ; AR status  . N AZ,ANM  . S AZ=+$O(^PRCA(430.3,"C",SV,"")) ; AR status ien  . S ANM=$P($G(^PRCA(430.3,AZ,0)),U,1) ; AR status description  . S SV=SV\_" - "\_ANM  . Q  I IBSORT1=8 S SV=SV\_" Days"  ;  S SV=$$SD^IBCEDS1(IBSORT1)\_": "\_SV  W !,SV SDX ;  Q  ; | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCEF3 | | | | | | | | |
| **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** | | | | | | | | | |
| IBCEF3 ;ALB/TMP - FORMATTER SPECIFIC BILL FLD FUNCTIONS ;17-JUNE-96  ;;2.0;INTEGRATED BILLING;\*\*52,84,121,51,152,210,155,348,349,389,488,516\*\*;21-MAR-94;Build 123  ;;Per VA Directive 6402, this routine should not be modified.  ; MPG(PG,FLDS,FORM) ; Set static flds on pages after page 1  ; for either 1500 or UB  ; PG = page #  ; FORM= 1 for UB, otherwise for 1500  ; FLDS: array passed by reference and containing lines OR  ; line/column from pg 1 to repeat on subsequent pages  ; Format: FLDS(LINE,COL) or FLDS(LINE) for whole line  ; CMS-1500: LINES 1-5,7-43,57 from col 1 to 50, 58-63  ; UB: see CKPGUB for lines and columns  ;  N Z,Z0,Z1,LPG  S FORM=$S($G(FORM)=1:3,1:2)  I FORM=2 D  ; print page # on each pg, totals on last page of 1500  . S LPG=+$O(^TMP("IBXDATA",$J,IBXREC,""),-1)  . S Z="[Page "\_PG\_" of "\_LPG\_"]"  . S Z=$$FO^IBCNEUT1(Z,17,"R")  . D SETGBL^IBCEFG(PG,6,61,Z,.IBXSIZE)  . I PG=2 S Z=$P(Z,"[",1)\_"[Page 1 of "\_LPG\_"]" D SETGBL^IBCEFG(1,6,61,Z,.IBXSIZE)  . I LPG=PG D  .. ;  .. ; esg - IB\*2\*348 - update dollar format for last page of 1500  .. ;  .. D SETGBL^IBCEFG(PG,57,51,$$DOL^IBCEF77($G(IBXSAVE("TOT")),9),.IBXSIZE)  .. D SETGBL^IBCEFG(PG,57,62,$$DOL^IBCEF77($G(IBXSAVE("PAID")),8),.IBXSIZE)  .. ;IB\*2.0\*516/DRF - Blank Box 30 on last page of multi-page claims  .. ;D SETGBL^IBCEFG(PG,57,71,$$DOL^IBCEF77($G(IBXSAVE("BDUE")),8),.IBXSIZE)  .. K IBXSAVE("PTOT"),IBXSAVE("TOT"),IBXSAVE("BDUE"),IBXSAVE("PAID")  ;  S Z=0 F  S Z=$O(FLDS(Z)) Q:'Z  D  . I $O(FLDS(Z,""))="" D  Q  ;repeats line  .. S Z0=0 F  S Z0=$O(^TMP("IBXDATA",$J,IBXREC,1,Z,Z0)) Q:'Z0  S Z1=$G(^(Z0)) I Z1'="" D SETGBL^IBCEFG(PG,Z,Z0,Z1,.IBXSIZE)  . S Z0=0 F  S Z0=$O(FLDS(Z,Z0)) Q:'Z0  S Z1=$G(^TMP("IBXDATA",$J,IBXREC,1,Z,Z0)) I Z1'="" D SETGBL^IBCEFG(PG,Z,Z0,Z1,.IBXSIZE)  . I FORM=2,LPG'=PG D  .. D SETGBL^IBCEFG(PG,57,51,"",.IBXSIZE)  .. D SETGBL^IBCEFG(PG,57,71,"",.IBXSIZE)  Q  ; NONSERV(Z,Z0) ; Set variable if non-service/non-text data is present for box  ; 24 of CMS-1500  ; Z = sequence of IBXSAVE being processed  ; Z0 = sequnce within IBXDATA to indicate actual line #  I $P(IBXSAVE("BOX24",Z),U)="" S IBXSAVE("NON-SERV",Z0)=""  Q  ; PG(VAL,LNCT) ;Set next pg for CMS-1500 lines  ;VAL = value of fld  ;LNCT = line # from IBXSAVE("BOX24") array  N IBP,IBL  S IBP=LNCT\12+(LNCT#12>0),IBL=LNCT-(12\*(IBP-1))-1  I IBL'<0 S VAL=$$FORMAT(VAL,$G(IBXLOOP("IBX0")),$G(IBXDA)) D SETGBL^IBCEFG(IBP,IBXLN+IBL,IBXCOL,VAL,.IBXSIZE)  K IBXDATA(LNCT)  Q  ; MPGUB(PG,OFFSET,VAL,IBLN,IBCOL,NOFORM) ; Set up pages > 1 for UB overflows  ; PG = Page # to set (REQUIRED)  ; OFFSET = offset from first line this should be extracted into  ; 0 = first line (REQUIRED)  ; VAL = value to set (REQUIRED)  ; IBLN = line to set data at (if null, uses IBXLN)  ; IBCOL = column to set data at (if null, uses IBXCOL)  ; NOFORM = don't format, just output data as passed  ; Assumes formatter IBXLN,IBXCOL variables exist  ;  I $G(IBLN)="" S IBLN=IBXLN  I $G(IBCOL)="" S IBCOL=IBXCOL  S:'$G(NOFORM) VAL=$$FORMAT(VAL,$G(IBXLOOP("IBX0")),$G(IBXDA))  D SETGBL^IBCEFG(PG,IBLN+OFFSET,IBCOL,VAL,.IBXSIZE)  Q  ; CKREV(CT,VAL) ; Check too many rev code lines to fit on page  ; This procedure is only called when CT>22 (i.e. 23 or more)  ;  D MPGUB((CT-1)\22+1,CT-1#22,VAL) ; 22 codes on a single page  Q  ; CKPGUB ; Check to see if multiple UB pages are needed then populate  ; static flds from page 1, add page numbers  ;  N FLDS,LPG,IBPG,IBP,Z,Z0,TOT1,TOT2  ;  S LPG=$O(^TMP("IBXDATA",$J,IBXREC,""),-1),IBP=0  S Z="" F  S Z=$O(^TMP("IBXDATA",$J,IBXREC,LPG,Z),-1) Q:'Z  S Z0=0 F  S Z0=$O(^TMP("IBXDATA",$J,IBXREC,LPG,Z,Z0)) Q:'Z0  I $G(^(Z0))'="" S IBP=1 Q  I 'IBP K ^TMP("IBXDATA",$J,IBXREC,LPG) S LPG=$O(^TMP("IBXDATA",$J,IBXREC,""),-1) Q:LPG=1  ;  ; Static flds  F Z=2:1:7 S FLDS(Z)=""    ; FL-1 thru FL-9  F Z=1,10,13,19,22,25,28,31 S FLDS(9,Z)=""    ; FL-10 thru FL-17  F Z=13:1:17 S FLDS(Z,1)=""   ; payer address in FL-38  S FLDS(41,46)=""     ; creation date  F Z=42,43,44,45,47,48,49,51,52,53 S FLDS(Z)=""   ; FL-50 thru FL-65  F Z=57,59,61,63 S (FLDS(Z,59),FLDS(Z,72),FLDS(Z,74))=""   ; FL-76-79 ID's  F Z=58,60,62,64 S (FLDS(Z,53),FLDS(Z,71))=""    ; FL-76-79 Names  ;  F IBPG=1:1:LPG D  . ; Add pg # to last line of rev codes if multiple pages  . N IB,IBP  . S IB=$G(^TMP("IBXDATA",$J,IBXREC,IBPG,41,6))  . D MPGUB(IBPG,0,IBPG,41,10,1)  . D MPGUB(IBPG,0,LPG,41,16,1)  . D:IBPG>1 MPG(IBPG,.FLDS,1)  . Q  ; print totals on line 41 of the last page  S (TOT1,TOT2)=0  F Z=1:1 Q:'$D(^TMP($J,"IBC-RC",Z)) S Z0=^(Z) I +Z0=1 S TOT1=TOT1+$P(Z0,U,7),TOT2=TOT2+$P(Z0,U,8)  ; Make sure totals are only 9 digits => baa IB\*2.0\*488  S TOT1=$$DOL^IBCEF77(TOT1,9)  S TOT1=$E(TOT1,1,9)  S TOT2=$$DOL^IBCEF77(TOT2,9)  S TOT2=$E(TOT2,1,9)  D MPGUB(IBPG,0,"0001",41,1,1)  D MPGUB(IBPG,0,TOT1,41,61,1)  D MPGUB(IBPG,0,TOT2,41,71,1)  ;End changes => baa IB\*2.0\*488  Q  ; HCPC(R) ;FORMAT HCPC fld FOR UB (returns formatted value)  ; R = flag for type of fld (1/2/3) being printed in rev code block  Q R  ;No longer used as of patch IB\*2.0\*51  ; PROS(IBIFN) ; Extract billable prosthetics for 837  N IBARRAY,Z,Z0,CT,PROS  D SET^IBCSC5B(IBIFN,.IBARRAY)  I '$P(IBARRAY,U,2) S CT="" G PROSQ  S Z="",CT=0  F  S Z=$O(IBARRAY(Z)) Q:Z=""  S Z0="" F  S Z0=$O(IBARRAY(Z,Z0)) Q:Z0=""  S CT=CT+1 D  .S PROS=$$PINB^IBCSC5B(+IBARRAY(Z,Z0)) ; P389 removed p2 - item ptr file 661  .;date^^short descr^entry # in file 362.5  .S IBXDATA(CT)=Z\_U\_U\_PROS\_U\_+IBARRAY(Z,Z0) PROSQ Q CT  ; B24(IBXSV,IBIFN,IBNOSHOW) ; Code to execute to set up IBXSV("BOX24") for  ; print or IBXSAVE("OUTPT") for transmit - called by output formatter  ; IBNOSHOW = 1 if not to show error/warning text lines  ; Pass IBXSV by reference  N IBSUB  S IBSUB=$S('$G(^TMP("IBTX",$J,IBIFN)):"BOX24",1:"OUTPT")  K IBXSV(IBSUB)  I '$D(IBIFN) S IBIFN=$G(IBXIEN)  I IBIFN D F^IBCEF("N-HCFA 1500 SERVICE"\_$S(IBSUB["24":"S (PRINT",1:" LINE (EDI")\_")",,,IBIFN)  I $S(IBSUB'["24":1,1:'$G(IBNOSHOW)) D  . M IBXSV(IBSUB)=IBXDATA  E  D  . N Z,CT  . S (Z,CT)=0 F  S Z=$O(IBXDATA(Z)) Q:'Z  I '$D(IBXDATA(Z,"ARX")) S CT=CT+1 M IBXSV(IBSUB,CT)=IBXDATA(Z)  Q  ;  ; esg - 11/14/03 - Moved the below functions due to space constraints  ; ALLTYP(IBIFN) Q $$ALLTYP^IBCEF31(IBIFN) INSTYP(IBIFN,SEQ) Q $$INSTYP^IBCEF31(IBIFN,$G(SEQ)) POLTYP(IBIFN,IBSEQ) Q $$POLTYP^IBCEF31(IBIFN,$G(IBSEQ)) ALLPTYP(IBIFN) Q $$ALLPTYP^IBCEF31(IBIFN)  ; FILL(Z) ;  Q  ;  ; \*\*\*\*\*   ; The following code performs the multi-page set up for  ; printing overflow data on the UB  ; \*\*\*\*\*  ; XPROC(DATA,CT) ; Output any UB procedures after 6 on new page(s)  ; DATA = output data from IBXSAVE("PROC",CT)  ; CT = array sequence # of the procedure being output  ; Only used for local prints  N OFFSET,PG,COL,PRCODE,Q  S Q=(CT-1)\3#2,OFFSET=$S('Q:0,1:2)  S PG=(CT-1)\6+1,COL=1+(CT-1#3\*15)  D MPGUB(PG,OFFSET,$P(DATA,U,1),58,COL)  D MPGUB(PG,OFFSET,$P(DATA,U,2),58,COL+9)  Q  ; XDIAG(DATA,CT) ; Output any UB other diagnoses after 8 on new page(s)  ; DATA = output data from IBXSAVE("DX",CT)  ; CT = array sequence # of the diagnosis being output  ; Only used for local prints  N COL,PG  S PG=(CT-1)\8+1,COL=8+(CT-1#9\*7)  S DATA=$P($$ICD9^IBACSV(+DATA),U,1)  D MPGUB(PG,0,DATA,56,COL)  Q  ; XVAL(DATA,CT) ; Output any UB value codes after 12 on new page(s)  ; DATA = output data from IBXSAVE("VC",CT)  ; CT = array sequence # of the value code being output  ;  N COL,PG,OFFSET  S PG=(CT-1)\12+1,COL=44+(CT-1#3\*13),OFFSET=(CT-(12\*(PG-1))-1)\3  D MPGUB(PG,OFFSET,$P(DATA,U,1),14,COL)  D MPGUB(PG,OFFSET,$P(DATA,U,2),14,COL+3)  Q  ; XCC(DATA,CT) ; Output any UB condition codes after 11 on new page(s)  ; 11 condition codes per page, starting columns 34 thru 64  ; DATA = output data from IBXSAVE("CC",CT)  ; CT = array sequence # of the condition code being output  ;  N COL,PG  S PG=(CT-1)\11+1,COL=34+(CT-1#11\*3)  D MPGUB(PG,0,DATA,9,COL)  Q  ; XOCC(DATA,CT,FL) ; Output any UB occurrence codes after 8 (2 per form  ; locators 31-34) on new page(s)  ; DATA = data from IBXSAVE("OCC",z) to be output  ; CT = array sequence # of occurrence code being output  ; FL = # of form locator being populated with the occ code  ;  N COL,PG,OFFSET  S PG=(CT-1)\2+1,COL=1+((FL-31)\*10),OFFSET=$S(CT#2:0,1:1)  D MPGUB(PG,OFFSET,$P(DATA,U,1),11,COL)  D MPGUB(PG,OFFSET,$P(DATA,U,2),11,COL+4)  Q  ; XOCCS(DATA,CT,FL) ; Output any UB occurrence span codes after 4 on new page(s)  ; DATA = data from IBXSAVE("OCCS",z) to be output  ; CT = array sequence # of occurrence span code being output  ; FL = # of form locator being populated (either FL 35 or 36)  ;  N COL,PG,OFFSET  S PG=(CT-1)\2+1,OFFSET=$S(CT#2:0,1:1)  S COL=41+((FL-35)\*17)  D MPGUB(PG,OFFSET,$P(DATA,U,1),11,COL)  D MPGUB(PG,OFFSET,$P(DATA,U,2),11,COL+4)  D MPGUB(PG,OFFSET,$P(DATA,U,3),11,COL+11)  Q  ; FORMAT(VAL,IBX0,IBXDA) ;  I IBX0'="",IBXDA S VAL=$$FORMAT^IBCEFG(VAL,$P($G(^IBA(364.6,+IBXDA,0)),U,9),$P(IBX0,U,7),IBX0)  Q VAL  ; OUTPDT(IBIFN,IBXSAVE,IBXDATA) ; Returns outpatient service to date  ; formatted CCYYMMDD for UB 837  ; IBIFN = ien of bill (file 399)  ; IBXSAVE = pass by reference for IBXSAVE("INPT") and IBXSAVE("DATE")  ; IBXDATA = array with formatted date or each line item - CCYYMMDD  N Z  S Z=0 F  S Z=$O(IBXSAVE("INPT",Z)) Q:'Z  S IBXDATA(Z)=$S($P(IBXSAVE("INPT",Z),U,10):$$DT^IBCEFG1($P(IBXSAVE("INPT",Z),U,10),,"D8"),1:IBXSAVE("DATE"))  K IBXSAVE("DATE")  Q  ; | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCEF3 ;ALB/TMP - FORMATTER SPECIFIC BILL FLD FUNCTIONS ;17-JUNE-96  ;;2.0;INTEGRATED BILLING;\*\*52,84,121,51,152,210,155,348,349,389,488,516,592\*\*;21-MAR-94;Build 123  ;;Per VA Directive 6402, this routine should not be modified.  ; MPG(PG,FLDS,FORM) ; Set static flds on pages after page 1  ; for either 1500 or UB  ; PG = page #  ; FORM= 1 for UB, otherwise for 1500  ; FLDS: array passed by reference and containing lines OR  ; line/column from pg 1 to repeat on subsequent pages  ; Format: FLDS(LINE,COL) or FLDS(LINE) for whole line  ; CMS-1500: LINES 1-5,7-43,57 from col 1 to 50, 58-63  ; UB: see CKPGUB for lines and columns  ;  N Z,Z0,Z1,LPG  S FORM=$S($G(FORM)=1:3,1:2)  ;JRA IB\*2.0\*592 Treat Dental Form 7 (J430D) same as the 1500  ;I FORM=2 D ; print page # on each pg, totals on last page of 1500 ;JRA IB\*2.0\*592 ';'  I FORM=2!(FORM=7) D  ; print page # on each pg, totals on last page of 1500 (or J430D) ;JRA IB\*2.0\*592  . S LPG=+$O(^TMP("IBXDATA",$J,IBXREC,""),-1)  . S Z="[Page "\_PG\_" of "\_LPG\_"]"  . S Z=$$FO^IBCNEUT1(Z,17,"R")  . D SETGBL^IBCEFG(PG,6,61,Z,.IBXSIZE)  . I PG=2 S Z=$P(Z,"[",1)\_"[Page 1 of "\_LPG\_"]" D SETGBL^IBCEFG(1,6,61,Z,.IBXSIZE)  . I LPG=PG D  .. ;  .. ; esg - IB\*2\*348 - update dollar format for last page of 1500  .. ;  .. D SETGBL^IBCEFG(PG,57,51,$$DOL^IBCEF77($G(IBXSAVE("TOT")),9),.IBXSIZE)  .. D SETGBL^IBCEFG(PG,57,62,$$DOL^IBCEF77($G(IBXSAVE("PAID")),8),.IBXSIZE)  .. ;IB\*2.0\*516/DRF - Blank Box 30 on last page of multi-page claims  .. ;D SETGBL^IBCEFG(PG,57,71,$$DOL^IBCEF77($G(IBXSAVE("BDUE")),8),.IBXSIZE)  .. K IBXSAVE("PTOT"),IBXSAVE("TOT"),IBXSAVE("BDUE"),IBXSAVE("PAID")  ;  S Z=0 F  S Z=$O(FLDS(Z)) Q:'Z  D  . I $O(FLDS(Z,""))="" D  Q  ;repeats line  .. S Z0=0 F  S Z0=$O(^TMP("IBXDATA",$J,IBXREC,1,Z,Z0)) Q:'Z0  S Z1=$G(^(Z0)) I Z1'="" D SETGBL^IBCEFG(PG,Z,Z0,Z1,.IBXSIZE)  . S Z0=0 F  S Z0=$O(FLDS(Z,Z0)) Q:'Z0  S Z1=$G(^TMP("IBXDATA",$J,IBXREC,1,Z,Z0)) I Z1'="" D SETGBL^IBCEFG(PG,Z,Z0,Z1,.IBXSIZE)  . I FORM=2,LPG'=PG D  .. D SETGBL^IBCEFG(PG,57,51,"",.IBXSIZE)  .. D SETGBL^IBCEFG(PG,57,71,"",.IBXSIZE)  Q  ; NONSERV(Z,Z0) ; Set variable if non-service/non-text data is present for box  ; 24 of CMS-1500  ; Z = sequence of IBXSAVE being processed  ; Z0 = sequnce within IBXDATA to indicate actual line #  I $P(IBXSAVE("BOX24",Z),U)="" S IBXSAVE("NON-SERV",Z0)=""  Q  ; PG(VAL,LNCT) ;Set next pg for CMS-1500 lines  ;VAL = value of fld  ;LNCT = line # from IBXSAVE("BOX24") array  N IBP,IBL  S IBP=LNCT\12+(LNCT#12>0),IBL=LNCT-(12\*(IBP-1))-1  I IBL'<0 S VAL=$$FORMAT(VAL,$G(IBXLOOP("IBX0")),$G(IBXDA)) D SETGBL^IBCEFG(IBP,IBXLN+IBL,IBXCOL,VAL,.IBXSIZE)  K IBXDATA(LNCT)  Q  ; MPGUB(PG,OFFSET,VAL,IBLN,IBCOL,NOFORM) ; Set up pages > 1 for UB overflows  ; PG = Page # to set (REQUIRED)  ; OFFSET = offset from first line this should be extracted into  ; 0 = first line (REQUIRED)  ; VAL = value to set (REQUIRED)  ; IBLN = line to set data at (if null, uses IBXLN)  ; IBCOL = column to set data at (if null, uses IBXCOL)  ; NOFORM = don't format, just output data as passed  ; Assumes formatter IBXLN,IBXCOL variables exist  ;  I $G(IBLN)="" S IBLN=IBXLN  I $G(IBCOL)="" S IBCOL=IBXCOL  S:'$G(NOFORM) VAL=$$FORMAT(VAL,$G(IBXLOOP("IBX0")),$G(IBXDA))  D SETGBL^IBCEFG(PG,IBLN+OFFSET,IBCOL,VAL,.IBXSIZE)  Q  ; CKREV(CT,VAL) ; Check too many rev code lines to fit on page  ; This procedure is only called when CT>22 (i.e. 23 or more)  ;  D MPGUB((CT-1)\22+1,CT-1#22,VAL) ; 22 codes on a single page  Q  ; CKPGUB ; Check to see if multiple UB pages are needed then populate  ; static flds from page 1, add page numbers  ;  N FLDS,LPG,IBPG,IBP,Z,Z0,TOT1,TOT2  ;  S LPG=$O(^TMP("IBXDATA",$J,IBXREC,""),-1),IBP=0  S Z="" F  S Z=$O(^TMP("IBXDATA",$J,IBXREC,LPG,Z),-1) Q:'Z  S Z0=0 F  S Z0=$O(^TMP("IBXDATA",$J,IBXREC,LPG,Z,Z0)) Q:'Z0  I $G(^(Z0))'="" S IBP=1 Q  I 'IBP K ^TMP("IBXDATA",$J,IBXREC,LPG) S LPG=$O(^TMP("IBXDATA",$J,IBXREC,""),-1) Q:LPG=1  ;  ; Static flds  F Z=2:1:7 S FLDS(Z)=""    ; FL-1 thru FL-9  F Z=1,10,13,19,22,25,28,31 S FLDS(9,Z)=""    ; FL-10 thru FL-17  F Z=13:1:17 S FLDS(Z,1)=""   ; payer address in FL-38  S FLDS(41,46)=""     ; creation date  F Z=42,43,44,45,47,48,49,51,52,53 S FLDS(Z)=""   ; FL-50 thru FL-65  F Z=57,59,61,63 S (FLDS(Z,59),FLDS(Z,72),FLDS(Z,74))=""   ; FL-76-79 ID's  F Z=58,60,62,64 S (FLDS(Z,53),FLDS(Z,71))=""    ; FL-76-79 Names  ;  F IBPG=1:1:LPG D  . ; Add pg # to last line of rev codes if multiple pages  . N IB,IBP  . S IB=$G(^TMP("IBXDATA",$J,IBXREC,IBPG,41,6))  . D MPGUB(IBPG,0,IBPG,41,10,1)  . D MPGUB(IBPG,0,LPG,41,16,1)  . D:IBPG>1 MPG(IBPG,.FLDS,1)  . Q  ; print totals on line 41 of the last page  S (TOT1,TOT2)=0  F Z=1:1 Q:'$D(^TMP($J,"IBC-RC",Z)) S Z0=^(Z) I +Z0=1 S TOT1=TOT1+$P(Z0,U,7),TOT2=TOT2+$P(Z0,U,8)  ; Make sure totals are only 9 digits => baa IB\*2.0\*488  S TOT1=$$DOL^IBCEF77(TOT1,9)  S TOT1=$E(TOT1,1,9)  S TOT2=$$DOL^IBCEF77(TOT2,9)  S TOT2=$E(TOT2,1,9)  D MPGUB(IBPG,0,"0001",41,1,1)  D MPGUB(IBPG,0,TOT1,41,61,1)  D MPGUB(IBPG,0,TOT2,41,71,1)  ;End changes => baa IB\*2.0\*488  Q  ; HCPC(R) ;FORMAT HCPC fld FOR UB (returns formatted value)  ; R = flag for type of fld (1/2/3) being printed in rev code block  Q R  ;No longer used as of patch IB\*2.0\*51  ; PROS(IBIFN) ; Extract billable prosthetics for 837  N IBARRAY,Z,Z0,CT,PROS  D SET^IBCSC5B(IBIFN,.IBARRAY)  I '$P(IBARRAY,U,2) S CT="" G PROSQ  S Z="",CT=0  F  S Z=$O(IBARRAY(Z)) Q:Z=""  S Z0="" F  S Z0=$O(IBARRAY(Z,Z0)) Q:Z0=""  S CT=CT+1 D  .S PROS=$$PINB^IBCSC5B(+IBARRAY(Z,Z0)) ; P389 removed p2 - item ptr file 661  .;date^^short descr^entry # in file 362.5  .S IBXDATA(CT)=Z\_U\_U\_PROS\_U\_+IBARRAY(Z,Z0) PROSQ Q CT  ; B24(IBXSV,IBIFN,IBNOSHOW) ; Code to execute to set up IBXSV("BOX24") for  ; print or IBXSAVE("OUTPT") for transmit - called by output formatter  ; IBNOSHOW = 1 if not to show error/warning text lines  ; Pass IBXSV by reference  N IBSUB  S IBSUB=$S('$G(^TMP("IBTX",$J,IBIFN)):"BOX24",1:"OUTPT")  K IBXSV(IBSUB)  I '$D(IBIFN) S IBIFN=$G(IBXIEN)  I IBIFN D F^IBCEF("N-HCFA 1500 SERVICE"\_$S(IBSUB["24":"S (PRINT",1:" LINE (EDI")\_")",,,IBIFN)  I $S(IBSUB'["24":1,1:'$G(IBNOSHOW)) D  . M IBXSV(IBSUB)=IBXDATA  E  D  . N Z,CT  . S (Z,CT)=0 F  S Z=$O(IBXDATA(Z)) Q:'Z  I '$D(IBXDATA(Z,"ARX")) S CT=CT+1 M IBXSV(IBSUB,CT)=IBXDATA(Z)  Q  ;  ; esg - 11/14/03 - Moved the below functions due to space constraints  ; ALLTYP(IBIFN) Q $$ALLTYP^IBCEF31(IBIFN) INSTYP(IBIFN,SEQ) Q $$INSTYP^IBCEF31(IBIFN,$G(SEQ)) POLTYP(IBIFN,IBSEQ) Q $$POLTYP^IBCEF31(IBIFN,$G(IBSEQ)) ALLPTYP(IBIFN) Q $$ALLPTYP^IBCEF31(IBIFN)  ; FILL(Z) ;  Q  ;  ; \*\*\*\*\*   ; The following code performs the multi-page set up for  ; printing overflow data on the UB  ; \*\*\*\*\*  ; XPROC(DATA,CT) ; Output any UB procedures after 6 on new page(s)  ; DATA = output data from IBXSAVE("PROC",CT)  ; CT = array sequence # of the procedure being output  ; Only used for local prints  N OFFSET,PG,COL,PRCODE,Q  S Q=(CT-1)\3#2,OFFSET=$S('Q:0,1:2)  S PG=(CT-1)\6+1,COL=1+(CT-1#3\*15)  D MPGUB(PG,OFFSET,$P(DATA,U,1),58,COL)  D MPGUB(PG,OFFSET,$P(DATA,U,2),58,COL+9)  Q  ; XDIAG(DATA,CT) ; Output any UB other diagnoses after 8 on new page(s)  ; DATA = output data from IBXSAVE("DX",CT)  ; CT = array sequence # of the diagnosis being output  ; Only used for local prints  N COL,PG  S PG=(CT-1)\8+1,COL=8+(CT-1#9\*7)  S DATA=$P($$ICD9^IBACSV(+DATA),U,1)  D MPGUB(PG,0,DATA,56,COL)  Q  ; XVAL(DATA,CT) ; Output any UB value codes after 12 on new page(s)  ; DATA = output data from IBXSAVE("VC",CT)  ; CT = array sequence # of the value code being output  ;  N COL,PG,OFFSET  S PG=(CT-1)\12+1,COL=44+(CT-1#3\*13),OFFSET=(CT-(12\*(PG-1))-1)\3  D MPGUB(PG,OFFSET,$P(DATA,U,1),14,COL)  D MPGUB(PG,OFFSET,$P(DATA,U,2),14,COL+3)  Q  ; XCC(DATA,CT) ; Output any UB condition codes after 11 on new page(s)  ; 11 condition codes per page, starting columns 34 thru 64  ; DATA = output data from IBXSAVE("CC",CT)  ; CT = array sequence # of the condition code being output  ;  N COL,PG  S PG=(CT-1)\11+1,COL=34+(CT-1#11\*3)  D MPGUB(PG,0,DATA,9,COL)  Q  ; XOCC(DATA,CT,FL) ; Output any UB occurrence codes after 8 (2 per form  ; locators 31-34) on new page(s)  ; DATA = data from IBXSAVE("OCC",z) to be output  ; CT = array sequence # of occurrence code being output  ; FL = # of form locator being populated with the occ code  ;  N COL,PG,OFFSET  S PG=(CT-1)\2+1,COL=1+((FL-31)\*10),OFFSET=$S(CT#2:0,1:1)  D MPGUB(PG,OFFSET,$P(DATA,U,1),11,COL)  D MPGUB(PG,OFFSET,$P(DATA,U,2),11,COL+4)  Q  ; XOCCS(DATA,CT,FL) ; Output any UB occurrence span codes after 4 on new page(s)  ; DATA = data from IBXSAVE("OCCS",z) to be output  ; CT = array sequence # of occurrence span code being output  ; FL = # of form locator being populated (either FL 35 or 36)  ;  N COL,PG,OFFSET  S PG=(CT-1)\2+1,OFFSET=$S(CT#2:0,1:1)  S COL=41+((FL-35)\*17)  D MPGUB(PG,OFFSET,$P(DATA,U,1),11,COL)  D MPGUB(PG,OFFSET,$P(DATA,U,2),11,COL+4)  D MPGUB(PG,OFFSET,$P(DATA,U,3),11,COL+11)  Q  ; FORMAT(VAL,IBX0,IBXDA) ;  I IBX0'="",IBXDA S VAL=$$FORMAT^IBCEFG(VAL,$P($G(^IBA(364.6,+IBXDA,0)),U,9),$P(IBX0,U,7),IBX0)  Q VAL  ; OUTPDT(IBIFN,IBXSAVE,IBXDATA) ; Returns outpatient service to date  ; formatted CCYYMMDD for UB 837  ; IBIFN = ien of bill (file 399)  ; IBXSAVE = pass by reference for IBXSAVE("INPT") and IBXSAVE("DATE")  ; IBXDATA = array with formatted date or each line item - CCYYMMDD  N Z  S Z=0 F  S Z=$O(IBXSAVE("INPT",Z)) Q:'Z  S IBXDATA(Z)=$S($P(IBXSAVE("INPT",Z),U,10):$$DT^IBCEFG1($P(IBXSAVE("INPT",Z),U,10),,"D8"),1:IBXSAVE("DATE"))  K IBXSAVE("DATE")  Q  ; | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCEF31 | | | | | | | | |
| **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** | | | | | | | | | |
| IBCEF31 ;ALB/ESG - FORMATTER SPECIFIC BILL FLD FUNCTIONS - CONT ;14-NOV-03  ;;2.0;INTEGRATED BILLING;\*\*155,296,349,400,432,488,516\*\*;21-MAR-94;Build 123  ;;Per VA Directive 6402, this routine should not be modified.  ;  Q  ; ALLTYP(IBIFN) ; returns codes to translate to ALL ins types on a bill  ; IBIFN = ien of bill  N IBX,Z  F Z=1:1:3 S $P(IBX,U,Z)=$$INSTYP(IBIFN,Z)  ; IBX = primary code^secondary code^tertiary code  Q IBX  ; INSTYP(IBIFN,SEQ) ; Returns insurance type code for an ins on a bill  ; IBIFN = ien of bill  ; SEQ = sequence (1,2,3) of insurance wanted - prim, second, tert  ; Default is current insurance co  ;  N IBA,Z  ;  I '$G(SEQ) S SEQ=$$COBN^IBCEF(IBIFN)  S Z=+$G(^DGCR(399,IBIFN,"I"\_SEQ))  ;Codes 1:HMO;2:COMMERCIAL;3:MEDICARE;4:MEDICAID;5:GROUP POLICY;9:OTHER  I Z D  . S IBA=$P($G(^DIC(36,Z,3)),U,9)  . I $$MCRWNR^IBEFUNC(Z) S IBA=3 ; force Medicare (WNR) definition to be correct  . I IBA="" S IBA=5 ;Default is group policy - 5 if blank  ;  Q $G(IBA)  ; POLTYP(IBIFN,IBSEQ) ; Returns ins electronic policy type code for one  ; ins policy on a bill  ; IBIFN = ien of bill  ; IBSEQ = sequence (1,2,3) of ins policy wanted - prim, second, tert  ; Default is current insurance co  ;  N IBPLAN,IBPLTYP  ;  I '$G(IBSEQ) S IBSEQ=+$$COBN^IBCEF(IBIFN)  S IBPLAN=$G(^IBA(355.3,+$P($G(^DGCR(399,IBIFN,"I"\_IBSEQ)),U,18),0))  S IBPLTYP=$P(IBPLAN,U,15)  ;  ; esg - 06/30/05 - IB\*2.0\*296 - Force Medicare (WNR) to be correct  I $$WNRBILL^IBEFUNC(IBIFN,IBSEQ),$$FT^IBCEF(IBIFN)=2 S IBPLTYP="MB"   ; CMS-1500 ----> Medicare Part B  I $$WNRBILL^IBEFUNC(IBIFN,IBSEQ),$$FT^IBCEF(IBIFN)=3 S IBPLTYP="MA"   ; UB-04 -------> Medicare Part A  ;  I IBPLTYP="" S IBPLTYP="CI" ;Default is commercial - 'CI'  I IBPLTYP="MX" D  . I $P(IBPLAN,U,14)'="","AB"[$P(IBPLAN,U,14) S IBPLTYP="M"\_$P(IBPLAN,U,14) Q  . S IBPLTYP="CI"  Q $G(IBPLTYP)  ; ALLPTYP(IBIFN) ; returns insurance policy type codes for ALL ins on a bill  ; IBIFN = ien of bill  N IBX,Z S IBX=""  F Z=1:1:3 I $D(^DGCR(399,IBIFN,"I"\_Z)) S $P(IBX,U,Z)=$$POLTYP(IBIFN,Z)  ; IBX = primary code^secondary code^tertiary code  Q IBX  ; PGDX(DXCNT,IBX0,IBXDA,IBXLN,IBXCOL,IBXSIZE,IBXSAVE) ; Subroutine - Checks for Diagnosis Codes (Dx) beyond   ; the first four, that relate to the current Dx position passed in DXCNT.  ; This subroutine stores the Diagnosis Codes in output global using display parameters (IBXLN,IBXCOL)  ; THE PAGE IS ALWAYS 1 NOW SO WE DON'T NEED 4 LINES BELOW BAA \*488\*  ; If DXCNT is 1, check for Dx's 5,9,...etc & display on pages 2,3,...etc  ; If DXCNT is 2, check for Dx's 6,10,...etc & display on pages 2,3,...etc  ; If DXCNT is 3, check for Dx's 7,11,...etc & display on pages 2,3,...etc  ; If DXCNT is 4, check for Dx's 8,12,...etc & display on pages 2,3,...etc  ;  ; Input: DXCNT= position of current Dx (from 1 to 4)  ; IBX0= zero-level of file 364.7 of current Dx  ; IBXDA= ien# of file 364.6 of current Dx  ; IBXLN IBXCOL= line# & Column# of current Dx  ; IBXSIZE= size counter  ; IBXSAVE("DX")= local array with all Dx's on current bill  ;  ; For patch \*488\*   ; S DXNM = 12 This is the number of diagnosis on a 1500 form   ; S IBPG=1 This is the page number. All 12 print on page 1  N IBPG,VAL  S IBPG=1  I '$D(IBXSAVE("DX",DXCNT)) Q  S VAL=$P($$ICD9^IBACSV(+IBXSAVE("DX",DXCNT)),U) ; resolve Dx pointer  S VAL=$$FORMAT^IBCEF3(VAL,$G(IBX0),$G(IBXDA)) ;format Dx value  D SETGBL^IBCEFG(IBPG,IBXLN,IBXCOL,VAL,.IBXSIZE) ;store in output global  Q  ;PGDX  ; DXSV(IB,IBXSAVE) ; output formatter subroutine  ; save off DX codes in IBXSAVE("DX")  N Z,IBCT  S (Z,IBCT)=0  F  S Z=$O(IB(Z)) Q:'Z  I $G(IB(Z)) S IBCT=IBCT+1 M IBXSAVE("DX",IBCT)=IB(Z)  Q  ; AUTRF(IBXIEN,IBL,Z) ; returns auth # and referral# if room for both, separated by a space - IB\*2.0\*432  ; IBXIEN= claim ien  ; IBL = field length-1 to allow for 1 blank space between numbers (28 for CMS 1500, 30 for UB-04)  ; Z = 1 for PRIMARY, 2 for SECONDARY, 3 for TERTIARY  ;   N IBXDATA,IBZ  Q:$G(IBXIEN)="" ""  ; if CMS 1500, find current codes  I $G(Z)="",$G(IBL)=28 S Z=$$COBN^IBCEF(IBXIEN)  Q:$G(Z)="" ""  ; if length not defined, default to shortest  S:IBL="" IBL=28  D F^IBCEF("N-"\_$P("PRIMARY^SECONDARY^TERTIARY",U,Z)\_" AUTH CODE",,,IBXIEN)  D F^IBCEF("N-"\_$P("PRIMARY^SECONDARY^TERTIARY",U,Z)\_" REFERRAL NUMBER","IBZ",,IBXIEN)  ; if length of auth and referral combined is too long, only return auth code  Q $S(IBZ="":IBXDATA,IBXDATA="":IBZ,$L(IBXDATA)+$L(IBZ)>IBL:IBXDATA,1:IBXDATA\_" "\_IBZ)  ; GRPNAME(IBIEN,IBXDATA) ; Populate IBXDATA with the Group Name(s).  ; MRD;IB\*2.0\*516 - Created this procedure as extract code for  ; ^IBA(364.5,199), N-ALL INSURANCE GROUP NAME.  N A,Z  F Z=1:1:3 I $D(^DGCR(399,IBIEN,"I"\_Z)) D  . S IBXDATA(Z)=$$POLICY^IBCEF(IBIEN,15,Z) I IBXDATA(Z)'="" Q  . S A=$$POLICY^IBCEF(IBIEN,1,Z) ; Pull piece 1, Ins. Type.  . I A'="" S IBXDATA(Z)=$P($G(^DIC(36,+A,0)),U)  . Q  Q  ; GRPNUM(IBXIEN,IBXDATA) ; Populate IBXDATA with the Group Number(s).  ; MRD;IB\*2.0\*516 - Created this procedure as extract code for  ; ^IBA(364.5,200), N-ALL INSURANCE GROUP NUMBER.  N Z  F Z=1:1:3 I $D(^DGCR(399,IBXIEN,"I"\_Z)) S IBXDATA(Z)=$$POLICY^IBCEF(IBXIEN,3,Z)  Q  ; | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCEF31 ;ALB/ESG - FORMATTER SPECIFIC BILL FLD FUNCTIONS - CONT ;14-NOV-03  ;;2.0;INTEGRATED BILLING;\*\*155,296,349,400,432,488,516,592\*\*;21-MAR-94;Build 123  ;;Per VA Directive 6402, this routine should not be modified.  ;  Q  ; ALLTYP(IBIFN) ; returns codes to translate to ALL ins types on a bill  ; IBIFN = ien of bill  N IBX,Z  F Z=1:1:3 S $P(IBX,U,Z)=$$INSTYP(IBIFN,Z)  ; IBX = primary code^secondary code^tertiary code  Q IBX  ; INSTYP(IBIFN,SEQ) ; Returns insurance type code for an ins on a bill  ; IBIFN = ien of bill  ; SEQ = sequence (1,2,3) of insurance wanted - prim, second, tert  ; Default is current insurance co  ;  N IBA,Z  ;  I '$G(SEQ) S SEQ=$$COBN^IBCEF(IBIFN)  S Z=+$G(^DGCR(399,IBIFN,"I"\_SEQ))  ;Codes 1:HMO;2:COMMERCIAL;3:MEDICARE;4:MEDICAID;5:GROUP POLICY;9:OTHER  I Z D  . S IBA=$P($G(^DIC(36,Z,3)),U,9)  . I $$MCRWNR^IBEFUNC(Z) S IBA=3 ; force Medicare (WNR) definition to be correct  . I IBA="" S IBA=5 ;Default is group policy - 5 if blank  ;  Q $G(IBA)  ; POLTYP(IBIFN,IBSEQ) ; Returns ins electronic policy type code for one  ; ins policy on a bill  ; IBIFN = ien of bill  ; IBSEQ = sequence (1,2,3) of ins policy wanted - prim, second, tert  ; Default is current insurance co  ;  N IBPLAN,IBPLTYP  ;  I '$G(IBSEQ) S IBSEQ=+$$COBN^IBCEF(IBIFN)  S IBPLAN=$G(^IBA(355.3,+$P($G(^DGCR(399,IBIFN,"I"\_IBSEQ)),U,18),0))  S IBPLTYP=$P(IBPLAN,U,15)  ;  ; esg - 06/30/05 - IB\*2.0\*296 - Force Medicare (WNR) to be correct  ;JRA IB\*2.0\*592 Treat Dental Form 7 (J430D) the same as CMS-1500  ;I $$WNRBILL^IBEFUNC(IBIFN,IBSEQ),$$FT^IBCEF(IBIFN)=2 S IBPLTYP="MB" ; CMS-1500 ----> Medicare Part B ;JRA IB\*2.0\*592 ';'  ;I $$WNRBILL^IBEFUNC(IBIFN,IBSEQ),$$FT^IBCEF(IBIFN)=3 S IBPLTYP="MA" ; UB-04 -------> Medicare Part A  N FT S FT=$$FT^IBCEF(IBIFN) ;JRA IB\*2.0\*592  I $$WNRBILL^IBEFUNC(IBIFN,IBSEQ),(FT=2!(FT=7)) S IBPLTYP="MB"   ; CMS-1500 ----> Medicare Part B ;JRA IB\*2.0\*592 same for J430D  I $$WNRBILL^IBEFUNC(IBIFN,IBSEQ),FT=3 S IBPLTYP="MA"   ; UB-04 -------> Medicare Part A ;JRA IB\*2.0\*592 Use 'FT' vs function call  ;  I IBPLTYP="" S IBPLTYP="CI" ;Default is commercial - 'CI'  I IBPLTYP="MX" D  . I $P(IBPLAN,U,14)'="","AB"[$P(IBPLAN,U,14) S IBPLTYP="M"\_$P(IBPLAN,U,14) Q  . S IBPLTYP="CI"  Q $G(IBPLTYP)  ; ALLPTYP(IBIFN) ; returns insurance policy type codes for ALL ins on a bill  ; IBIFN = ien of bill  N IBX,Z S IBX=""  F Z=1:1:3 I $D(^DGCR(399,IBIFN,"I"\_Z)) S $P(IBX,U,Z)=$$POLTYP(IBIFN,Z)  ; IBX = primary code^secondary code^tertiary code  Q IBX  ; PGDX(DXCNT,IBX0,IBXDA,IBXLN,IBXCOL,IBXSIZE,IBXSAVE) ; Subroutine - Checks for Diagnosis Codes (Dx) beyond   ; the first four, that relate to the current Dx position passed in DXCNT.  ; This subroutine stores the Diagnosis Codes in output global using display parameters (IBXLN,IBXCOL)  ; THE PAGE IS ALWAYS 1 NOW SO WE DON'T NEED 4 LINES BELOW BAA \*488\*  ; If DXCNT is 1, check for Dx's 5,9,...etc & display on pages 2,3,...etc  ; If DXCNT is 2, check for Dx's 6,10,...etc & display on pages 2,3,...etc  ; If DXCNT is 3, check for Dx's 7,11,...etc & display on pages 2,3,...etc  ; If DXCNT is 4, check for Dx's 8,12,...etc & display on pages 2,3,...etc  ;  ; Input: DXCNT= position of current Dx (from 1 to 4)  ; IBX0= zero-level of file 364.7 of current Dx  ; IBXDA= ien# of file 364.6 of current Dx  ; IBXLN IBXCOL= line# & Column# of current Dx  ; IBXSIZE= size counter  ; IBXSAVE("DX")= local array with all Dx's on current bill  ;  ; For patch \*488\*   ; S DXNM = 12 This is the number of diagnosis on a 1500 form   ; S IBPG=1 This is the page number. All 12 print on page 1  N IBPG,VAL  S IBPG=1  I '$D(IBXSAVE("DX",DXCNT)) Q  S VAL=$P($$ICD9^IBACSV(+IBXSAVE("DX",DXCNT)),U) ; resolve Dx pointer  S VAL=$$FORMAT^IBCEF3(VAL,$G(IBX0),$G(IBXDA)) ;format Dx value  D SETGBL^IBCEFG(IBPG,IBXLN,IBXCOL,VAL,.IBXSIZE) ;store in output global  Q  ;PGDX  ; DXSV(IB,IBXSAVE) ; output formatter subroutine  ; save off DX codes in IBXSAVE("DX")  N Z,IBCT  S (Z,IBCT)=0  F  S Z=$O(IB(Z)) Q:'Z  I $G(IB(Z)) S IBCT=IBCT+1 M IBXSAVE("DX",IBCT)=IB(Z)  Q  ; AUTRF(IBXIEN,IBL,Z) ; returns auth # and referral# if room for both, separated by a space - IB\*2.0\*432  ; IBXIEN= claim ien  ; IBL = field length-1 to allow for 1 blank space between numbers (28 for CMS 1500, 30 for UB-04)  ; Z = 1 for PRIMARY, 2 for SECONDARY, 3 for TERTIARY  ;   N IBXDATA,IBZ  Q:$G(IBXIEN)="" ""  ; if CMS 1500, find current codes  I $G(Z)="",$G(IBL)=28 S Z=$$COBN^IBCEF(IBXIEN)  Q:$G(Z)="" ""  ; if length not defined, default to shortest  S:IBL="" IBL=28  D F^IBCEF("N-"\_$P("PRIMARY^SECONDARY^TERTIARY",U,Z)\_" AUTH CODE",,,IBXIEN)  D F^IBCEF("N-"\_$P("PRIMARY^SECONDARY^TERTIARY",U,Z)\_" REFERRAL NUMBER","IBZ",,IBXIEN)  ; if length of auth and referral combined is too long, only return auth code  Q $S(IBZ="":IBXDATA,IBXDATA="":IBZ,$L(IBXDATA)+$L(IBZ)>IBL:IBXDATA,1:IBXDATA\_" "\_IBZ)  ; GRPNAME(IBIEN,IBXDATA) ; Populate IBXDATA with the Group Name(s).  ; MRD;IB\*2.0\*516 - Created this procedure as extract code for  ; ^IBA(364.5,199), N-ALL INSURANCE GROUP NAME.  N A,Z  F Z=1:1:3 I $D(^DGCR(399,IBIEN,"I"\_Z)) D  . S IBXDATA(Z)=$$POLICY^IBCEF(IBIEN,15,Z) I IBXDATA(Z)'="" Q  . S A=$$POLICY^IBCEF(IBIEN,1,Z) ; Pull piece 1, Ins. Type.  . I A'="" S IBXDATA(Z)=$P($G(^DIC(36,+A,0)),U)  . Q  Q  ; GRPNUM(IBXIEN,IBXDATA) ; Populate IBXDATA with the Group Number(s).  ; MRD;IB\*2.0\*516 - Created this procedure as extract code for  ; ^IBA(364.5,200), N-ALL INSURANCE GROUP NUMBER.  N Z  F Z=1:1:3 I $D(^DGCR(399,IBXIEN,"I"\_Z)) S IBXDATA(Z)=$$POLICY^IBCEF(IBXIEN,3,Z)  Q  ; | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCEF4 | | | | | | | | |
| **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** | | | | | | | | | |
| IBCEF4 ;ALB/TMP - MRA/EDI ACTIVATED UTILITIES ;06-FEB-96  ;;2.0;INTEGRATED BILLING;\*\*51,137,232,155,296,327,349\*\*;21-MAR-94;Build 46  ;;Per VHA Directive 2004-038, this routine should not be modified.  ; EDIACTV(IBEDIMRA) ; Returns 0 if EDI or MRA is not active,   ; otherwise, returns 1  ; IBEDIMRA : 1= checking if EDI is active, 2= checking if MRA is active  N IBEDI  S IBEDI=$P($G(^IBE(350.9,1,8)),U,10)  Q $S('IBEDI:0,IBEDI=3:1,1:IBEDI=IBEDIMRA)  ; RATEOK(IBIFN) ; Returns 1 if rate type of bill IBIFN is transmittable  Q +$P($G(^DGCR(399.3,+$P($G(^DGCR(399,IBIFN,0)),U,7),0)),U,10)  ; INSOK(INS) ; Determine EDI activation status of insurance co  Q +$G(^DIC(36,INS,3)) ;1 = TEST, 2 = LIVE, 0 = NOT ACTIVE FOR EDI  ; BSTATX(IBIFN) ; Returns internal value of bill's latest transmission status  N IBDA  Q $P($G(^IBA(364,+$$LAST364(IBIFN),0)),U,3)  ; LAST364(IBIFN) ; Determine ien of latest transmit bill record for a bill  Q +$O(^IBA(364,"ABDT",IBIFN,+$O(^IBA(364,"ABDT",IBIFN,""),-1),""),-1)  ; TXMT(IBIFN,IBWHY,IBNEW) ; Determine if bill # IBIFN is 'transmittable'  ; IBNEW = flag is 1 if new entry - don't check for entry in file 364  ; Function returns:  ; 0 if not transmittable  ; if transmittable, the entire node 3 of the insurance company  ; and, if passed by reference IBWHY = reason not transmittable  ; 1 if local print  ; 2 if EDI/MRA not active  ; 3 if rate type not transmittable  ; 4 if no transmit for insurance co  ; 5 if failed txmn rules  ; and IBWHY(0) = ien of rule failed  ; 6 if Rx with missing/invalid NDC format  ;  N IB,IB0,IBOK,IBCOB,IBMCR,X1  S IBOK=1,IB=IBIFN,IBWHY=""  ;  S IBCOB=$$COBN^IBCEF(IB),IB(.07)=+$G(^DGCR(399,IB,"I"\_IBCOB))  S IBMCR=$$MCRWNR^IBEFUNC(IB(.07))  ; Does bill have force local print flag set?  I 'IBMCR D  G:IBWHY TXMTQ  ; MCR WNR not curr ins  . I $S($$MRASEC(IBIFN):$P($G(^DGCR(399,IBIFN,"TX")),U,9)=1,1:$P($G(^DGCR(399,IBIFN,"TX")),U,8)=1) S IBOK=0,IBWHY=1  I '$G(IBNEW),'$O(^IBA(364,"B",IBIFN,0)),$P($G(^DGCR(399,IBIFN,0)),U,13)>2,'$$RETN^PRCAFN(IBIFN) S IBOK=0 G TXMTQ ; Not recognized as transmittable when it was authorized  I $O(^IBA(364,"B",IBIFN,0)),$$INSOK(IB(.07)),$$BSTATX(IBIFN)'="X" G TXMTQ ;Already determined to be transmittable - entry exists for bill in transmit bill file  S IB(.03)=$S('IBMCR:1,1:2) ; EDI(1) or MRA(2)  S IB(.04)=$S('$$INPAT^IBCEF(IB,1):1,1:2) ;Outpt(1) or Inpt(2)  S IB(.05)=$S($$FT^IBCEF(IB)=3:1,1:2) ;Inst(1) or Prof(2)  ; Execute unmodifiable, general edits  S X1=$$EDIACTV(IB(.03))  I 'X1 S IBWHY=2  I 'IBWHY S X1=$$RATEOK(IBIFN) S:'X1 IBWHY=3  I 'IBWHY S X1=$$INSOK(+IB(.07)) S:'X1 IBWHY=4  I 'IBWHY,$$ISRX^IBCEF1(IBIFN) D  ;S:'X1 IBWHY=6  . ; Check for Rxs and NDC # format valid (5-4-2)  . ;IF THIS IS A UB FORM DO NOT SEND ELECTRONIC  . I $$FT^IBCEF(IBIFN)=3 S IBWHY=1  . ;  . Q  ;;CHECK REMOVAL SO NON NDC FORMAT NUMBERS WILL GO  . N Z,Z0,Z00  . S Z="" F  S Z=$O(^IBA(362.4,"AIFN"\_IBIFN,Z)) Q:Z=""!'X1  D  Q:'X1  .. S Z0=0 F  S Z0=$O(^IBA(362.4,"AIFN"\_IBIFN,Z,Z0)) Q:'Z0  D  Q:'X1  ... S Z00=$G(^IBA(362.4,Z0,0))  ... Q:$S($P(Z00,U,8)="":1,1:$L($P(Z00,U,8))=11)  ... I $P(Z00,U,9)'=4 S X1=0  ; Only continue if general edits are passed  I $$COB^IBCEF(IB)="S" D  . S COBINS=$P($G(^DGCR(399,IB,"M")),U,IBCOB+1)  . I 'COBINS Q  . I IBMCR S IBWHY=1,$P(^DGCR(399,IBIFN,"TX"),U,8)=1  I IBWHY S IBOK=0 G TXMTQ  S IBOK=$$EDIT(IBIFN,.IB,.IBWHY)  G:'IBOK TXMTQ  ; TXMTQ ;  I IBOK S IBOK=$G(^DIC(36,+IB(.07),3))  Q IBOK  ; MRASEC(IBIFN) ; Returns 1 if current bill is secondary to MCR WNR  N IBSEQ,IB,Z  S IB=0  ; Chk if MCR WNR is prev insurer with MRA on file  S IBSEQ=$$COBN^IBCEF(IBIFN)-1  S Z=$$MCRONBIL^IBEFUNC(IBIFN,IBSEQ) I +Z=1,$P(Z,U,2)=1,$$CHK^IBCEMU1(IBIFN) S IB=1  Q IB  ; EDIT(IBIFN,IB,IBWHY) ; Find, execute edits applying to bill to see if transmittable  ; IBIFN = ien of bill in file 399  ; IB = array containing necessary data for xref search from bill  ; subscripted by field # in file 364.4  ;  ; Matrix entries:  ; IB(.03): 1=EDI specific, 2=MRA specific  ; IB(.04): 1=Outpatient or 2=inpatient only (currently defaults to 3)  ; IB(.05): 1=Only institutional or 2=only professional  ; X: Anything valid  ;  ; MRA-EDI IN-OUT INST-PROF  ; Level ------- ------ ---------  ; 1 X X X  ; 2 X X IB(.05)  ; 3 X IB(.04) X  ; 4 X IB(.04) IB(.05)  ; 5 IB(.03) X X  ; 6 IB(.03) X IB(.05)   ; 7 IB(.03) IB(.04) X  ; 8 IB(.03) IB(.04) IB(.05)  ;  N IB0,IB1,IB2,IB3,IB4,IBDA,IBFT,IBPASS,IBSEQ,IBT,IBNOCK  I '$G(IB(.03)) S IBPASS=0 G EDITQ  S IBFT=$$FT^IBCEF(IBIFN)  ;  S IBPASS=1  F IBSEQ=1:1:8 D  Q:'IBPASS  ; Loop thru levels in matrix  . F IB1=1:1:3 Q:'IBPASS  F IB2=1:1:3 Q:'IBPASS  F IB3=1:1:3 Q:'IBPASS  D  .. S IB4=0 F  S IB4=$O(^IBE(364.4,"AD",IB1,IB2,IB3,IB4)) Q:'IB4  I $O(^(IB4,0)) D  Q:'IBPASS  ... S IBDA=0  ... F   S IBDA=$O(^IBE(364.4,"AD",IB1,IB2,IB3,IB4,IBDA)) Q:'IBDA  S IB0=$G(^IBE(364.4,IBDA,0)) I IB0'="",'$D(IBNOCK(IBDA)) D  Q:'IBPASS  .... I $P(IB0,U,2)>DT S IBNOCK(IBDA)="" Q  ; Not activated yet  .... I $P(IB0,U,6),$P(IB0,U,6)'>DT  S IBNOCK(IBDA)="" Q  ; Inactive  .... I $P(IB0,U,11),IB3'=3,$S(IBFT=3:IB3'=1,IBFT=2:IB3'=2,1:0) S IBNOCK(IBDA)="" Q  ; Form type not included - not used for form type rule (0)  .... I IB4=1,'$D(^IBE(364.4,IBDA,3,"B",+IB(.07))) S IBNOCK(IBDA)="" Q  ; Ins not included for rule  .... I IB4=2,$D(^IBE(364.4,IBDA,2,"B",+IB(.07))) S IBNOCK(IBDA)="" Q  ; Ins is excluded from rule  .... S IBT=$G(^IBE(364.4,IBDA,1))  .... ; Code can assume IBIFN, IBDA and IB(.03 thru .05 and .07) exist  .... I IBT'="" X IBT I '$T S IBPASS=0,IBWHY(0)=IBDA,IBWHY=5 EDITQ Q IBPASS  ; STATUS(IBIFN) ; Function returns whether or not bill currently has a status  ; message or EOB message not yet fully reviewed -  ; (only for transmittable bills)  ; IBIFN = ien of bill in file 399  ; Returns:  ; 0 = None found  ; If found, returns a pieced string as follows:  ;  ; [1] ien of transmit bill entry (file 364) associated with an  ; entry in file 361 with an unreviewed status message  ; [2] ien of transmit bill entry (file 364) associated with an  ; entry in file 361.1 with an unreviewed EOB  ;  N IB,Z,Z0  S IB=""  S Z="" F  S Z=$O(^IBM(361,"B",IBIFN,Z),-1) Q:'Z  I $P($G(^IBM(361,Z,0)),U,9)<2,$P(^(0),U,11) S $P(IB,U)=$P(^(0),U,11) Q  ;  S Z="" F  S Z=$O(^IBM(361.1,"B",IBIFN,Z),-1) Q:'Z  I $P($G(^IBM(361.1,Z,0)),U,16)<2,$P(^(0),U,19) S $P(IB,U,2)=$P(^(0),U,19) Q  ;  Q IB  ; TEST(IBIFN) ; Returns 1 if bill IBIFN is a transmission test bill, 0 if not  Q +$S($G(^TMP("IBEDI\_TEST\_BATCH",$J)):1,1:+$P($G(^IBA(364,+$$LAST364(IBIFN),0)),U,7))  ; | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCEF4 ;ALB/TMP - MRA/EDI ACTIVATED UTILITIES ;06-FEB-96  ;;2.0;INTEGRATED BILLING;\*\*51,137,232,155,296,327,349,592\*\*;21-MAR-94;Build 46  ;;Per VHA Directive 2004-038, this routine should not be modified.  ; EDIACTV(IBEDIMRA) ; Returns 0 if EDI or MRA is not active,   ; otherwise, returns 1  ; IBEDIMRA : 1= checking if EDI is active, 2= checking if MRA is active  N IBEDI  S IBEDI=$P($G(^IBE(350.9,1,8)),U,10)  Q $S('IBEDI:0,IBEDI=3:1,1:IBEDI=IBEDIMRA)  ; RATEOK(IBIFN) ; Returns 1 if rate type of bill IBIFN is transmittable  Q +$P($G(^DGCR(399.3,+$P($G(^DGCR(399,IBIFN,0)),U,7),0)),U,10)  ; INSOK(INS) ; Determine EDI activation status of insurance co  Q +$G(^DIC(36,INS,3)) ;1 = TEST, 2 = LIVE, 0 = NOT ACTIVE FOR EDI  ; BSTATX(IBIFN) ; Returns internal value of bill's latest transmission status  N IBDA  Q $P($G(^IBA(364,+$$LAST364(IBIFN),0)),U,3)  ; LAST364(IBIFN) ; Determine ien of latest transmit bill record for a bill  Q +$O(^IBA(364,"ABDT",IBIFN,+$O(^IBA(364,"ABDT",IBIFN,""),-1),""),-1)  ; TXMT(IBIFN,IBWHY,IBNEW) ; Determine if bill # IBIFN is 'transmittable'  ; IBNEW = flag is 1 if new entry - don't check for entry in file 364  ; Function returns:  ; 0 if not transmittable  ; if transmittable, the entire node 3 of the insurance company  ; and, if passed by reference IBWHY = reason not transmittable  ; 1 if local print  ; 2 if EDI/MRA not active  ; 3 if rate type not transmittable  ; 4 if no transmit for insurance co  ; 5 if failed txmn rules  ; and IBWHY(0) = ien of rule failed  ; 6 if Rx with missing/invalid NDC format  ;  N IB,IB0,IBOK,IBCOB,IBMCR,X1  S IBOK=1,IB=IBIFN,IBWHY=""  ;  S IBCOB=$$COBN^IBCEF(IB),IB(.07)=+$G(^DGCR(399,IB,"I"\_IBCOB))  S IBMCR=$$MCRWNR^IBEFUNC(IB(.07))  ; Does bill have force local print flag set?  I 'IBMCR D  G:IBWHY TXMTQ  ; MCR WNR not curr ins  . I $S($$MRASEC(IBIFN):$P($G(^DGCR(399,IBIFN,"TX")),U,9)=1,1:$P($G(^DGCR(399,IBIFN,"TX")),U,8)=1) S IBOK=0,IBWHY=1  I '$G(IBNEW),'$O(^IBA(364,"B",IBIFN,0)),$P($G(^DGCR(399,IBIFN,0)),U,13)>2,'$$RETN^PRCAFN(IBIFN) S IBOK=0 G TXMTQ ; Not recognized as transmittable when it was authorized  I $O(^IBA(364,"B",IBIFN,0)),$$INSOK(IB(.07)),$$BSTATX(IBIFN)'="X" G TXMTQ ;Already determined to be transmittable - entry exists for bill in transmit bill file  S IB(.03)=$S('IBMCR:1,1:2) ; EDI(1) or MRA(2)  S IB(.04)=$S('$$INPAT^IBCEF(IB,1):1,1:2) ;Outpt(1) or Inpt(2)  S IB(.05)=$S($$FT^IBCEF(IB)=3:1,1:2) ;Inst(1) or Prof(2)  ; Execute unmodifiable, general edits  S X1=$$EDIACTV(IB(.03))  I 'X1 S IBWHY=2  I 'IBWHY S X1=$$RATEOK(IBIFN) S:'X1 IBWHY=3  I 'IBWHY S X1=$$INSOK(+IB(.07)) S:'X1 IBWHY=4  I 'IBWHY,$$ISRX^IBCEF1(IBIFN) D  ;S:'X1 IBWHY=6  . ; Check for Rxs and NDC # format valid (5-4-2)  . ;IF THIS IS A UB FORM DO NOT SEND ELECTRONIC  . I $$FT^IBCEF(IBIFN)=3 S IBWHY=1  . ;  . Q  ;;CHECK REMOVAL SO NON NDC FORMAT NUMBERS WILL GO  . N Z,Z0,Z00  . S Z="" F  S Z=$O(^IBA(362.4,"AIFN"\_IBIFN,Z)) Q:Z=""!'X1  D  Q:'X1  .. S Z0=0 F  S Z0=$O(^IBA(362.4,"AIFN"\_IBIFN,Z,Z0)) Q:'Z0  D  Q:'X1  ... S Z00=$G(^IBA(362.4,Z0,0))  ... Q:$S($P(Z00,U,8)="":1,1:$L($P(Z00,U,8))=11)  ... I $P(Z00,U,9)'=4 S X1=0  ; Only continue if general edits are passed  I $$COB^IBCEF(IB)="S" D  . S COBINS=$P($G(^DGCR(399,IB,"M")),U,IBCOB+1)  . I 'COBINS Q  . I IBMCR S IBWHY=1,$P(^DGCR(399,IBIFN,"TX"),U,8)=1  I IBWHY S IBOK=0 G TXMTQ  S IBOK=$$EDIT(IBIFN,.IB,.IBWHY)  G:'IBOK TXMTQ  ; TXMTQ ;  I IBOK S IBOK=$G(^DIC(36,+IB(.07),3))  Q IBOK  ; MRASEC(IBIFN) ; Returns 1 if current bill is secondary to MCR WNR  N IBSEQ,IB,Z  S IB=0  ; Chk if MCR WNR is prev insurer with MRA on file  S IBSEQ=$$COBN^IBCEF(IBIFN)-1  S Z=$$MCRONBIL^IBEFUNC(IBIFN,IBSEQ) I +Z=1,$P(Z,U,2)=1,$$CHK^IBCEMU1(IBIFN) S IB=1  Q IB  ; EDIT(IBIFN,IB,IBWHY) ; Find, execute edits applying to bill to see if transmittable  ; IBIFN = ien of bill in file 399  ; IB = array containing necessary data for xref search from bill  ; subscripted by field # in file 364.4  ;  ; Matrix entries:  ; IB(.03): 1=EDI specific, 2=MRA specific  ; IB(.04): 1=Outpatient or 2=inpatient only (currently defaults to 3)  ; IB(.05): 1=Only institutional or 2=only professional  ; X: Anything valid  ;  ; MRA-EDI IN-OUT INST-PROF  ; Level ------- ------ ---------  ; 1 X X X  ; 2 X X IB(.05)  ; 3 X IB(.04) X  ; 4 X IB(.04) IB(.05)  ; 5 IB(.03) X X  ; 6 IB(.03) X IB(.05)   ; 7 IB(.03) IB(.04) X  ; 8 IB(.03) IB(.04) IB(.05)  ;  N IB0,IB1,IB2,IB3,IB4,IBDA,IBFT,IBPASS,IBSEQ,IBT,IBNOCK  I '$G(IB(.03)) S IBPASS=0 G EDITQ  S IBFT=$$FT^IBCEF(IBIFN)  ;  S IBPASS=1  F IBSEQ=1:1:8 D  Q:'IBPASS  ; Loop thru levels in matrix  . F IB1=1:1:3 Q:'IBPASS  F IB2=1:1:3 Q:'IBPASS  F IB3=1:1:3 Q:'IBPASS  D  .. S IB4=0 F  S IB4=$O(^IBE(364.4,"AD",IB1,IB2,IB3,IB4)) Q:'IB4  I $O(^(IB4,0)) D  Q:'IBPASS  ... S IBDA=0  ... F   S IBDA=$O(^IBE(364.4,"AD",IB1,IB2,IB3,IB4,IBDA)) Q:'IBDA  S IB0=$G(^IBE(364.4,IBDA,0)) I IB0'="",'$D(IBNOCK(IBDA)) D  Q:'IBPASS  .... I $P(IB0,U,2)>DT S IBNOCK(IBDA)="" Q  ; Not activated yet  .... I $P(IB0,U,6),$P(IB0,U,6)'>DT  S IBNOCK(IBDA)="" Q  ; Inactive  .... ;JWS;IB\*2.0\*592;dental form #7, same as CMS-1500  .... I $P(IB0,U,11),IB3'=3,$S(IBFT=3:IB3'=1,IBFT=2:IB3'=2,IBFT=7:IB3'=2,1:0) S IBNOCK(IBDA)="" Q  ; Form type not included - not used for form type rule (0)  .... I IB4=1,'$D(^IBE(364.4,IBDA,3,"B",+IB(.07))) S IBNOCK(IBDA)="" Q  ; Ins not included for rule  .... I IB4=2,$D(^IBE(364.4,IBDA,2,"B",+IB(.07))) S IBNOCK(IBDA)="" Q  ; Ins is excluded from rule  .... S IBT=$G(^IBE(364.4,IBDA,1))  .... ; Code can assume IBIFN, IBDA and IB(.03 thru .05 and .07) exist  .... I IBT'="" X IBT I '$T S IBPASS=0,IBWHY(0)=IBDA,IBWHY=5 EDITQ Q IBPASS  ; STATUS(IBIFN) ; Function returns whether or not bill currently has a status  ; message or EOB message not yet fully reviewed -  ; (only for transmittable bills)  ; IBIFN = ien of bill in file 399  ; Returns:  ; 0 = None found  ; If found, returns a pieced string as follows:  ;  ; [1] ien of transmit bill entry (file 364) associated with an  ; entry in file 361 with an unreviewed status message  ; [2] ien of transmit bill entry (file 364) associated with an  ; entry in file 361.1 with an unreviewed EOB  ;  N IB,Z,Z0  S IB=""  S Z="" F  S Z=$O(^IBM(361,"B",IBIFN,Z),-1) Q:'Z  I $P($G(^IBM(361,Z,0)),U,9)<2,$P(^(0),U,11) S $P(IB,U)=$P(^(0),U,11) Q  ;  S Z="" F  S Z=$O(^IBM(361.1,"B",IBIFN,Z),-1) Q:'Z  I $P($G(^IBM(361.1,Z,0)),U,16)<2,$P(^(0),U,19) S $P(IB,U,2)=$P(^(0),U,19) Q  ;  Q IB  ; TEST(IBIFN) ; Returns 1 if bill IBIFN is a transmission test bill, 0 if not  Q +$S($G(^TMP("IBEDI\_TEST\_BATCH",$J)):1,1:+$P($G(^IBA(364,+$$LAST364(IBIFN),0)),U,7))  ; | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCEF7 | | | | | | | | |
| **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** | | | | | | | | | |
| IBCEF7 ;WOIFO/SS - FORMATTER AND EXTRACTOR SPECIFIC BILL FUNCTIONS ;8/6/03 10:56am  ;;2.0;INTEGRATED BILLING;\*\*232,349,432\*\*;21-MAR-94;Build 192  ;;Per VHA Directive 2004-038, this routine should not be modified.  ; ALLPROV ;called from #364.5 entry "N-ALL CUR/OTH PROVIDER INFO"  ;\*342/TAZ - Added call to LPRV^IBCEF80 for line level providers; restructured due to line length  I +$G(IBXSAVE("PROVINF",IBXIEN))=0 D  . N IBZ  . D PROVIDER(IBXIEN,"C",.IBZ),PROVIDER(IBXIEN,"O",.IBZ) S IBXSAVE("PROVINF",IBXIEN)=IBXIEN M IBXSAVE("PROVINF",IBXIEN)=IBZ  Q  ;for PRV1  ;Input:  ; IB399 ien of #399 PRV1(IB399) ;  N IBN,IBZ,IBZ1,IBZN,IBZD,IBRES,IBIND,IBDEF,IBDEFTYP,IBQ,IBFRMTYP,IBZNAME  S IBFRMTYP=+$$FT^IBCEF(IB399)  S IBN=0,IBIND=0,IBRES="",IBQ=0  S IBDEF=$P($G(^DGCR(399,IB399,"M1")),U,$$COBN^IBCEF(IB399)+1),IBDEFTYP=""  I IBDEF'="" S IBDEFTYP=$$SOP^IBCEP2B(IB399,"")  I IBDEFTYP'="",$$CHCKPRV1^IBCEF73($S(IBFRMTYP=2:2,IBFRMTYP=3:1,1:0),IBDEFTYP)=0 S (IBDEF,IBDEFTYP)=""  I IBDEF'="",IBDEFTYP'="" S IBIND=IBIND+2,$P(IBRES,U,IBIND)=(IBDEFTYP\_U\_IBDEF)  F  S IBN=$O(^IBE(355.97,IBN)) Q:+IBN=0!(IBQ=1) D  . S IBZ=$G(^IBE(355.97,IBN,0)),IBZ1=$G(^(1))  . Q:$P(IBZ,"^",4)=""!$P(IBZ1,U,9) ;if no FACILITY'S DEFAULT ID #  . Q:$P(IBZ1,"^",4)!(IBDEFTYP=$P(IBZ,U,3))  . S IBZN=$P(IBZ,"^",3),IBZNAME=$P(IBZ,"^",1)  . I IBFRMTYP=2 Q:IBZN="1A"!(IBZNAME="MEDICARE PART A") ;1500  . I IBFRMTYP=3 Q:IBZN="1B"!(IBZNAME="MEDICARE PART B") ;UB  . Q:$$CHCKPRV1^IBCEF73($S(IBFRMTYP=2:2,IBFRMTYP=3:1,1:0),IBZN)=0  . I $P(IBZ,"^",2)=0!($P(IBZ,"^",2)=2) D  . . S IBIND=IBIND+2  . . I IBIND>14 S IBQ=1 Q  . . S $P(IBRES,"^",IBIND)=IBZN\_"^"\_$P(IBZ,"^",4)  ;Remove any duplicate entries  N I,Q,QUAL,QUALC,IBRESTMP,SEQ  F I=2:2:($L(IBRES,"^")-1) D  . S QUAL=$P(IBRES,"^",I)  . I $G(IBRESTMP(QUAL))="" S IBRESTMP(QUAL)=$P(IBRES,"^",(I+1))  S Q=2  S I="",QUAL=""  K IBRES  S IBRES=""  S SEQ=0  F  S QUAL=$O(IBRESTMP(QUAL)) Q:QUAL=""  D  . S SEQ=SEQ+2  . S $P(IBRES,"^",SEQ)=QUAL,$P(IBRES,"^",(SEQ+1))=IBRESTMP(QUAL)  Q IBRES  ;  ; creates array of SUBSCR IDs for all "other insurances"  ;Input :  ; IBXIEN - ien in #399  ;Output:  ; IBZOUT(Z) - array with ien of #36  OTHSBID(IBXIEN,IBZOUT) ;  N Z,Z0,Z1,IBZ,C  D F^IBCEF("N-ALL INSURANCE CO 837 ID","IBZ")  F Z=1,2,3 S IBZ(Z)=$$POLICY^IBCEF(IBXIEN,2,$E("PST",Z))  K IBXDATA  S C=$$OTHINS1^IBCEF2(IBXIEN)  F Z=1,2 I $G(IBZ(Z))'="",$E(C,Z) D  . S IBZOUT(Z)=IBZ(+$E(C,Z))  Q  ;Input :  ; IBXIEN - ien in #399  ; IBP - # piece in address string : STR LINE1|STR LINE2|CITY|STATE|ZIP  ;Output:  ; IBARR - output array m by reference ELMADD2(IBXIEN,IBP,IBARR) ;  N IBZZZ,A,CHECK,IB1  I '$D(IBXSAVE("OTH\_INSURED\_ADDR")) D OTHADD2(IBXIEN,.IBZZZ) M IBXSAVE("OTH\_INSURED\_ADDR")=IBZZZ  S IB1=0  F  S IB1=$O(IBXSAVE("OTH\_INSURED\_ADDR",IB1)) Q:'IB1  D  . ;IF ANY PORTION OF ADDRESS IS NULL SET CHECK VALUE, ERASE ENTRY  . S CHECK=0  . F A=1,3,4,5 I $P(IBXSAVE("OTH\_INSURED\_ADDR",IB1),"|",A)="" S CHECK=1 K IBXSAVE("OTH\_INSURED\_ADDR",IB1) Q  . I 'CHECK D  . . I IBP=0 S IBARR(IB1)=$G(IBXSAVE("OTH\_INSURED\_ADDR",IB1)) Q  . . S IBARR(IB1)=$P($G(IBXSAVE("OTH\_INSURED\_ADDR",IB1)),"|",IBP)  Q  ;creates an array with address info for all other insured persons  ;Input :  ; IBXIEN - ien in #399  ;Output:  ; IBZOUT(Z) - array with STR LINE1|STR LINE2|CITY|STATE|ZIP  OTHADD2(IBXIEN,IBZOUT) ;  N C,Z,Z0,Z1,IBZ,IBZIP,IB1,IBDFN1  S IBZOUT=""  D OTHP36^IBCEF72(IBXIEN,.IBZ) ;array with iens of file #36  K IBXDATA  S C=$$OTHINS1^IBCEF2(IBXIEN)  F Z=1,2 I $G(IBZ(Z))'="",$E(C,Z) D  . S IBINS=+IBZ(+$E(C,Z))  . S IBDFN1=$P($G(^DGCR(399,IBXIEN,0)),"^",2)  . S IBZOUT(Z)=$$FR2PAT(IBDFN1,IBINS)  Q  ;Input:  ; IBDFN-patient ien  ; IBINS - input array with insurance pointers to 36  ;Output   ; STR LINE1|STR LINE2|CITY|STATE|ZIP FR2PAT(IBDFN,IBINS) ;information about "other insured" address  N Z3,Z4,Z5,IBZIP  S Z3=$O(^DPT(IBDFN,.312,"B",$G(IBINS),0))  Q:+Z3=0 "||||"  S Z4=$G(^DPT(IBDFN,.312,Z3,3))  S IBZIP=$P($G(^DIC(5,+$P(Z4,"^",9),0)),"^",2)  S Z5=$P(Z4,"^",6,8)\_"^"\_IBZIP\_"^"\_$P(Z4,"^",10)  Q $TR(Z5,"^","|")  ;  ;Input :  ; IBXIEN - ien in #399  ; IBP - # piece in address string : STR LINE1|STR LINE2|CITY|STATE|ZIP  ; if IBP=0 then returns whole string  ;Output:  ; IBARR - output array m by reference ELMADDR(IBXIEN,IBP,IBARR) ;  N IB1,A,CHECK  D:'$D(IBXSAVE("OTH\_PROV\_ADDR")) OTHADDR(IBXIEN)  S IB1=0  F  S IB1=$O(IBXSAVE("OTH\_PROV\_ADDR",IB1)) Q:'IB1  D  . S CHECK=0  . ;EXCLUDE ADD LINE 2 SECOND PC SINCE IT'S OK FOR THAT TO BE EMPTY  . F A=1,3,4,5 I $P(IBXSAVE("OTH\_PROV\_ADDR",IB1),"|",A)="" D  Q  . . ;IF ANY PORTION OF ADDRESS IS NULL SET CHECK VALUE, ERASE ENTRY  . . S CHECK=1 K IBXSAVE("OTH\_PROV\_ADDR",IB1)  . I 'CHECK D  . . I IBP=0 S IBARR(IB1)=$G(IBXSAVE("OTH\_PROV\_ADDR",IB1)) Q  . . S IBARR(IB1)=$P($G(IBXSAVE("OTH\_PROV\_ADDR",IB1)),"|",IBP)  Q  ;  ;creates an array with address info for all insurances  ;Input :  ; IBXIEN - ien in #399  ;Output:  ; IBXSAVE("OTH\_PROV\_ADDR",Z)  OTHADDR(IBXIEN) ;  N C,Z,Z0,Z1,IBZ,IBZIP,IB1,IBINS  D F^IBCEF("N-OTH INSURANCE CO IEN 36") ;array with iens of file #36  M IBZ=IBXDATA  K IBXDATA  S C=$$OTHINS1^IBCEF2(IBXIEN)  F Z=1,2 I $G(IBZ(Z))'="",$E(C,Z) D  . S IBINS=+IBZ(+$E(C,Z))  . S IBZIP=$P($G(^DIC(5,+$P($G(^DIC(36,IBINS,.11)),"^",5),0)),"^",2)  . S IB1=$P($G(^DIC(36,IBINS,.11)),"^",1,2)\_"^"\_$P($G(^DIC(36,IBINS,.11)),"^",4)\_"^"\_IBZIP\_"^"\_$P($G(^DIC(36,IBINS,.11)),"^",6)  . S IBXSAVE("OTH\_PROV\_ADDR",Z)=$TR(IB1,"^","|")  Q  ;  ;Retrieves pointer to get info about the service provider  ;IBIEN399 - ien in #399  ;IBFUNC -function (3-RENDERING,etc)  ;Output: VARIABLE POINTER (PTR;file\_root) PROVPTR(IBIEN399,IBFUNC) ;  ;\*432/TAZ - No longer used for IBXSAVE array setup  N IBN  S IBN=$O(^DGCR(399,IBIEN399,"PRV","B",IBFUNC,0))  I +IBN=0 Q 0  Q $P($G(^DGCR(399,IBIEN399,"PRV",+IBN,0)),"^",2)  ;  ;Retrieves SSN from #200  ;IBPTR- VARIABLE POINTER to #200 PROVSSN(IBIEN399) ;  N IBRETVAL S IBRETVAL=""  N IBPTR,IBFT  F IBFT=1:1:9 D  . S IBPTR=$$PROVPTR(IBIEN399,IBFT)  . S $P(IBRETVAL,"^",IBFT)=$$GETSSN^IBCEF72(IBPTR)  Q IBRETVAL  ;  ;Input:  ; IBPTR- ptr to ^VA(200 or ^IBA(355.93  ;Output:  ; SSN or null GETNMEL(IBFULL,IBEL) ;Get name element  D NAMECOMP^XLFNAME(.IBFULL)  Q $G(IBFULL(IBEL))  ;-  ;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)  ;  ;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) ;  N IBCURR,IBZ,IBRESARR  S IBRESARR=""  S IBCURR=$$COB^IBCEF(IB399) ;current bill payer sequence  Q:IBPROV="A"  ;PATIENT's bill  I IBPROV="C" D  . D:$$ISINSUR^IBCEF71(IBCURR,IB399) PROVINF(IB399,$S(IBCURR="T":3,IBCURR="S":2,IBCURR="P":1,1:1),.IBRESARR,1,IBPROV)  I IBPROV="O" D  . I IBCURR="P" D:$$ISINSUR^IBCEF71("S",IB399) PROVINF(IB399,2,.IBRESARR,1,IBPROV) D:$$ISINSUR^IBCEF71("T",IB399) PROVINF(IB399,3,.IBRESARR,2,IBPROV)  . I IBCURR="S" D:$$ISINSUR^IBCEF71("P",IB399) PROVINF(IB399,1,.IBRESARR,1,IBPROV) D:$$ISINSUR^IBCEF71("T",IB399) PROVINF(IB399,3,.IBRESARR,2,IBPROV)  . I IBCURR="T" D:$$ISINSUR^IBCEF71("P",IB399) PROVINF(IB399,1,.IBRESARR,1,IBPROV) D:$$ISINSUR^IBCEF71("S",IB399) PROVINF(IB399,2,.IBRESARR,2,IBPROV)  M IBRES(IBPROV)=IBRESARR  Q  ; PROVINF(IB399,IBPRNUM,IBRES,IBSORT,IBINSTP) ;  D PROVINF^IBCEF74(IB399,IBPRNUM,.IBRES,IBSORT,IBINSTP)  Q  ; PSPRV(IBIFN) ; Returns information for bill ien IBIFN for purchased svc   ; Returns 4 digit data in following format:  ; 1st digit: 0 if not outside facility  ; 1 if outside facility  ; 2nd digit: 0 if not non-VA provider for rendering/attending  ; 1 if non-VA provider for rendering/attending  ; 3rd digit: 0 if not purchased svc  ; 1 if purchased svc  ; 4th digit: 0 if 1500 bill  ; 1 if UB bill  N IBSVC,Z,Z0,IBU2  S IBSVC="000"\_+$$INSFT^IBCEU5(IBIFN),IBU2=$G(^DGCR(399,IBIFN,"U2"))  I $P(IBU2,U,10) S $E(IBSVC,1)=1 ; NON-VA FACILITY  S Z=($$FT^IBCEF(IBIFN)=3)+3,Z0=+$O(^DGCR(399,IBIFN,"PRV","B",Z,0))  I $P($G(^DGCR(399,IBIFN,"PRV",Z0,0)),U,2)["IBA(355.93" S $E(IBSVC,2)=1  I $P(IBU2,U,11)>0,$P(IBU2,U,11)'>2 S $E(IBSVC,3)=1 PSPRVQ Q IBSVC  ; CHKADD ;CHECK ALL ADDRESS ELEMENTS PRESENT IF NOT KILL ALL ADDRESS ELEMENTS  ;EXPECT IBXSAVE("CADR") AS SOURCE ARRAY  N Z,CHECK  S Z="",CHECK=0  F Z=1,4,5,6 D  . I $P($G(IBXSAVE("CADR")),"^",Z)="" S CHECK=1  I CHECK=1 S IBXSAVE("CADR")=""  Q  ; | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCEF7 ;WOIFO/SS - FORMATTER AND EXTRACTOR SPECIFIC BILL FUNCTIONS ;8/6/03 10:56am  ;;2.0;INTEGRATED BILLING;\*\*232,349,432,**592**\*\*;21-MAR-94;Build 192  ;;Per VHA Directive 2004-038, this routine should not be modified.  ; ALLPROV ;called from #364.5 entry "N-ALL CUR/OTH PROVIDER INFO"  ;\*342/TAZ - Added call to LPRV^IBCEF80 for line level providers; restructured due to line length  I +$G(IBXSAVE("PROVINF",IBXIEN))=0 D  . N IBZ  . D PROVIDER(IBXIEN,"C",.IBZ),PROVIDER(IBXIEN,"O",.IBZ) S IBXSAVE("PROVINF",IBXIEN)=IBXIEN M IBXSAVE("PROVINF",IBXIEN)=IBZ  Q  ;for PRV1  ;Input:  ; IB399 ien of #399 PRV1(IB399) ;  N IBN,IBZ,IBZ1,IBZN,IBZD,IBRES,IBIND,IBDEF,IBDEFTYP,IBQ,IBFRMTYP,IBZNAME  S IBFRMTYP=+$$FT^IBCEF(IB399)  S IBN=0,IBIND=0,IBRES="",IBQ=0  S IBDEF=$P($G(^DGCR(399,IB399,"M1")),U,$$COBN^IBCEF(IB399)+1),IBDEFTYP=""  I IBDEF'="" S IBDEFTYP=$$SOP^IBCEP2B(IB399,"")  **;JRA IB\*2.0\*592 Treat new Dental form 7 (J430D) same as CMS-1500**  **;I** IBDEFTYP'="",$$CHCKPRV1^IBCEF73($S(IBFRMTYP=2:2,IBFRMTYP=3:1,1:0),IBDEFTYP)=0 S (IBDEF,IBDEFTYP)="" ;**JRA IB\*2.0\*592 ';'**  I IBDEFTYP'="",$$CHCKPRV1^IBCEF73($S((IBFRMTYP=2**!(IBFRMTYP=7)**):2,IBFRMTYP=3:1,1:0),IBDEFTYP)=0 S (IBDEF,IBDEFTYP)=""  **;JRA IB\*2.0\*592**  I IBDEF'="",IBDEFTYP'="" S IBIND=IBIND+2,$P(IBRES,U,IBIND)=(IBDEFTYP\_U\_IBDEF)  F  S IBN=$O(^IBE(355.97,IBN)) Q:+IBN=0!(IBQ=1) D  . S IBZ=$G(^IBE(355.97,IBN,0)),IBZ1=$G(^(1))  . Q:$P(IBZ,"^",4)=""!$P(IBZ1,U,9) ;if no FACILITY'S DEFAULT ID #  . Q:$P(IBZ1,"^",4)!(IBDEFTYP=$P(IBZ,U,3))  . S IBZN=$P(IBZ,"^",3),IBZNAME=$P(IBZ,"^",1)  **. ;**I IBFRMTYP=2 Q:IBZN="1A"!(IBZNAME="MEDICARE PART A") ;1500 **;JRA IB\*2.0\*592 ';'**  . I IBFRMTYP=2**!(IBFRMTYP=7)** Q:IBZN="1A"!(IBZNAME="MEDICARE PART A") ;1500 **or J430D ;JRA IB\*2.0\*592**  . I IBFRMTYP=3 Q:IBZN="1B"!(IBZNAME="MEDICARE PART B") ;UB  . **;**Q:$$CHCKPRV1^IBCEF73($S(IBFRMTYP=2:2,IBFRMTYP=3:1,1:0),IBZN)=0 ;**JRA IB\*2.0\*592 ';'**  . Q:$$CHCKPRV1^IBCEF73($S((IBFRMTYP=2**!(IBFRMTYP=7)):**2,IBFRMTYP=3:1,1:0),IBZN)=0 **;JRA IB\*2.0\*592**  . I $P(IBZ,"^",2)=0!($P(IBZ,"^",2)=2) D  . . S IBIND=IBIND+2  . . I IBIND>14 S IBQ=1 Q  . . S $P(IBRES,"^",IBIND)=IBZN\_"^"\_$P(IBZ,"^",4)  ;Remove any duplicate entries  N I,Q,QUAL,QUALC,IBRESTMP,SEQ  F I=2:2:($L(IBRES,"^")-1) D  . S QUAL=$P(IBRES,"^",I)  . I $G(IBRESTMP(QUAL))="" S IBRESTMP(QUAL)=$P(IBRES,"^",(I+1))  S Q=2  S I="",QUAL=""  K IBRES  S IBRES=""  S SEQ=0  F  S QUAL=$O(IBRESTMP(QUAL)) Q:QUAL=""  D  . S SEQ=SEQ+2  . S $P(IBRES,"^",SEQ)=QUAL,$P(IBRES,"^",(SEQ+1))=IBRESTMP(QUAL)  Q IBRES  ;  ; creates array of SUBSCR IDs for all "other insurances"  ;Input :  ; IBXIEN - ien in #399  ;Output:  ; IBZOUT(Z) - array with ien of #36  OTHSBID(IBXIEN,IBZOUT) ;  N Z,Z0,Z1,IBZ,C  D F^IBCEF("N-ALL INSURANCE CO 837 ID","IBZ")  F Z=1,2,3 S IBZ(Z)=$$POLICY^IBCEF(IBXIEN,2,$E("PST",Z))  K IBXDATA  S C=$$OTHINS1^IBCEF2(IBXIEN)  F Z=1,2 I $G(IBZ(Z))'="",$E(C,Z) D  . S IBZOUT(Z)=IBZ(+$E(C,Z))  Q  ;Input :  ; IBXIEN - ien in #399  ; IBP - # piece in address string : STR LINE1|STR LINE2|CITY|STATE|ZIP  ;Output:  ; IBARR - output array m by reference ELMADD2(IBXIEN,IBP,IBARR) ;  N IBZZZ,A,CHECK,IB1  I '$D(IBXSAVE("OTH\_INSURED\_ADDR")) D OTHADD2(IBXIEN,.IBZZZ) M IBXSAVE("OTH\_INSURED\_ADDR")=IBZZZ  S IB1=0  F  S IB1=$O(IBXSAVE("OTH\_INSURED\_ADDR",IB1)) Q:'IB1  D  . ;IF ANY PORTION OF ADDRESS IS NULL SET CHECK VALUE, ERASE ENTRY  . S CHECK=0  . F A=1,3,4,5 I $P(IBXSAVE("OTH\_INSURED\_ADDR",IB1),"|",A)="" S CHECK=1 K IBXSAVE("OTH\_INSURED\_ADDR",IB1) Q  . I 'CHECK D  . . I IBP=0 S IBARR(IB1)=$G(IBXSAVE("OTH\_INSURED\_ADDR",IB1)) Q  . . S IBARR(IB1)=$P($G(IBXSAVE("OTH\_INSURED\_ADDR",IB1)),"|",IBP)  Q  ;creates an array with address info for all other insured persons  ;Input :  ; IBXIEN - ien in #399  ;Output:  ; IBZOUT(Z) - array with STR LINE1|STR LINE2|CITY|STATE|ZIP  OTHADD2(IBXIEN,IBZOUT) ;  N C,Z,Z0,Z1,IBZ,IBZIP,IB1,IBDFN1  S IBZOUT=""  D OTHP36^IBCEF72(IBXIEN,.IBZ) ;array with iens of file #36  K IBXDATA  S C=$$OTHINS1^IBCEF2(IBXIEN)  F Z=1,2 I $G(IBZ(Z))'="",$E(C,Z) D  . S IBINS=+IBZ(+$E(C,Z))  . S IBDFN1=$P($G(^DGCR(399,IBXIEN,0)),"^",2)  . S IBZOUT(Z)=$$FR2PAT(IBDFN1,IBINS)  Q  ;Input:  ; IBDFN-patient ien  ; IBINS - input array with insurance pointers to 36  ;Output   ; STR LINE1|STR LINE2|CITY|STATE|ZIP FR2PAT(IBDFN,IBINS) ;information about "other insured" address  N Z3,Z4,Z5,IBZIP  S Z3=$O(^DPT(IBDFN,.312,"B",$G(IBINS),0))  Q:+Z3=0 "||||"  S Z4=$G(^DPT(IBDFN,.312,Z3,3))  S IBZIP=$P($G(^DIC(5,+$P(Z4,"^",9),0)),"^",2)  S Z5=$P(Z4,"^",6,8)\_"^"\_IBZIP\_"^"\_$P(Z4,"^",10)  Q $TR(Z5,"^","|")  ;  ;Input :  ; IBXIEN - ien in #399  ; IBP - # piece in address string : STR LINE1|STR LINE2|CITY|STATE|ZIP  ; if IBP=0 then returns whole string  ;Output:  ; IBARR - output array m by reference ELMADDR(IBXIEN,IBP,IBARR) ;  N IB1,A,CHECK  D:'$D(IBXSAVE("OTH\_PROV\_ADDR")) OTHADDR(IBXIEN)  S IB1=0  F  S IB1=$O(IBXSAVE("OTH\_PROV\_ADDR",IB1)) Q:'IB1  D  . S CHECK=0  . ;EXCLUDE ADD LINE 2 SECOND PC SINCE IT'S OK FOR THAT TO BE EMPTY  . F A=1,3,4,5 I $P(IBXSAVE("OTH\_PROV\_ADDR",IB1),"|",A)="" D  Q  . . ;IF ANY PORTION OF ADDRESS IS NULL SET CHECK VALUE, ERASE ENTRY  . . S CHECK=1 K IBXSAVE("OTH\_PROV\_ADDR",IB1)  . I 'CHECK D  . . I IBP=0 S IBARR(IB1)=$G(IBXSAVE("OTH\_PROV\_ADDR",IB1)) Q  . . S IBARR(IB1)=$P($G(IBXSAVE("OTH\_PROV\_ADDR",IB1)),"|",IBP)  Q  ;  ;creates an array with address info for all insurances  ;Input :  ; IBXIEN - ien in #399  ;Output:  ; IBXSAVE("OTH\_PROV\_ADDR",Z)  OTHADDR(IBXIEN) ;  N C,Z,Z0,Z1,IBZ,IBZIP,IB1,IBINS  D F^IBCEF("N-OTH INSURANCE CO IEN 36") ;array with iens of file #36  M IBZ=IBXDATA  K IBXDATA  S C=$$OTHINS1^IBCEF2(IBXIEN)  F Z=1,2 I $G(IBZ(Z))'="",$E(C,Z) D  . S IBINS=+IBZ(+$E(C,Z))  . S IBZIP=$P($G(^DIC(5,+$P($G(^DIC(36,IBINS,.11)),"^",5),0)),"^",2)  . S IB1=$P($G(^DIC(36,IBINS,.11)),"^",1,2)\_"^"\_$P($G(^DIC(36,IBINS,.11)),"^",4)\_"^"\_IBZIP\_"^"\_$P($G(^DIC(36,IBINS,.11)),"^",6)  . S IBXSAVE("OTH\_PROV\_ADDR",Z)=$TR(IB1,"^","|")  Q  ;  ;Retrieves pointer to get info about the service provider  ;IBIEN399 - ien in #399  ;IBFUNC -function (3-RENDERING,etc)  ;Output: VARIABLE POINTER (PTR;file\_root) PROVPTR(IBIEN399,IBFUNC) ;  ;\*432/TAZ - No longer used for IBXSAVE array setup  N IBN  S IBN=$O(^DGCR(399,IBIEN399,"PRV","B",IBFUNC,0))  I +IBN=0 Q 0  Q $P($G(^DGCR(399,IBIEN399,"PRV",+IBN,0)),"^",2)  ;  ;Retrieves SSN from #200  ;IBPTR- VARIABLE POINTER to #200 PROVSSN(IBIEN399) ;  N IBRETVAL S IBRETVAL=""  N IBPTR,IBFT  F IBFT=1:1:9 D  . S IBPTR=$$PROVPTR(IBIEN399,IBFT)  . S $P(IBRETVAL,"^",IBFT)=$$GETSSN^IBCEF72(IBPTR)  Q IBRETVAL  ;  ;Input:  ; IBPTR- ptr to ^VA(200 or ^IBA(355.93  ;Output:  ; SSN or null GETNMEL(IBFULL,IBEL) ;Get name element  D NAMECOMP^XLFNAME(.IBFULL)  Q $G(IBFULL(IBEL))  ;-  ;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)  ;  ;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) ;  N IBCURR,IBZ,IBRESARR  S IBRESARR=""  S IBCURR=$$COB^IBCEF(IB399) ;current bill payer sequence  Q:IBPROV="A"  ;PATIENT's bill  I IBPROV="C" D  . D:$$ISINSUR^IBCEF71(IBCURR,IB399) PROVINF(IB399,$S(IBCURR="T":3,IBCURR="S":2,IBCURR="P":1,1:1),.IBRESARR,1,IBPROV)  I IBPROV="O" D  . I IBCURR="P" D:$$ISINSUR^IBCEF71("S",IB399) PROVINF(IB399,2,.IBRESARR,1,IBPROV) D:$$ISINSUR^IBCEF71("T",IB399) PROVINF(IB399,3,.IBRESARR,2,IBPROV)  . I IBCURR="S" D:$$ISINSUR^IBCEF71("P",IB399) PROVINF(IB399,1,.IBRESARR,1,IBPROV) D:$$ISINSUR^IBCEF71("T",IB399) PROVINF(IB399,3,.IBRESARR,2,IBPROV)  . I IBCURR="T" D:$$ISINSUR^IBCEF71("P",IB399) PROVINF(IB399,1,.IBRESARR,1,IBPROV) D:$$ISINSUR^IBCEF71("S",IB399) PROVINF(IB399,2,.IBRESARR,2,IBPROV)  M IBRES(IBPROV)=IBRESARR  Q  ; PROVINF(IB399,IBPRNUM,IBRES,IBSORT,IBINSTP) ;  D PROVINF^IBCEF74(IB399,IBPRNUM,.IBRES,IBSORT,IBINSTP)  Q  ; PSPRV(IBIFN) ; Returns information for bill ien IBIFN for purchased svc   ; Returns 4 digit data in following format:  ; 1st digit: 0 if not outside facility  ; 1 if outside facility  ; 2nd digit: 0 if not non-VA provider for rendering/attending  ; 1 if non-VA provider for rendering/attending  ; 3rd digit: 0 if not purchased svc  ; 1 if purchased svc  ; 4th digit: 0 if 1500 bill  ; 1 if UB bill  N IBSVC,Z,Z0,IBU2  S IBSVC="000"\_+$$INSFT^IBCEU5(IBIFN),IBU2=$G(^DGCR(399,IBIFN,"U2"))  I $P(IBU2,U,10) S $E(IBSVC,1)=1 ; NON-VA FACILITY  S Z=($$FT^IBCEF(IBIFN)=3)+3,Z0=+$O(^DGCR(399,IBIFN,"PRV","B",Z,0))  I $P($G(^DGCR(399,IBIFN,"PRV",Z0,0)),U,2)["IBA(355.93" S $E(IBSVC,2)=1  I $P(IBU2,U,11)>0,$P(IBU2,U,11)'>2 S $E(IBSVC,3)=1 PSPRVQ Q IBSVC  ; CHKADD ;CHECK ALL ADDRESS ELEMENTS PRESENT IF NOT KILL ALL ADDRESS ELEMENTS  ;EXPECT IBXSAVE("CADR") AS SOURCE ARRAY  N Z,CHECK  S Z="",CHECK=0  F Z=1,4,5,6 D  . I $P($G(IBXSAVE("CADR")),"^",Z)="" S CHECK=1  I CHECK=1 S IBXSAVE("CADR")=""  Q  ; | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCEF71 | | | | | | | | |
| **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** | | | | | | | | | |
| IBCEF71 ;WOIFO/SS - FORMATTER AND EXTRACTOR SPECIFIC BILL FUNCTIONS ;31-JUL-03  ;;2.0;INTEGRATED BILLING;\*\*232,155,288,320,349,432\*\*;21-MAR-94;Build 192  ;;Per VHA Directive 2004-038, this routine should not be modified.  ;  ;---------  ;OTHPAYC - from FORMAT code for OP1,OP2 ...  ;Input:  ;IBXIEN - ien #399  ;IBSAVE - "in" array (i.e. IBXSAVE)  ;IBDATA - "out" array (i.e. IBXDATA)  ;IBFUNC - FUNCTION from #399 (1-refering,2-operating,etc)  ;IBVAL - output value  ;Output:  ; IBDATA with formatted output OTHPAYC(IBXIEN,IBSAVE,IBDATA,IBFUNC,IBVAL) ;  N IB1,IB2,IBINS,IBFL  S IBFL=$S(IBFUNC=3!(IBFUNC=4):1,1:0)  F IB1=1,2 D  . I $$ISINSUR($G(IBSAVE("PROVINF",IBXIEN,"O",IB1)),IBXIEN) D  Q  ;don't create anything if no such insurance  .. ;\*432/TAZ Attending/Rendering is no longer either/or so there can be both  .. ;I IBFL S IBFUNC=$S($O(IBSAVE("PROVINF",IBXIEN,"O",IB1,3,0)):3,1:4)  .. S:$O(IBSAVE("PROVINF",IBXIEN,"O",IB1,IBFUNC,0)) IBDATA(IB1)=IBVAL  Q  ;----  ;OTHPAYV - called from FORMAT code for OP1,OP2 ...  ;Input:  ;IBXIEN - ien #399  ;IBSAVE - "in" array (i.e. IBXSAVE)  ;IBDATA - "out" array (i.e. IBXDATA)  ;IBFUNC - FUNCTION from #399 (1-refering, 2-operating, etc)  ;IBSEQN - seq # of ID/QUAL  ;IBFLDTYP  ; "I" - ID "Q" - ID QUAL  ;Output:  ; IBDATA with formatted output OTHPAYV(IBXIEN,IBSAVE,IBDATA,IBFUNC,IBFLDTYP,IBSEQN) ;  N IB1,IB2,IBPIECE,IBINS,IBFL  S IBFL=$S(IBFUNC=3!(IBFUNC=4):1,1:0)  S IBPIECE=$S(IBFLDTYP="I":4,IBFLDTYP="Q":3,1:3)  F IB1=1,2 D  . I $$ISINSUR($G(IBSAVE("PROVINF",IBXIEN,"O",IB1)),IBXIEN) D  Q  ;don't create anything if there is no such insurance  .. ;\*432/TAZ Attending/Rendering is no longer either/or so there can be both  .. ;I IBFL S IBFUNC=$S($O(IBSAVE("PROVINF",IBXIEN,"O",IB1,3,0)):3,1:4),IBFL=0  .. S IBDATA(IB1)=$P($G(IBSAVE("PROVINF",IBXIEN,"O",IB1,IBFUNC,IBSEQN)),U,IBPIECE)  Q  ;  ;chk for ins  ;Input:  ; IBINS = "P","S","T"  ; IBXIEN - ien file #399  ;Output:  ; returns 1-exists , 0-doesn't ISINSUR(IBINS,IBXIEN) ;  N IBINSNOD  S IBINSNOD=$S(IBINS="P":"I1",IBINS="S":"I2",IBINS="T":"I3",1:"")  I IBINSNOD="" Q 0  Q $D(^DGCR(399,IBXIEN,IBINSNOD))  ;  ;---PRACT----  ;Get list of all 355.9 or 355.93 records for prov  ;Input:   ;IB399INS - ins co for bill to match PRACTIONER from 355.9  ;IB399FRM - form type (0=unknwn/both,1=UB,2=1500) to   ; match PRACTIONER from 355.9  ;IB399CAR - BILL CARE (0=unknwn or both inp/outp,1=inpatient,  ; 2=outpatient/3=Rx) to match PROV from 355.9  ; OR - DIVISION PTR to file 40.8 for entries in file 355.92  ;IBPROV - VARIABLE PTR VA prov  ;IBARR - array by reference for result  ;IBPROVTP- function (2-operating, 3-RENDERING,etc 0-facility)  ;IBINSTP - "C" -current ins , "O"-other  ;IBFILE - 355.92 for facility ids or 355.9 (default) for provider ids  ;IBINS - 1 if to include ids for the ins co for all provs  ;Ouput:  ;IBARR - array by ref for result  ; prov var ptr^ins ptr^X12 id cd^ID^form typ^care typ or division ptr^st ptr^id rec ptr^id type ptr PRACT(IB399INS,IB399FRM,IB399CAR,IBPROV,IBARR,IBPROVTP,IBINSTP,IBFILE,IBINS) ;  N IB1,IB2,IBDAT,IBF,IBFX,IB3559,IBINSCO,IBFRMTYP,IBIDTYP,IBID,IBIDT,IBDIV,IBQ,IBS1,IBS2,IBARRX,Z,Z1,Z2,IBCARE  I $G(IBFILE)="" S IBFILE=355.9  S IBINS=$G(IBINS)  S (IBARR,IB3559,IB1)=0  F IBF="",1 Q:IBF=1&$S(IBFILE'=355.9:1,1:'IBINS) S IBFX=IBFILE\_IBF F IB2=1:1 S IB3559=$O(^IBA(IBFX,"B",$S(IBFILE=355.9&(IBF=""):IBPROV,1:IB399INS),IB3559)) Q:IB3559=""  D  . S IBINSCO=$P($G(^IBA(IBFX,IB3559,0)),"^",$S(IBFILE=355.9&(IBF=""):2,1:1)) ;ins co. ptr  . I IBINSCO'="" I IBINSCO'=IB399INS Q  ;exclude if different ins  . S:IBINSCO="" IBINSCO="NONE" ;NONE will be included in the array  . S IBFRMTYP=+$P($G(^IBA(IBFX,IB3559,0)),"^",4) ;form type (0=both,1=UB,2=1500)  . I '(IBFRMTYP=0!(IB399FRM=0)) Q:IBFRMTYP'=IB399FRM  ;exclude if not "both" and different  . S IBCARE=+$P($G(^IBA(IBFX,IB3559,0)),"^",5) ;0=both(inp and outp),1=inp,2=outp,3=prescr -- OR -- division ptr  . I $S(IBFILE=355.92:0,1:IBCARE=3) I IB399CAR'=3 Q  ; Id is only for Rx  . I $S(IBFILE=355.92:0,1:IBCARE=1!(IBCARE=2)) I IB399CAR=1!(IB399CAR=2) Q:IBCARE'=IB399CAR  ;both is OK  . I IBFILE=355.92,IBCARE Q:IB399CAR'=IBCARE  ; Division doesn't match  . S IBIDTYP=+$P($G(^IBA(IBFX,IB3559,0)),"^",6) ;prov ID type  . I IBFILE=355.9,IBIDTYP=$$TAXID^IBCEP8(),$S(IBPROV["VA(200":1,1:$P($G(^IBA(355.93,+IBPROV,0)),U,2)=2) Q  ; Don't extract tax id # id for indiv prov  . S IBIDT=IBIDTYP  . S IBIDTYP=$P($G(^IBE(355.97,IBIDTYP,0)),"^",3)  . Q:$P($G(^IBE(355.97,+IBIDT,1)),U,9)  . Q:IBFILE=355.9&(IBIDTYP="X4") ;exclude CLIA #  . S IBID=$P($G(^IBA(IBFX,IB3559,0)),"^",7) ;prov ID value  . I $G(IBPROVTP)'="",$G(IBINSTP)'="",IBPROVTP'=0 I '$$CHCKSEC^IBCEF73(IB399FRM,IBPROVTP,IBINSTP,IBIDTYP) Q  ; No qualifier chk for fac  . I IBID'="" S IBDAT=IBPROV\_"^"\_IBINSCO\_"^"\_IBIDTYP\_"^"\_IBID\_"^"\_IBFRMTYP\_"^"\_IBCARE\_"^"\_"^"\_IB3559\_U\_IBIDT,IBS2=$S(IBFX'=355.91:"",1:"INS DEF^")\_IB3559  . I IBFILE'=355.92,IBID'="",IB399CAR=3 S IBQ=0 D  Q:IBQ  .. I $G(IBARRX(IBIDT))!(IBCARE=1) S IBQ=1 Q  .. I IBCARE=3&(IB399CAR=3) S IBARRX(IBIDT)=1 Q  ; Rx match  .. I IBCARE=0!(IBCARE=2) S IBARRX(IBIDT,IBINSCO,IBS2)=IBDAT,IBQ=1 Q  . I IBID'="" S IBARR(IBINSCO,IBS2)=IBDAT  ;  I IB399CAR=3 S Z=0 F  S Z=$O(IBARRX(Z)) Q:'Z  I '$G(IBARRX(Z)) D  . S Z1="" F  S Z1=$O(IBARRX(Z,Z1)) Q:Z1=""  S Z2="" F  S Z2=$O(IBARRX(Z,Z1,Z2)) Q:Z2=""  S IBARR(Z1,Z2)=IBARRX(Z,Z1,Z2)  ;  I IBPROV["VA(200," D  ; Get lic #s from file 2 for VA providers  . N Z,IBLIC  . S IBLIC=+IBPROV,IBLIC=$$GETLIC^IBCEP5D(.IBLIC)  . S IBIDTYP=$P($G(^IBE(355.97,+$$STLIC^IBCEP8(),0)),U,3)  . S Z=0 F  S Z=$O(IBLIC(Z)) Q:'Z  S:$$CHCKSEC^IBCEF73(IB399FRM,IBPROVTP,IBINSTP,IBIDTYP) IBARR("NONE","LIC"\_Z\_"^"\_IBPROV)=IBPROV\_U\_"NONE"\_U\_IBIDTYP\_U\_IBLIC(Z)\_U\_"0"\_U\_"0"\_U\_Z\_U\_U\_+$$STLIC^IBCEP8()  I IBPROV["IBA(355.93" D  . Q:$P($G(^IBA(355.93,+IBPROV,0)),U,12)=""  . S IBIDTYP=$P($G(^IBE(355.97,+$$STLIC^IBCEP8(),0)),U,3)  . I $$CHCKSEC^IBCEF73(IB399FRM,IBPROVTP,IBINSTP,IBIDTYP) D  . . S IBARR("NONE","LIC"\_$P($G(^DIC(5,+$P(^IBA(355.93,+IBPROV,0),U,7),0)),U,2)\_"^"\_IBPROV)=IBPROV\_U\_"NONE"\_U\_IBIDTYP\_U\_$P(^IBA(355.93,+IBPROV,0),U,12)\_U\_"0"\_U\_"0"\_U\_$P(^IBA(355.93,+IBPROV,0),U,7)\_U\_U\_+IBPROV  Q  ; ALLPRFAC(IBXIEN,IBXSAVE) ; Return all non-VA/outside facility prov ids  ; and all VA alternate prov ids  ; IBXIEN = ien file 399  ; IBXSAVE = subscripted array returned  N IBPROV,IBFRMTYP,IBCARE,IBRETARR,IBRET1,IBCOBN,Z,Z0,Z1,ZZ  K IBXSAVE("PROVINF\_FAC",IBXIEN) ; Always rebuild this  S IBCOBN=+$$COBN^IBCEF(IBXIEN)  S IBFRMTYP=$$FT^IBCEF(IBXIEN),IBFRMTYP=$S(IBFRMTYP=2:2,IBFRMTYP=3:1,1:0)  S IBPROV=$P($G(^DGCR(399,IBXIEN,"U2")),U,10)  ; IB patch 320 - Build IBPROV variable better when a non-VA facility exists  I IBPROV S IBPROV=IBPROV\_";IBA(355.93,"  I 'IBPROV S IBCARE=$P($G(^DGCR(399,IBXIEN,0)),U,22)  I IBPROV D  . S IBCARE=$S($$ISRX^IBCEF1(IBXIEN):3,1:0) ;if Rx refill bill  . S:IBCARE=0 IBCARE=$$INPAT^IBCEF(IBXIEN,1) S:'IBCARE IBCARE=2 ;1-inp, 2-out  F Z=1:1:3 K IBRETARR I $G(^DGCR(399,IBXIEN,"I"\_Z)) D  . D PRACT(+^DGCR(399,IBXIEN,"I"\_Z),IBFRMTYP,IBCARE,IBPROV,.IBRETARR,0,$S(Z=IBCOBN:"C",1:"O"),$S('IBPROV:355.92,1:355.9))  . K IBRET1  . S Z0="" F  S Z0=$O(IBRETARR(Z0)) Q:Z0=""  S Z1="" F  S Z1=$O(IBRETARR(Z0,Z1)) Q:Z1=""  D  .. ; Sort by div/id type  .. S IBRET1($S(IBPROV:0,1:+$P(IBRETARR(Z0,Z1),U,6)),+$P(IBRETARR(Z0,Z1),U,9))=IBRETARR(Z0,Z1)  .. Q  . ;  . S Z0=$O(IBRET1(""),-1) Q:Z0=""  D  .. ; IB patch 320 - loop thru all ID's  .. S Z1="" F  S Z1=$O(IBRET1(Z0,Z1)) Q:Z1=""  D  ... I Z=IBCOBN S IBXSAVE("PROVINF\_FAC",IBXIEN,"C",1,0,$O(IBXSAVE("PROVINF\_FAC",IBXIEN,"C",1,0," "),-1)+1)=IBRET1(Z0,Z1) Q  ... S ZZ=$S(Z=1:1,Z=2:(IBCOBN=3)+1,1:2)  ... S IBXSAVE("PROVINF\_FAC",IBXIEN,"O",ZZ,0,$O(IBXSAVE("PROVINF\_FAC",IBXIEN,"O",ZZ,0," "),-1)+1)=IBRET1(Z0,Z1),IBXSAVE("PROVINF\_FAC",IBXIEN,"O",ZZ)=$E("PST",Z)  ... Q  .. Q  . Q  ;  S IBXSAVE("PROVINF\_FAC",IBXIEN)=IBXIEN,IBXSAVE("PROVINF\_FAC",IBXIEN,"C",1)=$E("PST",IBCOBN)  Q  ; OTHID(IBXSAVE,IBXDATA,IBXIEN,PRIDSEQ,PRTYP,IBQ,IBFAC) ; From data in IBXSAVE,  ; determine id or qualifier to output in the 837 records OP\*  ; Returns IBXDATA array IBXDATA(n)=data  ; IBXIEN = ien of the bill-file 399  ; PRIDSEQ = sequence of the payer id needed  ; PRTYP = provider type to check for data  ; IBQ = 1 if qualifier needed, 0/null if id needed  ; IBFAC = 1 if facility id, 0 for individual provider id  ;   N Z,Z0,Z1  S Z0="PROVINF"\_$S('$G(IBFAC):"",1:"\_FAC"),Z1=$S($G(IBQ):3,1:4)  S Z=0 F  S Z=$O(IBXSAVE("OSQ",Z)) Q:'Z  D  . I $P($G(IBXSAVE(Z0,IBXIEN,"O",Z,+$G(PRTYP),+$G(PRIDSEQ))),U,4)'="" S IBXDATA(IBXSAVE("OSQ",Z))=$P(IBXSAVE(Z0,IBXIEN,"O",Z,+$G(PRTYP),+$G(PRIDSEQ)),U,Z1)  Q  ; SETSEQ(IBXIEN,IBXSAVE,IBXDATA,PRTYP,IBFAC,IBOP) ; Sets up IBXSAVE("OSQ")  ; array for other id seq in 837 records OP\*  ; Returns IBXDATA(n)=cob seq indicator for ids  ; IBXIEN = ien of bill-399  ; PRTYP = the provider type to check for data for indiv provider  ; IBFAC = 1 if facility id, 0 for individual provider id  ; IBOP = segement # in OP being output  N C,Z,Z0,Z1,OK  S C=0,Z0="PROVINF"\_$S('$G(IBFAC):"",1:"\_FAC")  S:$G(IBFAC) PRTYP=0  S Z=0 F  S Z=$O(IBXSAVE(Z0,IBXIEN,"O",Z)) Q:'Z  S OK=0 D  . N Z1 F Z1=1:1 Q:'$D(IBXSAVE(Z0,IBXIEN,"O",Z,+$G(PRTYP),Z1)) I $P(IBXSAVE(Z0,IBXIEN,"O",Z,+$G(PRTYP),Z1),U,4)'="""" S OK=1 Q  . I OK S C=C+1,IBXSAVE("OSQ",Z)=C  S Z=0 F  S Z=$O(IBXSAVE("OSQ",Z)) Q:'Z  S IBXDATA(IBXSAVE("OSQ",Z))=$G(IBXSAVE(Z0,IBXIEN,"O",Z)) D:IBXSAVE("OSQ",Z)>1 ID^IBCEF2(IBXSAVE("OSQ",Z),"OP"\_$G(IBOP)\_" ")  Q  ; PSPRV(IBIFN) ;  Q $$PSPRV^IBCEF7(IBIFN) ; Moved  ; | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCEF71 ;WOIFO/SS - FORMATTER AND EXTRACTOR SPECIFIC BILL FUNCTIONS ;31-JUL-03  ;;2.0;INTEGRATED BILLING;\*\*232,155,288,320,349,432,**592**\*\*;21-MAR-94;Build 192  ;;Per VHA Directive 2004-038, this routine should not be modified.  ;  ;---------  ;OTHPAYC - from FORMAT code for OP1,OP2 ...  ;Input:  ;IBXIEN - ien #399  ;IBSAVE - "in" array (i.e. IBXSAVE)  ;IBDATA - "out" array (i.e. IBXDATA)  ;IBFUNC - FUNCTION from #399 (1-refering,2-operating,etc)  ;IBVAL - output value  ;Output:  ; IBDATA with formatted output OTHPAYC(IBXIEN,IBSAVE,IBDATA,IBFUNC,IBVAL) ;  N IB1,IB2,IBINS,IBFL  S IBFL=$S(IBFUNC=3!(IBFUNC=4):1,1:0)  F IB1=1,2 D  . I $$ISINSUR($G(IBSAVE("PROVINF",IBXIEN,"O",IB1)),IBXIEN) D  Q  ;don't create anything if no such insurance  .. ;\*432/TAZ Attending/Rendering is no longer either/or so there can be both  .. ;I IBFL S IBFUNC=$S($O(IBSAVE("PROVINF",IBXIEN,"O",IB1,3,0)):3,1:4)  .. S:$O(IBSAVE("PROVINF",IBXIEN,"O",IB1,IBFUNC,0)) IBDATA(IB1)=IBVAL  Q  ;----  ;OTHPAYV - called from FORMAT code for OP1,OP2 ...  ;Input:  ;IBXIEN - ien #399  ;IBSAVE - "in" array (i.e. IBXSAVE)  ;IBDATA - "out" array (i.e. IBXDATA)  ;IBFUNC - FUNCTION from #399 (1-refering, 2-operating, etc)  ;IBSEQN - seq # of ID/QUAL  ;IBFLDTYP  ; "I" - ID "Q" - ID QUAL  ;Output:  ; IBDATA with formatted output OTHPAYV(IBXIEN,IBSAVE,IBDATA,IBFUNC,IBFLDTYP,IBSEQN) ;  N IB1,IB2,IBPIECE,IBINS,IBFL  S IBFL=$S(IBFUNC=3!(IBFUNC=4):1,1:0)  S IBPIECE=$S(IBFLDTYP="I":4,IBFLDTYP="Q":3,1:3)  F IB1=1,2 D  . I $$ISINSUR($G(IBSAVE("PROVINF",IBXIEN,"O",IB1)),IBXIEN) D  Q  ;don't create anything if there is no such insurance  .. ;\*432/TAZ Attending/Rendering is no longer either/or so there can be both  .. ;I IBFL S IBFUNC=$S($O(IBSAVE("PROVINF",IBXIEN,"O",IB1,3,0)):3,1:4),IBFL=0  .. S IBDATA(IB1)=$P($G(IBSAVE("PROVINF",IBXIEN,"O",IB1,IBFUNC,IBSEQN)),U,IBPIECE)  Q  ;  ;chk for ins  ;Input:  ; IBINS = "P","S","T"  ; IBXIEN - ien file #399  ;Output:  ; returns 1-exists , 0-doesn't ISINSUR(IBINS,IBXIEN) ;  N IBINSNOD  S IBINSNOD=$S(IBINS="P":"I1",IBINS="S":"I2",IBINS="T":"I3",1:"")  I IBINSNOD="" Q 0  Q $D(^DGCR(399,IBXIEN,IBINSNOD))  ;  ;---PRACT----  ;Get list of all 355.9 or 355.93 records for prov  ;Input:   ;IB399INS - ins co for bill to match PRACTIONER from 355.9  ;IB399FRM - form type (0=unknwn/both,1=UB,2=1500) to   ; match PRACTIONER from 355.9  ;IB399CAR - BILL CARE (0=unknwn or both inp/outp,1=inpatient,  ; 2=outpatient/3=Rx) to match PROV from 355.9  ; OR - DIVISION PTR to file 40.8 for entries in file 355.92  ;IBPROV - VARIABLE PTR VA prov  ;IBARR - array by reference for result  ;IBPROVTP- function (2-operating, 3-RENDERING,etc 0-facility)  ;IBINSTP - "C" -current ins , "O"-other  ;IBFILE - 355.92 for facility ids or 355.9 (default) for provider ids  ;IBINS - 1 if to include ids for the ins co for all provs  ;Ouput:  ;IBARR - array by ref for result  ; prov var ptr^ins ptr^X12 id cd^ID^form typ^care typ or division ptr^st ptr^id rec ptr^id type ptr PRACT(IB399INS,IB399FRM,IB399CAR,IBPROV,IBARR,IBPROVTP,IBINSTP,IBFILE,IBINS) ;  N IB1,IB2,IBDAT,IBF,IBFX,IB3559,IBINSCO,IBFRMTYP,IBIDTYP,IBID,IBIDT,IBDIV,IBQ,IBS1,IBS2,IBARRX,Z,Z1,Z2,IBCARE  I $G(IBFILE)="" S IBFILE=355.9  S IBINS=$G(IBINS)  S (IBARR,IB3559,IB1)=0  F IBF="",1 Q:IBF=1&$S(IBFILE'=355.9:1,1:'IBINS) S IBFX=IBFILE\_IBF F IB2=1:1 S IB3559=$O(^IBA(IBFX,"B",$S(IBFILE=355.9&(IBF=""):IBPROV,1:IB399INS),IB3559)) Q:IB3559=""  D  . S IBINSCO=$P($G(^IBA(IBFX,IB3559,0)),"^",$S(IBFILE=355.9&(IBF=""):2,1:1)) ;ins co. ptr  . I IBINSCO'="" I IBINSCO'=IB399INS Q  ;exclude if different ins  . S:IBINSCO="" IBINSCO="NONE" ;NONE will be included in the array  . S IBFRMTYP=+$P($G(^IBA(IBFX,IB3559,0)),"^",4) ;form type (0=both,1=UB,2=1500 **or 4=J430D**) **;JWS;JRA IB\*2.0\*592 added J430D to comment**  . I '(IBFRMTYP=0!(IB399FRM=0)) Q:IBFRMTYP'=IB399FRM  ;exclude if not "both" and different  . S IBCARE=+$P($G(^IBA(IBFX,IB3559,0)),"^",5) ;0=both(inp and outp),1=inp,2=outp,3=prescr -- OR -- division ptr  . I $S(IBFILE=355.92:0,1:IBCARE=3) I IB399CAR'=3 Q  ; Id is only for Rx  . **;JWS;IB\*2.0\*592;Dental form = 4 in set of codes value**  . I $S(IBFILE=355.92:0,1:IBCARE=1!(IBCARE=2**)!(IBCARE=4))** I IB399CAR=1!(IB399CAR=2) Q:IBCARE'=IB399CAR  ;both is OK  . I IBFILE=355.92,IBCARE Q:IB399CAR'=IBCARE  ; Division doesn't match  . S IBIDTYP=+$P($G(^IBA(IBFX,IB3559,0)),"^",6) ;prov ID type  . I IBFILE=355.9,IBIDTYP=$$TAXID^IBCEP8(),$S(IBPROV["VA(200":1,1:$P($G(^IBA(355.93,+IBPROV,0)),U,2)=2) Q  ; Don't extract tax id # id for indiv prov  . S IBIDT=IBIDTYP  . S IBIDTYP=$P($G(^IBE(355.97,IBIDTYP,0)),"^",3)  . Q:$P($G(^IBE(355.97,+IBIDT,1)),U,9)  . Q:IBFILE=355.9&(IBIDTYP="X4") ;exclude CLIA #  . S IBID=$P($G(^IBA(IBFX,IB3559,0)),"^",7) ;prov ID value  . I $G(IBPROVTP)'="",$G(IBINSTP)'="",IBPROVTP'=0 I '$$CHCKSEC^IBCEF73(IB399FRM,IBPROVTP,IBINSTP,IBIDTYP) Q  ; No qualifier chk for fac  . I IBID'="" S IBDAT=IBPROV\_"^"\_IBINSCO\_"^"\_IBIDTYP\_"^"\_IBID\_"^"\_IBFRMTYP\_"^"\_IBCARE\_"^"\_"^"\_IB3559\_U\_IBIDT,IBS2=$S(IBFX'=355.91:"",1:"INS DEF^")\_IB3559  . I IBFILE'=355.92,IBID'="",IB399CAR=3 S IBQ=0 D  Q:IBQ  .. I $G(IBARRX(IBIDT))!(IBCARE=1) S IBQ=1 Q  .. I IBCARE=3&(IB399CAR=3) S IBARRX(IBIDT)=1 Q  ; Rx match  .. **;JWS;IB\*2.0\*592;Dental form**  .. I IBCARE=0!(IBCARE=2**)!(IBCARE=4)** S IBARRX(IBIDT,IBINSCO,IBS2)=IBDAT,IBQ=1 Q  . I IBID'="" S IBARR(IBINSCO,IBS2)=IBDAT  ;  I IB399CAR=3 S Z=0 F  S Z=$O(IBARRX(Z)) Q:'Z  I '$G(IBARRX(Z)) D  . S Z1="" F  S Z1=$O(IBARRX(Z,Z1)) Q:Z1=""  S Z2="" F  S Z2=$O(IBARRX(Z,Z1,Z2)) Q:Z2=""  S IBARR(Z1,Z2)=IBARRX(Z,Z1,Z2)  ;  I IBPROV["VA(200," D  ; Get lic #s from file 2 for VA providers  . N Z,IBLIC  . S IBLIC=+IBPROV,IBLIC=$$GETLIC^IBCEP5D(.IBLIC)  . S IBIDTYP=$P($G(^IBE(355.97,+$$STLIC^IBCEP8(),0)),U,3)  . S Z=0 F  S Z=$O(IBLIC(Z)) Q:'Z  S:$$CHCKSEC^IBCEF73(IB399FRM,IBPROVTP,IBINSTP,IBIDTYP) IBARR("NONE","LIC"\_Z\_"^"\_IBPROV)=IBPROV\_U\_"NONE"\_U\_IBIDTYP\_U\_IBLIC(Z)\_U\_"0"\_U\_"0"\_U\_Z\_U\_U\_+$$STLIC^IBCEP8()  I IBPROV["IBA(355.93" D  . Q:$P($G(^IBA(355.93,+IBPROV,0)),U,12)=""  . S IBIDTYP=$P($G(^IBE(355.97,+$$STLIC^IBCEP8(),0)),U,3)  . I $$CHCKSEC^IBCEF73(IB399FRM,IBPROVTP,IBINSTP,IBIDTYP) D  . . S IBARR("NONE","LIC"\_$P($G(^DIC(5,+$P(^IBA(355.93,+IBPROV,0),U,7),0)),U,2)\_"^"\_IBPROV)=IBPROV\_U\_"NONE"\_U\_IBIDTYP\_U\_$P(^IBA(355.93,+IBPROV,0),U,12)\_U\_"0"\_U\_"0"\_U\_$P(^IBA(355.93,+IBPROV,0),U,7)\_U\_U\_+IBPROV  Q  ; ALLPRFAC(IBXIEN,IBXSAVE) ; Return all non-VA/outside facility prov ids  ; and all VA alternate prov ids  ; IBXIEN = ien file 399  ; IBXSAVE = subscripted array returned  N IBPROV,IBFRMTYP,IBCARE,IBRETARR,IBRET1,IBCOBN,Z,Z0,Z1,ZZ  K IBXSAVE("PROVINF\_FAC",IBXIEN) ; Always rebuild this  S IBCOBN=+$$COBN^IBCEF(IBXIEN)  **;**S IBFRMTYP=$$FT^IBCEF(IBXIEN),IBFRMTYP=$S(IBFRMTYP=2:2,IBFRMTYP=3:1,1:0) **;JRA IB\*2.0\*592 ';'**  S IBFRMTYP=$$FT^IBCEF(IBXIEN),IBFRMTYP=$S(IBFRMTYP=2:2,**IBFRMTYP=7:4**,IBFRMTYP=3:1,1:0) **;JRA IB\*2.0\*592**  S IBPROV=$P($G(^DGCR(399,IBXIEN,"U2")),U,10)  ; IB patch 320 - Build IBPROV variable better when a non-VA facility exists  I IBPROV S IBPROV=IBPROV\_";IBA(355.93,"  I 'IBPROV S IBCARE=$P($G(^DGCR(399,IBXIEN,0)),U,22)  I IBPROV D  . S IBCARE=$S($$ISRX^IBCEF1(IBXIEN):3,1:0) ;if Rx refill bill  . S:IBCARE=0 IBCARE=$$INPAT^IBCEF(IBXIEN,1) S:'IBCARE IBCARE=2 ;1-inp, 2-out  F Z=1:1:3 K IBRETARR I $G(^DGCR(399,IBXIEN,"I"\_Z)) D  . D PRACT(+^DGCR(399,IBXIEN,"I"\_Z),IBFRMTYP,IBCARE,IBPROV,.IBRETARR,0,$S(Z=IBCOBN:"C",1:"O"),$S('IBPROV:355.92,1:355.9))  . K IBRET1  . S Z0="" F  S Z0=$O(IBRETARR(Z0)) Q:Z0=""  S Z1="" F  S Z1=$O(IBRETARR(Z0,Z1)) Q:Z1=""  D  .. ; Sort by div/id type  .. S IBRET1($S(IBPROV:0,1:+$P(IBRETARR(Z0,Z1),U,6)),+$P(IBRETARR(Z0,Z1),U,9))=IBRETARR(Z0,Z1)  .. Q  . ;  . S Z0=$O(IBRET1(""),-1) Q:Z0=""  D  .. ; IB patch 320 - loop thru all ID's  .. S Z1="" F  S Z1=$O(IBRET1(Z0,Z1)) Q:Z1=""  D  ... I Z=IBCOBN S IBXSAVE("PROVINF\_FAC",IBXIEN,"C",1,0,$O(IBXSAVE("PROVINF\_FAC",IBXIEN,"C",1,0," "),-1)+1)=IBRET1(Z0,Z1) Q  ... S ZZ=$S(Z=1:1,Z=2:(IBCOBN=3)+1,1:2)  ... S IBXSAVE("PROVINF\_FAC",IBXIEN,"O",ZZ,0,$O(IBXSAVE("PROVINF\_FAC",IBXIEN,"O",ZZ,0," "),-1)+1)=IBRET1(Z0,Z1),IBXSAVE("PROVINF\_FAC",IBXIEN,"O",ZZ)=$E("PST",Z)  ... Q  .. Q  . Q  ;  S IBXSAVE("PROVINF\_FAC",IBXIEN)=IBXIEN,IBXSAVE("PROVINF\_FAC",IBXIEN,"C",1)=$E("PST",IBCOBN)  Q  ; OTHID(IBXSAVE,IBXDATA,IBXIEN,PRIDSEQ,PRTYP,IBQ,IBFAC) ; From data in IBXSAVE,  ; determine id or qualifier to output in the 837 records OP\*  ; Returns IBXDATA array IBXDATA(n)=data  ; IBXIEN = ien of the bill-file 399  ; PRIDSEQ = sequence of the payer id needed  ; PRTYP = provider type to check for data  ; IBQ = 1 if qualifier needed, 0/null if id needed  ; IBFAC = 1 if facility id, 0 for individual provider id  ;   N Z,Z0,Z1  S Z0="PROVINF"\_$S('$G(IBFAC):"",1:"\_FAC"),Z1=$S($G(IBQ):3,1:4)  S Z=0 F  S Z=$O(IBXSAVE("OSQ",Z)) Q:'Z  D  . I $P($G(IBXSAVE(Z0,IBXIEN,"O",Z,+$G(PRTYP),+$G(PRIDSEQ))),U,4)'="" S IBXDATA(IBXSAVE("OSQ",Z))=$P(IBXSAVE(Z0,IBXIEN,"O",Z,+$G(PRTYP),+$G(PRIDSEQ)),U,Z1)  Q  ; SETSEQ(IBXIEN,IBXSAVE,IBXDATA,PRTYP,IBFAC,IBOP) ; Sets up IBXSAVE("OSQ")  ; array for other id seq in 837 records OP\*  ; Returns IBXDATA(n)=cob seq indicator for ids  ; IBXIEN = ien of bill-399  ; PRTYP = the provider type to check for data for indiv provider  ; IBFAC = 1 if facility id, 0 for individual provider id  ; IBOP = segement # in OP being output  N C,Z,Z0,Z1,OK  S C=0,Z0="PROVINF"\_$S('$G(IBFAC):"",1:"\_FAC")  S:$G(IBFAC) PRTYP=0  S Z=0 F  S Z=$O(IBXSAVE(Z0,IBXIEN,"O",Z)) Q:'Z  S OK=0 D  . N Z1 F Z1=1:1 Q:'$D(IBXSAVE(Z0,IBXIEN,"O",Z,+$G(PRTYP),Z1)) I $P(IBXSAVE(Z0,IBXIEN,"O",Z,+$G(PRTYP),Z1),U,4)'="""" S OK=1 Q  . I OK S C=C+1,IBXSAVE("OSQ",Z)=C  S Z=0 F  S Z=$O(IBXSAVE("OSQ",Z)) Q:'Z  S IBXDATA(IBXSAVE("OSQ",Z))=$G(IBXSAVE(Z0,IBXIEN,"O",Z)) D:IBXSAVE("OSQ",Z)>1 ID^IBCEF2(IBXSAVE("OSQ",Z),"OP"\_$G(IBOP)\_" ")  Q  ; PSPRV(IBIFN) ;  Q $$PSPRV^IBCEF7(IBIFN) ; Moved  ; | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCEF73 | | | | | | | | |
| **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** | | | | | | | | | |
| IBCEF73 ;WOIFO/SS - FORMATTER AND EXTRACTOR SPECIFIC BILL FUNCTIONS ;8/6/03 10:56am  ;;2.0;INTEGRATED BILLING;\*\*232,320,358,349,377\*\*;21-MAR-94;Build 23  ;;Per VHA Directive 2004-038, this routine should not be modified.  ;  ;check qualifier  ;IBFRM 0-both, 1=UB,2=1500  ;IBPROV - function in #399 (1-referring, 2-operating,etc)  ;IBTYPE - "C"-current insurance, "O"-other insurance  ;IBVAL - value to check CHCKSEC(IBFRM,IBPROV,IBTYPE,IBVAL) ;  I IBFRM=0 Q:$$CHSEC(1,IBPROV,IBTYPE,IBVAL) 1 Q $$CHSEC(2,IBPROV,IBTYPE,IBVAL)  Q $$CHSEC(IBFRM,IBPROV,IBTYPE,IBVAL)  ; CHSEC(IBFRM,IBPROV,IBTYPE,IBVAL) ;  N IBSTR S IBSTR=""  ;referring  I IBPROV=1 S IBSTR=$S(IBTYPE="C":$$OPR5(IBFRM),IBTYPE="O":$$OP4(IBFRM),1:"")  ;operating  I IBPROV=2 S IBSTR=$S(IBTYPE="C":$$OPR3(IBFRM),IBTYPE="O":$$OP2(IBFRM),1:"")  ;rendering  I IBPROV=3 S IBSTR=$S(IBTYPE="C":$$OPR2(IBFRM),IBTYPE="O":$$OP1(IBFRM),1:"")  ;attending  I IBPROV=4 S IBSTR=$S(IBTYPE="C":$$OPR2(IBFRM),IBTYPE="O":$$OP1(IBFRM),1:"")  ;supervising  I IBPROV=5 S IBSTR=$S(IBTYPE="C":$$OPR8(IBFRM),IBTYPE="O":$$OP8(IBFRM),1:"")  ;other  I IBPROV=9 S IBSTR=$S(IBTYPE="C":$$OPR4(IBFRM),IBTYPE="O":$$OP9(IBFRM),1:"")  Q:IBPROV=0!(IBSTR="") 1 ;if "" or facility id always return 1  Q IBSTR[("^"\_IBVAL\_"^")  ;  ;Filter invalid qualifier entries for records SUB1,SUB2,OP6,OP7,OP3  ; Rebuild the IBXSAVE("PROVINF" or IBXSAVE("PROVINF\_FAC" array with  ; only ids that have valid qualifiers  ;IBFRM 0-both, 1=UB,2=1500  ;IBREC record ID whose ids are being filtered (SUB1,SUB2,etc)  ;IBFAC - 1 if facility check, 0 if attending/rendering check  ;IBTYPE - "C"-current insurance, "O"-other insurance  ;IBXSAVE - the array of provider ids extracted, returned filtered -  ; passed by reference CHCKSUB(IBFRM,IBREC,IBFAC,IBTYPE,IBXSAVE) ;  N Z,Z0,Z1,Z2,CT,IBSAVE  S Z="PROVINF"\_$P("^\_FAC",U,$G(IBFAC)+1)  I '$G(IBXSAVE(Z,IBXIEN)) D  . D F^IBCEF("N-ALL "\_$S($G(IBFAC):"OUTSIDE FAC PROVIDER INF",1:"CUR/OTH PROVIDER INFO"))  M IBSAVE(Z,IBXIEN,IBTYPE)=IBXSAVE(Z,IBXIEN,IBTYPE) K IBXSAVE(Z,IBXIEN,IBTYPE)  S Z0=0 F  S Z0=$O(IBSAVE(Z,IBXIEN,IBTYPE,Z0)) Q:'Z0  S Z1="" F  S Z1=$O(IBSAVE(Z,IBXIEN,IBTYPE,Z0,Z1)) Q:Z1=""  S (Z2,CT)=0 F  S Z2=$O(IBSAVE(Z,IBXIEN,IBTYPE,Z0,Z1,Z2)) Q:'Z2  D  . N IBVAL  . S IBVAL=$P(IBSAVE(Z,IBXIEN,IBTYPE,Z0,Z1,Z2),U,3)  . I IBFRM=0 D  Q  .. I $S($$CHSUB(1,IBREC,IBVAL):1,1:$$CHSUB(2,IBPROV,IBTYPE,IBVAL)) D  ... S CT=CT+1,IBXSAVE(Z,IBXIEN,IBTYPE,Z0,Z1,CT)=IBSAVE(Z,IBXIEN,IBTYPE,Z0,Z1,Z2)  ... I $G(IBXSAVE(Z,IBXIEN,IBTYPE,Z0))="",$G(IBSAVE(Z,IBXIEN,IBTYPE,Z0))'="" S IBXSAVE(Z,IBXIEN,IBTYPE,Z0)=IBSAVE(Z,IBXIEN,IBTYPE,Z0)  . I $$CHSUB(IBFRM,IBREC,IBVAL) D  .. S CT=CT+1,IBXSAVE(Z,IBXIEN,IBTYPE,Z0,Z1,CT)=IBSAVE(Z,IBXIEN,IBTYPE,Z0,Z1,Z2)  .. I $G(IBXSAVE(Z,IBXIEN,IBTYPE,Z0))="",$G(IBSAVE(Z,IBXIEN,IBTYPE,Z0))'="" S IBXSAVE(Z,IBXIEN,IBTYPE,Z0)=IBSAVE(Z,IBXIEN,IBTYPE,Z0)  Q  ;  ; Check if valid qualifier  ;IBFRM 0-both, 1=UB,2=1500  ;IBREC record ID whose ids are being filtered (SUB1,SUB2,etc)  ;IBVAL - value to check CHSUB(IBFRM,IBREC,IBVAL) ;  N IBSTR  I IBREC="SUB1" S IBSTR=$$SUB1(IBFRM)  I IBREC="SUB2" S IBSTR=$$SUB2(IBFRM)  I IBREC="OP7" S IBSTR=$$OP7(IBFRM)  I IBREC="OP3" S IBSTR=$$OP3(IBFRM)  I IBREC="OP6" S IBSTR=$$OP6(IBFRM)  Q:$G(IBSTR)="" 1 ;if "" always return 1  Q IBSTR[("^"\_IBVAL\_"^")  ;  ;IBFRM 0-both, 1=UB,2=1500 OPR2(IBFRM) ;  Q:IBFRM=1 "^0B^1A^1B^1C^1D^1G^1H^EI^G2^LU^N5^SY^X5^"  Q:IBFRM=2 "^0B^1B^1C^1D^1G^1H^EI^G2^LU^N5^SY^X5^"  Q ""  ;  ;IBFRM 0-both, 1=UB,2=1500 OP1(IBFRM) ;  Q:IBFRM=1 "^1A^1B^1C^1D^1G^1H^EI^G2^LU^N5^"  Q:IBFRM=2 "^1B^1C^1D^EI^G2^LU^N5^"  Q ""  ;  ;IBFRM 0-both, 1=UB,2=1500 OPR3(IBFRM) ;  Q:IBFRM=1 "^0B^1A^1B^1C^1D^1G^1H^EI^G2^LU^N5^SY^X5^"  Q ""  ;  ;IBFRM 0-both, 1=UB,2=1500 OP2(IBFRM) ;  Q:IBFRM=1 "^1A^1B^1C^1D^1G^1H^EI^G2^LU^N5^"  Q ""  ;  ;IBFRM 0-both, 1=UB,2=1500 SUB1(IBFRM) ;  Q:IBFRM=1 "^0B^1A^1B^1C^1D^1G^1H^EI^G2^LU^N5^SY^X5^"  Q:IBFRM=2 "^0B^1A^1B^1C^1D^1G^1H^EI^G2^LU^N5^U3^SY^X5^"  Q ""  ;  ;IBFRM 0-both, 1=UB,2=1500 OPR4(IBFRM) ;  Q:IBFRM=1 "^0B^1A^1B^1C^1D^1G^1H^EI^G2^LU^N5^SY^X5^"  Q ""  ;  ;IBFRM 0-both, 1=UB,2=1500 OP9(IBFRM) ;  Q:IBFRM=1 "^1A^1B^1C^1D^1G^1H^EI^G2^LU^N5^"  Q ""  ;  ;IBFRM 0-both, 1=UB,2=1500 SUB2(IBFRM) ;  Q:IBFRM=1 "^0B^1A^1B^1C^1G^1H^1J^EI^FH^G2^G5^LU^N5^X5^TJ^B3^BQ^SY^U3^"  Q:IBFRM=2 "^0B^X4^1A^1B^1C^1G^1H^G2^LU^X5^TJ^B3^BQ^SY^U3^"  Q ""  ;  ;IBFRM 0-both, 1=UB,2=1500 OP3(IBFRM) ;  Q:IBFRM=1 "^1B^1C^EI^G2^LU^N5^"  Q ""  ;  ;IBFRM 0-both, 1=UB,2=1500 OPR5(IBFRM) ;  Q:IBFRM=2 "^0B^1B^1C^1D^1G^1H^EI^G2^LU^N5^SY^X5^"  Q ""  ;  ;IBFRM 0-both, 1=UB,2=1500 OPR8(IBFRM) ;  Q:IBFRM=2 "^0B^1B^1C^1D^1G^1H^EI^G2^LU^N5^SY^X5^"  Q ""  ;  ;IBFRM 0-both, 1=UB,2=1500 OP4(IBFRM) ;  Q:IBFRM=2 "^1B^1C^1D^EI^G2^LU^N5^"  Q ""  ;  ;IBFRM 0-both, 1=UB,2=1500 OP8(IBFRM) ;  Q:IBFRM=2 "^1B^1C^1D^EI^G2^N5^"  Q ""  ;  ;IBFRM 0-both, 1=UB,2=1500 OP6(IBFRM) ;  Q:IBFRM=2 "^1A^1B^1C^G2^LU^N5^"  Q ""  ;  ;IBFRM 0-both, 1=UB,2=1500 OP7(IBFRM) ;  Q:IBFRM=2 "^1A^1B^1C^G2^LU^N5^"  Q ""  ;  ;check qualifier for PRV1  ;IBFRM 0-both, 1=UB,2=1500  ;IBVAL - value to check CHCKPRV1(IBFRM,IBVAL) ;  I IBFRM=0 Q:$$CHPRV1(1,IBVAL) 1 Q $$CHPRV1(2,IBVAL)  Q $$CHPRV1(IBFRM,IBVAL)  ;IBFRM 0-both, 1=UB,2=1500 CHPRV1(IBFRM,IBVAL) ;  N IBSTR S IBSTR=""  S IBSTR=$$PRV1(IBFRM)  Q:IBSTR="" 1  Q IBSTR[("^"\_IBVAL\_"^")  ; PRV1(IBFRM) ;  Q:IBFRM=1 "^1A^1C^1D^1G^1H^1J^B3^BQ^EI^FH^G2^G5^LU^SY^X5^"  Q:IBFRM=2 "^1B^1C^1D^1G^1H^1J^B3^BQ^EI^FH^G2^G5^LU^U3^SY^X5^"  Q ""  ; PTSELF ;This tag is for the CI2 segment. If the IBXSAVE("IADR") is empty  ;check to see if the relationship to pt is 18 (self) if so pull info  ;from PT1 calls  ;See if relationship to insured is 18 if not or if "" quit  N IBZ  D F^IBCEF("N-ALL INSURED PT RELATION","IBZ",,IBXIEN)  S IBZ=$G(IBZ(+$$COBN^IBCEF(IBXIEN)))  S IBZ=$$PRELCNV^IBCNSP1(IBZ,1)  I IBZ'="18" S IBXDATA="" Q  N IBZ D F^IBCEF("N-PATIENT STREET ADDRESS 1-3","IBZ",,IBXIEN)  S IBXDATA="18"  Q  ; NOPUNCT(X,SPACE,EXC) ; Strip punctuation from data in X  ; SPACE = flag if 1 strip SPACES  ; EXC = list of punct not to strip  ;  N PUNCT,Z  S PUNCT=".,-+(){}[]\/><:;?|=\_\*&%$#@!~`^'"""  I $G(SPACE) S PUNCT=PUNCT\_" "  I $G(EXC)'="" S PUNCT=$TR(PUNCT,EXC)  N L S L=""  F  S L=$O(X(L)) Q:L=""  D  . S X(L)=$TR(X(L),PUNCT)  I $G(X)'="" D  . S X=$TR(X,PUNCT)  Q  ; PROVID(IBXIEN) ;This modified version of prov id call is to acquire the SSN  ;first, if the ssn is not available then we need to get the tax id.  ;we also need to provide the modifier for which value it is  Q:+$G(IBXIEN)=0 ""  S IBXSAVE("ID")=""  S IBXSAVE=""  S IBXSAVE=$$PROVSSN^IBCEF7(IBXIEN)  N I  F I=1:1:9 D  . I $P(IBXSAVE,"^",I)]"" S $P(IBXSAVE("ID"),U,I)="34"  ;If no ibxdata go look in 355.97 for 24   N IBRETVAL S IBRETVAL=""  N IBPTR,IBFT  F IBFT=1:1:9 D  . Q:$P(IBXSAVE,U,IBFT)]""  . S IBPTR=$$PROVPTR^IBCEF7(IBXIEN,IBFT)  . S $P(IBRETVAL,"^",IBFT)=$$TAX3559(IBPTR)  . I $P(IBRETVAL,U,IBFT)]"" D  . . S $P(IBXSAVE,U,IBFT)=$P(IBRETVAL,U,IBFT)  . . S $P(IBXSAVE("ID"),U,IBFT)="24"  Q IBXSAVE  ; TAX3559(IBPROV) ;  I $P(IBPROV,";",2)'["IBA(355.9" Q ""  N IB2,IB3559,IBIDTYP,IBID,IBQFL  S (IB3559,IBQFL)=0  S IBID=""  Q:+$G(IBPROV)=0 ""  F IB2=1:1 S IB3559=$O(^IBA(355.9,"B",IBPROV,IB3559)) Q:IB3559=""!IBQFL  D  . S IBIDTYP=+$P($G(^IBA(355.9,IB3559,0)),"^",6) ;provider ID type, ptr to #355.97  . S IBIDTYP=$P($G(^IBE(355.97,IBIDTYP,0)),"^",3)  . S:IBIDTYP="EI" IBID=$P($G(^IBA(355.9,IB3559,0)),"^",7),IBQFL=1  ; if nothing found yet, look in file 355.93 for Facility Default ID  I IBID="",IBPROV["IBA(355.93" D  .N IB0,IBFID,IBQ  .S IB0=$G(^IBA(355.93,+IBPROV,0)) Q:IB0=""!($P(IB0,U,2)'=1) ; not a facility - bail out  .S IBFID=$P(IB0,U,9) Q:IBFID=""  ; no default id on file - bail out  .S IBQ=$P(IB0,U,13) I +IBQ>0,$P($G(^IBE(355.97,IBQ,0)),U,3)=24 S IBID=IBFID  .Q  Q $$NOPUNCT^IBCEF(IBID)  ;  ;IBFULL-full name  ;IBEL - Name element : "FAMILY","GIVEN","MIDDLE","SUFFIX"  ; SSN200(IBPTR) ;  I $P(IBPTR,";",2)'="VA(200," Q ""  Q $$NOPUNCT^IBCEF($$GET1^DIQ(200,+$P(IBPTR,";")\_",",9))  ;  ;Input:  ; IBIEN399 - ien in #399  ;Output:  ; returns a string with "^" delimiters that contains SSNs (if any)  ; in the position that equal to FUNCTION number  ; i.e. if RENDERING function # is 3 then SSN will be  ; in $P(return value,"^",3), etc.  ; SSN3559(IBPROV) ;  N IB2,IB3559,IBIDTYP,IBID,IBQFL  S (IB3559,IBQFL)=0  S IBID=""  Q:+$G(IBPROV)=0 ""  F IB2=1:1 S IB3559=$O(^IBA(355.9,"B",IBPROV,IB3559)) Q:IB3559=""!IBQFL  D  . S IBIDTYP=+$P($G(^IBA(355.9,IB3559,0)),"^",6)  . S IBIDTYP=$P($G(^IBE(355.97,IBIDTYP,0)),"^",3)  . S:IBIDTYP="SY" IBID=$P($G(^IBA(355.9,IB3559,0)),"^",7),IBQFL=1  Q $$NOPUNCT^IBCEF(IBID)  ;  ;IBIDTYP-provider ID type, ptr to #355.97  ;IBFULL-full name  ;IBEL - Name element : "FAMILY","GIVEN","MIDDLE","SUFFIX"  ; PRV1FMT(P) ;FORMAT CODE FOR PRV1 SEGMENT THAT WON'T FIT ON LINE  K IBXDATA  S:'$D(IBXSAVE("BIL-PROV-SEC")) IBXSAVE("BIL-PROV-SEC")=$$PRV1^IBCEF7(IBXIEN)  S IBXDATA=$P($G(IBXSAVE("BIL-PROV-SEC")),"^",P)  I $G(IBXDATA)'="" S IBXDATA=$$NOPUNCT^IBCEF(IBXDATA,1)  Q  ; | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCEF73 ;WOIFO/SS - FORMATTER AND EXTRACTOR SPECIFIC BILL FUNCTIONS ;8/6/03 10:56am  ;;2.0;INTEGRATED BILLING;\*\*232,320,358,349,377,**592**\*\*;21-MAR-94;Build 23  ;;Per VHA Directive 2004-038, this routine should not be modified.  ;  ;check qualifier  ;IBFRM 0-both, 1=UB,2=1500, 4=J430D  ;IBPROV - function in #399 (1-referring, 2-operating,etc)  ;IBTYPE - "C"-current insurance, "O"-other insurance  ;IBVAL - value to check CHCKSEC(IBFRM,IBPROV,IBTYPE,IBVAL) ;  **;JWS;IB\*2.0\*592; J430D form 4**  I IBFRM=0 Q:$$CHSEC(1,IBPROV,IBTYPE,IBVAL) 1 **Q:$$CHSEC(4,IBPROV,IBTYPE,IBVAL)** Q $$CHSEC(2,IBPROV,IBTYPE,IBVAL)  Q $$CHSEC(IBFRM,IBPROV,IBTYPE,IBVAL)  ; CHSEC(IBFRM,IBPROV,IBTYPE,IBVAL) ;  N IBSTR S IBSTR=""  ;referring  I IBPROV=1 S IBSTR=$S(IBTYPE="C":$$OPR5(IBFRM),IBTYPE="O":$$OP4(IBFRM),1:"")  ;operating  I IBPROV=2 S IBSTR=$S(IBTYPE="C":$$OPR3(IBFRM),IBTYPE="O":$$OP2(IBFRM),1:"")  ;rendering  I IBPROV=3 S IBSTR=$S(IBTYPE="C":$$OPR2(IBFRM),IBTYPE="O":$$OP1(IBFRM),1:"")  ;attending  I IBPROV=4 S IBSTR=$S(IBTYPE="C":$$OPR2(IBFRM),IBTYPE="O":$$OP1(IBFRM),1:"")  ;supervising  I IBPROV=5 S IBSTR=$S(IBTYPE="C":$$OPR8(IBFRM),IBTYPE="O":$$OP8(IBFRM),1:"")  **;JWS;IB\*2.0\*592;assistant surgeon Dental  I IBPROV=6 S IBSTR=$S(IBTYPE="C":$$OPRB(IBFRM),IBTYPE="O":$$OPRB(IBFRM),1:"")**  ;other  I IBPROV=9 S IBSTR=$S(IBTYPE="C":$$OPR4(IBFRM),IBTYPE="O":$$OP9(IBFRM),1:"")  Q:IBPROV=0!(IBSTR="") 1 ;if "" or facility id always return 1  Q IBSTR[("^"\_IBVAL\_"^")  ;  ;Filter invalid qualifier entries for records SUB1,SUB2,OP6,OP7,OP3  ; Rebuild the IBXSAVE("PROVINF" or IBXSAVE("PROVINF\_FAC" array with  ; only ids that have valid qualifiers  ;IBFRM 0-both, 1=UB,2=1500, **4=J430D**  ;IBREC record ID whose ids are being filtered (SUB1,SUB2,etc)  ;IBFAC - 1 if facility check, 0 if attending/rendering check  ;IBTYPE - "C"-current insurance, "O"-other insurance  ;IBXSAVE - the array of provider ids extracted, returned filtered -  ; passed by reference CHCKSUB(IBFRM,IBREC,IBFAC,IBTYPE,IBXSAVE) ;  N Z,Z0,Z1,Z2,CT,IBSAVE  S Z="PROVINF"\_$P("^\_FAC",U,$G(IBFAC)+1)  I '$G(IBXSAVE(Z,IBXIEN)) D  . D F^IBCEF("N-ALL "\_$S($G(IBFAC):"OUTSIDE FAC PROVIDER INF",1:"CUR/OTH PROVIDER INFO"))  M IBSAVE(Z,IBXIEN,IBTYPE)=IBXSAVE(Z,IBXIEN,IBTYPE) K IBXSAVE(Z,IBXIEN,IBTYPE)  S Z0=0 F  S Z0=$O(IBSAVE(Z,IBXIEN,IBTYPE,Z0)) Q:'Z0  S Z1="" F  S Z1=$O(IBSAVE(Z,IBXIEN,IBTYPE,Z0,Z1)) Q:Z1=""  S (Z2,CT)=0 F  S Z2=$O(IBSAVE(Z,IBXIEN,IBTYPE,Z0,Z1,Z2)) Q:'Z2  D  . N IBVAL  . S IBVAL=$P(IBSAVE(Z,IBXIEN,IBTYPE,Z0,Z1,Z2),U,3)  . I IBFRM=0 D  Q  .. I $S($$CHSUB(1,IBREC,IBVAL):1,1:$$CHSUB(2,IBPROV,IBTYPE,IBVAL)) D  ... S CT=CT+1,IBXSAVE(Z,IBXIEN,IBTYPE,Z0,Z1,CT)=IBSAVE(Z,IBXIEN,IBTYPE,Z0,Z1,Z2)  ... I $G(IBXSAVE(Z,IBXIEN,IBTYPE,Z0))="",$G(IBSAVE(Z,IBXIEN,IBTYPE,Z0))'="" S IBXSAVE(Z,IBXIEN,IBTYPE,Z0)=IBSAVE(Z,IBXIEN,IBTYPE,Z0)  . I $$CHSUB(IBFRM,IBREC,IBVAL) D  .. S CT=CT+1,IBXSAVE(Z,IBXIEN,IBTYPE,Z0,Z1,CT)=IBSAVE(Z,IBXIEN,IBTYPE,Z0,Z1,Z2)  .. I $G(IBXSAVE(Z,IBXIEN,IBTYPE,Z0))="",$G(IBSAVE(Z,IBXIEN,IBTYPE,Z0))'="" S IBXSAVE(Z,IBXIEN,IBTYPE,Z0)=IBSAVE(Z,IBXIEN,IBTYPE,Z0)  Q  ;  ; Check if valid qualifier  ;IBFRM 0-both, 1=UB,2=1500**, 4=J430D**  ;IBREC record ID whose ids are being filtered (SUB1,SUB2,etc)  ;IBVAL - value to check CHSUB(IBFRM,IBREC,IBVAL) ;  N IBSTR  I IBREC="SUB1" S IBSTR=$$SUB1(IBFRM)  I IBREC="SUB2" S IBSTR=$$SUB2(IBFRM)  I IBREC="OP7" S IBSTR=$$OP7(IBFRM)  I IBREC="OP3" S IBSTR=$$OP3(IBFRM)  I IBREC="OP6" S IBSTR=$$OP6(IBFRM)  Q:$G(IBSTR)="" 1 ;if "" always return 1  Q IBSTR[("^"\_IBVAL\_"^")  ;  ;IBFRM 0-both, 1=UB,2=1500**, 4=J430D** OPR2(IBFRM) ;  Q:IBFRM=1 "^0B^1A^1B^1C^1D^1G^1H^EI^G2^LU^N5^SY^X5^"  **;JRA IB\*2.0\*592 Modify for Dental form 7**  **;**Q:IBFRM=2 "^0B^1B^1C^1D^1G^1H^EI^G2^LU^N5^SY^X5^" **;JRA IB\*2.0\*592 ';'**  Q:(IBFRM=2**!(IBFRM=4))** "^0B^1B^1C^1D^1G^1H^EI^G2^LU^N5^SY^X5^"  **;JWS;JRA IB\*2.0\*592**  Q ""  ;  ;IBFRM 0-both, 1=UB,2=1500**, 4=J430D** OP1(IBFRM) ;  Q:IBFRM=1 "^1A^1B^1C^1D^1G^1H^EI^G2^LU^N5^"  **;JRA IB\*2.0\*592 Modify for Dental form 7**  **;**Q:IBFRM=2 "^1B^1C^1D^EI^G2^LU^N5^" **;JRA IB\*2.0\*592 ';'**  Q:(IBFRM=2**!(IBFRM=4))** "^1B^1C^1D^EI^G2^LU^N5^"  **;JWS;JRA IB\*2.0\*592**  Q ""  ;  ;IBFRM 0-both, 1=UB,2=1500**, 4=J430D** OPR3(IBFRM) ;  Q:IBFRM=1 "^0B^1A^1B^1C^1D^1G^1H^EI^G2^LU^N5^SY^X5^"  Q ""  ;  ;IBFRM 0-both, 1=UB,2=1500**, 4=J430D** OP2(IBFRM) ;  Q:IBFRM=1 "^1A^1B^1C^1D^1G^1H^EI^G2^LU^N5^"  Q ""  ;  ;IBFRM 0-both, 1=UB,2=1500**, 4=J430D** SUB1(IBFRM) ;  Q:IBFRM=1 "^0B^1A^1B^1C^1D^1G^1H^EI^G2^LU^N5^SY^X5^"  **;JRA IB\*2.0\*592 Modify for Dental form 7  ;**Q:IBFRM=2 "^0B^1A^1B^1C^1D^1G^1H^EI^G2^LU^N5^U3^SY^X5^" **;JRA IB\*2.0\*592 ';'**  Q:(IBFRM=2**!(IBFRM=4))** "^0B^1A^1B^1C^1D^1G^1H^EI^G2^LU^N5^U3^SY^X5^"  **;JWS;JRA IB\*2.0\*592**  Q ""  ;  ;IBFRM 0-both, 1=UB,2=1500**, 4=J430D** OPR4(IBFRM) ;  Q:IBFRM=1 "^0B^1A^1B^1C^1D^1G^1H^EI^G2^LU^N5^SY^X5^"  Q ""  ;  ;IBFRM 0-both, 1=UB,2=1500**, 4=J430D** OP9(IBFRM) ;  Q:IBFRM=1 "^1A^1B^1C^1D^1G^1H^EI^G2^LU^N5^"  Q ""  ;  ;IBFRM 0-both, 1=UB,2=1500**, 4=J430D** SUB2(IBFRM) ;  Q:IBFRM=1 "^0B^1A^1B^1C^1G^1H^1J^EI^FH^G2^G5^LU^N5^X5^TJ^B3^BQ^SY^U3^"  **;JRA IB\*2.0\*592 Modify for Dental form 7**  **;**Q:IBFRM=2 "^0B^X4^1A^1B^1C^1G^1H^G2^LU^X5^TJ^B3^BQ^SY^U3^" **;JRA IB\*2.0\*592 ';'**  Q:(IBFRM=2**!(IBFRM=4))** "^0B^X4^1A^1B^1C^1G^1H^G2^LU^X5^TJ^B3^BQ^SY^U3^"  **;JWS;JRA IB\*2.0\*592**  Q ""  ;  ;IBFRM 0-both, 1=UB,2=1500, **4=J430D** OP3(IBFRM) ;  Q:IBFRM=1 "^1B^1C^EI^G2^LU^N5^"  Q ""  ;  ;IBFRM 0-both, 1=UB,2=1500**, 4=J430D** OPR5(IBFRM) ;  **;JRA IB\*2.0\*592 Modify for Dental form 7**  **;**Q:IBFRM=2 "^0B^1B^1C^1D^1G^1H^EI^G2^LU^N5^SY^X5^" **;JRA IB\*2.0\*592 ';'**  Q:(IBFRM=2**!(IBFRM=4))** "^0B^1B^1C^1D^1G^1H^EI^G2^LU^N5^SY^X5^"  **;JWS;JRA IB\*2.0\*592**  Q ""  ;  ;IBFRM 0-both, 1=UB,2=1500, **4=J430D** OPR8(IBFRM) ; **;JRA IB\*2.0\*592 Modify for Dental form 7**  **;**Q:IBFRM=2 "^0B^1B^1C^1D^1G^1H^EI^G2^LU^N5^SY^X5^" **;JRA IB\*2.0\*592 ';'**  Q:(IBFRM=2**!(IBFRM=4))** "^0B^1B^1C^1D^1G^1H^EI^G2^LU^N5^SY^X5^"  **;JWS;JRA IB\*2.0\*592**  Q ""  ;  ;IBFRM 0-both, 1=UB,2=1500, **4=J430D** OP4(IBFRM) ; **;JRA IB\*2.0\*592 Modify for Dental form 7**  **;**Q:IBFRM=2 "^1B^1C^1D^EI^G2^LU^N5^" **;JRA IB\*2.0\*592 ';'**  Q:(IBFRM=2**!(IBFRM=4))** "^1B^1C^1D^EI^G2^LU^N5^"  **;JWS;JRA IB\*2.0\*592**  Q ""  ;  ;IBFRM 0-both, 1=UB,2=1500**, 4=J430D** OP8(IBFRM) ;  **;JRA IB\*2.0\*592 Modify for Dental form 7**  **;**Q:IBFRM=2 "^1B^1C^1D^EI^G2^N5^" **;JRA IB\*2.0\*592 ';'**  Q:(IBFRM=2**!(IBFRM=4))** "^1B^1C^1D^EI^G2^N5^"  **;JWS;JRA IB\*2.0\*592**  Q ""  ;  ;IBFRM 0-both, 1=UB,2=1500**, 4=J430D** OP6(IBFRM) ; **;JRA IB\*2.0\*592 Modify for Dental form 7**  **;**Q:IBFRM=2 "^1A^1B^1C^G2^LU^N5^" **;JRA IB\*2.0\*592 ';'**  Q:(IBFRM=2**!(IBFRM=4))** "^1A^1B^1C^G2^LU^N5^"  **;JWS;JRA IB\*2.0\*592**  Q ""  ;  ;IBFRM 0-both, 1=UB,2=1500**, 4=J430D** OP7(IBFRM) ; **;JRA IB\*2.0\*592 Modify for Dental form 7**  **;**Q:IBFRM=2 "^1A^1B^1C^G2^LU^N5^" **;JRA IB\*2.0\*592 ';'**  Q:(IBFRM=2**!(IBFRM=4))** "^1A^1B^1C^G2^LU^N5^"  **;JWS;JRA IB\*2.0\*592**  Q ""  ;  ;IBFRM 0-both, 1=UB,2=1500**, 4=J430D** OPRB(IBFRM) ;  Q:IBFRM=4 "^0B^1G^G2^LU^"  Q ""  ;  ;check qualifier for PRV1  ;IBFRM 0-both, 1=UB,2=1500**, 4=J430D**  ;IBVAL - value to check CHCKPRV1(IBFRM,IBVAL) ;  I IBFRM=0 Q:$$CHPRV1(1,IBVAL) 1 Q $$CHPRV1(2,IBVAL)  Q $$CHPRV1(IBFRM,IBVAL)  ;IBFRM 0-both, 1=UB,2=1500, **4=J430D** CHPRV1(IBFRM,IBVAL) ;  N IBSTR S IBSTR=""  S IBSTR=$$PRV1(IBFRM)  Q:IBSTR="" 1  Q IBSTR[("^"\_IBVAL\_"^")  ; PRV1(IBFRM) ;  Q:IBFRM=1 "^1A^1C^1D^1G^1H^1J^B3^BQ^EI^FH^G2^G5^LU^SY^X5^"  **;JRA IB\*2.0\*592 Modify for Dental form 7**  **;**Q:IBFRM=2 "^1B^1C^1D^1G^1H^1J^B3^BQ^EI^FH^G2^G5^LU^U3^SY^X5^" **;JRA IB\*2.0\*592 ';'**  Q:(IBFRM=2**!(IBFRM=4))** "^1B^1C^1D^1G^1H^1J^B3^BQ^EI^FH^G2^G5^LU^U3^SY^X5^"  **;JWS;JRA IB\*2.0\*592**  Q ""  ; PTSELF ;This tag is for the CI2 segment. If the IBXSAVE("IADR") is empty  ;check to see if the relationship to pt is 18 (self) if so pull info  ;from PT1 calls  ;See if relationship to insured is 18 if not or if "" quit  N IBZ  D F^IBCEF("N-ALL INSURED PT RELATION","IBZ",,IBXIEN)  S IBZ=$G(IBZ(+$$COBN^IBCEF(IBXIEN)))  S IBZ=$$PRELCNV^IBCNSP1(IBZ,1)  I IBZ'="18" S IBXDATA="" Q  N IBZ D F^IBCEF("N-PATIENT STREET ADDRESS 1-3","IBZ",,IBXIEN)  S IBXDATA="18"  Q  ; NOPUNCT(X,SPACE,EXC) ; Strip punctuation from data in X  ; SPACE = flag if 1 strip SPACES  ; EXC = list of punct not to strip  ;  N PUNCT,Z  S PUNCT=".,-+(){}[]\/><:;?|=\_\*&%$#@!~`^'"""  I $G(SPACE) S PUNCT=PUNCT\_" "  I $G(EXC)'="" S PUNCT=$TR(PUNCT,EXC)  N L S L=""  F  S L=$O(X(L)) Q:L=""  D  . S X(L)=$TR(X(L),PUNCT)  I $G(X)'="" D  . S X=$TR(X,PUNCT)  Q  ; PROVID(IBXIEN) ;This modified version of prov id call is to acquire the SSN  ;first, if the ssn is not available then we need to get the tax id.  ;we also need to provide the modifier for which value it is  Q:+$G(IBXIEN)=0 ""  S IBXSAVE("ID")=""  S IBXSAVE=""  S IBXSAVE=$$PROVSSN^IBCEF7(IBXIEN)  N I  F I=1:1:9 D  . I $P(IBXSAVE,"^",I)]"" S $P(IBXSAVE("ID"),U,I)="34"  ;If no ibxdata go look in 355.97 for 24   N IBRETVAL S IBRETVAL=""  N IBPTR,IBFT  F IBFT=1:1:9 D  . Q:$P(IBXSAVE,U,IBFT)]""  . S IBPTR=$$PROVPTR^IBCEF7(IBXIEN,IBFT)  . S $P(IBRETVAL,"^",IBFT)=$$TAX3559(IBPTR)  . I $P(IBRETVAL,U,IBFT)]"" D  . . S $P(IBXSAVE,U,IBFT)=$P(IBRETVAL,U,IBFT)  . . S $P(IBXSAVE("ID"),U,IBFT)="24"  Q IBXSAVE  ; TAX3559(IBPROV) ;  I $P(IBPROV,";",2)'["IBA(355.9" Q ""  N IB2,IB3559,IBIDTYP,IBID,IBQFL  S (IB3559,IBQFL)=0  S IBID=""  Q:+$G(IBPROV)=0 ""  F IB2=1:1 S IB3559=$O(^IBA(355.9,"B",IBPROV,IB3559)) Q:IB3559=""!IBQFL  D  . S IBIDTYP=+$P($G(^IBA(355.9,IB3559,0)),"^",6) ;provider ID type, ptr to #355.97  . S IBIDTYP=$P($G(^IBE(355.97,IBIDTYP,0)),"^",3)  . S:IBIDTYP="EI" IBID=$P($G(^IBA(355.9,IB3559,0)),"^",7),IBQFL=1  ; if nothing found yet, look in file 355.93 for Facility Default ID  I IBID="",IBPROV["IBA(355.93" D  .N IB0,IBFID,IBQ  .S IB0=$G(^IBA(355.93,+IBPROV,0)) Q:IB0=""!($P(IB0,U,2)'=1) ; not a facility - bail out  .S IBFID=$P(IB0,U,9) Q:IBFID=""  ; no default id on file - bail out  .S IBQ=$P(IB0,U,13) I +IBQ>0,$P($G(^IBE(355.97,IBQ,0)),U,3)=24 S IBID=IBFID  .Q  Q $$NOPUNCT^IBCEF(IBID)  ;  ;IBFULL-full name  ;IBEL - Name element : "FAMILY","GIVEN","MIDDLE","SUFFIX"  ; SSN200(IBPTR) ;  I $P(IBPTR,";",2)'="VA(200," Q ""  Q $$NOPUNCT^IBCEF($$GET1^DIQ(200,+$P(IBPTR,";")\_",",9))  ;  ;Input:  ; IBIEN399 - ien in #399  ;Output:  ; returns a string with "^" delimiters that contains SSNs (if any)  ; in the position that equal to FUNCTION number  ; i.e. if RENDERING function # is 3 then SSN will be  ; in $P(return value,"^",3), etc.  ; SSN3559(IBPROV) ;  N IB2,IB3559,IBIDTYP,IBID,IBQFL  S (IB3559,IBQFL)=0  S IBID=""  Q:+$G(IBPROV)=0 ""  F IB2=1:1 S IB3559=$O(^IBA(355.9,"B",IBPROV,IB3559)) Q:IB3559=""!IBQFL  D  . S IBIDTYP=+$P($G(^IBA(355.9,IB3559,0)),"^",6)  . S IBIDTYP=$P($G(^IBE(355.97,IBIDTYP,0)),"^",3)  . S:IBIDTYP="SY" IBID=$P($G(^IBA(355.9,IB3559,0)),"^",7),IBQFL=1  Q $$NOPUNCT^IBCEF(IBID)  ;  ;IBIDTYP-provider ID type, ptr to #355.97  ;IBFULL-full name  ;IBEL - Name element : "FAMILY","GIVEN","MIDDLE","SUFFIX"  ; PRV1FMT(P) ;FORMAT CODE FOR PRV1 SEGMENT THAT WON'T FIT ON LINE  K IBXDATA  S:'$D(IBXSAVE("BIL-PROV-SEC")) IBXSAVE("BIL-PROV-SEC")=$$PRV1^IBCEF7(IBXIEN)  S IBXDATA=$P($G(IBXSAVE("BIL-PROV-SEC")),"^",P)  I $G(IBXDATA)'="" S IBXDATA=$$NOPUNCT^IBCEF(IBXDATA,1)  Q  ; | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCEF74 | | | | | | | | |
| **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** | | | | | | | | | |
| IBCEF74 ;WOIFO/SS - FORMATTER/EXTRACT BILL FUNCTIONS ;31-JUL-03  ;;2.0;INTEGRATED BILLING;\*\*232,280,155,290,291,320,358,343,374,432\*\*;21-MAR-94;Build 192  ;;Per VHA Directive 2004-038, this routine should not be modified.  ; SORT(IBPRNUM,IBPRTYP,IB399,IBSRC,IBDST,IBN,IBEXC,IBSEQ,IBLIMIT) ;  D SORT^IBCEF77($G(IBPRNUM),$G(IBPRTYP),$G(IB399),.IBSRC,.IBDST,$G(IBN),$G(IBEXC),$G(IBSEQ),$G(IBLIMIT))  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  ;Output:  ; IBRES(PRNUM,PRTYPE,SEQ#)=PROV^INSUR^IDTYPE^ID^FORMTYP^CARETYP  ; where:(see PROVIDER) PROVINF(IB399,IBPRNUM,IBRES,IBSORT,IBINSTP) ;  I $G(IB399)="" Q  I +$G(IBSORT)=0 S IBSORT=$G(IBPRNUM)  N IBPRTYP,IBINSCO,IBPROV,IBFRMTYP,IBCARE,IB35591,IBN,IBCURR,IBEXC,IBLIMIT  S IBN=0  S IBINSCO=+$P($G(^DGCR(399,IB399,"M")),"^",IBPRNUM)  S IBFRMTYP=$$FT^IBCEF(IB399),IBFRMTYP=$S(IBFRMTYP=2:2,IBFRMTYP=3:1,1:0)  S IBCARE=$S($$ISRX^IBCEF1(IB399):3,1:0) ;if an Rx refill bill  S:IBCARE=0 IBCARE=$$INPAT^IBCEF(IB399,1) S:'IBCARE IBCARE=2 ;1-inp,2-out  S IBLIMIT=$S($G(IBINSTP)="C":5,1:3) ; Limits on secondary IDs  F IBPRTYP=1:1:9 D  . N Z,IB355OV  . S IBPROV=$$PROVPTR^IBCEF7(IB399,IBPRTYP)  . Q:+IBPROV=0  . ;don't create anything if form type not CMS-1500 or UB  . Q:IBFRMTYP=0  . N IBRETARR S IBRETARR=0  . D PRACT^IBCEF71(IBINSCO,IBFRMTYP,IBCARE,IBPROV,.IBRETARR,IBPRTYP,$G(IBINSTP))  . S IB355OV="",IBEXC=""  . S Z=$O(^DGCR(399,IB399,"PRV","B",IBPRTYP,0))  . I Z S Z=$G(^DGCR(399,IB399,"PRV",Z,0)) D  .. I $P(Z,U,IBPRNUM+4)'="",$P(Z,U,IBPRNUM+11)'="" S IB355OV=$P(Z,U,IBPRNUM+4)\_U\_$P(Z,U,IBPRNUM+11)  . S IBCURR=$$COB^IBCEF(IB399)  . S IBN=0,IB35591=$$CH35591^IBCEF72(IBINSCO,IBFRMTYP,IBCARE)  . I $G(IBINSTP)="C",$G(IBPRNUM)=1,"34"[$G(IBPRTYP),"P"[$G(IBCURR),$G(IBFRMTYP)=2,$$MCRONBIL^IBEFUNC(IB399) S IB355OV=$$MCR24K^IBCEU3(IB399)\_"^12"  . I $G(IBINSTP)="O","34"[$G(IBPRTYP),"ST"[$G(IBCURR),$G(IBFRMTYP)=2,$$MCRONBIL^IBEFUNC(IB399) S IB355OV=$$MCR24K^IBCEU3(IB399)\_"^12" ;Calculate MEDICARE (WNR) specific provider qualifier and ID for CMS-1500 secondary claims  . I $P(IB355OV,U,2) D  .. I $$CHCKSEC^IBCEF73(IBFRMTYP,IBPRTYP,$G(IBINSTP),$P($G(^IBE(355.97,+$P(IB355OV,U,2),0)),U,3)) D  ... S IBEXC=$P(IB355OV,U,2),IBN=IBN+1,IBRES(IBSORT,IBPRTYP,IBN)="OVERRIDE^"\_IBINSCO\_U\_$P($G(^IBE(355.97,+IBEXC,0)),U,3)\_U\_$P(IB355OV,U)\_"^^^^^"\_+IBEXC  . I IB35591'="",IBEXC'=$P(IB35591,U,3) S:$$CHCKSEC^IBCEF73(IBFRMTYP,IBPRTYP,$G(IBINSTP),$P(IB35591,"^")) IBN=IBN+1,IBRES(IBSORT,IBPRTYP,IBN)="DEFAULT^"\_IBINSCO\_"^"\_IB35591\_"^^",$P(IBRES(IBSORT,IBPRTYP,IBN),U,9)=$P(IB35591,U,3)  . D SORT(IBSORT,IBPRTYP,IB399,.IBRETARR,.IBRES,IBN,IBEXC,IBPRNUM,IBLIMIT)  . S IBRES(IBSORT,IBPRTYP)=IBPROV  S IBRES(IBSORT)=$S(IBPRNUM=3:"T",IBPRNUM=2:"S",1:"P")  Q  ; SECIDCK(IBIFN,IBSEQ,IBTYP,IBIFN1) ; Function returns 1 if ID type ptr in  ; IBTYP is valid X12 code for the claim/prov function (IBPROVF)  ; as a sec id  ; IBSEQ = COB seq being checked  ; IBIFN1 = entry # in PRV multiple being checked  ; Called from input transform of fields .12-.14, subfile 399.0222  I $G(IBIFN)="" Q  N IBOK,IBFRM,IBCOBN,IBX12,IBPROVF  S IBPROVF=+$G(^DGCR(399,IBIFN,"PRV",IBIFN1,0))  S IBFRM=$$FT^IBCEF(IBIFN),IBFRM=$S(IBFRM=3:1,1:2) ; Form type  S IBCOBN=$$COBN^IBCEF(IBIFN) S:'IBCOBN IBCOBN=1 ; Current COB seq  S IBX12=$P($G(^IBE(355.97,+IBTYP,0)),U,3) ; X12 code for prov id typ  Q $$CHSEC^IBCEF73(IBFRM,IBPROVF,$S(IBSEQ=IBCOBN:"C",1:"O"),IBX12)  ; DEFID(IBIFN,IBPRV) ;  ; IBIFN = ien of bill  ; IBPRV = ien of entry subfile 399.0222  ; Function returns default ids: prim id def^sec id def^tert id def  ; SSN cannot be the default ID  I $G(IBIFN)="" Q ""  N Z,Z1,ID,IBZ,IBINS,IBINS4,IBUB  S IBZ=""  S IBUB=($$FT^IBCEF(IBIFN)=3)  D F^IBCEF("N-ALL ATT/REND PROV SSN/EI","IBZ","",IBIFN)  S Z=$G(^DGCR(399,IBIFN,"PRV",IBPRV,0)),ID=$P(Z,U,5,7)  F Z1=1:1:3 I $P(ID,U,Z1)="" D  . Q:'$G(^DGCR(399,IBIFN,"I"\_Z1)) S IBINS=+^("I"\_Z1)  . S $P(ID,U,Z1)=$$GETID^IBCEP2(IBIFN,2,$P(Z,U,2),Z1)  . ; Set default if null  . I $P(ID,U,Z1)="" S $P(ID,U,Z1)="VAD000"  Q ID  ; DISPID(IBXIEN) ; Display list of all prov and fac ids that will  ; extract for this bill if transmitted electronically  I $G(IBXIEN)="" Q  N IBID,IBID1,IBZ,IBCT,IBFRM,IBCOBN,IBQUIT,IBTYP,DIR,IBIFN,X,Y,Z,Z0,Z1,CO,IBN,IBCODE  S IBIFN=IBXIEN  S IBFRM=$$FT^IBCEF(IBIFN),IBCOBN=$$COBN^IBCEF(IBIFN)  W @IOF  W !,"If this bill is transmitted electronically, the following IDs will be sent:"  ; Returns all prov sec ids to be transmitted in indicated segments  S Z=+$G(^DGCR(399,IBIFN,"I1")) I Z W !," Primary Ins Co: ",$$EXTERNAL^DILFD(399,101,"",Z) I IBCOBN=1 W ?54,"<<<Current Ins"  S Z=+$G(^DGCR(399,IBIFN,"I2")) I Z W !,"Secondary Ins Co: ",$$EXTERNAL^DILFD(399,101,"",Z) I IBCOBN=2 W ?54,"<<<Current Ins"  S Z=+$G(^DGCR(399,IBIFN,"I3")) I Z W !," Tertiary Ins Co: ",$$EXTERNAL^DILFD(399,101,"",Z) I IBCOBN=3 W ?54,"<<<Current Ins"  W !!,"Provider IDs: (VistA Records OP1,OP2,OP4,OP8,OP9,OPR2,OPR3,OPR4,OPR5,OPR8):"  ;F Z=1:1:3 I $G(^DGCR(399,IBIFN,"I"\_Z)) D PROVINF(IBIFN,Z,.IBID,"",$S(IBCOBN=Z:"C",1:"O"))  ;\*432/TAZ - Added call to gather line providers and apply business rules  D ALLIDS^IBCEFP(IBIFN,.IBID)  ;\*432/TAZ - Rewrote following code to take info from the IBID array instead of File 399. This allows changes from the application of the business rules.  S IBQUIT=0  ;  F IBPRV=4,3,1,2,5,9 D  ; Process providers in order: Attending, Rendering, Referring, Operating, Supervising, and Other Operating if they exist  . I '$D(IBID("PROVINF",IBIFN,"C",1,IBPRV)) Q  . I ($Y+5)>IOSL S IBQUIT=$$NOMORE() Q:IBQUIT  . W !!?5,$$EXTERNAL^DILFD(399.0222,.01,"",IBPRV),": "\_$$EXTERNAL^DILFD(399.0222,.02,"",$P(IBID("PROVINF",IBIFN,"C",1,IBPRV),U))  . W !?8,"NPI: ",?40,$S($P($G(IBID("PROVINF",IBIFN,"C",1,IBPRV,0)),U,4)]"":$P(IBID("PROVINF",IBIFN,"C",1,IBPRV,0),U,4),1:"\*\*\*MISSING\*\*\*")  . K IBTYP  . F CO="C","O" D  .. F IBN=1,2 I $D(IBID("PROVINF",IBIFN,CO,IBN,IBPRV)) D  ... F Z0=1:1 Q:'$D(IBID("PROVINF",IBIFN,CO,IBN,IBPRV,Z0))!IBQUIT  D  .... S IBCODE=+$P(IBID("PROVINF",IBIFN,CO,IBN,IBPRV,Z0),U,9)  .... Q:$D(IBTYP(IBCODE)) ;1st of each type transmits  .... I ($Y+5)>IOSL S IBQUIT=$$NOMORE() Q:IBQUIT  .... S IBTYP(IBCODE)=""  .... W !,?8,"(",IBID("PROVINF",IBIFN,CO,IBN),") ",$$EXTERNAL^DILFD(36,4.01,"",IBCODE),?40,$P(IBID("PROVINF",IBIFN,CO,IBN,IBPRV,Z0),U,4)  ;  I IBQUIT G DISPIDX  ;  ; IB\*2\*320 - display additional IDs for ?ID  D EN^IBCEF74A(IBIFN,.IBQUIT,.IBID)  ; DISPIDX ;  I '$G(IBQUIT) S DIR(0)="EA",DIR("A")="Press RETURN to continue " W ! D ^DIR K DIR  Q  ; NOMORE() ;  S DIR(0)="EA",DIR("A")="Press RETURN for more IDs or '^' to exit: " W ! D ^DIR  W @IOF  Q (Y'=1)  ; DEFSEC(IBIFN,IBARR) ; Returns array in IBARR for default prov sec ids for ien IBIFN  ; IBARR if passed by ref is returned IBARR(prov function,COBN)=def id  I $G(IBIFN)=""  N IBCAR,IBCOBN,IBPC,IBINS,IBARRX,Q,Z,Z0,ZINS,X  K IBARR  S ZINS="",IBCOBN=$$COBN^IBCEF(IBIFN),IBPC=$S($$FT^IBCEF(IBIFN)=3:2,1:1)  S IBCAR=$$INPAT^IBCEF(IBIFN,1),IBCAR=$S('IBCAR:2,1:1)  F Z=1:1:3 S ZINS=ZINS\_+$G(^DGCR(399,IBIFN,"I"\_Z))\_U  F Z=1:1:3 I $P(ZINS,U,Z),'$P($G(^DIC(36,+$P(ZINS,U,Z),4)),U,IBPC) S $P(ZINS,U,Z)=""  S Z=0 F  S Z=$O(^DGCR(399,IBIFN,"PRV",Z)) Q:'Z  S Z0=$G(^(Z,0)) D  . F Q=1:1:3 D  .. I $P(Z0,U,Q+4)'="" S IBARR(+Z0,Q)=$P(Z0,U,Q+4) Q  ; Override  .. S IBINS=$P(ZINS,U,Q)  .. Q:'IBINS  .. S X=$$IDFIND^IBCEP2(IBIFN,"",$P(Z0,U,2),Q,1)  .. I X'="" S IBARR(+Z0,Q)=X  Q  ; | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCEF74 ;WOIFO/SS - FORMATTER/EXTRACT BILL FUNCTIONS ;31-JUL-03  ;;2.0;INTEGRATED BILLING;\*\*232,280,155,290,291,320,358,343,374,432,**592**\*\*;21-MAR-94;Build 192  ;;Per VHA Directive 2004-038, this routine should not be modified.  ; SORT(IBPRNUM,IBPRTYP,IB399,IBSRC,IBDST,IBN,IBEXC,IBSEQ,IBLIMIT) ;  D SORT^IBCEF77($G(IBPRNUM),$G(IBPRTYP),$G(IB399),.IBSRC,.IBDST,$G(IBN),$G(IBEXC),$G(IBSEQ),$G(IBLIMIT))  Q  ;  ;-- PROVINF --  ;Create array with prov info  ;Input:  ; IB399 - ien #399  ; IBPRNUM - 1=prim ins, 2= sec, 3 -tert  ; IBRES - for results  ; IBSORT - to sort OTHER INSURANCE data   ; if PROVINF is called for "C" mode of PROVIDER subroutine then   ; IBSORT can be any (say 1)  ; if PROVINF is called for "O" mode then can be more than set of data  ; - need to sort array to use it (like IBXDATA(1) and IBXDATA(2))  ; for mode "O" it should be 1 or 2 (see PROVIDER section)  ;IBINSTP - "C" -current ins, "O"-other  ;Output:  ; IBRES(PRNUM,PRTYPE,SEQ#)=PROV^INSUR^IDTYPE^ID^FORMTYP^CARETYP  ; where:(see PROVIDER) PROVINF(IB399,IBPRNUM,IBRES,IBSORT,IBINSTP) ;  I $G(IB399)="" Q  I +$G(IBSORT)=0 S IBSORT=$G(IBPRNUM)  N IBPRTYP,IBINSCO,IBPROV,IBFRMTYP,IBCARE,IB35591,IBN,IBCURR,IBEXC,IBLIMIT  S IBN=0  S IBINSCO=+$P($G(^DGCR(399,IB399,"M")),"^",IBPRNUM) **;JRA IB\*2.0\*592 Modify for Dental form 7 - treat the same as CMS-1500**  **;S** IBFRMTYP=$$FT^IBCEF(IB399),IBFRMTYP=$S(IBFRMTYP=2:2,IBFRMTYP=3:1,1:0) **;JRA IB\*2.0\*592 ';'**  S IBFRMTYP=$$FT^IBCEF(IB399),IBFRMTYP=$S((IBFRMTYP=2**!(IBFRMTYP=7)):**2,IBFRMTYP=3:1,1:0) **;JRA IB\*2.0\*592**  S IBCARE=$S($$ISRX^IBCEF1(IB399):3,1:0) ;if an Rx refill bill  S:IBCARE=0 IBCARE=$$INPAT^IBCEF(IB399,1) S:'IBCARE IBCARE=2 ;1-inp,2-out  S IBLIMIT=$S($G(IBINSTP)="C":5,1:3) ; Limits on secondary IDs  F IBPRTYP=1:1:9 D  . N Z,IB355OV  . S IBPROV=$$PROVPTR^IBCEF7(IB399,IBPRTYP)  . Q:+IBPROV=0  . ;don't create anything if form type not CMS-1500 or UB  . Q:IBFRMTYP=0  . N IBRETARR S IBRETARR=0  . D PRACT^IBCEF71(IBINSCO,IBFRMTYP,IBCARE,IBPROV,.IBRETARR,IBPRTYP,$G(IBINSTP))  . S IB355OV="",IBEXC=""  . S Z=$O(^DGCR(399,IB399,"PRV","B",IBPRTYP,0))  . I Z S Z=$G(^DGCR(399,IB399,"PRV",Z,0)) D  .. I $P(Z,U,IBPRNUM+4)'="",$P(Z,U,IBPRNUM+11)'="" S IB355OV=$P(Z,U,IBPRNUM+4)\_U\_$P(Z,U,IBPRNUM+11)  . S IBCURR=$$COB^IBCEF(IB399)  . S IBN=0,IB35591=$$CH35591^IBCEF72(IBINSCO,IBFRMTYP,IBCARE)  . **;JRA IB\*2.0\*592 Modify for Dental form 7 - treat the same as CMS-1500**  . I $G(IBINSTP)="C",$G(IBPRNUM)=1,"34"[$G(IBPRTYP),"P"[$G(IBCURR),($G(IBFRMTYP)=2!**($G(IBFRMTYP)=7)**),$$MCRONBIL^IBEFUNC(IB399) S IB355OV=$$MCR24K^IBCEU3(IB399)\_"^12"  **;JRA IB\*2.0\*592**  . ;Calculate MEDICARE (WNR) specific provider qualifier and ID for CMS-1500 secondary claim **;JRA IB\*2.0\*592**  . I $G(IBINSTP)="O","34"[$G(IBPRTYP),"ST"[$G(IBCURR),($G(IBFRMTYP)=2!($G(IBFRMTYP)=7)),$$MCRONBIL^IBEFUNC(IB399) S IB355OV=$$MCR24K^IBCEU3(IB399)\_"^12"  . I $P(IB355OV,U,2) D  .. I $$CHCKSEC^IBCEF73(IBFRMTYP,IBPRTYP,$G(IBINSTP),$P($G(^IBE(355.97,+$P(IB355OV,U,2),0)),U,3)) D  ... S IBEXC=$P(IB355OV,U,2),IBN=IBN+1,IBRES(IBSORT,IBPRTYP,IBN)="OVERRIDE^"\_IBINSCO\_U\_$P($G(^IBE(355.97,+IBEXC,0)),U,3)\_U\_$P(IB355OV,U)\_"^^^^^"\_+IBEXC  . I IB35591'="",IBEXC'=$P(IB35591,U,3) S:$$CHCKSEC^IBCEF73(IBFRMTYP,IBPRTYP,$G(IBINSTP),$P(IB35591,"^")) IBN=IBN+1,IBRES(IBSORT,IBPRTYP,IBN)="DEFAULT^"\_IBINSCO\_"^"\_IB35591\_"^^",$P(IBRES(IBSORT,IBPRTYP,IBN),U,9)=$P(IB35591,U,3)  . D SORT(IBSORT,IBPRTYP,IB399,.IBRETARR,.IBRES,IBN,IBEXC,IBPRNUM,IBLIMIT)  . S IBRES(IBSORT,IBPRTYP)=IBPROV  S IBRES(IBSORT)=$S(IBPRNUM=3:"T",IBPRNUM=2:"S",1:"P")  Q  ; SECIDCK(IBIFN,IBSEQ,IBTYP,IBIFN1) ; Function returns 1 if ID type ptr in  ; IBTYP is valid X12 code for the claim/prov function (IBPROVF)  ; as a sec id  ; IBSEQ = COB seq being checked  ; IBIFN1 = entry # in PRV multiple being checked  ; Called from input transform of fields .12-.14, subfile 399.0222  I $G(IBIFN)="" Q  N IBOK,IBFRM,IBCOBN,IBX12,IBPROVF  S IBPROVF=+$G(^DGCR(399,IBIFN,"PRV",IBIFN1,0))  S IBFRM=$$FT^IBCEF(IBIFN),IBFRM=$S(IBFRM=3:1,1:2) ; Form type  S IBCOBN=$$COBN^IBCEF(IBIFN) S:'IBCOBN IBCOBN=1 ; Current COB seq  S IBX12=$P($G(^IBE(355.97,+IBTYP,0)),U,3) ; X12 code for prov id typ  Q $$CHSEC^IBCEF73(IBFRM,IBPROVF,$S(IBSEQ=IBCOBN:"C",1:"O"),IBX12)  ; DEFID(IBIFN,IBPRV) ;  ; IBIFN = ien of bill  ; IBPRV = ien of entry subfile 399.0222  ; Function returns default ids: prim id def^sec id def^tert id def  ; SSN cannot be the default ID  I $G(IBIFN)="" Q ""  N Z,Z1,ID,IBZ,IBINS,IBINS4,IBUB  S IBZ=""  S IBUB=($$FT^IBCEF(IBIFN)=3)  D F^IBCEF("N-ALL ATT/REND PROV SSN/EI","IBZ","",IBIFN)  S Z=$G(^DGCR(399,IBIFN,"PRV",IBPRV,0)),ID=$P(Z,U,5,7)  F Z1=1:1:3 I $P(ID,U,Z1)="" D  . Q:'$G(^DGCR(399,IBIFN,"I"\_Z1)) S IBINS=+^("I"\_Z1)  . S $P(ID,U,Z1)=$$GETID^IBCEP2(IBIFN,2,$P(Z,U,2),Z1)  . ; Set default if null  . I $P(ID,U,Z1)="" S $P(ID,U,Z1)="VAD000"  Q ID  ; DISPID(IBXIEN) ; Display list of all prov and fac ids that will  ; extract for this bill if transmitted electronically  I $G(IBXIEN)="" Q  N IBID,IBID1,IBZ,IBCT,IBFRM,IBCOBN,IBQUIT,IBTYP,DIR,IBIFN,X,Y,Z,Z0,Z1,CO,IBN,IBCODE  S IBIFN=IBXIEN  S IBFRM=$$FT^IBCEF(IBIFN),IBCOBN=$$COBN^IBCEF(IBIFN)  W @IOF  W !,"If this bill is transmitted electronically, the following IDs will be sent:"  ; Returns all prov sec ids to be transmitted in indicated segments  S Z=+$G(^DGCR(399,IBIFN,"I1")) I Z W !," Primary Ins Co: ",$$EXTERNAL^DILFD(399,101,"",Z) I IBCOBN=1 W ?54,"<<<Current Ins"  S Z=+$G(^DGCR(399,IBIFN,"I2")) I Z W !,"Secondary Ins Co: ",$$EXTERNAL^DILFD(399,101,"",Z) I IBCOBN=2 W ?54,"<<<Current Ins"  S Z=+$G(^DGCR(399,IBIFN,"I3")) I Z W !," Tertiary Ins Co: ",$$EXTERNAL^DILFD(399,101,"",Z) I IBCOBN=3 W ?54,"<<<Current Ins"  **;JWS;IB\*2.0\*592;added Assistant Surgeon records to header display**  W !!,"Provider IDs: (VistA Records OP1,OP2,OP4,OP8,OP9,**OP10,OPR,OPR1**,OPR2,OPR3,OPR4,",!?29,"OPR5,**OPR7**,OPR8,**OPR9,OPRA,OPRB,OPRC**):"  ;F Z=1:1:3 I $G(^DGCR(399,IBIFN,"I"\_Z)) D PROVINF(IBIFN,Z,.IBID,"",$S(IBCOBN=Z:"C",1:"O"))  ;\*432/TAZ - Added call to gather line providers and apply business rules  D ALLIDS^IBCEFP(IBIFN,.IBID)  ;\*432/TAZ - Rewrote following code to take info from the IBID array instead of File 399. This allows changes from the application of the business rules.  S IBQUIT=0  ;  **;JWS;IB\*2.0\*592; added assistant surgeon**  F IBPRV=4,3,1,2,5,**6**,9 D  ; Process providers in order: Attending, Rendering, Referring, Operating, Supervising, and Other Operating if they exist  . I '$D(IBID("PROVINF",IBIFN,"C",1,IBPRV)) Q  . I ($Y+5)>IOSL S IBQUIT=$$NOMORE() Q:IBQUIT  . W !!?5,$$EXTERNAL^DILFD(399.0222,.01,"",IBPRV),": "\_$$EXTERNAL^DILFD(399.0222,.02,"",$P(IBID("PROVINF",IBIFN,"C",1,IBPRV),U))  . W !?8,"NPI: ",?40,$S($P($G(IBID("PROVINF",IBIFN,"C",1,IBPRV,0)),U,4)]"":$P(IBID("PROVINF",IBIFN,"C",1,IBPRV,0),U,4),1:"\*\*\*MISSING\*\*\*")  . K IBTYP  . F CO="C","O" D  .. F IBN=1,2 I $D(IBID("PROVINF",IBIFN,CO,IBN,IBPRV)) D  ... F Z0=1:1 Q:'$D(IBID("PROVINF",IBIFN,CO,IBN,IBPRV,Z0))!IBQUIT  D  .... S IBCODE=+$P(IBID("PROVINF",IBIFN,CO,IBN,IBPRV,Z0),U,9)  .... Q:$D(IBTYP(IBCODE)) ;1st of each type transmits  .... I ($Y+5)>IOSL S IBQUIT=$$NOMORE() Q:IBQUIT  .... S IBTYP(IBCODE)=""  .... W !,?8,"(",IBID("PROVINF",IBIFN,CO,IBN),") ",$$EXTERNAL^DILFD(36,4.01,"",IBCODE),?40,$P(IBID("PROVINF",IBIFN,CO,IBN,IBPRV,Z0),U,4)  ;  I IBQUIT G DISPIDX  ;  ; IB\*2\*320 - display additional IDs for ?ID  D EN^IBCEF74A(IBIFN,.IBQUIT,.IBID)  ; DISPIDX ;  I '$G(IBQUIT) S DIR(0)="EA",DIR("A")="Press RETURN to continue " W ! D ^DIR K DIR  Q  ; NOMORE() ;  S DIR(0)="EA",DIR("A")="Press RETURN for more IDs or '^' to exit: " W ! D ^DIR  W @IOF  Q (Y'=1)  ; DEFSEC(IBIFN,IBARR) ; Returns array in IBARR for default prov sec ids for ien IBIFN  ; IBARR if passed by ref is returned IBARR(prov function,COBN)=def id  I $G(IBIFN)=""  N IBCAR,IBCOBN,IBPC,IBINS,IBARRX,Q,Z,Z0,ZINS,X  K IBARR  S ZINS="",IBCOBN=$$COBN^IBCEF(IBIFN),IBPC=$S($$FT^IBCEF(IBIFN)=3:2,1:1)  S IBCAR=$$INPAT^IBCEF(IBIFN,1),IBCAR=$S('IBCAR:2,1:1)  F Z=1:1:3 S ZINS=ZINS\_+$G(^DGCR(399,IBIFN,"I"\_Z))\_U  F Z=1:1:3 I $P(ZINS,U,Z),'$P($G(^DIC(36,+$P(ZINS,U,Z),4)),U,IBPC) S $P(ZINS,U,Z)=""  S Z=0 F  S Z=$O(^DGCR(399,IBIFN,"PRV",Z)) Q:'Z  S Z0=$G(^(Z,0)) D  . F Q=1:1:3 D  .. I $P(Z0,U,Q+4)'="" S IBARR(+Z0,Q)=$P(Z0,U,Q+4) Q  ; Override  .. S IBINS=$P(ZINS,U,Q)  .. Q:'IBINS  .. S X=$$IDFIND^IBCEP2(IBIFN,"",$P(Z0,U,2),Q,1)  .. I X'="" S IBARR(+Z0,Q)=X  Q  ; | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCEF74A | | | | | | | | |
| **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** | | | | | | | | | |
| IBCEF74A ;ALB/ESG - Provider ID maint ?ID continuation ;7 Mar 2006  ;;2.0;INTEGRATED BILLING;\*\*320,343,349,395,400,432,516\*\*;21-MAR-94;Build 123  ;;Per VA Directive 6402, this routine should not be modified.  ;  Q  ; EN(IBIFN,IBQUIT,IBID) ; Display billing provider and service provider IDs as part  ; of the ?ID display/help in the billing screens.  ; Called from DISPID^IBCEF74.  NEW IBX,Z,ZI,ZN,SEQ,PSIN,DATA,QUALNM,IDNUM,FACNAME,IBZ,ORGNPI,BPZ,BPNAME,BPNPI,BPTAX,SFNPI,SFTAX  ;  ;D ALLIDS^IBCEF75(IBIFN,.IBID)  ;  ; Re-sort array by insurance sequence (P/S/T)  K IBX  F Z="BILLING PRV","LAB/FAC" F ZI="C","O" S ZN=0 F  S ZN=$O(IBID(Z,IBIFN,ZI,ZN)) Q:'ZN  D  . S SEQ=$P($G(IBID(Z,IBIFN,ZI,ZN)),U,1) Q:SEQ=""  . S IBX(Z,SEQ,ZI,ZN)=""  . Q  ;  ; Display billing provider information - IB\*2\*400  S BPZ=$$B^IBCEF79(IBIFN)  D GETBP^IBCEF79(IBIFN,"",+BPZ,"?ID",.IBZ)  S ORGNPI=$$ORGNPI^IBCEF73A(IBIFN)  I ($Y+5)>IOSL S IBQUIT=$$NOMORE^IBCEF74() I IBQUIT G EX  W !!,"Billing Provider Name and ID Information"  S BPNAME=$G(IBZ("?ID","NAME"))  I BPNAME="" S BPNAME="\*\*\*MISSING\*\*\*"  I ($Y+5)>IOSL S IBQUIT=$$NOMORE^IBCEF74() I IBQUIT G EX  W !,"Billing Provider: ",BPNAME  ;  S BPNPI=$P(ORGNPI,U,3)  I BPNPI="" S BPNPI="\*\*\*MISSING\*\*\*"  I ($Y+5)>IOSL S IBQUIT=$$NOMORE^IBCEF74() I IBQUIT G EX  W !?5,"Billing Provider NPI: ",BPNPI  ;  S BPTAX=$$NOPUNCT^IBCEF($P($G(^IBE(350.9,1,1)),U,5),1)  I BPTAX="" S BPTAX="\*\*\*MISSING\*\*\*"  I ($Y+5)>IOSL S IBQUIT=$$NOMORE^IBCEF74() I IBQUIT G EX  W !?5,"Billing Provider Tax ID (VistA Record PRV): ",BPTAX  ;  ; Display billing provider secondary ID's (current ins only)  I ($Y+5)>IOSL S IBQUIT=$$NOMORE^IBCEF74() I IBQUIT G EX  W !?5,"Billing Provider Secondary IDs (VistA Record CI1A):"  S Z="BILLING PRV"  D SECID(Z,.IBQUIT)  I IBQUIT G EX  ;  ; Now display the lab or facility primary and secondary IDs  ; This is the service facility information  ; IB\*2\*400 - check to make sure there is a service facility  ;  I $P(BPZ,U,3)="" G LPRV     ; no service facility information to display  ;  ; Service facility name, similar code as found in SUB-2  I ($Y+5)>IOSL S IBQUIT=$$NOMORE^IBCEF74() I IBQUIT G EX  W !!,"Service Facility Name and ID Information"  ;  ; MRD;IB\*2.0\*516 - Due to fields being marked for deletion, the  ; function $$SENDSF^IBCEF79 will always return '1'. Refer to  ; that function and INSFLGS^IBCEF79 for more information.  ;  ; Display note if ins co flag to suppress lab/fac data is set (only applies in switchback mode)  ;I '$$SENDSF^IBCEF79(IBIFN) D I IBQUIT G EX  ;. I ($Y+5)>IOSL S IBQUIT=$$NOMORE^IBCEF74() Q:IBQUIT  ;. W !!,"Note: Service Facility Data not sent for Current Insurance"  ;. W !," 'Send VA Lab/Facility IDs or Facility Data for VAMC?' is set to NO",!  ;. Q  ;  S FACNAME=$$GETFAC^IBCEP8(+$P(BPZ,U,4),$P(BPZ,U,3),0)  I FACNAME="" S FACNAME="\*\*\*MISSING\*\*\*"  I ($Y+5)>IOSL S IBQUIT=$$NOMORE^IBCEF74() I IBQUIT G EX  W !?5,"Facility: ",FACNAME  ;  S SFNPI=$P(ORGNPI,U,1)  I SFNPI="" S SFNPI="\*\*\*MISSING\*\*\*"  I ($Y+5)>IOSL S IBQUIT=$$NOMORE^IBCEF74() I IBQUIT G EX  W !?5,"Lab or Facility NPI: ",SFNPI  ;  S SFTAX=$$NOPUNCT^IBCEF($$EIN^IBCEP8A(IBIFN),1)  I SFTAX="" S SFTAX="\*\*\*MISSING\*\*\*"  I ($Y+5)>IOSL S IBQUIT=$$NOMORE^IBCEF74() I IBQUIT G EX  W !?5,"Lab or Facility Tax ID (VistA Record SUB): ",SFTAX  ;  ; lab/fac secondary IDs  I ($Y+5)>IOSL S IBQUIT=$$NOMORE^IBCEF74() I IBQUIT G EX  W !?5,"Lab or Facility Secondary IDs (VistA Records SUB1,SUB2,OP3,OP6,OP7):"  S Z="LAB/FAC"  D SECID(Z,.IBQUIT)  I IBQUIT G EX  ; LPRV ;Service Line Providers  I '$D(IBID("L-PROV")) G EX  ; No Line Level Providers  N IBSLC,IBN,CO,IBCODE,IBTYP,IBPRTYP,Z0  S IBSLC=0  W !!,"Service Line Providers"  F  S IBSLC=$O(IBID("L-PROV",IBIFN,IBSLC)) Q:'IBSLC  D  I IBQUIT Q  . I ($Y+6)>IOSL S IBQUIT=$$NOMORE^IBCEF74() I IBQUIT Q  . W !!?5,"Service Line: ",IBSLC  . F IBPRTYP=4,3,1,2,5,9 I $D(IBID("L-PROV",IBIFN,IBSLC,"C",1,IBPRTYP)) D  ; Process providers in order: Attending, Rendering, Referring, Operating, Supervising, and Other Operating if they exist  .. I ($Y+5)>IOSL S IBQUIT=$$NOMORE^IBCEF74() I IBQUIT Q  .. W !?5,$$EXTERNAL^DILFD(399.0404,.01,"",IBPRTYP),": ",$$EXTERNAL^DILFD(399.0404,.02,"",$P(IBID("L-PROV",IBIFN,IBSLC,"C",1,IBPRTYP),U,1))  .. W !?8,"NPI:",?40,$S($P(IBID("L-PROV",IBIFN,IBSLC,"C",1,IBPRTYP,0),U,4)]"":$P(IBID("L-PROV",IBIFN,IBSLC,"C",1,IBPRTYP,0),U,4),1:"\*\*\*MISSING\*\*\*")  .. K IBTYP  .. F CO="C","O" D  ... F IBN=1,2 D  .... F Z0=1:1 Q:'$D(IBID("L-PROV",IBIFN,IBSLC,CO,IBN,IBPRTYP,Z0))!IBQUIT  D  ..... S IBCODE=$P(IBID("L-PROV",IBIFN,IBSLC,CO,IBN,IBPRTYP,Z0),U,9)  ..... Q:$D(IBTYP(IBCODE)) ; 1st of each type transmits  ..... I ($Y+5)>IOSL S IBQUIT=$$NOMORE^IBCEF74() Q:IBQUIT  ..... S IBTYP(IBCODE)=""  ..... W !,?8,"(",IBID("L-PROV",IBIFN,IBSLC,CO,IBN),") ",$$EXTERNAL^DILFD(36,4.01,"",IBCODE),?40,$P(IBID("L-PROV",IBIFN,IBSLC,CO,IBN,IBPRTYP,Z0),U,4)  ; EX ;  Q  ; QUAL(Z,FORMTYPE) ; turn the qualifier code into a qualifier description  NEW QUAL,IEN  S QUAL=""  I $G(Z)="" G QUALX  I Z="1C" D  G QUALX   ; qualifier for Medicare Part ?  . I $G(FORMTYPE)=2 S QUAL="MEDICARE PART B"   ; 1500  . I $G(FORMTYPE)=3 S QUAL="MEDICARE PART A"   ; ub  . Q  I Z=34 S Z="SY"       ; qualifier for SSN  S IEN=+$O(^IBE(355.97,"C",Z,"")) I 'IEN G QUALX  S QUAL=$P($G(^IBE(355.97,IEN,0)),U,1) QUALX ;  Q QUAL  ; SECID(Z,IBQUIT) ; Display secondary ID and qualifier information  ; Z is the type of IDs passed in; either BILLING PRV or LAB/FAC  ; IBQUIT is returned if passed by reference  NEW SEQ,ZI,ZN,PSIN,DATA,QUALNM,IDNUM,NODATA  S IBQUIT=0,NODATA=1  F SEQ="P","S","T" D  Q:IBQUIT  . ;  . ; current ins only for billing provider secondary IDs  . I Z="BILLING PRV",SEQ'=$$COB^IBCEF(IBIFN) Q  . S ZI=""  . F  S ZI=$O(IBX(Z,SEQ,ZI)) Q:ZI=""  D  Q:IBQUIT  .. S ZN=0  .. F  S ZN=$O(IBX(Z,SEQ,ZI,ZN)) Q:'ZN  D  Q:IBQUIT  ... S PSIN=0 ; start at 0 to skip primary IDs  ... ;\*432/TAZ - Changed Q:PSIN="" to Q:'PSIN to prevent "CONTACTS" node from printing as secondary ID  ... F  S PSIN=$O(IBID(Z,IBIFN,ZI,ZN,PSIN)) Q:'PSIN  D  Q:IBQUIT  .... S DATA=$G(IBID(Z,IBIFN,ZI,ZN,PSIN))  .... S QUALNM=$$QUAL($P(DATA,U,1),$$FT^IBCEF(IBIFN))  .... S IDNUM=$P(DATA,U,2)  .... I ($Y+5)>IOSL S IBQUIT=$$NOMORE^IBCEF74() Q:IBQUIT  .... S NODATA=0  .... W !?8,"(",SEQ,") ",QUALNM,?40,IDNUM  .... I Z="LAB/FAC",$D(^DGCR(399,IBIFN,"I2")),SEQ=$$COB^IBCEF(IBIFN) W ?54,"<<<Current Ins"  .... I Z="BILLING PRV",PSIN=1 W ?54,"<<<System Generated ID"  .... Q  ... Q  .. Q  . Q  I NODATA,'IBQUIT W !?8,"(-) None Found" SECIDX ;  Q  ; | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCEF74A ;ALB/ESG - Provider ID maint ?ID continuation ;7 Mar 2006  ;;2.0;INTEGRATED BILLING;\*\*320,343,349,395,400,432,516,**592**\*\*;21-MAR-94;Build 123  ;;Per VA Directive 6402, this routine should not be modified.  ;  Q  ; EN(IBIFN,IBQUIT,IBID) ; Display billing provider and service provider IDs as part  ; of the ?ID display/help in the billing screens.  ; Called from DISPID^IBCEF74.  NEW IBX,Z,ZI,ZN,SEQ,PSIN,DATA,QUALNM,IDNUM,FACNAME,IBZ,ORGNPI,BPZ,BPNAME,BPNPI,BPTAX,SFNPI,SFTAX  ;  ;D ALLIDS^IBCEF75(IBIFN,.IBID)  ;  ; Re-sort array by insurance sequence (P/S/T)  K IBX  F Z="BILLING PRV","LAB/FAC" F ZI="C","O" S ZN=0 F  S ZN=$O(IBID(Z,IBIFN,ZI,ZN)) Q:'ZN  D  . S SEQ=$P($G(IBID(Z,IBIFN,ZI,ZN)),U,1) Q:SEQ=""  . S IBX(Z,SEQ,ZI,ZN)=""  . Q  ;  ; Display billing provider information - IB\*2\*400  S BPZ=$$B^IBCEF79(IBIFN)  D GETBP^IBCEF79(IBIFN,"",+BPZ,"?ID",.IBZ)  S ORGNPI=$$ORGNPI^IBCEF73A(IBIFN)  I ($Y+5)>IOSL S IBQUIT=$$NOMORE^IBCEF74() I IBQUIT G EX  W !!,"Billing Provider Name and ID Information"  S BPNAME=$G(IBZ("?ID","NAME"))  I BPNAME="" S BPNAME="\*\*\*MISSING\*\*\*"  I ($Y+5)>IOSL S IBQUIT=$$NOMORE^IBCEF74() I IBQUIT G EX  W !,"Billing Provider: ",BPNAME  ;  S BPNPI=$P(ORGNPI,U,3)  I BPNPI="" S BPNPI="\*\*\*MISSING\*\*\*"  I ($Y+5)>IOSL S IBQUIT=$$NOMORE^IBCEF74() I IBQUIT G EX  W !?5,"Billing Provider NPI: ",BPNPI  ;  S BPTAX=$$NOPUNCT^IBCEF($P($G(^IBE(350.9,1,1)),U,5),1)  I BPTAX="" S BPTAX="\*\*\*MISSING\*\*\*"  I ($Y+5)>IOSL S IBQUIT=$$NOMORE^IBCEF74() I IBQUIT G EX  W !?5,"Billing Provider Tax ID (VistA Record PRV): ",BPTAX  ;  ; Display billing provider secondary ID's (current ins only)  I ($Y+5)>IOSL S IBQUIT=$$NOMORE^IBCEF74() I IBQUIT G EX  W !?5,"Billing Provider Secondary IDs (VistA Record CI1A):"  S Z="BILLING PRV"  D SECID(Z,.IBQUIT)  I IBQUIT G EX  ;  ; Now display the lab or facility primary and secondary IDs  ; This is the service facility information  ; IB\*2\*400 - check to make sure there is a service facility  ;  I $P(BPZ,U,3)="" G LPRV     ; no service facility information to display  ;  ; Service facility name, similar code as found in SUB-2  I ($Y+5)>IOSL S IBQUIT=$$NOMORE^IBCEF74() I IBQUIT G EX  W !!,"Service Facility Name and ID Information"  ;  ; MRD;IB\*2.0\*516 - Due to fields being marked for deletion, the  ; function $$SENDSF^IBCEF79 will always return '1'. Refer to  ; that function and INSFLGS^IBCEF79 for more information.  ;  ; Display note if ins co flag to suppress lab/fac data is set (only applies in switchback mode)  ;I '$$SENDSF^IBCEF79(IBIFN) D I IBQUIT G EX  ;. I ($Y+5)>IOSL S IBQUIT=$$NOMORE^IBCEF74() Q:IBQUIT  ;. W !!,"Note: Service Facility Data not sent for Current Insurance"  ;. W !," 'Send VA Lab/Facility IDs or Facility Data for VAMC?' is set to NO",!  ;. Q  ;  S FACNAME=$$GETFAC^IBCEP8(+$P(BPZ,U,4),$P(BPZ,U,3),0)  I FACNAME="" S FACNAME="\*\*\*MISSING\*\*\*"  I ($Y+5)>IOSL S IBQUIT=$$NOMORE^IBCEF74() I IBQUIT G EX  W !?5,"Facility: ",FACNAME  ;  S SFNPI=$P(ORGNPI,U,1)  I SFNPI="" S SFNPI="\*\*\*MISSING\*\*\*"  I ($Y+5)>IOSL S IBQUIT=$$NOMORE^IBCEF74() I IBQUIT G EX  W !?5,"Lab or Facility NPI: ",SFNPI  ;  S SFTAX=$$NOPUNCT^IBCEF($$EIN^IBCEP8A(IBIFN),1)  I SFTAX="" S SFTAX="\*\*\*MISSING\*\*\*"  I ($Y+5)>IOSL S IBQUIT=$$NOMORE^IBCEF74() I IBQUIT G EX  W !?5,"Lab or Facility Tax ID (VistA Record SUB): ",SFTAX  ;  ; lab/fac secondary IDs  I ($Y+5)>IOSL S IBQUIT=$$NOMORE^IBCEF74() I IBQUIT G EX  W !?5,"Lab or Facility Secondary IDs (VistA Records SUB1,SUB2,OP3,OP6,OP7):"  S Z="LAB/FAC"  D SECID(Z,.IBQUIT)  I IBQUIT G EX  ; LPRV ;Service Line Providers  I '$D(IBID("L-PROV")) G EX  ; No Line Level Providers  N IBSLC,IBN,CO,IBCODE,IBTYP,IBPRTYP,Z0  S IBSLC=0  W !!,"Service Line Providers"  F  S IBSLC=$O(IBID("L-PROV",IBIFN,IBSLC)) Q:'IBSLC  D  I IBQUIT Q  . I ($Y+6)>IOSL S IBQUIT=$$NOMORE^IBCEF74() I IBQUIT Q  . W !!?5,"Service Line: ",IBSLC  **. ;JWS;IB\*2.0\*592; 6 - Assistant Surgeon**  . F IBPRTYP=4,3,1,2,5,**6**,9 I $D(IBID("L-PROV",IBIFN,IBSLC,"C",1,IBPRTYP)) D  ; Process providers in order: Attending, Rendering, Referring, Operating, Supervising**, Assistant Surgeon** and Other Operating if they exist  .. I ($Y+5)>IOSL S IBQUIT=$$NOMORE^IBCEF74() I IBQUIT Q  .. W !?5,$$EXTERNAL^DILFD(399.0404,.01,"",IBPRTYP),": ",$$EXTERNAL^DILFD(399.0404,.02,"",$P(IBID("L-PROV",IBIFN,IBSLC,"C",1,IBPRTYP),U,1))  .. W !?8,"NPI:",?40,$S($P(IBID("L-PROV",IBIFN,IBSLC,"C",1,IBPRTYP,0),U,4)]"":$P(IBID("L-PROV",IBIFN,IBSLC,"C",1,IBPRTYP,0),U,4),1:"\*\*\*MISSING\*\*\*")  .. K IBTYP  .. F CO="C","O" D  ... F IBN=1,2 D  .... F Z0=1:1 Q:'$D(IBID("L-PROV",IBIFN,IBSLC,CO,IBN,IBPRTYP,Z0))!IBQUIT  D  ..... S IBCODE=$P(IBID("L-PROV",IBIFN,IBSLC,CO,IBN,IBPRTYP,Z0),U,9)  ..... Q:$D(IBTYP(IBCODE)) ; 1st of each type transmits  ..... I ($Y+5)>IOSL S IBQUIT=$$NOMORE^IBCEF74() Q:IBQUIT  ..... S IBTYP(IBCODE)=""  ..... W !,?8,"(",IBID("L-PROV",IBIFN,IBSLC,CO,IBN),") ",$$EXTERNAL^DILFD(36,4.01,"",IBCODE),?40,$P(IBID("L-PROV",IBIFN,IBSLC,CO,IBN,IBPRTYP,Z0),U,4)  ; EX ;  Q  ; QUAL(Z,FORMTYPE) ; turn the qualifier code into a qualifier description  NEW QUAL,IEN  S QUAL=""  I $G(Z)="" G QUALX  I Z="1C" D  G QUALX   ; qualifier for Medicare Part ?  . I $G(FORMTYPE)=2 S QUAL="MEDICARE PART B"   ; 1500  . I $G(FORMTYPE)=3 S QUAL="MEDICARE PART A"   ; ub  . Q  I Z=34 S Z="SY"       ; qualifier for SSN  S IEN=+$O(^IBE(355.97,"C",Z,"")) I 'IEN G QUALX  S QUAL=$P($G(^IBE(355.97,IEN,0)),U,1) QUALX ;  Q QUAL  ; SECID(Z,IBQUIT) ; Display secondary ID and qualifier information  ; Z is the type of IDs passed in; either BILLING PRV or LAB/FAC  ; IBQUIT is returned if passed by reference  NEW SEQ,ZI,ZN,PSIN,DATA,QUALNM,IDNUM,NODATA  S IBQUIT=0,NODATA=1  F SEQ="P","S","T" D  Q:IBQUIT  . ;  . ; current ins only for billing provider secondary IDs  . I Z="BILLING PRV",SEQ'=$$COB^IBCEF(IBIFN) Q  . S ZI=""  . F  S ZI=$O(IBX(Z,SEQ,ZI)) Q:ZI=""  D  Q:IBQUIT  .. S ZN=0  .. F  S ZN=$O(IBX(Z,SEQ,ZI,ZN)) Q:'ZN  D  Q:IBQUIT  ... S PSIN=0 ; start at 0 to skip primary IDs  ... ;\*432/TAZ - Changed Q:PSIN="" to Q:'PSIN to prevent "CONTACTS" node from printing as secondary ID  ... F  S PSIN=$O(IBID(Z,IBIFN,ZI,ZN,PSIN)) Q:'PSIN  D  Q:IBQUIT  .... S DATA=$G(IBID(Z,IBIFN,ZI,ZN,PSIN))  .... S QUALNM=$$QUAL($P(DATA,U,1),$$FT^IBCEF(IBIFN))  .... S IDNUM=$P(DATA,U,2)  .... I ($Y+5)>IOSL S IBQUIT=$$NOMORE^IBCEF74() Q:IBQUIT  .... S NODATA=0  .... W !?8,"(",SEQ,") ",QUALNM,?40,IDNUM  .... I Z="LAB/FAC",$D(^DGCR(399,IBIFN,"I2")),SEQ=$$COB^IBCEF(IBIFN) W ?54,"<<<Current Ins"  .... I Z="BILLING PRV",PSIN=1 W ?54,"<<<System Generated ID"  .... Q  ... Q  .. Q  . Q  I NODATA,'IBQUIT W !?8,"(-) None Found" SECIDX ;  Q  ; | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCEF75 | | | | | | | | |
| **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** | | | | | | | | | |
| IBCEF75 ;ALB/WCJ - Provider ID functions ;13 Feb 2006  ;;2.0;INTEGRATED BILLING;\*\*320,371,400,432\*\*;21-MAR-94;Build 192  ;;Per VHA Directive 2004-038, this routine should not be modified.  ;  G AWAY AWAY 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  ;  S IBXIEN=IBIFN  D ALLPROV^IBCEF7 ; Get the Person ID's (Returns IBXSAVE)  S DAT=$$PROVID^IBCEF73(IBIFN)  S DAT("QUAL")=IBXSAVE("ID") ; this value was also passed back by above function  S SORT1="" F  S SORT1=$O(IBXSAVE("PROVINF",IBIFN,SORT1)) Q:SORT1=""  D  . S SORT2=0 F  S SORT2=$O(IBXSAVE("PROVINF",IBIFN,SORT1,SORT2)) Q:SORT2=""  D  .. S SORT3=0 F  S SORT3=$O(IBXSAVE("PROVINF",IBIFN,SORT1,SORT2,SORT3)) Q:SORT3=""  D  ... ;\*432/TAZ - Primary node now points to NPI  ... N IBPRVPTR,IBNPI  ... S IBPRVPTR=IBXSAVE("PROVINF",IBIFN,SORT1,SORT2,SORT3),IBNPI=$$GETNPI^IBCEF73A(IBPRVPTR)  ... S IBXSAVE("PROVINF",IBIFN,SORT1,SORT2,SORT3,0)="PRIMARY"\_U\_U\_$$STRIP^IBCEF76($S(IBNPI]"":"XX",1:"")\_U\_IBNPI,1,U,IBSTRIP)  ... F I=1:1 Q:'$D(IBXSAVE("PROVINF",IBIFN,SORT1,SORT2,SORT3,I)) D  .... S $P(IBXSAVE("PROVINF",IBIFN,SORT1,SORT2,SORT3,I),U,3,4)=$$STRIP^IBCEF76($P(IBXSAVE("PROVINF",IBIFN,SORT1,SORT2,SORT3,I),U,3,4),1,U,IBSTRIP)  ;  D LFIDS^IBCEF76(IBIFN,.IBXSAVE,IBSTRIP,SEG) ; Get the Lab/Facility IDs  ;  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=$G(IBXSAVE("PROVINF",IBIFN,SORT1,SORT2,ARIEN,1))  . ;  . D BPIDS(IBIFN,.IBXSAVE,SORT1,SORT2,COB,IBSTRIP,SEG)  Q  ;  BPIDS(IBIFN,IDS,SORT1,SORT2,COB,IBSTRIP,SEG) ; Get all the billing provider IDs and qualifiers from the claim and file 355.92  N DAT,IBFRMTYP,IBCARE,IBDIV,IBINS,MAIN,IBCCOB,USED,PLANTYPE,I,CNT,QUAL,ARF,M1,DEF,IDDIV,IBLIMIT,IEN,ID,IB2  ;  S DAT=$G(^DGCR(399,IBIFN,0))  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=$$INPAT^IBCEF(IBIFN) S:'IBCARE IBCARE=2 ;1-inp,2-out  S IBDIV=+$P(DAT,U,22)  S MAIN=$$MAIN^IBCEP2B() ; get the IEN for main Division  S IBCCOB=$$COBN^IBCEF(IBIFN) ; Current Insurance  S IBINS=$P($G(^DGCR(399,IBIFN,"I"\_COB)),U)  Q:IBINS=""  ;  S IDS("BILLING PRV",IBIFN,SORT1,SORT2)=$E("PST",COB)  ;  ; Primary ID  S IDS("BILLING PRV",IBIFN,SORT1,SORT2,0)=$$STRIP^IBCEF76($$TAXID(),1,U,IBSTRIP)  S USED($P(IDS("BILLING PRV",IBIFN,SORT1,SORT2,0),U))=""  ;  ; Secondary #1 - This is the ID Emdeon uses for sorting  S IDS("BILLING PRV",IBIFN,SORT1,SORT2,1)=$$STRIP^IBCEF76($$BPSID1(IBDIV),1,U,IBSTRIP)  S USED($P(IDS("BILLING PRV",IBIFN,SORT1,SORT2,1),U))=""  ;  ; Check if this is a plan type which gets no secondary IDs  S M1=$G(^DGCR(399,IBIFN,"M1"))  ; the following check is the current value of the flag, not when the claim was created.   S PLANTYPE=$$POLTYP^IBCEF3(IBIFN,COB)  I PLANTYPE]"",$D(^DIC(36,IBINS,13,"B",PLANTYPE)) Q   ;  ; Secondary #2  ; If there is a ID send with quailifer (stored or computed)  I $TR($P(M1,U,COB+1)," ")]"" D  . S QUAL=""  . S DAT=$P(M1,U,COB+9)  . I DAT S QUAL=$$STRIP^IBCEF76($P($G(^IBE(355.97,DAT,0)),U,3),1,,IBSTRIP)  . ; the null check is needed to be backwards compatible  . I QUAL=""!(QUAL="1J") S QUAL=$$STRIP^IBCEF76($$OLDWAY(IBIFN,COB),1,,IBSTRIP)  . S IB2=QUAL\_U\_$$STRIP^IBCEF76($P(M1,U,COB+1),1,,IBSTRIP)  ;  ;WCJ;IB\*2.0\*432;START  ;I $TR($P(M1,U,COB+1)," ")="" S IB2=$$STRIP^IBCEF76($$OLDWAY(IBIFN,COB),1,,IBSTRIP)\_U\_$$STRIP^IBCEF76($$GET1^DIQ(350.9,1,1.05),1,,IBSTRIP)  ;  I $G(IB2)]"",$P(IB2,U)]"",$P(IB2,U,2)]"" D  ;TAZ - Changed $G(IB2) to $G(IB2)]""  . S IDS("BILLING PRV",IBIFN,SORT1,SORT2,2)=IB2  . ;S IDS("BILLING PRV",IBIFN,SORT1,SORT2,2,"PTQ")=$$OLDWAY(IBIFN,COB)  . S USED($P(IB2,U))=""  ;WCJ;IB\*2.0\*432  ;  S CNT=$S('$D(IDS("BILLING PRV",IBIFN,SORT1,SORT2,2)):2,1:3)  S IBLIMIT=8  S IEN=0 F  S IEN=$O(^IBA(355.92,"B",IBINS,IEN)) Q:IEN=""  D  Q:CNT>IBLIMIT  . S DAT=$G(^IBA(355.92,IEN,0))  . Q:$P(DAT,U,8)'="A"   ; only allow additional IDs  . Q:$P(DAT,U,7)=""  ; No Provider ID  . Q:$P(DAT,U,6)=""  ; No ID Qualifier  . I IBFRMTYP=1 Q:$P(DAT,U,4)=2  . I IBFRMTYP=2 Q:$P(DAT,U,4)=1  . ;  . ; Check if we already have one of these  . S QUAL=$$STRIP^IBCEF76($P(DAT,U,6),1,,IBSTRIP)  . S QUAL=$P($G(^IBE(355.97,QUAL,0)),U,3)  . Q:QUAL=""  . Q:$D(USED(QUAL))  . ;  . S IDS("BILLING PRV",IBIFN,SORT1,SORT2,CNT)=QUAL\_U\_$$STRIP^IBCEF76($P(DAT,U,7),1,,IBSTRIP)  . S CNT=CNT+1,USED(QUAL)=""  ;  Q  ; OLDWAY(IBIFN,COB) ; Figure out the qualifier the old way if it's not stored with the claim.  ; It's based on the plan type. This is used for Billing Provider Secondary ID #2  N PLANTYPE  S PLANTYPE=$$POLTYP^IBCEF3(IBIFN,COB)  Q $$SOP^IBCEP2B(IBIFN,PLANTYPE)  ; BPSID1(DIV) ; Return the Billing Provider Secondary ID #1 and qualifier which Emdeon uses to sort IBIFNs  N DATA  S DATA=$P($$SITE^VASITE(DT,$S(DIV:DIV,1:+$$PRIM^VASITE(DT))),U,3)  S DATA=$E("0000",1,7-$L(DATA))\_$E(DATA,4,7)  Q "G5"\_U\_DATA  ; TAXID() ; Return the Billing Provider Primary ID and qualifier which is the TAXID for the site and also the qualifier  N DATA  S DATA=$P($G(^IBE(350.9,1,1)),U,5)  S DATA=$$NOPUNCT^IBCEF(DATA,1)  Q 24\_U\_DATA  ; CLEANUP(IBXSAVE) ; Clean up   K IBXSAVE("PROVINF")  K IBXSAVE("LAB/FAC")  K IBXSAVE("BILLING PRV")  K IBXSAVE("ID")  Q | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCEF75 ;ALB/WCJ - Provider ID functions ;13 Feb 2006  ;;2.0;INTEGRATED BILLING;\*\*320,371,400,432,**592**\*\*;21-MAR-94;Build 192  ;;Per VHA Directive 2004-038, this routine should not be modified.  ;  G AWAY AWAY 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  ;  S IBXIEN=IBIFN  D ALLPROV^IBCEF7 ; Get the Person ID's (Returns IBXSAVE)  S DAT=$$PROVID^IBCEF73(IBIFN)  S DAT("QUAL")=IBXSAVE("ID") ; this value was also passed back by above function  S SORT1="" F  S SORT1=$O(IBXSAVE("PROVINF",IBIFN,SORT1)) Q:SORT1=""  D  . S SORT2=0 F  S SORT2=$O(IBXSAVE("PROVINF",IBIFN,SORT1,SORT2)) Q:SORT2=""  D  .. S SORT3=0 F  S SORT3=$O(IBXSAVE("PROVINF",IBIFN,SORT1,SORT2,SORT3)) Q:SORT3=""  D  ... ;\*432/TAZ - Primary node now points to NPI  ... N IBPRVPTR,IBNPI  ... S IBPRVPTR=IBXSAVE("PROVINF",IBIFN,SORT1,SORT2,SORT3),IBNPI=$$GETNPI^IBCEF73A(IBPRVPTR)  ... S IBXSAVE("PROVINF",IBIFN,SORT1,SORT2,SORT3,0)="PRIMARY"\_U\_U\_$$STRIP^IBCEF76($S(IBNPI]"":"XX",1:"")\_U\_IBNPI,1,U,IBSTRIP)  ... F I=1:1 Q:'$D(IBXSAVE("PROVINF",IBIFN,SORT1,SORT2,SORT3,I)) D  .... S $P(IBXSAVE("PROVINF",IBIFN,SORT1,SORT2,SORT3,I),U,3,4)=$$STRIP^IBCEF76($P(IBXSAVE("PROVINF",IBIFN,SORT1,SORT2,SORT3,I),U,3,4),1,U,IBSTRIP)  ;  D LFIDS^IBCEF76(IBIFN,.IBXSAVE,IBSTRIP,SEG) ; Get the Lab/Facility IDs  ;  S IBFRMTYP=$$FT^IBCEF(IBIFN)  **;JWS;IB\*2.0\*592; Dental form 7**  S ARIEN=$S(IBFRMTYP=2:3,**IBFRMTYP=7:3**,1:4)  S IBCCOB=$$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=$G(IBXSAVE("PROVINF",IBIFN,SORT1,SORT2,ARIEN,1))  . ;  . D BPIDS(IBIFN,.IBXSAVE,SORT1,SORT2,COB,IBSTRIP,SEG)  Q  ;  BPIDS(IBIFN,IDS,SORT1,SORT2,COB,IBSTRIP,SEG) ; Get all the billing provider IDs and qualifiers from the claim and file 355.92  N DAT,IBFRMTYP,IBCARE,IBDIV,IBINS,MAIN,IBCCOB,USED,PLANTYPE,I,CNT,QUAL,ARF,M1,DEF,IDDIV,IBLIMIT,IEN,ID,IB2  ;  S DAT=$G(^DGCR(399,IBIFN,0))  **;JWS;IB\*2.0\*592;Dental form 7**  S IBFRMTYP=$$FT^IBCEF(IBIFN),IBFRMTYP=$S(IBFRMTYP=2:2,**IBFRMTYP=7:4**,IBFRMTYP=3:1,1:0)  S IBCARE=$S($$ISRX^IBCEF1(IBIFN):3,1:0) ;if an Rx refill bill  S:IBCARE=0 IBCARE=$$INPAT^IBCEF(IBIFN) S:'IBCARE IBCARE=2 ;1-inp,2-out  S IBDIV=+$P(DAT,U,22)  S MAIN=$$MAIN^IBCEP2B() ; get the IEN for main Division  S IBCCOB=$$COBN^IBCEF(IBIFN) ; Current Insurance  S IBINS=$P($G(^DGCR(399,IBIFN,"I"\_COB)),U)  Q:IBINS=""  ;  S IDS("BILLING PRV",IBIFN,SORT1,SORT2)=$E("PST",COB)  ;  ; Primary ID  S IDS("BILLING PRV",IBIFN,SORT1,SORT2,0)=$$STRIP^IBCEF76($$TAXID(),1,U,IBSTRIP)  S USED($P(IDS("BILLING PRV",IBIFN,SORT1,SORT2,0),U))=""  ;  ; Secondary #1 - This is the ID Emdeon uses for sorting  S IDS("BILLING PRV",IBIFN,SORT1,SORT2,1)=$$STRIP^IBCEF76($$BPSID1(IBDIV),1,U,IBSTRIP)  S USED($P(IDS("BILLING PRV",IBIFN,SORT1,SORT2,1),U))=""  ;  ; Check if this is a plan type which gets no secondary IDs  S M1=$G(^DGCR(399,IBIFN,"M1"))  ; the following check is the current value of the flag, not when the claim was created.   S PLANTYPE=$$POLTYP^IBCEF3(IBIFN,COB)  I PLANTYPE]"",$D(^DIC(36,IBINS,13,"B",PLANTYPE)) Q   ;  ; Secondary #2  ; If there is a ID send with quailifer (stored or computed)  I $TR($P(M1,U,COB+1)," ")]"" D  . S QUAL=""  . S DAT=$P(M1,U,COB+9)  . I DAT S QUAL=$$STRIP^IBCEF76($P($G(^IBE(355.97,DAT,0)),U,3),1,,IBSTRIP)  . ; the null check is needed to be backwards compatible  . I QUAL=""!(QUAL="1J") S QUAL=$$STRIP^IBCEF76($$OLDWAY(IBIFN,COB),1,,IBSTRIP)  . S IB2=QUAL\_U\_$$STRIP^IBCEF76($P(M1,U,COB+1),1,,IBSTRIP)  ;  ;WCJ;IB\*2.0\*432;START  ;I $TR($P(M1,U,COB+1)," ")="" S IB2=$$STRIP^IBCEF76($$OLDWAY(IBIFN,COB),1,,IBSTRIP)\_U\_$$STRIP^IBCEF76($$GET1^DIQ(350.9,1,1.05),1,,IBSTRIP)  ;  I $G(IB2)]"",$P(IB2,U)]"",$P(IB2,U,2)]"" D  ;TAZ - Changed $G(IB2) to $G(IB2)]""  . S IDS("BILLING PRV",IBIFN,SORT1,SORT2,2)=IB2  . ;S IDS("BILLING PRV",IBIFN,SORT1,SORT2,2,"PTQ")=$$OLDWAY(IBIFN,COB)  . S USED($P(IB2,U))=""  ;WCJ;IB\*2.0\*432  ;  S CNT=$S('$D(IDS("BILLING PRV",IBIFN,SORT1,SORT2,2)):2,1:3)  S IBLIMIT=8  S IEN=0 F  S IEN=$O(^IBA(355.92,"B",IBINS,IEN)) Q:IEN=""  D  Q:CNT>IBLIMIT  . S DAT=$G(^IBA(355.92,IEN,0))  . Q:$P(DAT,U,8)'="A"   ; only allow additional IDs  . Q:$P(DAT,U,7)=""  ; No Provider ID  . Q:$P(DAT,U,6)=""  ; No ID Qualifier  **. ;JWS;IB\*2.0\*592;exclude dental now**  . I IBFRMTYP=1 Q:$P(DAT,U,4)=2 **Q:$P(DAT,U,4)=4**  . I IBFRMTYP=2 Q:$P(DAT,U,4)=1 **Q:$P(DAT,U,4)=4**  **. ;JWS;IB\*2.0\*592;Dental form**  . **I IBFRMTYP=4 Q:$P(DAT,U,4)=1 Q:$P(DAT,U,4)=2**  . ;  . ; Check if we already have one of these  . S QUAL=$$STRIP^IBCEF76($P(DAT,U,6),1,,IBSTRIP)  . S QUAL=$P($G(^IBE(355.97,QUAL,0)),U,3)  . Q:QUAL=""  . Q:$D(USED(QUAL))  . ;  . S IDS("BILLING PRV",IBIFN,SORT1,SORT2,CNT)=QUAL\_U\_$$STRIP^IBCEF76($P(DAT,U,7),1,,IBSTRIP)  . S CNT=CNT+1,USED(QUAL)=""  ;  Q  ; OLDWAY(IBIFN,COB) ; Figure out the qualifier the old way if it's not stored with the claim.  ; It's based on the plan type. This is used for Billing Provider Secondary ID #2  N PLANTYPE  S PLANTYPE=$$POLTYP^IBCEF3(IBIFN,COB)  Q $$SOP^IBCEP2B(IBIFN,PLANTYPE)  ; BPSID1(DIV) ; Return the Billing Provider Secondary ID #1 and qualifier which Emdeon uses to sort IBIFNs  N DATA  S DATA=$P($$SITE^VASITE(DT,$S(DIV:DIV,1:+$$PRIM^VASITE(DT))),U,3)  S DATA=$E("0000",1,7-$L(DATA))\_$E(DATA,4,7)  Q "G5"\_U\_DATA  ; TAXID() ; Return the Billing Provider Primary ID and qualifier which is the TAXID for the site and also the qualifier  N DATA  S DATA=$P($G(^IBE(350.9,1,1)),U,5)  S DATA=$$NOPUNCT^IBCEF(DATA,1)  Q 24\_U\_DATA  ; CLEANUP(IBXSAVE) ; Clean up   K IBXSAVE("PROVINF")  K IBXSAVE("LAB/FAC")  K IBXSAVE("BILLING PRV")  K IBXSAVE("ID")  Q | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCEF76 | | | | | | | | |
| **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** | | | | | | | | | |
| IBCEF76 ;ALB/WCJ - Provider ID functions ;13 Feb 2006  ;;2.0;INTEGRATED BILLING;\*\*320,349,400,432,516\*\*;21-MAR-94;Build 123  ;;Per VA Directive 6402, this routine should not be modified.  ;  G AWAY AWAY Q  ; LFIDS(IBIFN,IDS,IBSTRIP,SEG) ;  ; Pass in the the internal claim number and return the array of IDS.  ; IDS("C"urrent or "O"ther, Order of Insurance within subscript 1, order of ID within subscript 2)  ; IDS("C",1)="P"  ; IDS("C",1,0)=Qualifier^Primary ID  ; IDS("C",1,1)=Qualifier^Sec ID #1  ; IDS("C",1,2)=Qualifier^Sec ID #2  ;  N DAT,IBFRMTYP,IBCARE,IBDIV,IBINS,OUTFAC,MAIN,IBCCOB,TMPIDS,COB,IBSORT1,IBSORT2,IBLIMIT,IBLF  ;  S DAT=$G(^DGCR(399,IBIFN,0))  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=$$INPAT^IBCEF(IBIFN) S:'IBCARE IBCARE=2 ;1-inp,2-out  S IBDIV=+$P(DAT,U,22)  S OUTFAC=$P($G(^DGCR(399,IBIFN,"U2")),U,10)  S MAIN=$$MAIN^IBCEP2B() ; get the IEN for main Division  ;  S IBCCOB=$$COBN^IBCEF(IBIFN)  F COB=1:1:3 D  . S IBSORT1=$S(COB=IBCCOB:"C",1:"O")  . S IBSORT2=$S(IBSORT1="C":1,COB=1:1,COB=2&(IBCCOB=1):1,1:2)  . S IBLIMIT=$S(IBSORT1="C":5,1:3) ; Limit secondary IDs  . S DAT=$G(^DGCR(399,IBIFN,"I"\_COB))  . ;  . S IBINS=$P(DAT,U) ; insurance PTR 36  . Q:IBINS=""  . ;  . ; IB\*2\*400 - esg - 9/24/08, 2/24/09 - if there is no service facility for this claim at this COB, then get out  . S IBLF=$$B^IBCEF79(IBIFN,COB) ; billing provider/service facility function  . I $P(IBLF,U,3)="" Q                 ; no service facility data at this COB, don't build this "LAB/FAC" area  . ;  . I OUTFAC]"" D  Q  .. D NONVALF(IBIFN,OUTFAC\_";IBA(355.93,",IBINS,IBFRMTYP,IBCARE,.IDS,IBSORT1,IBSORT2,COB,IBLIMIT,IBSTRIP,SEG)  . ;  . I OUTFAC="" D  .. ;  .. ; MRD;IB\*2.0\*516 - Due to fields being marked for deletion, the  .. ; function $$SENDSF^IBCEF79 will always return '1'. Refer to  .. ; that function and INSFLGS^^IBCEF79 for more information.  .. ;  .. ; if ins co flag says to not send svc fac data and we're sending an EDI claim, then get out  .. ;I '$$SENDSF^IBCEF79(IBIFN,COB),$G(^TMP("IBTX",$J,IBIFN)) Q  .. ;  .. ;IB\*2.0\*432/TAZ Moved Taxid setup inside VALF look to send as secondary ID for Medicare claims.  .. ;S IDS("LAB/FAC",IBIFN,IBSORT1,IBSORT2,0)=$$STRIP($$TAXID^IBCEF75(),1,U,IBSTRIP)  .. D VALF(IBIFN,IBINS,IBFRMTYP,IBDIV,.IDS,IBSORT1,IBSORT2,COB,IBLIMIT,IBSTRIP,SEG)  Q  ; VALF(IBIFN,INS,FT,DIV,IDS,SORT1,SORT2,COB,IBLIMIT,IBSTRIP,SEG) ; Get VA Lab/Fac Secondary IDs  ; Pass in INS - IEN to file 36  ; FT - 1 = UB 2 = 1500  ; DIV - PTR to 40.8  ;  N Z,Z0,ID,QUAL,MAIN,IDTBL,CNT,Z,IBMCR  S MAIN=$$MAIN^IBCEP2B() ; get the IEN for main Division  S Z=0 F  S Z=$O(^IBA(355.92,"B",INS,Z)) Q:'Z  D  . S Z0=$G(^IBA(355.92,Z,0))  . Q:$P(Z0,U,8)'="LF"   ; Screen out anything other than Lab or Facility  . I +$P(Z0,U,4) Q:$P(Z0,U,4)'=FT   ; Form type must match that passed in or be a 0 which allows both  . S ID=$$STRIP($P(Z0,U,7),1,,IBSTRIP)  . S QUAL=$$STRIP($P(Z0,U,6),1,,IBSTRIP)  . Q:QUAL=""   ; Needs a qualifier  . S QUAL=$P($G(^IBE(355.97,QUAL,0)),U,3)  . I FT=1,SORT1="O" Q:$$OP3^IBCEF73(FT)'[(U\_QUAL\_U) ; Institutional  . I FT=2,SORT1="O" Q:$$OP7^IBCEF73(FT)'[(U\_QUAL\_U) ; Professional  . I $P(Z0,U,5)=""!($P(Z0,U,5)=0)!($P(Z0,U,5)=MAIN) S IDTBL("DEF",QUAL)=ID  ; set up default for main division  . I $P(Z0,U,5)=DIV S IDTBL("DIV",QUAL)=ID  ; set up default for division  S CNT=0  S IDS("LAB/FAC",IBIFN,SORT1,SORT2)=$E("PST",COB)  ;IB\*2.0\*432/TAZ If Medicare send Tax ID as 1st Secondary ID ; only if it's not a printed form  S IBMCR=""  I '(($G(IBXFORM)=2)!($G(IBXFORM)=3)) S IBMCR=$$MCRONBIL^IBEFUNC(IBIFN)  I IBMCR S CNT=CNT+1,IDS("LAB/FAC",IBIFN,SORT1,SORT2,CNT)="LU"\_U\_$$STRIP($P($$TAXID^IBCEF75(),U,2),1,U,IBSTRIP)  I $D(IDTBL("DIV")) D  Q  . S Z="" F  S Z=$O(IDTBL("DIV",Z)) Q:Z=""  D  .. ;IB\*2.0\*432/TAZ If Medicare, screen out Tax ID  .. I IBMCR,(Z=24) Q  .. S CNT=CNT+1,IDS("LAB/FAC",IBIFN,SORT1,SORT2,CNT)=Z\_U\_IDTBL("DIV",Z) Q:CNT=IBLIMIT  I $D(IDTBL("DEF")) D  Q  . S Z="" F  S Z=$O(IDTBL("DEF",Z)) Q:Z=""  D  .. ;IB\*2.0\*432/TAZ If Medicare, screen out Tax ID  .. I IBMCR,(Z=24) Q  .. S CNT=CNT+1,IDS("LAB/FAC",IBIFN,SORT1,SORT2,CNT)=Z\_U\_IDTBL("DEF",Z) Q:CNT=IBLIMIT  Q  ; NONVALF(IBIFN,PRV,INS,FT,PT,IDS,SORT1,SORT2,COB,IBLIMIT,IBSTRIP,SEG) ; Get Non VA Lab/Fac Secondary IDs  ; Pass in PRV - VPTR - PTR to 355.93 (in format of variabel pointer IEN;IBA(355.93,  ; Pass in INS - PTR to 36 of null (not provide by insurance company)  ; FT - 1 = UB 2 = 1500  ; PT - Patient Type - 1 inpatient 2 outpatient  ; IDS array being returned  ; SORT1 - "C"urrent or "O"ther  ; SORT2 - 1 if current or (1 or 2 if other)  N Z,Z0,ID,QUAL,IDTBL,CNT,IBMCR  S Z=0 F  S Z=$O(^IBA(355.9,"B",PRV,Z)) Q:'Z  D  . S Z0=$G(^IBA(355.9,Z,0))  . I +$P(Z0,U,4) Q:$P(Z0,U,4)'=FT   ; Form type must match that passed in or be a 0 which allows both UB and 1500  . I +$P(Z0,U,5) Q:$P(Z0,U,5)'=PT   ; Patient type must match that passed in or be a 0 which allows both in patient and outpatient  . I INS]"",$P(Z0,U,2)]"",INS'=$P(Z0,U,2) Q  . S ID=$$STRIP($P(Z0,U,7),1,,IBSTRIP)  . Q:ID=""  . S QUAL=$$STRIP($P(Z0,U,6),1,,IBSTRIP)  . Q:QUAL=""   ; Needs a qualifier  . S QUAL=$P($G(^IBE(355.97,QUAL,0)),U,3)  . Q:QUAL=""  . I FT=1,SORT1="O" Q:$$OP3^IBCEF73(FT)'[(U\_QUAL\_U) ; Institutional  . I FT=2,SORT1="O" Q:$$OP7^IBCEF73(FT)'[(U\_QUAL\_U) ; Professional  . I $G(SEG)="SUB1" Q:$$SUB1^IBCEF73(FT)'[(U\_QUAL\_U)  . I $P(Z0,U,2)="" S IDTBL("OWN",QUAL)=ID  ; set up default of lab or facilities own ids  . I $P(Z0,U,2)=INS S IDTBL("INS",QUAL)=ID  ; set up default for division  ;  S CNT=0  S IDS("LAB/FAC",IBIFN,SORT1,SORT2)=$E("PST",COB)\_U\_PRV  S IDS("LAB/FAC",IBIFN,SORT1,SORT2,"CONTACT")=$G(^IBA(355.93,+PRV,1))  ; get primary  S Z0=$G(^IBA(355.93,+PRV,0))  ;IB\*2.0\*432/TAZ If Medicare send Tax ID as 1st Secondary ID  S IBMCR=""  I '(($G(IBXFORM)=2)!($G(IBXFORM)=3)) S IBMCR=$$MCRONBIL^IBEFUNC(IBIFN)  ;I $P(Z0,U,9)]"",$P(Z0,U,13)]"",IBMCR S CNT=CNT+1,IDS("LAB/FAC",IBIFN,SORT1,SORT2,CNT)="LU"\_U\_$$STRIP($P($G(^IBE(355.97,$P(Z0,U,13),0)),U,3)\_U\_$P(Z0,U,9),1,U,IBSTRIP)  I $P(Z0,U,9)]"",$P(Z0,U,13)]"",IBMCR S CNT=CNT+1,IDS("LAB/FAC",IBIFN,SORT1,SORT2,CNT)="LU"\_U\_$$STRIP($P(Z0,U,9),1,U,IBSTRIP)  ; get secondarys in order  I $D(IDTBL("INS")) D  . N Z S Z="" F  S Z=$O(IDTBL("INS",Z)) Q:Z=""  D  .. ;IB\*2.0\*432/TAZ If Medicare, screen out Tax ID  .. I IBMCR,(Z=24) Q  .. S CNT=CNT+1,IDS("LAB/FAC",IBIFN,SORT1,SORT2,CNT)=Z\_U\_IDTBL("INS",Z) Q:CNT=IBLIMIT  I $D(IDTBL("OWN")),CNT'=IBLIMIT D  . N Z S Z="" F  S Z=$O(IDTBL("OWN",Z)) Q:Z=""  D  .. ;IB\*2.0\*432/TAZ If Medicare, screen out Tax ID  .. I IBMCR,(Z=24) Q  .. I '$D(IDTBL("INS",Z)) S CNT=CNT+1,IDS("LAB/FAC",IBIFN,SORT1,SORT2,CNT)=Z\_U\_IDTBL("OWN",Z) Q:CNT=IBLIMIT  Q  ; STRIP(X,SPACE,EXC,IBSTRIP) ;  ; Strip punctuation from data in X  ; SPACE = flag if 1 strip SPACES  ; EXC = list of punct not to strip  ;   Q:'$G(IBSTRIP) X  Q $$NOPUNCT^IBCEF(X,$G(SPACE),$G(EXC))  ; OTH(IBIFN,IBXSAVE,IBXDATA,COND,SEG) ; Procedure used in piece 2 of some output  ; formatter segments for other insurance  ; COND = 0/1 value passed in that determines whether or not to call the  ; provider ID function  ; SEG = name of segment for use in calling ID^IBCEF2 (4 characters)  ;  N Z  ;\*432/TAZ - Changed Clean up and Setup routines to IBCEFP\*  ;D CLEANUP^IBCEF75(.IBXSAVE)  ;I COND D ALLIDS^IBCEF75(IBIFN,.IBXSAVE,1)  D CLEANUP^IBCEFP1(.IBXSAVE)  I COND D ALLIDS^IBCEFP(IBIFN,.IBXSAVE,1)  ;  ; Special Check: if Other Insurance #2 has secondary ID's while Other  ; Insurance #1 does not, then move up #2 to be #1 here. This is to  ; ensure the output formatter IBXDATA array is built properly.  ;  I $O(IBXSAVE("LAB/FAC",IBIFN,"O",2,0)),'$O(IBXSAVE("LAB/FAC",IBIFN,"O",1,0)) D  . K IBXSAVE("LAB/FAC",IBIFN,"O",1)  . M IBXSAVE("LAB/FAC",IBIFN,"O",1)=IBXSAVE("LAB/FAC",IBIFN,"O",2)  . K IBXSAVE("LAB/FAC",IBIFN,"O",2)  . Q  ;  K IBXDATA  S Z=0  F  S Z=$O(IBXSAVE("LAB/FAC",IBIFN,"O",Z)) Q:'Z  D  . I '$O(IBXSAVE("LAB/FAC",IBIFN,"O",Z,0)) Q  . S IBXDATA(Z)=$P($G(IBXSAVE("LAB/FAC",IBIFN,"O",Z)),U,1)  . I Z>1 D ID^IBCEF2(Z,SEG)  . Q OTHX ;  Q  ; | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCEF76 ;ALB/WCJ - Provider ID functions ;13 Feb 2006  ;;2.0;INTEGRATED BILLING;\*\*320,349,400,432,516,**592**\*\*;21-MAR-94;Build 123  ;;Per VA Directive 6402, this routine should not be modified.  ;  G AWAY AWAY Q  ; LFIDS(IBIFN,IDS,IBSTRIP,SEG) ;  ; Pass in the the internal claim number and return the array of IDS.  ; IDS("C"urrent or "O"ther, Order of Insurance within subscript 1, order of ID within subscript 2)  ; IDS("C",1)="P"  ; IDS("C",1,0)=Qualifier^Primary ID  ; IDS("C",1,1)=Qualifier^Sec ID #1  ; IDS("C",1,2)=Qualifier^Sec ID #2  ;  N DAT,IBFRMTYP,IBCARE,IBDIV,IBINS,OUTFAC,MAIN,IBCCOB,TMPIDS,COB,IBSORT1,IBSORT2,IBLIMIT,IBLF  ;  S DAT=$G(^DGCR(399,IBIFN,0)) **;JWS;IB\*2.0\*592;Dental form 7, same as form 2**  S IBFRMTYP=$$FT^IBCEF(IBIFN),IBFRMTYP=$S(IBFRMTYP=2:2,**IBFRMTYP=7:4**,IBFRMTYP=3:1,1:0)  S IBCARE=$S($$ISRX^IBCEF1(IBIFN):3,1:0) ;if an Rx refill bill  S:IBCARE=0 IBCARE=$$INPAT^IBCEF(IBIFN) S:'IBCARE IBCARE=2 ;1-inp,2-out  S IBDIV=+$P(DAT,U,22)  S OUTFAC=$P($G(^DGCR(399,IBIFN,"U2")),U,10)  S MAIN=$$MAIN^IBCEP2B() ; get the IEN for main Division  ;  S IBCCOB=$$COBN^IBCEF(IBIFN)  F COB=1:1:3 D  . S IBSORT1=$S(COB=IBCCOB:"C",1:"O")  . S IBSORT2=$S(IBSORT1="C":1,COB=1:1,COB=2&(IBCCOB=1):1,1:2)  . S IBLIMIT=$S(IBSORT1="C":5,1:3) ; Limit secondary IDs  . S DAT=$G(^DGCR(399,IBIFN,"I"\_COB))  . ;  . S IBINS=$P(DAT,U) ; insurance PTR 36  . Q:IBINS=""  . ;  . ; IB\*2\*400 - esg - 9/24/08, 2/24/09 - if there is no service facility for this claim at this COB, then get out  . S IBLF=$$B^IBCEF79(IBIFN,COB) ; billing provider/service facility function  . I $P(IBLF,U,3)="" Q                 ; no service facility data at this COB, don't build this "LAB/FAC" area  . ;  . I OUTFAC]"" D  Q  .. D NONVALF(IBIFN,OUTFAC\_";IBA(355.93,",IBINS,IBFRMTYP,IBCARE,.IDS,IBSORT1,IBSORT2,COB,IBLIMIT,IBSTRIP,SEG)  . ;  . I OUTFAC="" D  .. ;  .. ; MRD;IB\*2.0\*516 - Due to fields being marked for deletion, the  .. ; function $$SENDSF^IBCEF79 will always return '1'. Refer to  .. ; that function and INSFLGS^^IBCEF79 for more information.  .. ;  .. ; if ins co flag says to not send svc fac data and we're sending an EDI claim, then get out  .. ;I '$$SENDSF^IBCEF79(IBIFN,COB),$G(^TMP("IBTX",$J,IBIFN)) Q  .. ;  .. ;IB\*2.0\*432/TAZ Moved Taxid setup inside VALF look to send as secondary ID for Medicare claims.  .. ;S IDS("LAB/FAC",IBIFN,IBSORT1,IBSORT2,0)=$$STRIP($$TAXID^IBCEF75(),1,U,IBSTRIP)  .. D VALF(IBIFN,IBINS,IBFRMTYP,IBDIV,.IDS,IBSORT1,IBSORT2,COB,IBLIMIT,IBSTRIP,SEG)  Q  ; VALF(IBIFN,INS,FT,DIV,IDS,SORT1,SORT2,COB,IBLIMIT,IBSTRIP,SEG) ; Get VA Lab/Fac Secondary IDs  ; Pass in INS - IEN to file 36  ; FT - 1 = UB 2 = 1500 4 = J430D  ; DIV - PTR to 40.8  ;  N Z,Z0,ID,QUAL,MAIN,IDTBL,CNT,Z,IBMCR  S MAIN=$$MAIN^IBCEP2B() ; get the IEN for main Division  S Z=0 F  S Z=$O(^IBA(355.92,"B",INS,Z)) Q:'Z  D  . S Z0=$G(^IBA(355.92,Z,0))  . Q:$P(Z0,U,8)'="LF"   ; Screen out anything other than Lab or Facility  . I +$P(Z0,U,4) Q:$P(Z0,U,4)'=FT   ; Form type must match that passed in or be a 0 which allows both  . S ID=$$STRIP($P(Z0,U,7),1,,IBSTRIP)  . S QUAL=$$STRIP($P(Z0,U,6),1,,IBSTRIP)  . Q:QUAL=""   ; Needs a qualifier  . S QUAL=$P($G(^IBE(355.97,QUAL,0)),U,3)  . I FT=1,SORT1="O" Q:$$OP3^IBCEF73(FT)'[(U\_QUAL\_U) ; Institutional  . I FT=2,SORT1="O" Q:$$OP7^IBCEF73(FT)'[(U\_QUAL\_U) ; Professional  **. ;JWS;IB\*2.0\*592;Dental form**  **. I FT=4,SORT1="O" Q:$$OP7^IBCEF73(FT)'[(U\_QUAL\_U) ; Professional (Dental)**  . I $P(Z0,U,5)=""!($P(Z0,U,5)=0)!($P(Z0,U,5)=MAIN) S IDTBL("DEF",QUAL)=ID  ; set up default for main division  . I $P(Z0,U,5)=DIV S IDTBL("DIV",QUAL)=ID  ; set up default for division  S CNT=0  S IDS("LAB/FAC",IBIFN,SORT1,SORT2)=$E("PST",COB)  ;IB\*2.0\*432/TAZ If Medicare send Tax ID as 1st Secondary ID ; only if it's not a printed form  S IBMCR=""  **;JWS;IB\*2.0\*592;Dental**  I '(($G(IBXFORM)=2)!($G(IBXFORM)=3**)!($G(IBXFORM)=7))** S IBMCR=$$MCRONBIL^IBEFUNC(IBIFN)  I IBMCR S CNT=CNT+1,IDS("LAB/FAC",IBIFN,SORT1,SORT2,CNT)="LU"\_U\_$$STRIP($P($$TAXID^IBCEF75(),U,2),1,U,IBSTRIP)  I $D(IDTBL("DIV")) D  Q  . S Z="" F  S Z=$O(IDTBL("DIV",Z)) Q:Z=""  D  .. ;IB\*2.0\*432/TAZ If Medicare, screen out Tax ID  .. I IBMCR,(Z=24) Q  .. S CNT=CNT+1,IDS("LAB/FAC",IBIFN,SORT1,SORT2,CNT)=Z\_U\_IDTBL("DIV",Z) Q:CNT=IBLIMIT  I $D(IDTBL("DEF")) D  Q  . S Z="" F  S Z=$O(IDTBL("DEF",Z)) Q:Z=""  D  .. ;IB\*2.0\*432/TAZ If Medicare, screen out Tax ID  .. I IBMCR,(Z=24) Q  .. S CNT=CNT+1,IDS("LAB/FAC",IBIFN,SORT1,SORT2,CNT)=Z\_U\_IDTBL("DEF",Z) Q:CNT=IBLIMIT  Q  ; NONVALF(IBIFN,PRV,INS,FT,PT,IDS,SORT1,SORT2,COB,IBLIMIT,IBSTRIP,SEG) ; Get Non VA Lab/Fac Secondary IDs  ; Pass in PRV - VPTR - PTR to 355.93 (in format of variabel pointer IEN;IBA(355.93,  ; Pass in INS - PTR to 36 of null (not provide by insurance company)  ; FT - 1 = UB 2 = 1500 **4 = J430D**  ; PT - Patient Type - 1 inpatient 2 outpatient  ; IDS array being returned  ; SORT1 - "C"urrent or "O"ther  ; SORT2 - 1 if current or (1 or 2 if other)  N Z,Z0,ID,QUAL,IDTBL,CNT,IBMCR  S Z=0 F  S Z=$O(^IBA(355.9,"B",PRV,Z)) Q:'Z  D  . S Z0=$G(^IBA(355.9,Z,0))  . I +$P(Z0,U,4) Q:$P(Z0,U,4)'=FT   ; Form type must match that passed in or be a 0 which allows both UB and 1500  . I +$P(Z0,U,5) Q:$P(Z0,U,5)'=PT   ; Patient type must match that passed in or be a 0 which allows both in patient and outpatient  . I INS]"",$P(Z0,U,2)]"",INS'=$P(Z0,U,2) Q  . S ID=$$STRIP($P(Z0,U,7),1,,IBSTRIP)  . Q:ID=""  . S QUAL=$$STRIP($P(Z0,U,6),1,,IBSTRIP)  . Q:QUAL=""   ; Needs a qualifier  . S QUAL=$P($G(^IBE(355.97,QUAL,0)),U,3)  . Q:QUAL=""  . I FT=1,SORT1="O" Q:$$OP3^IBCEF73(FT)'[(U\_QUAL\_U) ; Institutional  . I FT=2,SORT1="O" Q:$$OP7^IBCEF73(FT)'[(U\_QUAL\_U) ; Professional  . **;JWS;IB\*2.0\*592;Dental - professional**  **. I FT=4,SORT1="O" Q:$$OP7^IBCEF73(FT)'[(U\_QUAL\_U) ; Professional - Dental**  . I $G(SEG)="SUB1" Q:$$SUB1^IBCEF73(FT)'[(U\_QUAL\_U)  . I $P(Z0,U,2)="" S IDTBL("OWN",QUAL)=ID  ; set up default of lab or facilities own ids  . I $P(Z0,U,2)=INS S IDTBL("INS",QUAL)=ID  ; set up default for division  ;  S CNT=0  S IDS("LAB/FAC",IBIFN,SORT1,SORT2)=$E("PST",COB)\_U\_PRV  S IDS("LAB/FAC",IBIFN,SORT1,SORT2,"CONTACT")=$G(^IBA(355.93,+PRV,1))  ; get primary  S Z0=$G(^IBA(355.93,+PRV,0))  ;IB\*2.0\*432/TAZ If Medicare send Tax ID as 1st Secondary ID  S IBMCR=""  **;JWS;IB\*2.0\*592;Dental**  I '(($G(IBXFORM)=2)!($G(IBXFORM)=3**)!($G(IBXFORM)=7))** S IBMCR=$$MCRONBIL^IBEFUNC(IBIFN)  ;I $P(Z0,U,9)]"",$P(Z0,U,13)]"",IBMCR S CNT=CNT+1,IDS("LAB/FAC",IBIFN,SORT1,SORT2,CNT)="LU"\_U\_$$STRIP($P($G(^IBE(355.97,$P(Z0,U,13),0)),U,3)\_U\_$P(Z0,U,9),1,U,IBSTRIP)  I $P(Z0,U,9)]"",$P(Z0,U,13)]"",IBMCR S CNT=CNT+1,IDS("LAB/FAC",IBIFN,SORT1,SORT2,CNT)="LU"\_U\_$$STRIP($P(Z0,U,9),1,U,IBSTRIP)  ; get secondarys in order  I $D(IDTBL("INS")) D  . N Z S Z="" F  S Z=$O(IDTBL("INS",Z)) Q:Z=""  D  .. ;IB\*2.0\*432/TAZ If Medicare, screen out Tax ID  .. I IBMCR,(Z=24) Q  .. S CNT=CNT+1,IDS("LAB/FAC",IBIFN,SORT1,SORT2,CNT)=Z\_U\_IDTBL("INS",Z) Q:CNT=IBLIMIT  I $D(IDTBL("OWN")),CNT'=IBLIMIT D  . N Z S Z="" F  S Z=$O(IDTBL("OWN",Z)) Q:Z=""  D  .. ;IB\*2.0\*432/TAZ If Medicare, screen out Tax ID  .. I IBMCR,(Z=24) Q  .. I '$D(IDTBL("INS",Z)) S CNT=CNT+1,IDS("LAB/FAC",IBIFN,SORT1,SORT2,CNT)=Z\_U\_IDTBL("OWN",Z) Q:CNT=IBLIMIT  Q  ; STRIP(X,SPACE,EXC,IBSTRIP) ;  ; Strip punctuation from data in X  ; SPACE = flag if 1 strip SPACES  ; EXC = list of punct not to strip  ;   Q:'$G(IBSTRIP) X  Q $$NOPUNCT^IBCEF(X,$G(SPACE),$G(EXC))  ; OTH(IBIFN,IBXSAVE,IBXDATA,COND,SEG) ; Procedure used in piece 2 of some output  ; formatter segments for other insurance  ; COND = 0/1 value passed in that determines whether or not to call the  ; provider ID function  ; SEG = name of segment for use in calling ID^IBCEF2 (4 characters)  ;  N Z  ;\*432/TAZ - Changed Clean up and Setup routines to IBCEFP\*  ;D CLEANUP^IBCEF75(.IBXSAVE)  ;I COND D ALLIDS^IBCEF75(IBIFN,.IBXSAVE,1)  D CLEANUP^IBCEFP1(.IBXSAVE)  I COND D ALLIDS^IBCEFP(IBIFN,.IBXSAVE,1)  ;  ; Special Check: if Other Insurance #2 has secondary ID's while Other  ; Insurance #1 does not, then move up #2 to be #1 here. This is to  ; ensure the output formatter IBXDATA array is built properly.  ;  I $O(IBXSAVE("LAB/FAC",IBIFN,"O",2,0)),'$O(IBXSAVE("LAB/FAC",IBIFN,"O",1,0)) D  . K IBXSAVE("LAB/FAC",IBIFN,"O",1)  . M IBXSAVE("LAB/FAC",IBIFN,"O",1)=IBXSAVE("LAB/FAC",IBIFN,"O",2)  . K IBXSAVE("LAB/FAC",IBIFN,"O",2)  . Q  ;  K IBXDATA  S Z=0  F  S Z=$O(IBXSAVE("LAB/FAC",IBIFN,"O",Z)) Q:'Z  D  . I '$O(IBXSAVE("LAB/FAC",IBIFN,"O",Z,0)) Q  . S IBXDATA(Z)=$P($G(IBXSAVE("LAB/FAC",IBIFN,"O",Z)),U,1)  . I Z>1 D ID^IBCEF2(Z,SEG)  . Q OTHX ;  Q  ; | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCEF77 | | | | | | | | |
| **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** | | | | | | | | | |
| IBCEF77 ;WOIFO/SS - FORMATTER/EXTRACT BILL FUNCTIONS ;31-JUL-03  ;;2.0;INTEGRATED BILLING;\*\*232,280,155,290,291,320,348,349,516\*\*;21-MAR-94;Build 123  ;;Per VA Directive 6402, this routine should not be modified.  ; SORT(IBPRNUM,IBPRTYP,IB399,IBSRC,IBDST,IBN,IBEXC,IBSEQ,IBLIMIT) ;  N IBXIEN,IBXDATA,IBNET,IBTRI,IB1,IB2,IBID,Z,IBZ,IBZ1,IBSVP  S (IB1,IB2,IBZ,IBZ1,IBTRI)=""  D F^IBCEF("N-ALL ATT/RENDERING PROV SSN","IBZ",,IB399)  S IBZ1=$$ALLPTYP^IBCEF3(IB399)  F Z=1:1:3 S $P(IBZ1,U,Z)=$S($P(IBZ1,U,Z)="CH":1,1:"") S:$P(IBZ1,U,Z) IBTRI=1  S IBNET=$$NETID^IBCEP() ; netwrk id type  I $G(IBN) D  . S Z=0 F  S Z=$O(IBDST(IBPRNUM,IBPRTYP,Z)) Q:'Z  S IBID(+$P(IBDST(IBPRNUM,IBPRTYP,Z),U,9))=""  F  S IB1=$O(IBSRC(IB1)) Q:IB1=""  D  Q:IBN=IBLIMIT  . N OK,IBSTLIC  . S IBSTLIC=""  . F  S IB2=$O(IBSRC(IB1,IB2)) Q:IB2=""  D  Q:IBN=IBLIMIT  . . S IBSVP=$P(IBSRC(IB1,IB2),U)  . . ; If ID overridden, output no others of this type  . . I $G(IBEXC),$P($G(IBSRC(IB1,IB2)),U,9)=IBEXC Q  . . ; Ck state of care/lic match if st lic#  . . I $P($G(IBSRC(IB1,IB2)),U,3)="0B" S OK=1 D  Q:'OK  . . . I +$$CAREST^IBCEP2A(IB399)'=$P(IBSRC(IB1,IB2),U,7) S IBSTLIC=1 Q  . . . I $G(IBSTLIC(0))'="" S OK=0 Q  . . . S IBSTLIC(0)=$G(IBSRC(IB1,IB2)),OK=0  . . ; Exclude SSN from sec ids unless required  . . I $P($G(IBSRC(IB1,IB2)),U,3)="SY" Q  . . ; Only 1 of each prov id type  . . Q:$D(IBID(+$P($G(IBSRC(IB1,IB2)),U,9)))  . . S IBN=IBN+1,IBID(+$P($G(IBSRC(IB1,IB2)),U,9))=""  . . S IBDST(IBPRNUM,IBPRTYP,IBN)=$G(IBSRC(IB1,IB2))  . I IBN'=IBLIMIT,'$G(IBSTLIC),$G(IBSTLIC(0))'="" S IBN=IBN+1,IBDST(IBPRNUM,IBPRTYP,IBN)=IBSTLIC(0)  I $$FT^IBCEF(IB399)=2,$G(IBID(IBNET))="",IBTRI,$P(IBZ1,U,IBSEQ) D    ; WCJ 02/13/2006  . Q:$P(IBZ,U,IBPRTYP)=""  . ; here, no network id & TRICARE ins co.  . N Z  . S Z=+$O(^DGCR(399,IB399,"PRV","B",IBPRTYP,0)),Z=$P($G(^DGCR(399,IB399,"PRV",Z,0)),U,2)  . S IBN=IBN+1,IBDST(IBPRNUM,IBPRTYP,IBN)=Z\_U\_+$$POLICY^IBCEF(IB399,1,IBSEQ)\_U\_$P($G(^IBE(355.97,IBNET,0)),U,3)\_U\_$P(IBZ,U,IBPRTYP)\_U\_"0^0^^^"\_IBNET  Q  ;  ; esg - 8/25/06 - IB\*2\*348 - CFIDS function  ; CFIDS(IBIFN,PRVTYP,ALLOWIDS) ; Claim Form IDs for human providers  ; Function returns a 3 piece string: [1] default secondary ID qual  ; [2] default secondary ID  ; [3] NPI  ; Input: IBIFN - internal claim#  ; PRVTYP - internal provider type ID number  ; - 1:REFER;2:OPER;3:REND;4:ATT;5:SUPER;9:OTHER  ; - if blank, then default Att/Rend based on form type  ; ALLOWIDS - List of allowable Secondary IDS ^ delimited.   ; ex "^1A^1B^1C^1H^G2^LU^N5^"  ; UB-04 only wants IDs provided by the payer, not the providers own IDS  ; Also, they want the qualifier to be G2 (Commercial)  ; if it is a payer provided ID  NEW ID,FT,IBZ,IBQ,IBSID,IBNPI,I,OK  S ID=""  I '$G(IBIFN) G CFIDSX  S FT=$$FT^IBCEF(IBIFN)  I '$G(PRVTYP) S PRVTYP=3 I FT=3 S PRVTYP=4  D ALLIDS^IBCEF75(IBIFN,.IBZ,1)  S OK=0 I $G(ALLOWIDS)="" S OK=1  F I=1:1 D  Q:OK  . S IBQ=$P($G(IBZ("PROVINF",IBIFN,"C",1,PRVTYP,I)),U,3) ; qualifier  . S IBSID=$P($G(IBZ("PROVINF",IBIFN,"C",1,PRVTYP,I)),U,4) ; ID#  . I IBQ="",IBSID="" S OK=1 Q  . Q:OK  . I $G(ALLOWIDS)[(U\_IBQ\_U) S OK=1,IBQ="G2" Q  . S (IBQ,IBSID)=""  S IBNPI=""  D F^IBCEF("N-PROVIDER NPI CODES","IBNPI",,IBIFN)  S IBNPI=$P(IBNPI,U,PRVTYP) ; NPI  ;  ; special check for the referring doc  I PRVTYP=1,$D(IBZ("PROVINF",IBIFN,"C",1,PRVTYP)),IBQ="",IBSID="" S IBQ="1G",IBSID="VAD000"  ;  ; If UB-04 and no IDs, use VA UPIN as deafult  I $D(IBZ("PROVINF",IBIFN,"C",1,PRVTYP)),FT=3,IBQ="",IBSID="" S IBQ="1G",IBSID="VAD000"  ;  ; determine if legacy ID's should be displayed  I '$$PRTLID(IBIFN,IBNPI) S (IBQ,IBSID)=""  ;  S ID=IBQ\_U\_IBSID\_U\_IBNPI CFIDSX ;  Q ID  ; DOL(AMT,LEN,DEC) ; format dollar amounts for printed claim forms  ; AMT = amount to be formatted  ; LEN = length of field - right justified to this length  ; DEC = flag to include the decimal point or not  ; DEFAULT value is to not include the decimal point  ; if DEC is not defined or 0, assume no decimal point  ; so 15 will be returned as 1500, 6.77 will be returned as 677  ; if DEC is 1, then the decimal point will be included  ;  S LEN=$G(LEN,10),DEC=$G(DEC,0) ; defaults  S AMT=$FN(+$G(AMT),"",2) ; format # with 2 decimals  I 'DEC S AMT=$TR(AMT,".") ; strip or leave decimal  S AMT=$J(AMT,LEN) ; right justify  Q AMT  ; PRTLID(IBIFN,NPI) ; YMG; Print Legacy IDs on the CMS-1500 or UB-04 form  ; Function fetches form type associated with given claim number  ; (values: 2 - CMS-1500 form, 3 - UB-04 form), then looks at  ; "Print Legacy ID" site parameter for this particular form type.  ;   ; Possible site parameter values are:  ; "Y" - always print Legacy ID  ; "N" - never print Legacy ID  ; "C" - only print Legacy ID if NPI is not available.  ;   ; This information is used to determine if Legacy ID should be printed  ; for claim number in question.  ;   ; Note: Situation when "Print Legacy ID" site parameter is not set is treated  ; as if this parameter was set to "Y" - always print Legacy ID.  ;   ; Input:  ; IBIFN - internal claim number  ; NPI - NPI number (or "" if no NPI is available)  ;   ; Returns:  ; 0 - Legacy ID should not be printed  ; 1 - Legacy ID should be printed  ;  Q $S(NPI="":"YC",1:"Y")[$P($G(^IBE(350.9,1,1)),U,$S($$FT^IBCEF(IBIFN)=2:32,1:33))  ; REMARK(IBIFN,IBXDATA,OFLG) ; procedure to return array of UB-04 remark text  ; for claim IBIFN. Data pulled from field# 402 of file 399 and  ; formatted into an array IBXDATA(n) where each line is not greater  ; than 24 characters long. This will fit into UB-04 FL-80.  ;  ; OFLG=1 only when called in the output formatter. In this case, only  ; 4 lines in IBXDATA will be returned.  ;  NEW TEXT,LEN,IBZ,J,PCE,CHS,NEWCHS,IBK,J,TX,IBCP1  K IBXDATA  ;  ; MRD;IB\*2.0\*516 - Pull the Bill Remarks for the claim. If this was  ; called from the Output Formatter, then look at lines of claim for  ; NDC's. If any are found, they should be added to the end of TEXT.  ;  S TEXT=$P($G(^DGCR(399,+$G(IBIFN),"UF2")),U,3)  I $G(OFLG) D  . S J=0  . F  S J=$O(^DGCR(399,+$G(IBIFN),"CP",J)) Q:'J  S IBCP1=$G(^(J,1)) I $P(IBCP1,U,7)'="" D  . . I TEXT'="" S TEXT=TEXT\_" "  . . S TEXT=TEXT\_"N4"\_$TR($P(IBCP1,U,7),"-")\_" UN"\_$P(IBCP1,U,8)  . . Q  . Q  ;  ; If there's nothing in TEXT, then Quit.  ;  I TEXT="" Q  ;  ; need to break up large words for word wrapping purposes to get  ; as many characters as possible in the box.  S LEN=17  F PCE=1:1 Q:PCE>$L(TEXT," ") S CHS=$P(TEXT," ",PCE) I $L(CHS)>LEN D  . S NEWCHS=$E(CHS,1,LEN)\_" "\_$E(CHS,LEN+1,999)  . S $P(TEXT," ",PCE)=NEWCHS  . Q  ;  ; When calling FSTRNG^IBJU1 which calls ^DIWP, FileMan builds the  ; array with strings of max length=1 less than what you tell it.  ;  S LEN=20 ; line 1 is 19 chars  D FSTRNG^IBJU1(TEXT,LEN,.IBZ) ; build IBZ array  S IBK=$$TRIM^XLFSTR($G(IBZ(1))) ; save off the first line  S TEXT=$P(TEXT,IBK,2,99) ; restore the rest of the text  S TEXT=$$TRIM^XLFSTR(TEXT) ; trim spaces  ;  S LEN=25 ; the rest is 24 chars  D FSTRNG^IBJU1(TEXT,LEN,.IBZ) ; build IBZ array  S IBXDATA(1)=" "\_IBK             ; line 1  S J=0 F  S J=$O(IBZ(J)) Q:'J  D      ; lines 2-n  . I J>3,$G(OFLG) Q                   ; only 4 lines for output formatter  . S TX=$$TRIM^XLFSTR($G(IBZ(J)))  . I TX'="" S IBXDATA(J+1)=TX  . Q  Q  ; | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCEF77 ;WOIFO/SS - FORMATTER/EXTRACT BILL FUNCTIONS ;31-JUL-03  ;;2.0;INTEGRATED BILLING;\*\*232,280,155,290,291,320,348,349,516,577,**592**\*\*;21-MAR-94;Build 1  ;;Per VA Directive 6402, this routine should not be modified.  ; SORT(IBPRNUM,IBPRTYP,IB399,IBSRC,IBDST,IBN,IBEXC,IBSEQ,IBLIMIT) ;  N IBXIEN,IBXDATA,IBNET,IBTRI,IB1,IB2,IBID,Z,IBZ,IBZ1,IBSVP  S (IB1,IB2,IBZ,IBZ1,IBTRI)=""  D F^IBCEF("N-ALL ATT/RENDERING PROV SSN","IBZ",,IB399)  S IBZ1=$$ALLPTYP^IBCEF3(IB399)  F Z=1:1:3 S $P(IBZ1,U,Z)=$S($P(IBZ1,U,Z)="CH":1,1:"") S:$P(IBZ1,U,Z) IBTRI=1  S IBNET=$$NETID^IBCEP() ; netwrk id type  I $G(IBN) D  . S Z=0 F  S Z=$O(IBDST(IBPRNUM,IBPRTYP,Z)) Q:'Z  S IBID(+$P(IBDST(IBPRNUM,IBPRTYP,Z),U,9))=""  F  S IB1=$O(IBSRC(IB1)) Q:IB1=""  D  Q:IBN=IBLIMIT  . N OK,IBSTLIC  . S IBSTLIC=""  . F  S IB2=$O(IBSRC(IB1,IB2)) Q:IB2=""  D  Q:IBN=IBLIMIT  . . S IBSVP=$P(IBSRC(IB1,IB2),U)  . . ; If ID overridden, output no others of this type  . . I $G(IBEXC),$P($G(IBSRC(IB1,IB2)),U,9)=IBEXC Q  . . ; Ck state of care/lic match if st lic#  . . I $P($G(IBSRC(IB1,IB2)),U,3)="0B" S OK=1 D  Q:'OK  . . . I +$$CAREST^IBCEP2A(IB399)'=$P(IBSRC(IB1,IB2),U,7) S IBSTLIC=1 Q  . . . I $G(IBSTLIC(0))'="" S OK=0 Q  . . . S IBSTLIC(0)=$G(IBSRC(IB1,IB2)),OK=0  . . ; Exclude SSN from sec ids unless required  . . I $P($G(IBSRC(IB1,IB2)),U,3)="SY" Q  . . ; Only 1 of each prov id type  . . Q:$D(IBID(+$P($G(IBSRC(IB1,IB2)),U,9)))  . . S IBN=IBN+1,IBID(+$P($G(IBSRC(IB1,IB2)),U,9))=""  . . S IBDST(IBPRNUM,IBPRTYP,IBN)=$G(IBSRC(IB1,IB2))  . I IBN'=IBLIMIT,'$G(IBSTLIC),$G(IBSTLIC(0))'="" S IBN=IBN+1,IBDST(IBPRNUM,IBPRTYP,IBN)=IBSTLIC(0)  **;JRA IB\*2.0\*592 Treat Dental Form 7 (J430D) same as CMS-1500 - added 'FT'**  **;I** $$FT^IBCEF(IB399)=2,$G(IBID(IBNET))="",IBTRI,$P(IBZ1,U,IBSEQ) D ; WCJ 02/13/2006 **;JRA IB\*2.0\*592 ';'**  **N FT S FT=$$FT^IBCEF(IB399) ;JRA IB\*2.0\*592**  I **(FT=2!(FT=7)**),$G(IBID(IBNET))="",IBTRI,$P(IBZ1,U,IBSEQ) D  **;JRA IB\*2.0\*592**  . Q:$P(IBZ,U,IBPRTYP)=""  . ; here, no network id & TRICARE ins co.  . N Z  . S Z=+$O(^DGCR(399,IB399,"PRV","B",IBPRTYP,0)),Z=$P($G(^DGCR(399,IB399,"PRV",Z,0)),U,2)  . S IBN=IBN+1,IBDST(IBPRNUM,IBPRTYP,IBN)=Z\_U\_+$$POLICY^IBCEF(IB399,1,IBSEQ)\_U\_$P($G(^IBE(355.97,IBNET,0)),U,3)\_U\_$P(IBZ,U,IBPRTYP)\_U\_"0^0^^^"\_IBNET  Q  ;  ; esg - 8/25/06 - IB\*2\*348 - CFIDS function  ; CFIDS(IBIFN,PRVTYP,ALLOWIDS) ; Claim Form IDs for human providers  ; Function returns a 3 piece string: [1] default secondary ID qual  ; [2] default secondary ID  ; [3] NPI  ; Input: IBIFN - internal claim#  ; PRVTYP - internal provider type ID number  ; - 1:REFER;2:OPER;3:REND;4:ATT;5:SUPER;9:OTHER  ; - if blank, then default Att/Rend based on form type  ; ALLOWIDS - List of allowable Secondary IDS ^ delimited.   ; ex "^1A^1B^1C^1H^G2^LU^N5^"  ; UB-04 only wants IDs provided by the payer, not the providers own IDS  ; Also, they want the qualifier to be G2 (Commercial)  ; if it is a payer provided ID  NEW ID,FT,IBZ,IBQ,IBSID,IBNPI,I,OK  S ID=""  I '$G(IBIFN) G CFIDSX  S FT=$$FT^IBCEF(IBIFN)  I '$G(PRVTYP) S PRVTYP=3 I FT=3 S PRVTYP=4  D ALLIDS^IBCEF75(IBIFN,.IBZ,1)  S OK=0 I $G(ALLOWIDS)="" S OK=1  F I=1:1 D  Q:OK  . S IBQ=$P($G(IBZ("PROVINF",IBIFN,"C",1,PRVTYP,I)),U,3) ; qualifier  . S IBSID=$P($G(IBZ("PROVINF",IBIFN,"C",1,PRVTYP,I)),U,4) ; ID#  . I IBQ="",IBSID="" S OK=1 Q  . Q:OK  . I $G(ALLOWIDS)[(U\_IBQ\_U) S OK=1,IBQ="G2" Q  . S (IBQ,IBSID)=""  S IBNPI=""  D F^IBCEF("N-PROVIDER NPI CODES","IBNPI",,IBIFN)  S IBNPI=$P(IBNPI,U,PRVTYP) ; NPI  ;  ; special check for the referring doc  I PRVTYP=1,$D(IBZ("PROVINF",IBIFN,"C",1,PRVTYP)),IBQ="",IBSID="" S IBQ="1G",IBSID="VAD000"  ;  ; If UB-04 and no IDs, use VA UPIN as deafult  I $D(IBZ("PROVINF",IBIFN,"C",1,PRVTYP)),FT=3,IBQ="",IBSID="" S IBQ="1G",IBSID="VAD000"  ;  ; determine if legacy ID's should be displayed  I '$$PRTLID(IBIFN,IBNPI) S (IBQ,IBSID)=""  ;  S ID=IBQ\_U\_IBSID\_U\_IBNPI CFIDSX ;  Q ID  ; DOL(AMT,LEN,DEC) ; format dollar amounts for printed claim forms  ; AMT = amount to be formatted  ; LEN = length of field - right justified to this length  ; DEC = flag to include the decimal point or not  ; DEFAULT value is to not include the decimal point  ; if DEC is not defined or 0, assume no decimal point  ; so 15 will be returned as 1500, 6.77 will be returned as 677  ; if DEC is 1, then the decimal point will be included  ;  S LEN=$G(LEN,10),DEC=$G(DEC,0) ; defaults  S AMT=$FN(+$G(AMT),"",2) ; format # with 2 decimals  I 'DEC S AMT=$TR(AMT,".") ; strip or leave decimal  S AMT=$J(AMT,LEN) ; right justify  Q AMT  ; PRTLID(IBIFN,NPI) ; YMG; Print Legacy IDs on the CMS-1500 or UB-04 form  ; Function fetches form type associated with given claim number  ; (values: 2 - CMS-1500 form, 3 - UB-04 form), then looks at  ; "Print Legacy ID" site parameter for this particular form type.  ;   ; Possible site parameter values are:  ; "Y" - always print Legacy ID  ; "N" - never print Legacy ID  ; "C" - only print Legacy ID if NPI is not available.  ;   ; This information is used to determine if Legacy ID should be printed  ; for claim number in question.  ;   ; Note: Situation when "Print Legacy ID" site parameter is not set is treated  ; as if this parameter was set to "Y" - always print Legacy ID.  ;   ; Input:  ; IBIFN - internal claim number  ; NPI - NPI number (or "" if no NPI is available)  ;   ; Returns:  ; 0 - Legacy ID should not be printed  ; 1 - Legacy ID should be printed  ; **;JRA IB\*2.0\*592 Treat Dental Form 7 (J430D) same as CMS-1500 - added 'FT'  ;**Q $S(NPI="":"YC",1:"Y")[$P($G(^IBE(350.9,1,1)),U,$S($$FT^IBCEF(IBIFN)=2:32,1:33**)) ;JRA IB\*2.0\*592 ';'**  **N FT S FT=$$FT^IBCEF(IBIFN) ;JRA IB\*2.0\*592**  Q $S(NPI="":"YC",1:"Y")[$P($G(^IBE(350.9,1,1)),U,$S((**FT=2!(FT=7)):**32,1:33)) **;JRA IB\*2.0\*592**  ; REMARK(IBIFN,IBXDATA,OFLG) ; procedure to return array of UB-04 remark text  ; for claim IBIFN. Data pulled from field# 402 of file 399 and  ; formatted into an array IBXDATA(n) where each line is not greater  ; than 24 characters long. This will fit into UB-04 FL-80.  ;  ; OFLG=1 only when called in the output formatter. In this case, only  ; 4 lines in IBXDATA will be returned.  ;  NEW TEXT,LEN,IBZ,J,PCE,CHS,NEWCHS,IBK,J,TX,IBCP1  K IBXDATA  ;  ; MRD;IB\*2.0\*516 - Pull the Bill Remarks for the claim. If this was  ; called from the Output Formatter, then look at lines of claim for  ; NDC's. If any are found, they should be added to the end of TEXT.  ;  S TEXT=$P($G(^DGCR(399,+$G(IBIFN),"UF2")),U,3)  ; VAD/ Begin of IB\*2\*577 changes  ; NDC, Quantity, and Unit of Measure now printed in FL-43  ; instead of here in FL-80  ;I $G(OFLG) D  ;. S J=0  ;. F S J=$O(^DGCR(399,+$G(IBIFN),"CP",J)) Q:'J S IBCP1=$G(^(J,1)) I $P(IBCP1,U,7)'="" D  ;. . I TEXT'="" S TEXT=TEXT\_" "  ;. . S TEXT=TEXT\_"N4"\_$TR($P(IBCP1,U,7),"-")\_" UN"\_$P(IBCP1,U,8)  ;. . Q  ;. Q  ; VAD/ End of IB\*2\*577 changes  ;  ; If there's nothing in TEXT, then Quit.  ;  I TEXT="" Q  ;  ; need to break up large words for word wrapping purposes to get  ; as many characters as possible in the box.  S LEN=17  F PCE=1:1 Q:PCE>$L(TEXT," ") S CHS=$P(TEXT," ",PCE) I $L(CHS)>LEN D  . S NEWCHS=$E(CHS,1,LEN)\_" "\_$E(CHS,LEN+1,999)  . S $P(TEXT," ",PCE)=NEWCHS  . Q  ;  ; When calling FSTRNG^IBJU1 which calls ^DIWP, FileMan builds the  ; array with strings of max length=1 less than what you tell it.  ;  S LEN=20 ; line 1 is 19 chars  D FSTRNG^IBJU1(TEXT,LEN,.IBZ) ; build IBZ array  S IBK=$$TRIM^XLFSTR($G(IBZ(1))) ; save off the first line  S TEXT=$P(TEXT,IBK,2,99) ; restore the rest of the text  S TEXT=$$TRIM^XLFSTR(TEXT) ; trim spaces  ;  S LEN=25 ; the rest is 24 chars  D FSTRNG^IBJU1(TEXT,LEN,.IBZ) ; build IBZ array  S IBXDATA(1)=" "\_IBK             ; line 1  S J=0 F  S J=$O(IBZ(J)) Q:'J  D      ; lines 2-n  . I J>3,$G(OFLG) Q                   ; only 4 lines for output formatter  . S TX=$$TRIM^XLFSTR($G(IBZ(J)))  . I TX'="" S IBXDATA(J+1)=TX  . Q  Q  ; B43(NDCDATA) ; This is passed a string and properly formats if there is NDC drug information.  ; The drug information is in pieces 21-23 of that string.  ; It was part of the output formatter entry 364.7[1406] used for FL43 but that got too big for a FileMan Mumps data element  ; It returns a string with N4 - the NDC Drug qualifier  ; NDC Code without the hyphens  ; a space  ; Units qualifier  ; Units  ; Ex "N412345678901 ML1.5"  I NDCDATA="" Q ""  S NDCDATA=$P(NDCDATA,U,21,23)  Q:$P(NDCDATA,U)="" ""  Q "N4"\_$TR($P(NDCDATA,U),"-")\_" "\_$TR($P(NDCDATA,U,2,3),U)  ; | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCEF78 | | | | | | | | |
| **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** | | | | | | | | | |
| IBCEF78 ;ALB/WCJ - Provider ID functions ;13 May 2007  ;;2.0;INTEGRATED BILLING;\*\*371,516\*\*;21-MAR-94;Build 123  ;;Per VA Directive 6402, this routine should not be modified.  ;;  G AWAY AWAY Q  ; PAYERIDS(IBXIEN,IBRET) ; This function returns all the PAYER IDS for the current and other insurance(s)  ;   D PRIPAYID(IBXIEN,.IBRET)  D SECPAYID(IBXIEN,.IBRET)  Q  ; PRIPAYID(IBXIEN,IBXRET) ; Primary Payer IDs  ; Incoming:  ; IBXIEN = IEN for File # 399  ; IBXRET = Return Array for Qualifiers and IDs  ;  ; Outgoing  ; IBXRET("CI\_PID",1)=QUAL^ID  ; IBXRET("OI\_PID",#)=QUAL^ID  ;   N RET,I  S RET=$$PAYERID^IBCEF2(IBXIEN)  I RET]"" S IBXRET("CI\_PID",1)="PI"\_U\_RET  ;  ; MRD;IB\*2.0\*516 - Added HPID here (CI) and below (OI).  S RET=$$HPID(IBXIEN)  I RET]"" S IBXRET("CI\_HPID",1)="XV"\_U\_RET  ;  S RET=""  D OTHINSID^IBCEF72(IBXIEN,.RET)  F I=1,2 D  . I $P($G(RET(I)),U)]"" S IBXRET("OI\_PID",I)="PI"\_U\_$P(RET(I),U)  . I $P($G(RET(I)),U,2)]"" S IBXRET("OI\_HPID",I)="XV"\_U\_$P(RET(I),U,2)  . Q  Q  ; SECPAYID(IBXIEN,IBXRET) ; This returns all of the secondary payer IDs from file #36   ; for the insurance companies on a given claim  ;   ; Incoming:  ; IBXIEN = IEN for File # 399  ; IBXRET = Return Array for Qualifiers and IDs  ;  ; Outgoing  ; IBXRET("CI\_PSIDS",1)=QUAL^ID^QUAL^ID  ; IBXRET("OI\_PSIDS",#)=QUAL^ID^QUAL^ID  ;  N Z,C,IBZ,Z0,FT  F Z=1:1:3 S IBZ(Z)=$$POLICY^IBCEF(IBXIEN,1,Z)  S Z0=0,C=$$COBN^IBCEF(IBXIEN),FT=$$FT^IBCEF(IBXIEN)  F Z=1:1:3 S:C'=Z Z0=Z0+1 S IBXRET($S(C=Z:"CI\_PSIDS",1:"OI\_PSIDS"),$S(C=Z:1,1:Z0))=$$SPIDS(+IBZ(Z),FT)  Q  ; SPIDS(INS,FT) ;  ; FT = FORM TYPE (2 PROFESSIONAL 3 INSTITUTIONAL)  ; INS = INSURANCE COMPANY (FILE #36) IEN  ; Returns String (^ delimited)  ; [1] = QUAL 1  ; [2] = PAYER ID 1  ; [3] = QUAL 2  ; [4] = PAYER ID 2  Q:'+INS ""  ;  N DATA,PCE  S DATA=$S(FT=3:$P($G(^DIC(36,+INS,6)),U,1,4),FT=2:$P($G(^DIC(36,+INS,6)),U,5,8),1:"")  ;  ; Check for dangling IDs/Qualifiers  F PCE=1,3 D  . I $P(DATA,U,PCE)'="",$P(DATA,U,PCE+1)'="" Q  . S ($P(DATA,U,PCE),$P(DATA,U,PCE+1))=""  ;  ; fill in the gap if there is one  I $P(DATA,U,1)="",$P(DATA,U,3)'="" D  . S $P(DATA,U,1)=$P(DATA,U,3)  . S $P(DATA,U,2)=$P(DATA,U,4)  . S ($P(DATA,U,3),$P(DATA,U,4))=""  ;  Q DATA  ; HPID(IBXIEN) ; Determine HPID for current payer.  ; MRD;IB\*2.0\*516 - Added HPID.  ;  N IBHPID,IBSEQ  S IBSEQ=$$COBN^IBCEF(IBXIEN) ; IBSEQ should be 1, 2 or 3.  I IBSEQ S IBHPID=$P($G(^DGCR(399,IBXIEN,"M1")),U,12+IBSEQ) ; Pull piece 13, 14 or 15.  Q IBHPID  ; CLEANUP(IBRET) ;  K IBRET("CI\_PID"),IBRET("OI\_PID"),IBRET("CI\_PSIDS"),IBRET("OI\_PSIDS"),IBRET("CI\_HPID"),IBRET("OI\_HPID")  Q  ; | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCEF78 ;ALB/WCJ - Provider ID functions ;13 May 2007  ;;2.0;INTEGRATED BILLING;\*\*371,516,**592**\*\*;21-MAR-94;Build 123  ;;Per VA Directive 6402, this routine should not be modified.  ;;  G AWAY AWAY Q  ; PAYERIDS(IBXIEN,IBRET) ; This function returns all the PAYER IDS for the current and other insurance(s)  ;   D PRIPAYID(IBXIEN,.IBRET)  D SECPAYID(IBXIEN,.IBRET)  Q  ; PRIPAYID(IBXIEN,IBXRET) ; Primary Payer IDs  ; Incoming:  ; IBXIEN = IEN for File # 399  ; IBXRET = Return Array for Qualifiers and IDs  ;  ; Outgoing  ; IBXRET("CI\_PID",1)=QUAL^ID  ; IBXRET("OI\_PID",#)=QUAL^ID  ;   N RET,I  S RET=$$PAYERID^IBCEF2(IBXIEN)  I RET]"" S IBXRET("CI\_PID",1)="PI"\_U\_RET  ;  ; MRD;IB\*2.0\*516 - Added HPID here (CI) and below (OI).  S RET=$$HPID(IBXIEN)  I RET]"" S IBXRET("CI\_HPID",1)="XV"\_U\_RET  ;  S RET=""  D OTHINSID^IBCEF72(IBXIEN,.RET)  F I=1,2 D  . I $P($G(RET(I)),U)]"" S IBXRET("OI\_PID",I)="PI"\_U\_$P(RET(I),U)  . I $P($G(RET(I)),U,2)]"" S IBXRET("OI\_HPID",I)="XV"\_U\_$P(RET(I),U,2)  . Q  Q  ; SECPAYID(IBXIEN,IBXRET) ; This returns all of the secondary payer IDs from file #36   ; for the insurance companies on a given claim  ;   ; Incoming:  ; IBXIEN = IEN for File # 399  ; IBXRET = Return Array for Qualifiers and IDs  ;  ; Outgoing  ; IBXRET("CI\_PSIDS",1)=QUAL^ID^QUAL^ID  ; IBXRET("OI\_PSIDS",#)=QUAL^ID^QUAL^ID  ;  N Z,C,IBZ,Z0,FT  F Z=1:1:3 S IBZ(Z)=$$POLICY^IBCEF(IBXIEN,1,Z)  S Z0=0,C=$$COBN^IBCEF(IBXIEN),FT=$$FT^IBCEF(IBXIEN)  F Z=1:1:3 S:C'=Z Z0=Z0+1 S IBXRET($S(C=Z:"CI\_PSIDS",1:"OI\_PSIDS"),$S(C=Z:1,1:Z0))=$$SPIDS(+IBZ(Z),FT)  Q  ; SPIDS(INS,FT) ;  ; FT = FORM TYPE (2 PROFESSIONAL 3 INSTITUTIONAL)  ; INS = INSURANCE COMPANY (FILE #36) IEN  ; Returns String (^ delimited)  ; [1] = QUAL 1  ; [2] = PAYER ID 1  ; [3] = QUAL 2  ; [4] = PAYER ID 2  Q:'+INS ""  ;  N DATA,PCE **;JWS;IB\*2.0\*592;Dental form 7 same as form 2 - no secondaries for Dental**  **S DATA=$S(FT=3:$P($G(^DIC(36,+INS,6)),U,1,4),FT=2:$P($G(^DIC(36,+INS,6)),U,5,8),1:"")**  ;  ; Check for dangling IDs/Qualifiers  F PCE=1,3 D  . I $P(DATA,U,PCE)'="",$P(DATA,U,PCE+1)'="" Q  . S ($P(DATA,U,PCE),$P(DATA,U,PCE+1))=""  ;  ; fill in the gap if there is one  I $P(DATA,U,1)="",$P(DATA,U,3)'="" D  . S $P(DATA,U,1)=$P(DATA,U,3)  . S $P(DATA,U,2)=$P(DATA,U,4)  . S ($P(DATA,U,3),$P(DATA,U,4))=""  ;  Q DATA  ; HPID(IBXIEN) ; Determine HPID for current payer.  ; MRD;IB\*2.0\*516 - Added HPID.  ;  N IBHPID,IBSEQ  S IBSEQ=$$COBN^IBCEF(IBXIEN) ; IBSEQ should be 1, 2 or 3.  I IBSEQ S IBHPID=$P($G(^DGCR(399,IBXIEN,"M1")),U,12+IBSEQ) ; Pull piece 13, 14 or 15.  Q IBHPID  ; CLEANUP(IBRET) ;  K IBRET("CI\_PID"),IBRET("OI\_PID"),IBRET("CI\_PSIDS"),IBRET("OI\_PSIDS"),IBRET("CI\_HPID"),IBRET("OI\_HPID")  Q  ; | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCEF81 | | | | | | | | |
| **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** | | | | | | | | | |
| IBCEF81 ;ALB/BI - PROVIDER ADJUSTMENTS ;11-OCT-2010  ;;2.0;INTEGRATED BILLING;\*\*432,473\*\*;21-MAR-94;Build 29  ;;Per VHA Directive 2004-038, this routine should not be modified.  ;  Q  ; EN(INPUT) ; FIRST ENTRY POINT  N INSLEVEL,PRTYPE,OUTPUT,IBIEN,CMODE,CPRNUM,STATUS  S STATUS=1  I $D(INPUT)=0 S STATUS=0 Q STATUS  I (($G(IBXFORM)=2)!($G(IBXFORM)=3)) D EN^IBCEF82(.INPUT) Q STATUS    ; PERFORM LOCAL PRINT BUSINESS RULES  K OUTPUT M OUTPUT=INPUT  D CINIT1 Q:IBIEN="" STATUS  F INSLEVEL="P","S","T" D    ; P=PRIMARY, S=SECONDARY, T=TERTIARY  . D CINIT2  . F PRTYPE=1,2,3,5,9 D    ; 1=REFERRING, 2=OPERATING, 3=RENDERING, 5=SUPERVISING, 9=OTHER OPERATING  .. D START(INSLEVEL,PRTYPE,.OUTPUT)  K INPUT M INPUT=OUTPUT  Q STATUS START(INSLEVEL,PRTYPE,OUTPUT) ; START PROCESSING  N INTERM,PROVINFO,MAXAINFO,FIRSTINF  S INTERM="A"  S INTERM=INTERM\_$$TEST1  ; Does Claim Level Provider Exist, 0=NO, 1=YES  S INTERM=INTERM\_$$TEST2  ; All procedures have a line level provider, 0=NO, 1=YES  S INTERM=INTERM\_$$TEST3  ; One Line Level provider is most significant, 0=NO, 1=YES  S INTERM=INTERM\_$$TEST4  ; At least one line level provider matches the claim level provider, 0=NO, 1=YES  S INTERM=INTERM\_$$TEST5  ; There is only one procedure without a line level provider, 0=NO, 1=YES  D @INTERM  Q   ; TEST1() ; Does Claim Level Provider Exist, 0=NO, 1=YES  N PROVX,PROVY  I $D(CMODE(INSLEVEL))#10=0 Q 0  I $D(CPRNUM(INSLEVEL))#10=0 Q 0  S (PROVX,PROVY)=$G(INPUT("PROVINF",IBIEN,CMODE(INSLEVEL),CPRNUM(INSLEVEL),PRTYPE)) Q:PROVX="" 0  S PROVX="^"\_$P(PROVX,";",2)\_$P(PROVX,";",1)\_")"  I $D(@PROVX) D  Q 1 ;CLAIM PROVIDER EXISTS, RETURN TRUE.  . ; LOAD CLAIM LEVEL PROVIDER INFORMATION  . S PROVINFO=PROVY  . S PROVINFO("PROVINF",IBIEN)=IBIEN  . S PROVINFO("PROVINF",IBIEN,CMODE(INSLEVEL))=""  . S PROVINFO("PROVINF",IBIEN,CMODE(INSLEVEL),CPRNUM(INSLEVEL))=INSLEVEL  . M PROVINFO("PROVINF",IBIEN,CMODE(INSLEVEL),CPRNUM(INSLEVEL),PRTYPE)=INPUT("PROVINF",IBIEN,CMODE(INSLEVEL),CPRNUM(INSLEVEL),PRTYPE)  Q 0  ; TEST2() ; All procedures have a line level provider, 0=NO, 1=YES  N SLC,RESULT,LMODE,LPRNUM,PROVX,LINECNT  S SLC=0,RESULT=1,LINECNT=0  F  S SLC=$$LINIT1(SLC) Q:+SLC=0 D  . S LINECNT=LINECNT+1  . D LINIT2  . I $D(LMODE(INSLEVEL))#10=0 S RESULT=0 Q  . I $D(LPRNUM(INSLEVEL))#10=0 S RESULT=0 Q  . S PROVX=$G(INPUT("L-PROV",IBIEN,SLC,LMODE(INSLEVEL),LPRNUM(INSLEVEL),PRTYPE))  . I PROVX="" D  Q  .. S RESULT=RESULT\*0  . S PROVX="^"\_$P(PROVX,";",2)\_$P(PROVX,";",1)\_")"  . S RESULT=RESULT\*($D(@PROVX)'=0)  I +$G(INPUT("SLC"))'=0,INPUT("SLC")>LINECNT S RESULT=0  Q RESULT  ; TEST3() ; One Line Level provider is most significant, 0=NO, 1=YES  N SLC,RESULT,LMODE,LPRNUM,PCOUNT,PCOUNTF,PCOUNTL,PROVX,TEMPNODE  S SLC=0,RESULT=0  F  S SLC=$$LINIT1(SLC) Q:+SLC=0 D  . D LINIT2  . I $D(LMODE(INSLEVEL))#10=0 Q  . I $D(LPRNUM(INSLEVEL))#10=0 Q  . S PROVX=$G(INPUT("L-PROV",IBIEN,SLC,LMODE(INSLEVEL),LPRNUM(INSLEVEL),PRTYPE)) Q:PROVX=""  . I $D(FIRSTINF)=0 D  .. ; LOAD FIRST AVAILABLE PROVIDER INFORMATION  .. S FIRSTINF=$G(INPUT("L-PROV",IBIEN,SLC,LMODE(INSLEVEL),LPRNUM(INSLEVEL),PRTYPE))  .. S FIRSTINF("L-PROV",IBIEN)=IBIEN  .. S FIRSTINF("L-PROV",IBIEN,LMODE(INSLEVEL),LPRNUM(INSLEVEL))=INSLEVEL  .. M FIRSTINF("L-PROV",IBIEN,LMODE(INSLEVEL),LPRNUM(INSLEVEL),PRTYPE)=INPUT("L-PROV",IBIEN,SLC,LMODE(INSLEVEL),LPRNUM(INSLEVEL),PRTYPE)  . S PCOUNT(PROVX)=$P($G(PCOUNT(PROVX)),"^",1)+1\_"^"\_SLC\_"^"\_LMODE(INSLEVEL)\_"^"\_LPRNUM(INSLEVEL)\_"^"\_PRTYPE  S PROVX="" F  S PROVX=$O(PCOUNT(PROVX)) Q:PROVX=""  D  . S PCOUNTF($P(PCOUNT(PROVX),"^",1),PROVX)=$P(PCOUNT(PROVX),"^",2,5)  S PCOUNTL(1)=$O(PCOUNTF(""),-1) Q:PCOUNTL(1)="" RESULT  S PCOUNTL(2,1)=$O(PCOUNTF(PCOUNTL(1),""),-1) Q:PCOUNTL(2,1)="" RESULT  S PCOUNTL(2,2)=$O(PCOUNTF(PCOUNTL(1),PCOUNTL(2,1)),-1)  I PCOUNTL(2,2)="" D  . S RESULT=1  . ; LOAD MOST SIGNIFICANT LINE LEVEL PROVIDER INFORMATION  . S MAXAINFO=PCOUNTL(2,1)  . S TEMPNODE=PCOUNTF(PCOUNTL(1),PCOUNTL(2,1))  . S MAXAINFO("L-PROV",IBIEN)=IBIEN  . S MAXAINFO("L-PROV",IBIEN,$P(TEMPNODE,"^",2),$P(TEMPNODE,"^",3))=INSLEVEL  . M MAXAINFO("L-PROV",IBIEN,$P(TEMPNODE,"^",2),$P(TEMPNODE,"^",3),$P(TEMPNODE,"^",4))=INPUT("L-PROV",IBIEN,$P(TEMPNODE,"^",1),$P(TEMPNODE,"^",2),$P(TEMPNODE,"^",3),$P(TEMPNODE,"^",4))  Q RESULT  ; TEST4() ; At least one line level provider matches the claim level provider, 0=NO, 1=YES  N CPROV,RESULT,LMODE,LPRNUM,LPROV,SLC  I $D(CMODE(INSLEVEL))#10=0 Q 0  I $D(CPRNUM(INSLEVEL))#10=0 Q 0  S CPROV=$G(INPUT("PROVINF",IBIEN,CMODE(INSLEVEL),CPRNUM(INSLEVEL),PRTYPE)) Q:CPROV="" 0  S SLC=0,RESULT=0  F  S SLC=$$LINIT1(SLC) Q:+SLC=0 D  . D LINIT2  . I $D(LMODE(INSLEVEL))#10=0 Q  . I $D(LPRNUM(INSLEVEL))#10=0 Q  . S LPROV=$G(INPUT("L-PROV",IBIEN,SLC,LMODE(INSLEVEL),LPRNUM(INSLEVEL),PRTYPE)) Q:LPROV=""  . I LPROV=CPROV S RESULT=1  Q RESULT  ; TEST5() ; There is only one procedure without a line level provider, 0=NO, 1=YES  N SLC,LMODE,LPRNUM,PROVCNT,RESULT  S SLC=0,PROVCNT=0,RESULT=0  F  S SLC=$$LINIT1(SLC) Q:+SLC=0 D  . D LINIT2  . I $D(LMODE(INSLEVEL))#10=0 Q  . I $D(LPRNUM(INSLEVEL))#10=0 Q  . S PROVX=$G(INPUT("L-PROV",IBIEN,SLC,LMODE(INSLEVEL),LPRNUM(INSLEVEL),PRTYPE))  . S:PROVX'="" PROVCNT=PROVCNT+1  I +$G(INPUT("SLC"))'=0,INPUT("SLC")=(PROVCNT+1) S RESULT=1  Q RESULT  ; A00000  ; Case 1  ; TESTS: Does Claim Level Provider Exist: 0=NO  ; All procedures have a line level provider: 0=NO  ; One Line Level provider is most significant: 0=NO  ; At least one line level provider matches the claim level provider: 0=NO  ; There is only one procedure without a line level provider: 0=NO  ;  ; Move the first available line level provider to the claim level.  I $G(FIRSTINF)="" Q  M OUTPUT("PROVINF",IBIEN)=FIRSTINF("L-PROV",IBIEN)  ;  ; Remove the claim lines associated with the primary provider.  S PROVINFO=FIRSTINF  D REMOVELN  Q  ; A00001  ; Case 2  ; TESTS: Does Claim Level Provider Exist: 0=NO  ; All procedures have a line level provider: 0=NO  ; One Line Level provider is most significant: 0=NO  ; At least one line level provider matches the claim level provider: 0=NO  ; There is only one procedure without a line level provider: 1=YES  ;  ; Move the first available line level provider to the claim level.  I $G(FIRSTINF)="" Q  M OUTPUT("PROVINF",IBIEN)=FIRSTINF("L-PROV",IBIEN)  ;  ; Remove the claim lines associated with the primary provider.  S PROVINFO=FIRSTINF  D REMOVELN  Q  ; A00010  ; Case 3 - This case can never happen!  ; ACTIONS: N/A - Transmit as is.  Q  ; A00011  ; Case 4 - This case can never happen!  ; ACTIONS: N/A - Transmit as is.  Q  ; A00100  ; Case 5  ; TESTS: Does Claim Level Provider Exist: 0=NO  ; All procedures have a line level provider: 0=NO  ; One Line Level provider is most significant: 1=YES  ; At least one line level provider matches the claim level provider: 0=NO  ; There is only one procedure without a line level provider: 0=NO  ;  ; Set the claim level provider equal to the most significant line level provider.  I $G(MAXAINFO)="" Q  M OUTPUT("PROVINF",IBIEN)=MAXAINFO("L-PROV",IBIEN)  ;  ; Remove the claim lines associated with the primary provider.  S PROVINFO=MAXAINFO  D REMOVELN  ;  Q  ; A00101  ; Case 6  ; TESTS: Does Claim Level Provider Exist: 0=NO  ; All procedures have a line level provider: 0=NO  ; One Line Level provider is most significant: 1=YES  ; At least one line level provider matches the claim level provider: 0=NO  ; There is only one procedure without a line level provider: 1=YES  ;  ; Set the claim level provider equal to the most significant line level provider.  I $G(MAXAINFO)="" Q  M OUTPUT("PROVINF",IBIEN)=MAXAINFO("L-PROV",IBIEN)  ;  ; Remove the claim lines associated with the primary provider.  S PROVINFO=MAXAINFO  D REMOVELN  ;  Q  ; A00110  ; Case 7 - This case can never happen!  ; ACTIONS: N/A - Transmit as is.  Q  ; A00111  ; Case 8 - This case can never happen!  ; ACTIONS: N/A - Transmit as is.  Q  ; A01000  ; Case 9  ; TESTS: Does Claim Level Provider Exist: 0=NO  ; All procedures have a line level provider: 1=YES  ; One Line Level provider is most significant: 0=NO  ; At least one line level provider matches the claim level provider: 0=NO  ; There is only one procedure without a line level provider: 0=NO  ;  ; Move the first available line level provider to the claim level.  ; Set the claim level provider equal to the most significant line level provider.  I $G(FIRSTINF)="" Q  M OUTPUT("PROVINF",IBIEN)=FIRSTINF("L-PROV",IBIEN)  ;  ; Remove the claim lines associated with the primary provider.  S PROVINFO=FIRSTINF  D REMOVELN  Q  ; A01001  ; Case 10 - This case can never happen!  ; ACTIONS: N/A - Transmit as is.  Q  ; A01010  ; Case 11 - This case can never happen!  ; ACTIONS: N/A - Transmit as is.  Q  ; A01011  ; Case 12 - This case can never happen!  ; ACTIONS: N/A - Transmit as is.  Q  ; A01100  ; Case 13  ; TESTS: Does Claim Level Provider Exist: 0=NO  ; All procedures have a line level provider: 1=YES  ; One Line Level provider is most significant: 1=YES  ; At least one line level provider matches the claim level provider: 0=NO  ; There is only one procedure without a line level provider: 0=NO  ;  ; Set the claim level provider equal to the most significant line level provider.  I $G(MAXAINFO)="" Q  M OUTPUT("PROVINF",IBIEN)=MAXAINFO("L-PROV",IBIEN)  ;  ; Remove the claim lines associated with the primary provider.  S PROVINFO=MAXAINFO  D REMOVELN  ;  Q  ; A01101  ; Case 14 - This case can never happen!  ; ACTIONS: N/A - Transmit as is.  Q  ; A01110  ; Case 15 - This case can never happen!  ; ACTIONS: N/A - Transmit as is.  Q  ; A01111  ; Case 16 - This case can never happen!  ; ACTIONS: N/A - Transmit as is.  Q  ; A10000  ; Case 17  ; TESTS: Does Claim Level Provider Exist: 1=YES  ; All procedures have a line level provider: 0=NO  ; One Line Level provider is most significant: 0=NO  ; At least one line level provider matches the claim level provider: 0=NO  ; There is only one procedure without a line level provider: 0=NO  ;  ; ACTIONS: Transmit as is.  ;  Q  ; A10001   ; Case 18  ; TESTS: Does Claim Level Provider Exist: 1=YES  ; All procedures have a line level provider: 0=NO  ; One Line Level provider is most significant: 0=NO  ; At least one line level provider matches the claim level provider: 0=NO  ; There is only one procedure without a line level provider: 1=YES  ;  ; ACTIONS: Transmit as is.  ;  Q  ; A10010   ; Case 19  ; TESTS: Does Claim Level Provider Exist: 1=YES  ; All procedures have a line level provider: 0=NO  ; One Line Level provider is most significant: 0=NO  ; At least one line level provider matches the claim level provider: 1=YES  ; There is only one procedure without a line level provider: 0=NO  ;  ; Remove the claim lines associated with the claim level provider.  D REMOVELN  ;  Q  ; A10011   ; Case 20  ; TESTS: Does Claim Level Provider Exist: 1=YES  ; All procedures have a line level provider: 0=NO  ; One Line Level provider is most significant: 0=NO  ; At least one line level provider matches the claim level provider: 1=YES  ; There is only one procedure without a line level provider: 1=YES  ;  ; Remove the claim lines associated with the claim level provider.  D REMOVELN  ;  Q  ; A10100   ; Case 21  ; TESTS: Does Claim Level Provider Exist: 1=YES  ; All procedures have a line level provider: 0=NO  ; One Line Level provider is most significant: 1=YES  ; At least one line level provider matches the claim level provider: 0=NO  ; There is only one procedure without a line level provider: 0=NO  ;  ; ACTIONS: Transmit as is.  ;  Q  ; A10101   ; Case 22  ; TESTS: Does Claim Level Provider Exist: 1=YES  ; All procedures have a line level provider: 0=NO  ; One Line Level provider is most significant: 1=YES  ; At least one line level provider matches the claim level provider: 0=NO  ; There is only one procedure without a line level provider: 1=YES  ;  ; ACTIONS: Transmit as is.  ;  Q  ; A10110   ; Case 23  ; TESTS: Does Claim Level Provider Exist: 1=YES  ; All procedures have a line level provider: 0=NO  ; One Line Level provider is most significant: 1=YES  ; At least one line level provider matches the claim level provider: 1=YES  ; There is only one procedure without a line level provider: 0=NO  ;  ; Remove the claim lines associated with the claim level provider.  D REMOVELN  ;  Q  ; A10111   ; Case 24  ; TESTS: Does Claim Level Provider Exist: 1=YES  ; All procedures have a line level provider: 0=NO  ; One Line Level provider is most significant: 1=YES  ; At least one line level provider matches the claim level provider: 1=YES  ; There is only one procedure without a line level provider: 1=YES  ;  ; Remove the claim lines associated with the claim level provider.  D REMOVELN  ;  Q  ; A11000   ; Case 25  ; TESTS: Does Claim Level Provider Exist: 1=YES  ; All procedures have a line level provider: 1=YES  ; One Line Level provider is most significant: 0=NO  ; At least one line level provider matches the claim level provider: 0=NO  ; There is only one procedure without a line level provider: 0=NO  ;  S STATUS="0^CASE 25 ERROR"  ;  Q  ; A11001   ; Case 26 - This case can never happen!  ; ACTIONS: N/A - Transmit as is.  Q  ; A11010   ; Case 27  ; TESTS: Does Claim Level Provider Exist: 1=YES  ; All procedures have a line level provider: 1=YES  ; One Line Level provider is most significant: 0=NO  ; At least one line level provider matches the claim level provider: 1=YES  ; There is only one procedure without a line level provider: 0=NO  ;  ; Remove the claim lines associated with the claim level provider.  D REMOVELN  ;  Q  ; A11011   ; Case 28 - This case can never happen!  ; ACTIONS: N/A - Transmit as is.  Q  ; A11100   ; Case 29  ; TESTS: Does Claim Level Provider Exist: 1=YES  ; All procedures have a line level provider: 1=YES  ; One Line Level provider is most significant: 1=YES  ; At least one line level provider matches the claim level provider: 0=NO  ; There is only one procedure without a line level provider: 0=NO  ;  S STATUS="0^CASE 29 ERROR"  ;  Q  ; A11101   ; Case 30 - This case can never happen!  ; ACTIONS: N/A - Transmit as is.  Q  ; A11110   ; Case 31  ; TESTS: Does Claim Level Provider Exist: 1=YES  ; All procedures have a line level provider: 1=YES  ; One Line Level provider is most significant: 1=YES  ; At least one line level provider matches the claim level provider: 1=YES  ; There is only one procedure without a line level provider: 0=NO  ;  ; Remove the claim lines associated with the claim level provider.  D REMOVELN  ;  Q  ; A11111   ; Case 32 - This case can never happen!  ; ACTIONS: N/A - Transmit as is.  Q  ; CINIT1    ; Claim level initiation  S IBIEN=$O(INPUT("L-PROV",0))  I IBIEN="" S IBIEN=$O(INPUT("PROVINF",0))  I IBIEN="" S IBIEN=$O(INPUT("LAB/FAC",0))  Q  ; CINIT2    ; Claim level initiation  N MODEX,PRNUMX,PROVX  F MODEX="C","O" D  . S PRNUMX=0 F  S PRNUMX=$O(INPUT("PROVINF",IBIEN,MODEX,PRNUMX)) Q:+PRNUMX=0 D  .. I $G(INPUT("PROVINF",IBIEN,MODEX,PRNUMX))="" Q  .. I INPUT("PROVINF",IBIEN,MODEX,PRNUMX)=INSLEVEL S CMODE(INSLEVEL)=MODEX,CPRNUM(INSLEVEL)=PRNUMX  Q  ; LINIT1(SLC) ; Line level initiation  Q $O(INPUT("L-PROV",IBIEN,SLC))  ; LINIT2    ; Line level initiation  N MODEX,PRNUMX,PROVX  F MODEX="C","O" D  . S PRNUMX=0 F  S PRNUMX=$O(INPUT("L-PROV",IBIEN,SLC,MODEX,PRNUMX)) Q:+PRNUMX=0 D  .. I INPUT("L-PROV",IBIEN,SLC,MODEX,PRNUMX)=INSLEVEL S LMODE(INSLEVEL)=MODEX,LPRNUM(INSLEVEL)=PRNUMX  Q  ; REMOVELN    ; Remove the claim lines associated with the claim level provider.  N MODEX,PRNUMX,PROVX  S SLC=0 F  S SLC=$O(OUTPUT("L-PROV",IBIEN,SLC)) Q:+SLC=0 D  . F MODEX="C","O" D  .. S PRNUMX=0 F  S PRNUMX=$O(OUTPUT("L-PROV",IBIEN,SLC,MODEX,PRNUMX)) Q:+PRNUMX=0 D  ... Q:$G(PROVINFO)=""  ... I $G(OUTPUT("L-PROV",IBIEN,SLC,MODEX,PRNUMX,PRTYPE))=PROVINFO D  .... K OUTPUT("L-PROV",IBIEN,SLC,MODEX,PRNUMX,PRTYPE)  .... I $D(OUTPUT("L-PROV",IBIEN,SLC,MODEX,PRNUMX))=1 K OUTPUT("L-PROV",IBIEN,SLC,MODEX,PRNUMX)  .... I $D(OUTPUT("L-PROV",IBIEN,SLC,MODEX))=1 K OUTPUT("L-PROV",IBIEN,SLC,MODEX)  .... I $D(OUTPUT("L-PROV",IBIEN,SLC))=1 K OUTPUT("L-PROV",IBIEN,SLC)  Q | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCEF81 ;ALB/BI - PROVIDER ADJUSTMENTS ;11-OCT-2010  ;;2.0;INTEGRATED BILLING;\*\*432,473,**592**\*\*;21-MAR-94;Build 29  ;;Per VHA Directive 2004-038, this routine should not be modified.  ;  Q  ; EN(INPUT) ; FIRST ENTRY POINT  N INSLEVEL,PRTYPE,OUTPUT,IBIEN,CMODE,CPRNUM,STATUS  S STATUS=1  I $D(INPUT)=0 S STATUS=0 Q STATUS  I (($G(IBXFORM)=2)!($G(IBXFORM)=3)) D EN^IBCEF82(.INPUT) Q STATUS    ; PERFORM LOCAL PRINT BUSINESS RULES  K OUTPUT M OUTPUT=INPUT  D CINIT1 Q:IBIEN="" STATUS  F INSLEVEL="P","S","T" D    ; P=PRIMARY, S=SECONDARY, T=TERTIARY  . D CINIT2  **. ;JWS;IB\*2.0\*592; 6 assistant surgeon for dental**  . F PRTYPE=1,2,3,5,**6**,9 D    ; 1=REFERRING, 2=OPERATING, 3=RENDERING, 5=SUPERVISING, **6=ASSISTANT SURGEON**, 9=OTHER OPERATING  .. D START(INSLEVEL,PRTYPE,.OUTPUT)  K INPUT M INPUT=OUTPUT  Q STATUS START(INSLEVEL,PRTYPE,OUTPUT) ; START PROCESSING  N INTERM,PROVINFO,MAXAINFO,FIRSTINF  S INTERM="A"  S INTERM=INTERM\_$$TEST1  ; Does Claim Level Provider Exist, 0=NO, 1=YES  S INTERM=INTERM\_$$TEST2  ; All procedures have a line level provider, 0=NO, 1=YES  S INTERM=INTERM\_$$TEST3  ; One Line Level provider is most significant, 0=NO, 1=YES  S INTERM=INTERM\_$$TEST4  ; At least one line level provider matches the claim level provider, 0=NO, 1=YES  S INTERM=INTERM\_$$TEST5  ; There is only one procedure without a line level provider, 0=NO, 1=YES  D @INTERM  Q   ; TEST1() ; Does Claim Level Provider Exist, 0=NO, 1=YES  N PROVX,PROVY  I $D(CMODE(INSLEVEL))#10=0 Q 0  I $D(CPRNUM(INSLEVEL))#10=0 Q 0  S (PROVX,PROVY)=$G(INPUT("PROVINF",IBIEN,CMODE(INSLEVEL),CPRNUM(INSLEVEL),PRTYPE)) Q:PROVX="" 0  S PROVX="^"\_$P(PROVX,";",2)\_$P(PROVX,";",1)\_")"  I $D(@PROVX) D  Q 1 ;CLAIM PROVIDER EXISTS, RETURN TRUE.  . ; LOAD CLAIM LEVEL PROVIDER INFORMATION  . S PROVINFO=PROVY  . S PROVINFO("PROVINF",IBIEN)=IBIEN  . S PROVINFO("PROVINF",IBIEN,CMODE(INSLEVEL))=""  . S PROVINFO("PROVINF",IBIEN,CMODE(INSLEVEL),CPRNUM(INSLEVEL))=INSLEVEL  . M PROVINFO("PROVINF",IBIEN,CMODE(INSLEVEL),CPRNUM(INSLEVEL),PRTYPE)=INPUT("PROVINF",IBIEN,CMODE(INSLEVEL),CPRNUM(INSLEVEL),PRTYPE)  Q 0  ; TEST2() ; All procedures have a line level provider, 0=NO, 1=YES  N SLC,RESULT,LMODE,LPRNUM,PROVX,LINECNT  S SLC=0,RESULT=1,LINECNT=0  F  S SLC=$$LINIT1(SLC) Q:+SLC=0 D  . S LINECNT=LINECNT+1  . D LINIT2  . I $D(LMODE(INSLEVEL))#10=0 S RESULT=0 Q  . I $D(LPRNUM(INSLEVEL))#10=0 S RESULT=0 Q  . S PROVX=$G(INPUT("L-PROV",IBIEN,SLC,LMODE(INSLEVEL),LPRNUM(INSLEVEL),PRTYPE))  . I PROVX="" D  Q  .. S RESULT=RESULT\*0  . S PROVX="^"\_$P(PROVX,";",2)\_$P(PROVX,";",1)\_")"  . S RESULT=RESULT\*($D(@PROVX)'=0)  I +$G(INPUT("SLC"))'=0,INPUT("SLC")>LINECNT S RESULT=0  Q RESULT  ; TEST3() ; One Line Level provider is most significant, 0=NO, 1=YES  N SLC,RESULT,LMODE,LPRNUM,PCOUNT,PCOUNTF,PCOUNTL,PROVX,TEMPNODE  S SLC=0,RESULT=0  F  S SLC=$$LINIT1(SLC) Q:+SLC=0 D  . D LINIT2  . I $D(LMODE(INSLEVEL))#10=0 Q  . I $D(LPRNUM(INSLEVEL))#10=0 Q  . S PROVX=$G(INPUT("L-PROV",IBIEN,SLC,LMODE(INSLEVEL),LPRNUM(INSLEVEL),PRTYPE)) Q:PROVX=""  . I $D(FIRSTINF)=0 D  .. ; LOAD FIRST AVAILABLE PROVIDER INFORMATION  .. S FIRSTINF=$G(INPUT("L-PROV",IBIEN,SLC,LMODE(INSLEVEL),LPRNUM(INSLEVEL),PRTYPE))  .. S FIRSTINF("L-PROV",IBIEN)=IBIEN  .. S FIRSTINF("L-PROV",IBIEN,LMODE(INSLEVEL),LPRNUM(INSLEVEL))=INSLEVEL  .. M FIRSTINF("L-PROV",IBIEN,LMODE(INSLEVEL),LPRNUM(INSLEVEL),PRTYPE)=INPUT("L-PROV",IBIEN,SLC,LMODE(INSLEVEL),LPRNUM(INSLEVEL),PRTYPE)  . S PCOUNT(PROVX)=$P($G(PCOUNT(PROVX)),"^",1)+1\_"^"\_SLC\_"^"\_LMODE(INSLEVEL)\_"^"\_LPRNUM(INSLEVEL)\_"^"\_PRTYPE  S PROVX="" F  S PROVX=$O(PCOUNT(PROVX)) Q:PROVX=""  D  . S PCOUNTF($P(PCOUNT(PROVX),"^",1),PROVX)=$P(PCOUNT(PROVX),"^",2,5)  S PCOUNTL(1)=$O(PCOUNTF(""),-1) Q:PCOUNTL(1)="" RESULT  S PCOUNTL(2,1)=$O(PCOUNTF(PCOUNTL(1),""),-1) Q:PCOUNTL(2,1)="" RESULT  S PCOUNTL(2,2)=$O(PCOUNTF(PCOUNTL(1),PCOUNTL(2,1)),-1)  I PCOUNTL(2,2)="" D  . S RESULT=1  . ; LOAD MOST SIGNIFICANT LINE LEVEL PROVIDER INFORMATION  . S MAXAINFO=PCOUNTL(2,1)  . S TEMPNODE=PCOUNTF(PCOUNTL(1),PCOUNTL(2,1))  . S MAXAINFO("L-PROV",IBIEN)=IBIEN  . S MAXAINFO("L-PROV",IBIEN,$P(TEMPNODE,"^",2),$P(TEMPNODE,"^",3))=INSLEVEL  . M MAXAINFO("L-PROV",IBIEN,$P(TEMPNODE,"^",2),$P(TEMPNODE,"^",3),$P(TEMPNODE,"^",4))=INPUT("L-PROV",IBIEN,$P(TEMPNODE,"^",1),$P(TEMPNODE,"^",2),$P(TEMPNODE,"^",3),$P(TEMPNODE,"^",4))  Q RESULT  ; TEST4() ; At least one line level provider matches the claim level provider, 0=NO, 1=YES  N CPROV,RESULT,LMODE,LPRNUM,LPROV,SLC  I $D(CMODE(INSLEVEL))#10=0 Q 0  I $D(CPRNUM(INSLEVEL))#10=0 Q 0  S CPROV=$G(INPUT("PROVINF",IBIEN,CMODE(INSLEVEL),CPRNUM(INSLEVEL),PRTYPE)) Q:CPROV="" 0  S SLC=0,RESULT=0  F  S SLC=$$LINIT1(SLC) Q:+SLC=0 D  . D LINIT2  . I $D(LMODE(INSLEVEL))#10=0 Q  . I $D(LPRNUM(INSLEVEL))#10=0 Q  . S LPROV=$G(INPUT("L-PROV",IBIEN,SLC,LMODE(INSLEVEL),LPRNUM(INSLEVEL),PRTYPE)) Q:LPROV=""  . I LPROV=CPROV S RESULT=1  Q RESULT  ; TEST5() ; There is only one procedure without a line level provider, 0=NO, 1=YES  N SLC,LMODE,LPRNUM,PROVCNT,RESULT  S SLC=0,PROVCNT=0,RESULT=0  F  S SLC=$$LINIT1(SLC) Q:+SLC=0 D  . D LINIT2  . I $D(LMODE(INSLEVEL))#10=0 Q  . I $D(LPRNUM(INSLEVEL))#10=0 Q  . S PROVX=$G(INPUT("L-PROV",IBIEN,SLC,LMODE(INSLEVEL),LPRNUM(INSLEVEL),PRTYPE))  . S:PROVX'="" PROVCNT=PROVCNT+1  I +$G(INPUT("SLC"))'=0,INPUT("SLC")=(PROVCNT+1) S RESULT=1  Q RESULT  ; A00000  ; Case 1  ; TESTS: Does Claim Level Provider Exist: 0=NO  ; All procedures have a line level provider: 0=NO  ; One Line Level provider is most significant: 0=NO  ; At least one line level provider matches the claim level provider: 0=NO  ; There is only one procedure without a line level provider: 0=NO  ;  ; Move the first available line level provider to the claim level.  I $G(FIRSTINF)="" Q  M OUTPUT("PROVINF",IBIEN)=FIRSTINF("L-PROV",IBIEN)  ;  ; Remove the claim lines associated with the primary provider.  S PROVINFO=FIRSTINF  D REMOVELN  Q  ; A00001  ; Case 2  ; TESTS: Does Claim Level Provider Exist: 0=NO  ; All procedures have a line level provider: 0=NO  ; One Line Level provider is most significant: 0=NO  ; At least one line level provider matches the claim level provider: 0=NO  ; There is only one procedure without a line level provider: 1=YES  ;  ; Move the first available line level provider to the claim level.  I $G(FIRSTINF)="" Q  M OUTPUT("PROVINF",IBIEN)=FIRSTINF("L-PROV",IBIEN)  ;  ; Remove the claim lines associated with the primary provider.  S PROVINFO=FIRSTINF  D REMOVELN  Q  ; A00010  ; Case 3 - This case can never happen!  ; ACTIONS: N/A - Transmit as is.  Q  ; A00011  ; Case 4 - This case can never happen!  ; ACTIONS: N/A - Transmit as is.  Q  ; A00100  ; Case 5  ; TESTS: Does Claim Level Provider Exist: 0=NO  ; All procedures have a line level provider: 0=NO  ; One Line Level provider is most significant: 1=YES  ; At least one line level provider matches the claim level provider: 0=NO  ; There is only one procedure without a line level provider: 0=NO  ;  ; Set the claim level provider equal to the most significant line level provider.  I $G(MAXAINFO)="" Q  M OUTPUT("PROVINF",IBIEN)=MAXAINFO("L-PROV",IBIEN)  ;  ; Remove the claim lines associated with the primary provider.  S PROVINFO=MAXAINFO  D REMOVELN  ;  Q  ; A00101  ; Case 6  ; TESTS: Does Claim Level Provider Exist: 0=NO  ; All procedures have a line level provider: 0=NO  ; One Line Level provider is most significant: 1=YES  ; At least one line level provider matches the claim level provider: 0=NO  ; There is only one procedure without a line level provider: 1=YES  ;  ; Set the claim level provider equal to the most significant line level provider.  I $G(MAXAINFO)="" Q  M OUTPUT("PROVINF",IBIEN)=MAXAINFO("L-PROV",IBIEN)  ;  ; Remove the claim lines associated with the primary provider.  S PROVINFO=MAXAINFO  D REMOVELN  ;  Q  ; A00110  ; Case 7 - This case can never happen!  ; ACTIONS: N/A - Transmit as is.  Q  ; A00111  ; Case 8 - This case can never happen!  ; ACTIONS: N/A - Transmit as is.  Q  ; A01000  ; Case 9  ; TESTS: Does Claim Level Provider Exist: 0=NO  ; All procedures have a line level provider: 1=YES  ; One Line Level provider is most significant: 0=NO  ; At least one line level provider matches the claim level provider: 0=NO  ; There is only one procedure without a line level provider: 0=NO  ;  ; Move the first available line level provider to the claim level.  ; Set the claim level provider equal to the most significant line level provider.  I $G(FIRSTINF)="" Q  M OUTPUT("PROVINF",IBIEN)=FIRSTINF("L-PROV",IBIEN)  ;  ; Remove the claim lines associated with the primary provider.  S PROVINFO=FIRSTINF  D REMOVELN  Q  ; A01001  ; Case 10 - This case can never happen!  ; ACTIONS: N/A - Transmit as is.  Q  ; A01010  ; Case 11 - This case can never happen!  ; ACTIONS: N/A - Transmit as is.  Q  ; A01011  ; Case 12 - This case can never happen!  ; ACTIONS: N/A - Transmit as is.  Q  ; A01100  ; Case 13  ; TESTS: Does Claim Level Provider Exist: 0=NO  ; All procedures have a line level provider: 1=YES  ; One Line Level provider is most significant: 1=YES  ; At least one line level provider matches the claim level provider: 0=NO  ; There is only one procedure without a line level provider: 0=NO  ;  ; Set the claim level provider equal to the most significant line level provider.  I $G(MAXAINFO)="" Q  M OUTPUT("PROVINF",IBIEN)=MAXAINFO("L-PROV",IBIEN)  ;  ; Remove the claim lines associated with the primary provider.  S PROVINFO=MAXAINFO  D REMOVELN  ;  Q  ; A01101  ; Case 14 - This case can never happen!  ; ACTIONS: N/A - Transmit as is.  Q  ; A01110  ; Case 15 - This case can never happen!  ; ACTIONS: N/A - Transmit as is.  Q  ; A01111  ; Case 16 - This case can never happen!  ; ACTIONS: N/A - Transmit as is.  Q  ; A10000  ; Case 17  ; TESTS: Does Claim Level Provider Exist: 1=YES  ; All procedures have a line level provider: 0=NO  ; One Line Level provider is most significant: 0=NO  ; At least one line level provider matches the claim level provider: 0=NO  ; There is only one procedure without a line level provider: 0=NO  ;  ; ACTIONS: Transmit as is.  ;  Q  ; A10001   ; Case 18  ; TESTS: Does Claim Level Provider Exist: 1=YES  ; All procedures have a line level provider: 0=NO  ; One Line Level provider is most significant: 0=NO  ; At least one line level provider matches the claim level provider: 0=NO  ; There is only one procedure without a line level provider: 1=YES  ;  ; ACTIONS: Transmit as is.  ;  Q  ; A10010   ; Case 19  ; TESTS: Does Claim Level Provider Exist: 1=YES  ; All procedures have a line level provider: 0=NO  ; One Line Level provider is most significant: 0=NO  ; At least one line level provider matches the claim level provider: 1=YES  ; There is only one procedure without a line level provider: 0=NO  ;  ; Remove the claim lines associated with the claim level provider.  D REMOVELN  ;  Q  ; A10011   ; Case 20  ; TESTS: Does Claim Level Provider Exist: 1=YES  ; All procedures have a line level provider: 0=NO  ; One Line Level provider is most significant: 0=NO  ; At least one line level provider matches the claim level provider: 1=YES  ; There is only one procedure without a line level provider: 1=YES  ;  ; Remove the claim lines associated with the claim level provider.  D REMOVELN  ;  Q  ; A10100   ; Case 21  ; TESTS: Does Claim Level Provider Exist: 1=YES  ; All procedures have a line level provider: 0=NO  ; One Line Level provider is most significant: 1=YES  ; At least one line level provider matches the claim level provider: 0=NO  ; There is only one procedure without a line level provider: 0=NO  ;  ; ACTIONS: Transmit as is.  ;  Q  ; A10101   ; Case 22  ; TESTS: Does Claim Level Provider Exist: 1=YES  ; All procedures have a line level provider: 0=NO  ; One Line Level provider is most significant: 1=YES  ; At least one line level provider matches the claim level provider: 0=NO  ; There is only one procedure without a line level provider: 1=YES  ;  ; ACTIONS: Transmit as is.  ;  Q  ; A10110   ; Case 23  ; TESTS: Does Claim Level Provider Exist: 1=YES  ; All procedures have a line level provider: 0=NO  ; One Line Level provider is most significant: 1=YES  ; At least one line level provider matches the claim level provider: 1=YES  ; There is only one procedure without a line level provider: 0=NO  ;  ; Remove the claim lines associated with the claim level provider.  D REMOVELN  ;  Q  ; A10111   ; Case 24  ; TESTS: Does Claim Level Provider Exist: 1=YES  ; All procedures have a line level provider: 0=NO  ; One Line Level provider is most significant: 1=YES  ; At least one line level provider matches the claim level provider: 1=YES  ; There is only one procedure without a line level provider: 1=YES  ;  ; Remove the claim lines associated with the claim level provider.  D REMOVELN  ;  Q  ; A11000   ; Case 25  ; TESTS: Does Claim Level Provider Exist: 1=YES  ; All procedures have a line level provider: 1=YES  ; One Line Level provider is most significant: 0=NO  ; At least one line level provider matches the claim level provider: 0=NO  ; There is only one procedure without a line level provider: 0=NO  ;  S STATUS="0^CASE 25 ERROR"  ;  Q  ; A11001   ; Case 26 - This case can never happen!  ; ACTIONS: N/A - Transmit as is.  Q  ; A11010   ; Case 27  ; TESTS: Does Claim Level Provider Exist: 1=YES  ; All procedures have a line level provider: 1=YES  ; One Line Level provider is most significant: 0=NO  ; At least one line level provider matches the claim level provider: 1=YES  ; There is only one procedure without a line level provider: 0=NO  ;  ; Remove the claim lines associated with the claim level provider.  D REMOVELN  ;  Q  ; A11011   ; Case 28 - This case can never happen!  ; ACTIONS: N/A - Transmit as is.  Q  ; A11100   ; Case 29  ; TESTS: Does Claim Level Provider Exist: 1=YES  ; All procedures have a line level provider: 1=YES  ; One Line Level provider is most significant: 1=YES  ; At least one line level provider matches the claim level provider: 0=NO  ; There is only one procedure without a line level provider: 0=NO  ;  S STATUS="0^CASE 29 ERROR"  ;  Q  ; A11101   ; Case 30 - This case can never happen!  ; ACTIONS: N/A - Transmit as is.  Q  ; A11110   ; Case 31  ; TESTS: Does Claim Level Provider Exist: 1=YES  ; All procedures have a line level provider: 1=YES  ; One Line Level provider is most significant: 1=YES  ; At least one line level provider matches the claim level provider: 1=YES  ; There is only one procedure without a line level provider: 0=NO  ;  ; Remove the claim lines associated with the claim level provider.  D REMOVELN  ;  Q  ; A11111   ; Case 32 - This case can never happen!  ; ACTIONS: N/A - Transmit as is.  Q  ; CINIT1    ; Claim level initiation  S IBIEN=$O(INPUT("L-PROV",0))  I IBIEN="" S IBIEN=$O(INPUT("PROVINF",0))  I IBIEN="" S IBIEN=$O(INPUT("LAB/FAC",0))  Q  ; CINIT2    ; Claim level initiation  N MODEX,PRNUMX,PROVX  F MODEX="C","O" D  . S PRNUMX=0 F  S PRNUMX=$O(INPUT("PROVINF",IBIEN,MODEX,PRNUMX)) Q:+PRNUMX=0 D  .. I $G(INPUT("PROVINF",IBIEN,MODEX,PRNUMX))="" Q  .. I INPUT("PROVINF",IBIEN,MODEX,PRNUMX)=INSLEVEL S CMODE(INSLEVEL)=MODEX,CPRNUM(INSLEVEL)=PRNUMX  Q  ; LINIT1(SLC) ; Line level initiation  Q $O(INPUT("L-PROV",IBIEN,SLC))  ; LINIT2    ; Line level initiation  N MODEX,PRNUMX,PROVX  F MODEX="C","O" D  . S PRNUMX=0 F  S PRNUMX=$O(INPUT("L-PROV",IBIEN,SLC,MODEX,PRNUMX)) Q:+PRNUMX=0 D  .. I INPUT("L-PROV",IBIEN,SLC,MODEX,PRNUMX)=INSLEVEL S LMODE(INSLEVEL)=MODEX,LPRNUM(INSLEVEL)=PRNUMX  Q  ; REMOVELN    ; Remove the claim lines associated with the claim level provider.  N MODEX,PRNUMX,PROVX  S SLC=0 F  S SLC=$O(OUTPUT("L-PROV",IBIEN,SLC)) Q:+SLC=0 D  . F MODEX="C","O" D  .. S PRNUMX=0 F  S PRNUMX=$O(OUTPUT("L-PROV",IBIEN,SLC,MODEX,PRNUMX)) Q:+PRNUMX=0 D  ... Q:$G(PROVINFO)=""  ... I $G(OUTPUT("L-PROV",IBIEN,SLC,MODEX,PRNUMX,PRTYPE))=PROVINFO D  .... K OUTPUT("L-PROV",IBIEN,SLC,MODEX,PRNUMX,PRTYPE)  .... I $D(OUTPUT("L-PROV",IBIEN,SLC,MODEX,PRNUMX))=1 K OUTPUT("L-PROV",IBIEN,SLC,MODEX,PRNUMX)  .... I $D(OUTPUT("L-PROV",IBIEN,SLC,MODEX))=1 K OUTPUT("L-PROV",IBIEN,SLC,MODEX)  .... I $D(OUTPUT("L-PROV",IBIEN,SLC))=1 K OUTPUT("L-PROV",IBIEN,SLC)  Q | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCEM03 | | | | | | | | |
| **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** | | | | | | | | | |
| IBCEM03 ;ALB/TMP - 837 EDI RESUBMIT INDIVIDUAL BILL PROCESSING ;17-SEP-96  ;;2.0;INTEGRATED BILLING;\*\*137,199,296,348,349\*\*;21-MAR-94;Build 46  ;;Per VHA Directive 2004-038, this routine should not be modified.  Q  ; BILL2 ; Resubmit a transmitted bill with a new batch #  N DIC,DIR,DIE,DA,DR,IB,IB0,IBDA,IBDA1,IBE,IBSTAT,IBBDA,IBOK,IBNEW,Y,ZTSK,IBTEST  K ^TMP("IBEDI\_TEST\_BATCH",$J)  ;  S DIR("A")="ARE YOU RESUBMITTING CLAIMS FOR TESTING?: ",DIR("B")="NO",DIR(0)="YA" D ^DIR K DIR  I $D(DTOUT)!$D(DUOUT) Q  I +Y S ^TMP("IBEDI\_TEST\_BATCH",$J)=1 ASK N DPTNOFZY S DPTNOFZY=1 ;Suppress PATIENT file fuzzy lookups  S IBTEST=+$G(^TMP("IBEDI\_TEST\_BATCH",$J))  ; Only auth or printed transmittable bill valid for non-test  ; All previously transmitted valid for test  S DIC="^DGCR(399,",DIC(0)="AEMQ",DIC("S")=$S('IBTEST:"I $P($G(^(""TX"")),U,2),$P($G(^(0)),U,13)'="""",""234""[$P($G(^(0)),U,13)",1:"I $O(^IBA(364,""B"",+Y,0))")  I IBTEST S DIC("A")="Select BILL/CLAIMS BILL NUMBER (FOR RESUBMIT AS TEST): "  D ^DIC K DIC  I Y<0 D  Q  . Q:'IBTEST  . I $O(^TMP("IBEDI\_TEST\_BATCH",$J,0)) D  .. M ^TMP("IBRESUBMIT",$J)=^TMP("IBEDI\_TEST\_BATCH",$J)  .. D ONE^IBCE837  . ;  . K ^TMP("IBEDI\_TEST\_BATCH",$J),^TMP("IBRESUBMIT",$J)  ;  S IBIFN=+Y,IBDA=+$$LAST364^IBCEF4(IBIFN),IB0=$G(^IBA(364,IBDA,0)),IBSTAT=$P(IB0,U,3)  ;  I IB0="" W !,"Bill does not exist in BILL TRANSMISSION file" G ASK  I IBTEST,$D(^TMP("IBEDI\_TEST\_BATCH",$J,IBDA)) W !,"Bill already selected for test transmission" G ASK  I $$COBN^IBCEF(IBIFN)=1,IBTEST S IBOK=1 D  G:'IBOK ASK  . S DIR("A")="BILL IS A PRIMARY BILL, ARE YOU SURE YOU WANT TO SEND IT AS A TEST CLAIM?: "  . S DIR("B")="NO",DIR(0)="YA" W ! D ^DIR K DIR  . I Y'=1 S IBOK=0  ;  I 'IBTEST,IBSTAT="X" W !,"Bill is currently awaiting extract - will be submitted with next batch run" G ASK  S IBBDA=+$P(IB0,U,2),IB=$P($G(^IBA(364.1,IBBDA,0)),U,9)  ;  I IB,'IBTEST D  G:'IBOK ASK  . S IBOK=1,ZTSK=IB D STAT^%ZTLOAD  . I ZTSK(0)=0 S DIE="^IBA(364.1,",DA=IBBDA,DR=".09///@" D ^DIE Q  ;Task not scheduled - delete task #  . I "125"[ZTSK(1) W \*7,!,"Cannot resubmit this bill.",!,"This bill's current batch is already ",$S("2"[ZTSK(1):"being resubmitted",1:"scheduled for resubmission")," - Task # is: ",IB,! S IBOK=0  ;  W !  S DIR("A",1)=" Previously In Batch #: "\_$$EXPAND^IBTRE(364,.02,$P(IB0,U,2))  S DIR("A",2)="Bill Transmission Status: "\_$$EXPAND^IBTRE(364,.03,IBSTAT)  S DIR("A",3)=" Status Date: "\_$$FMTE^XLFDT($P(IB0,U,4),2)  S DIR("A",5)=" "  S DIR("A",4)=" Current Bill Status: "\_$$EXPAND^IBTRE(399,.13,$P($G(^DGCR(399,+IBIFN,0)),U,13))  I 'IBTEST,IBSTAT'="P" S DIR("A",11)="WARNING - BILL TRANSMITTED PREVIOUSLY" S:IBSTAT?1"A".E DIR("A",11)=DIR("A",11)\_" & CONFIRMED AS RECEIVED BY "\_$P("AUSTIN^GENTRAN^INTERMEDIARY^CARRIER",U,$TR(IBSTAT,"A")+1)  S DIR("A")="ARE YOU SURE YOU WANT TO RESUBMIT THIS BILL"\_$S('IBTEST:"",1:" AS A TEST CLAIM")\_"?: "  S DIR(0)="YA",DIR("B")="NO"  D ^DIR K DIR  ;  W ! G:'Y ASK  ;  I IBTEST S ^TMP("IBEDI\_TEST\_BATCH",$J,IBDA)="" G ASK  ;  S IBDA1=+$$ADDTBILL^IBCB1(IBIFN) ;Add a new transmit bill record  ;  S Y=$$TX1^IBCB1(IBDA1,1)  ;  I 'Y D  G ASK  . W !,\*7,"An error has occurred ... bill NOT re-submitted!!"  . S DIK="^IBA(364,",DA=IBDA1 D:DA ^DIK  . L -^IBA(364,IBDA)  ;  S IBNEW=$P($G(^IBA(364,+IBDA1,0)),U,2)  ;  ;Update the old transmit bill record  D UPDEDI^IBCEM(IBDA,"R")  ;  W !,"Bill # ",$P($G(^DGCR(399,+IB0,0)),U)," was re-submitted in batch # ",$P($G(^IBA(364.1,+IBNEW,0)),U)  ;  L -^IBA(364,IBDA)  G ASK  ; PRINT1(IBIFN,IBDA,IB364,IBRESUB) ; Print bill, submit manually as resolution  ; for a returned message  ; IBIFN = ien of bill in file 399  ; IBDA = array returned from selection of message  ; IB364 = ien of transmit bill entry in file 364  ; IBRESUB = flag to indicate if bill is being resubmitted via print  ;  N IBAC,IBV,IB399,DFN,ZTSK,PRCASV,IBHOLD,IBTXPRT  W !  I IBIFN="" S IBDA="" G PRINT1Q  S IB399=$G(^DGCR(399,IBIFN,0))  I "34"'[$P(IB399,U,13) W !,\*7,"Bill status must be AUTHORIZED or PRNT/TX to print the bill" S IBDA="" G PRINT1Q  ;  I $P($G(^DGCR(399,IBIFN,"S")),U,14)=DT W !,\*7,"This bill was last printed today. You must wait at least 1 day from the last",!,"print date to print this bill using this function." S IBDA="" D PAUSE^VALM1 G PRINT1Q  ;  S IBV=1,IBAC=4,DFN=$P(IB399,U,2),IBTXPRT=0  M IBHOLD("IBDA")=IBDA  D 4^IBCB1,ENS^%ZISS  M IBDA=IBHOLD("IBDA")  ;  I 'IBTXPRT W !,"Bill was not printed" S IBDA="" G PRINT1Q  ;  D UPDEDI^IBCEM(IB364,"P")  ; PRINT1Q Q  ; SUB1 ; Select bills in ready for extract status to transmit individually  N IB0,IB399,IBDA,IBIFN,IBSEL,IBU,X,Y,DA,DIC,Z,DIR  K ^TMP("IBSELX",$J)  ;  S IBSEL=""  F  D  Q:'IBSEL  . S DIR("S")="I $P(^(0),U,3)=""X"""  . S DIR(0)="PAO^364:AEMQ",DIR("A")="SELECT "\_$S($D(^TMP("IBSELX",$J)):"NEXT ",1:"")\_"BILL TO TRANSMIT: "  . S DIR("?")="ONLY BILLS IN 'READY FOR EXTRACT' STATUS CAN BE TRANSMITTED WITH THIS OPTION"  . D ^DIR K DIR  . I Y'>0 K:Y=U ^TMP("IBSELX",$J) S IBSEL="" Q  . S IBSEL=+Y  . S IBDA=+Y,IB0=$G(^IBA(364,IBDA,0)),IBIFN=+IB0,IBU=$G(^DGCR(399,IBIFN,"U")),IB399=$G(^(0))  . S Z=+$$NEEDMRA^IBEFUNC(IBIFN)  . I '$$TXMT^IBCEF4(IBIFN,.IBNOTX),IBNOTX=2 D  Q  .. W !,$S(Z:"MRA",1:"EDI")\_" TRANSMISSION PARAMETER HAS BEEN TURNED OFF",!!,"BILL CANNOT BE SELECTED"  . ;  . W !  . S DIR("A",1)=" YOU HAVE SELECTED BILL #: "\_$P(IB399,U)\_" ("\_$S($$INPAT^IBCEF(IBIFN):"INPATIENT",1:"OUTPATIENT")\_"/"\_$S($$FT^IBCEF(IBIFN)=3:"UB-04",1:"CMS-1500")\_" FORMAT)"  . S DIR("A",2)=" PATIENT NAME: "\_$E($P($G(^DPT(+$P(IB399,U,2),0)),U)\_$J("",28),1,28)\_" SSN: "\_$P($G(^DPT(+$P(IB399,U,2),0)),U,9)  . S DIR("A",3)=" CARE DATE(S): "\_$$EXPAND^IBTRE(399,151,$P(IBU,U))\_" - "\_$$EXPAND^IBTRE(399,152,$P(IBU,U,2))  . S DIR("A",4)="'READY TO EXTRACT' STATUS DATE: "\_$$EXPAND^IBTRE(364,.04,$P(IB0,U,4))  . S DIR("?",1)=" "  . S DIR("A",5)=" ",DIR("?")="IF THIS IS THE BILL YOU WANT TO TRANSMIT, RESPOND YES, OTHERWISE, RESPOND NO"  . S DIR("A")="ARE YOU SURE THIS IS THE CORRECT BILL TO TRANSMIT?: "  . S DIR(0)="YAO",DIR("B")="NO" D ^DIR K DIR W !  . I Y'=1 W !,"BILL NOT SELECTED" Q  . ;  . S ^TMP("IBSELX",$J,IBDA)=""  ;  I '$O(^TMP("IBSELX",$J,0)) G SUB1Q  ;  W !,"Bills to be transmitted: "  S Z=0 F  S Z=$O(^TMP("IBSELX",$J,Z)) Q:'Z  W !,?8,$P($G(^DGCR(399,+$G(^IBA(364,Z,0)),0)),U)  W !  S DIR("A")="OK TO TRANSMIT NOW?: ",DIR(0)="YA0",DIR("B")="NO" D ^DIR K DIR  G:Y'=1 SUB1Q  W !  S ^TMP("IBSELX",$J)=0  D ONE^IBCE837  W !,"BILL(s) TRANSMITTED ... BATCH #(s): "  S Z=0 F  S Z=$O(^TMP("IBCE-BATCH",$J,Z)) Q:'Z  W Z,$S($O(^(Z)):", ",1:"")  I '$O(^TMP("IBCE-BATCH",$J,0)) W !,"NO BILL(S) TRANSMITTED - CHECK ALERTS/MAIL FOR DETAILS"  ; SUB1Q D PAUSE^VALM1  K ^TMP("IBSELX",$J),^TMP("IBCE-BATCH",$J)  Q  ; | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCEM03 ;ALB/TMP - 837 EDI RESUBMIT INDIVIDUAL BILL PROCESSING ;17-SEP-96  ;;2.0;INTEGRATED BILLING;\*\*137,199,296,348,349,**592**\*\*;21-MAR-94;Build 46  ;;Per VHA Directive 2004-038, this routine should not be modified.  Q  ; BILL2 ; Resubmit a transmitted bill with a new batch #  N DIC,DIR,DIE,DA,DR,IB,IB0,IBDA,IBDA1,IBE,IBSTAT,IBBDA,IBOK,IBNEW,Y,ZTSK,IBTEST  K ^TMP("IBEDI\_TEST\_BATCH",$J)  ;  S DIR("A")="ARE YOU RESUBMITTING CLAIMS FOR TESTING?: ",DIR("B")="NO",DIR(0)="YA" D ^DIR K DIR  I $D(DTOUT)!$D(DUOUT) Q  I +Y S ^TMP("IBEDI\_TEST\_BATCH",$J)=1 ASK N DPTNOFZY S DPTNOFZY=1 ;Suppress PATIENT file fuzzy lookups  S IBTEST=+$G(^TMP("IBEDI\_TEST\_BATCH",$J))  ; Only auth or printed transmittable bill valid for non-test  ; All previously transmitted valid for test  S DIC="^DGCR(399,",DIC(0)="AEMQ",DIC("S")=$S('IBTEST:"I $P($G(^(""TX"")),U,2),$P($G(^(0)),U,13)'="""",""234""[$P($G(^(0)),U,13)",1:"I $O(^IBA(364,""B"",+Y,0))")  I IBTEST S DIC("A")="Select BILL/CLAIMS BILL NUMBER (FOR RESUBMIT AS TEST): "  D ^DIC K DIC  I Y<0 D  Q  . Q:'IBTEST  . I $O(^TMP("IBEDI\_TEST\_BATCH",$J,0)) D  .. M ^TMP("IBRESUBMIT",$J)=^TMP("IBEDI\_TEST\_BATCH",$J)  .. D ONE^IBCE837  . ;  . K ^TMP("IBEDI\_TEST\_BATCH",$J),^TMP("IBRESUBMIT",$J)  ;  S IBIFN=+Y,IBDA=+$$LAST364^IBCEF4(IBIFN),IB0=$G(^IBA(364,IBDA,0)),IBSTAT=$P(IB0,U,3)  ;  I IB0="" W !,"Bill does not exist in BILL TRANSMISSION file" G ASK  I IBTEST,$D(^TMP("IBEDI\_TEST\_BATCH",$J,IBDA)) W !,"Bill already selected for test transmission" G ASK  I $$COBN^IBCEF(IBIFN)=1,IBTEST S IBOK=1 D  G:'IBOK ASK  . S DIR("A")="BILL IS A PRIMARY BILL, ARE YOU SURE YOU WANT TO SEND IT AS A TEST CLAIM?: "  . S DIR("B")="NO",DIR(0)="YA" W ! D ^DIR K DIR  . I Y'=1 S IBOK=0  ;  I 'IBTEST,IBSTAT="X" W !,"Bill is currently awaiting extract - will be submitted with next batch run" G ASK  S IBBDA=+$P(IB0,U,2),IB=$P($G(^IBA(364.1,IBBDA,0)),U,9)  ;  I IB,'IBTEST D  G:'IBOK ASK  . S IBOK=1,ZTSK=IB D STAT^%ZTLOAD  . I ZTSK(0)=0 S DIE="^IBA(364.1,",DA=IBBDA,DR=".09///@" D ^DIE Q  ;Task not scheduled - delete task #  . I "125"[ZTSK(1) W \*7,!,"Cannot resubmit this bill.",!,"This bill's current batch is already ",$S("2"[ZTSK(1):"being resubmitted",1:"scheduled for resubmission")," - Task # is: ",IB,! S IBOK=0  ;  W !  S DIR("A",1)=" Previously In Batch #: "\_$$EXPAND^IBTRE(364,.02,$P(IB0,U,2))  S DIR("A",2)="Bill Transmission Status: "\_$$EXPAND^IBTRE(364,.03,IBSTAT)  S DIR("A",3)=" Status Date: "\_$$FMTE^XLFDT($P(IB0,U,4),2)  S DIR("A",5)=" "  S DIR("A",4)=" Current Bill Status: "\_$$EXPAND^IBTRE(399,.13,$P($G(^DGCR(399,+IBIFN,0)),U,13))  I 'IBTEST,IBSTAT'="P" S DIR("A",11)="WARNING - BILL TRANSMITTED PREVIOUSLY" S:IBSTAT?1"A".E DIR("A",11)=DIR("A",11)\_" & CONFIRMED AS RECEIVED BY "\_$P("AUSTIN^GENTRAN^INTERMEDIARY^CARRIER",U,$TR(IBSTAT,"A")+1)  S DIR("A")="ARE YOU SURE YOU WANT TO RESUBMIT THIS BILL"\_$S('IBTEST:"",1:" AS A TEST CLAIM")\_"?: "  S DIR(0)="YA",DIR("B")="NO"  D ^DIR K DIR  ;  W ! G:'Y ASK  ;  I IBTEST S ^TMP("IBEDI\_TEST\_BATCH",$J,IBDA)="" G ASK  ;  S IBDA1=+$$ADDTBILL^IBCB1(IBIFN) ;Add a new transmit bill record  ;  S Y=$$TX1^IBCB1(IBDA1,1)  ;  I 'Y D  G ASK  . W !,\*7,"An error has occurred ... bill NOT re-submitted!!"  . S DIK="^IBA(364,",DA=IBDA1 D:DA ^DIK  . L -^IBA(364,IBDA)  ;  S IBNEW=$P($G(^IBA(364,+IBDA1,0)),U,2)  ;  ;Update the old transmit bill record  D UPDEDI^IBCEM(IBDA,"R")  ;  W !,"Bill # ",$P($G(^DGCR(399,+IB0,0)),U)," was re-submitted in batch # ",$P($G(^IBA(364.1,+IBNEW,0)),U)  ;  L -^IBA(364,IBDA)  G ASK  ; PRINT1(IBIFN,IBDA,IB364,IBRESUB) ; Print bill, submit manually as resolution  ; for a returned message  ; IBIFN = ien of bill in file 399  ; IBDA = array returned from selection of message  ; IB364 = ien of transmit bill entry in file 364  ; IBRESUB = flag to indicate if bill is being resubmitted via print  ;  N IBAC,IBV,IB399,DFN,ZTSK,PRCASV,IBHOLD,IBTXPRT  W !  I IBIFN="" S IBDA="" G PRINT1Q  S IB399=$G(^DGCR(399,IBIFN,0))  I "34"'[$P(IB399,U,13) W !,\*7,"Bill status must be AUTHORIZED or PRNT/TX to print the bill" S IBDA="" G PRINT1Q  ;  I $P($G(^DGCR(399,IBIFN,"S")),U,14)=DT W !,\*7,"This bill was last printed today. You must wait at least 1 day from the last",!,"print date to print this bill using this function." S IBDA="" D PAUSE^VALM1 G PRINT1Q  ;  S IBV=1,IBAC=4,DFN=$P(IB399,U,2),IBTXPRT=0  M IBHOLD("IBDA")=IBDA  D 4^IBCB1,ENS^%ZISS  M IBDA=IBHOLD("IBDA")  ;  I 'IBTXPRT W !,"Bill was not printed" S IBDA="" G PRINT1Q  ;  D UPDEDI^IBCEM(IB364,"P")  ; PRINT1Q Q  ; SUB1 ; Select bills in ready for extract status to transmit individually  N IB0,IB399,IBDA,IBIFN,IBSEL,IBU,X,Y,DA,DIC,Z,DIR  K ^TMP("IBSELX",$J)  ;  S IBSEL=""  F  D  Q:'IBSEL  . S DIR("S")="I $P(^(0),U,3)=""X"""  . S DIR(0)="PAO^364:AEMQ",DIR("A")="SELECT "\_$S($D(^TMP("IBSELX",$J)):"NEXT ",1:"")\_"BILL TO TRANSMIT: "  . S DIR("?")="ONLY BILLS IN 'READY FOR EXTRACT' STATUS CAN BE TRANSMITTED WITH THIS OPTION"  . D ^DIR K DIR  . I Y'>0 K:Y=U ^TMP("IBSELX",$J) S IBSEL="" Q  . S IBSEL=+Y  . S IBDA=+Y,IB0=$G(^IBA(364,IBDA,0)),IBIFN=+IB0,IBU=$G(^DGCR(399,IBIFN,"U")),IB399=$G(^(0))  . S Z=+$$NEEDMRA^IBEFUNC(IBIFN)  . I '$$TXMT^IBCEF4(IBIFN,.IBNOTX),IBNOTX=2 D  Q  .. W !,$S(Z:"MRA",1:"EDI")\_" TRANSMISSION PARAMETER HAS BEEN TURNED OFF",!!,"BILL CANNOT BE SELECTED"  . ;  . W !  **. ;JWS;IB\*2.0\*592; added form #7 J430D to display**  . S DIR("A",1)=" YOU HAVE SELECTED BILL #: "\_$P(IB399,U)\_" ("\_$S($$INPAT^IBCEF(IBIFN):"INPATIENT",1:"OUTPATIENT")\_"/"\_$S($$FT^IBCEF(IBIFN)=3:"UB-04**",$$FT^IBCEF(IBIFN)=7:"J430D",**1:"CMS-1500")\_" FORMAT)"  . S DIR("A",2)=" PATIENT NAME: "\_$E($P($G(^DPT(+$P(IB399,U,2),0)),U)\_$J("",28),1,28)\_" SSN: "\_$P($G(^DPT(+$P(IB399,U,2),0)),U,9)  . S DIR("A",3)=" CARE DATE(S): "\_$$EXPAND^IBTRE(399,151,$P(IBU,U))\_" - "\_$$EXPAND^IBTRE(399,152,$P(IBU,U,2))  . S DIR("A",4)="'READY TO EXTRACT' STATUS DATE: "\_$$EXPAND^IBTRE(364,.04,$P(IB0,U,4))  . S DIR("?",1)=" "  . S DIR("A",5)=" ",DIR("?")="IF THIS IS THE BILL YOU WANT TO TRANSMIT, RESPOND YES, OTHERWISE, RESPOND NO"  . S DIR("A")="ARE YOU SURE THIS IS THE CORRECT BILL TO TRANSMIT?: "  . S DIR(0)="YAO",DIR("B")="NO" D ^DIR K DIR W !  . I Y'=1 W !,"BILL NOT SELECTED" Q  . ;  . S ^TMP("IBSELX",$J,IBDA)=""  ;  I '$O(^TMP("IBSELX",$J,0)) G SUB1Q  ;  W !,"Bills to be transmitted: "  S Z=0 F  S Z=$O(^TMP("IBSELX",$J,Z)) Q:'Z  W !,?8,$P($G(^DGCR(399,+$G(^IBA(364,Z,0)),0)),U)  W !  S DIR("A")="OK TO TRANSMIT NOW?: ",DIR(0)="YA0",DIR("B")="NO" D ^DIR K DIR  G:Y'=1 SUB1Q  W !  S ^TMP("IBSELX",$J)=0  D ONE^IBCE837  W !,"BILL(s) TRANSMITTED ... BATCH #(s): "  S Z=0 F  S Z=$O(^TMP("IBCE-BATCH",$J,Z)) Q:'Z  W Z,$S($O(^(Z)):", ",1:"")  I '$O(^TMP("IBCE-BATCH",$J,0)) W !,"NO BILL(S) TRANSMITTED - CHECK ALERTS/MAIL FOR DETAILS"  ; SUB1Q D PAUSE^VALM1  K ^TMP("IBSELX",$J),^TMP("IBCE-BATCH",$J)  Q  ; | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCEMU4 | | | | | | | | |
| **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** | | | | | | | | | |
| IBCEMU4 ;ALB/ESG - MRA UTILITIES ;25-OCT-2004  ;;2.0;INTEGRATED BILLING;\*\*288,432,447\*\*;21-MAR-94;Build 80  ;;Per VHA Directive 2004-038, this routine should not be modified.  ;  Q  ; DENDUP(IBEOB,IBMRANOT) ; Denied for Duplicate Function ;WCJ IB\*2.0\*432  ; Function returns true if MRA is Denied AND Reason code 18 is present (Duplicate claim/service)  NEW IBX,IBM,LINE,DUP,ADJ  S IBX=0,IBM=$G(^IBM(361.1,+$G(IBEOB),0))  I '$G(IBMRANOT),$P(IBM,U,4)'=1 G DENDUPX    ; not an MRA ;WCJ IB\*2.0\*432  I $G(IBMRANOT),$P(IBM,U,4)'=0 G DENDUPX    ; not an EOB ;WCJ IB\*2.0\*432  I $P(IBM,U,13)'=2 G DENDUPX   ; not Denied  ;  ; check line item adjustments for reason code 18  S LINE=0,DUP=0  F  S LINE=$O(^IBM(361.1,IBEOB,15,LINE)) Q:'LINE  D  Q:DUP  . S ADJ=0  . F  S ADJ=$O(^IBM(361.1,IBEOB,15,LINE,1,ADJ)) Q:'ADJ  D  Q:DUP  .. I $D(^IBM(361.1,IBEOB,15,LINE,1,ADJ,1,"B",18)) S DUP=1 Q  .. Q  . Q  ;  I DUP S IBX=1 DENDUPX ;  Q IBX  ;  ; the remaining functions are all new w/ IB\*2.0\*447 and have to do with calculating  ; different amounts based on percentages stored in the effective date multiple of  ; the TYPE OF PLAN file (#355.1) for Medicare Supplemental plans  ; MSPRE(IBIFN,IBEXF,IBTYPLAN) ; Medicare supplemental PR and Excess calculations  ; determine PR amount in order to calculate balance due after medicare for secondary/tertiary  ; if type of plan is a Medicare supplemental or EGHP plan secondary to Medicare, PR   ; calculations are determined based on the effective date multiple in the TYPE OF PLAN file  ; and may or may not included Excess charges (CO-45), based on Plan Type.  ; need to pass in:  ; IBIFN (REQUIRED) = claim ien  ; IBEXF = Excess Flag, set to 1 if NOT to include excess charges in calculation but to  ; return "e" (IBE) for excess indicator if plan allows excess and there are   ; excess charges. Used by PR column of MRW screen to show PR without excess  ; amounts included in calculation.  ; IBTYPLAN = ien in TYPE OF PLAN file (355.1)  ; returns "" if no effective date for type of plan to calculate on  ;  N IBFRMTYP,IBPNCAT,IBINPAT,IBMGBD,IBEOB,LNLVL,EOBADJ,IBPCE,IBEDT,IBE,IBTOT  Q:$G(IBIFN)="" ""  S:$G(IBTYPLAN)="" IBTYPLAN=$$TYPLN(IBIFN)  S IBEDT=$$MSEDT(IBIFN,IBTYPLAN) Q:IBEDT="" ""  S IBINPAT=$$INPAT^IBCEF(IBIFN) ;Inpat/Outpat Flag  S IBFRMTYP=$P($G(^DGCR(399,IBIFN,0)),U,19) ; Form Type 2=1500, 3=UB  ; plan category - PART A is Inpatient Institutional, B is all Outpatient and Inpatient Professional  S IBPNCAT="B"  I IBINPAT=1,IBFRMTYP=3 S IBPNCAT="A" Q:IBPNCAT="" ""  ; Medicare supplemental plan Offset amount = total charges - what medicare secondary plan will pay  ; so balance due = whatever medicare secondary will pay  ;  ; plan category - PART A =1st piece of AEDT Index, B =2nd  S IBPCE=$S(IBPNCAT="B":2,1:1)  S IBMGBD=0,IBEOB=0  F  S IBEOB=$O(^IBM(361.1,"B",IBIFN,IBEOB)) Q:'IBEOB  D  .N I  .F I=0,1,2 S IBEOB(I)=$G(^IBM(361.1,IBEOB,I))  .I $P(IBEOB(0),U,4)'=1 Q  ;make sure it's an MRA  .;  .; Handle CMS-1500 Form Type and UB Outpatient:  .I IBFRMTYP=2!('IBINPAT) D  Q  ..; calculate Medicare unpaid amount from line-level (outpatient)  ..S LNLVL=0 F  S LNLVL=$O(^IBM(361.1,IBEOB,15,LNLVL)) Q:'LNLVL  D  ;  ...K EOBADJ  ...M EOBADJ=^IBM(361.1,IBEOB,15,LNLVL,1)  ...; Total up the Medicare Contract Adjustment across ALL Service Lines to find  ...; Medicare supplemental Balance Due  ...S IBTOT=$$CALC(.EOBADJ,IBTYPLAN,IBPCE,IBEDT,$G(IBEXF)),IBE=$P(IBTOT,U,2)  ...S IBMGBD=$G(IBMGBD)+$P(IBTOT,U)  .;  .; Handle Inpatient UB Form Type Next: Calculate from Claim level data  .K EOBADJ  .M EOBADJ=^IBM(361.1,IBEOB,10)  .S IBTOT=$$CALC(.EOBADJ,IBTYPLAN,IBPCE,IBEDT,$G(IBEXF)),IBE=$P(IBTOT,U,2)  .S IBMGBD=$G(IBMGBD)+$P(IBTOT,U)  Q IBMGBD\_$G(IBE)  ; CALC(EOBADJ,IBTYPLAN,IBPCE,IBEDT,IBEXF) ; FUNCTION - Calculate Medicare Supplemental Balance due  ; Sums up Amounts on ALL Reason Codes under ALL Group Codes = 'PR' and CO/Reason code=45.  ; If those reason codes have an entry in the effective date mutliple, multiples that  ; reason amount by the % the Type of plan will pay. If no entry, assume 100% payment for PR.  ; any other Group and reason codes would be 0%.   ; Adds up all those sums and returns that value as the total PR&CO the Medicare   ; Supplemental plan will pay.  ;  ; Input EOBADJ = Array of Group Codes & Reason Codes from either the Claim   ; Level (10) or Service Line Level (15) of EOB file (#361.1)  ; IBTYPLAN = ien in TYPE OF PLAN file   ; IBPCE = 2 for PART A, 3 for PART B - REQUIRED  ; IBEDT = effective date of plan rates   ; IBEXF = Excess Flag, set to 1 if NOT to include excess charges in calculation but to  ; return "e" for excess indicator if plan allows excess and there are excess   ; charges. Used by PR column of MRW screen to show PR without excess  ; amounts included in calculation.  ; Output amount that Medicare supplemental plan will pay  ;  N GRPLVL,RSNLVL,RSNAMT,MCA,GRPCD,RSNCD,RSN0,CALC,IBIND  Q:$G(IBPCE)="" ""  S:$G(IBTYPLAN)="" IBTYPLAN=$$TYPLN(IBIFN)  I $G(IBEDT)="" S IBEDT=$$MSEDT(IBIFN,IBTYPLAN) Q:IBEDT="" ""  S (GRPLVL,MCA)=0  F  S GRPLVL=$O(EOBADJ(GRPLVL)) Q:'GRPLVL  D   .S GRPCD=$P($G(EOBADJ(GRPLVL,0)),U)  .; For now they want to calculate all PR but only apply %age calcs to PR-1,2 & 3   .I GRPCD'="PR" Q:'$D(^IBE(355.1,IBTYPLAN,14,"AEDT",IBEDT,GRPCD))  .S RSNLVL=0  .F  S RSNLVL=$O(EOBADJ(GRPLVL,1,RSNLVL)) Q:'RSNLVL  D  ;  ..S RSN0=$G(EOBADJ(GRPLVL,1,RSNLVL,0)),RSNAMT=$P(RSN0,U,2),RSNCD=$P(RSN0,U)  ..I GRPCD="PR",RSNCD="AAA" Q  ; ignore PR-AAA  ..; For now they want to calculate all PR but only apply %age calcs to PR-1,2 & 3  ..I GRPCD="PR","1^2^3"'[RSNCD,'$D(^IBE(355.1,IBTYPLAN,14,"AEDT",IBEDT,GRPCD,RSNCD)) S MCA=MCA+RSNAMT Q  ..Q:'$D(^IBE(355.1,IBTYPLAN,14,"AEDT",IBEDT,GRPCD,RSNCD))  ..; if there is an entry in the effective date multiple for this grp/rsn code use it to calculate amount for PART A and B.  ..; for MRW, don't add up excess charges if IBEXF=1, just send back an "e" indicator to alert user of excess   ..I $G(IBEXF)=1,GRPCD="CO",RSNCD=45,$P($G(^IBE(355.1,IBTYPLAN,14,"AEDT",IBEDT,GRPCD,RSNCD)),U,IBPCE)>0 S IBIND="e" Q  ..S CALC=$P($G(^IBE(355.1,IBTYPLAN,14,"AEDT",IBEDT,GRPCD,RSNCD)),U,IBPCE)/100  ..S MCA=MCA+(RSNAMT\*CALC)  Q MCA\_U\_$G(IBIND)  ; MSEDT(IBIFN,IBTYPLAN) ; does this claim's TYPE OF PLAN have an effective date multiple on or before the  ; claim 'statement covers from' date  ; IBIFN = claim ien - REQUIRED  ; IBTYPLAN = Type of Plan ien  ; returns eff.date calculation multiple to use or null  ; called from SKIP^IBCCCB, BLD^IBCECOB1, TOT^IBCECOB2, CRIT^IBCEMQC, & SECOND^IBCEMSR  ;  ; IB\*2.0\*447: the below quit statement has been added because CBO has decided not to implement  ; these changes with patch 447 after all. Once a long-term maintenance plan for the plan type  ; calculations can be worked out and CBO is ready to implement the special calculations, the  ; below quit statement and these comments should be removed and the type of plan special calculations   ; will immediately take effect. For now, returning a null will allow existing code to bypass   ; the special calculation table in file 355.1 and calculate everything as 100% of Patient Responsibility (PR).  Q ""  ;  N IBSVDT  Q:$G(IBIFN)="" ""  S:$G(IBTYPLAN)="" IBTYPLAN=+$$TYPLN(IBIFN)  S IBSVDT=+$P($G(^DGCR(399,IBIFN,"U")),U)  Q:$D(^IBE(355.1,IBTYPLAN,14,"B",IBSVDT)) IBSVDT  Q $O(^IBE(355.1,IBTYPLAN,14,"B",IBSVDT),-1)  ; TYPLN(IBIFN) ; find type of plan for claim  ; IBIFN = claim ien - REQUIRED  ; returns ien from file 355.1 or null if none found  ;  Q:$G(IBIFN)="" ""  N IBCOBN,IBGRPNO  S IBCOBN=$$COBN^IBCEF(IBIFN)+1 ;find next payer  S IBGRPNO=+$P($G(^DGCR(399,IBIFN,"I"\_IBCOBN)),U,18) ; group plan number  Q $P($G(^IBA(355.3,IBGRPNO,0)),U,9) ; type of plan - IEN  ; | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCEMU4 ;ALB/ESG - MRA UTILITIES ;25-OCT-2004  ;;2.0;INTEGRATED BILLING;\*\*288,432,447,**592\***\*;21-MAR-94;Build 80  ;;Per VHA Directive 2004-038, this routine should not be modified.  ;  Q  ; DENDUP(IBEOB,IBMRANOT) ; Denied for Duplicate Function ;WCJ IB\*2.0\*432  ; Function returns true if MRA is Denied AND Reason code 18 is present (Duplicate claim/service)  NEW IBX,IBM,LINE,DUP,ADJ  S IBX=0,IBM=$G(^IBM(361.1,+$G(IBEOB),0))  I '$G(IBMRANOT),$P(IBM,U,4)'=1 G DENDUPX    ; not an MRA ;WCJ IB\*2.0\*432  I $G(IBMRANOT),$P(IBM,U,4)'=0 G DENDUPX    ; not an EOB ;WCJ IB\*2.0\*432  I $P(IBM,U,13)'=2 G DENDUPX   ; not Denied  ;  ; check line item adjustments for reason code 18  S LINE=0,DUP=0  F  S LINE=$O(^IBM(361.1,IBEOB,15,LINE)) Q:'LINE  D  Q:DUP  . S ADJ=0  . F  S ADJ=$O(^IBM(361.1,IBEOB,15,LINE,1,ADJ)) Q:'ADJ  D  Q:DUP  .. I $D(^IBM(361.1,IBEOB,15,LINE,1,ADJ,1,"B",18)) S DUP=1 Q  .. Q  . Q  ;  I DUP S IBX=1 DENDUPX ;  Q IBX  ;  ; the remaining functions are all new w/ IB\*2.0\*447 and have to do with calculating  ; different amounts based on percentages stored in the effective date multiple of  ; the TYPE OF PLAN file (#355.1) for Medicare Supplemental plans  ; MSPRE(IBIFN,IBEXF,IBTYPLAN) ; Medicare supplemental PR and Excess calculations  ; determine PR amount in order to calculate balance due after medicare for secondary/tertiary  ; if type of plan is a Medicare supplemental or EGHP plan secondary to Medicare, PR   ; calculations are determined based on the effective date multiple in the TYPE OF PLAN file  ; and may or may not included Excess charges (CO-45), based on Plan Type.  ; need to pass in:  ; IBIFN (REQUIRED) = claim ien  ; IBEXF = Excess Flag, set to 1 if NOT to include excess charges in calculation but to  ; return "e" (IBE) for excess indicator if plan allows excess and there are   ; excess charges. Used by PR column of MRW screen to show PR without excess  ; amounts included in calculation.  ; IBTYPLAN = ien in TYPE OF PLAN file (355.1)  ; returns "" if no effective date for type of plan to calculate on  ;  N IBFRMTYP,IBPNCAT,IBINPAT,IBMGBD,IBEOB,LNLVL,EOBADJ,IBPCE,IBEDT,IBE,IBTOT  Q:$G(IBIFN)="" ""  S:$G(IBTYPLAN)="" IBTYPLAN=$$TYPLN(IBIFN)  S IBEDT=$$MSEDT(IBIFN,IBTYPLAN) Q:IBEDT="" ""  S IBINPAT=$$INPAT^IBCEF(IBIFN) ;Inpat/Outpat Flag  S IBFRMTYP=$P($G(^DGCR(399,IBIFN,0)),U,19) ; Form Type 2=1500, 3=UB, **7=J430D ;JRA IB\*2.0\*592 Add Dental form 7**  ; plan category - PART A is Inpatient Institutional, B is all Outpatient and Inpatient Professional  S IBPNCAT="B"  I IBINPAT=1,IBFRMTYP=3 S IBPNCAT="A" Q:IBPNCAT="" ""  ; Medicare supplemental plan Offset amount = total charges - what medicare secondary plan will pay  ; so balance due = whatever medicare secondary will pay  ;  ; plan category - PART A =1st piece of AEDT Index, B =2nd  S IBPCE=$S(IBPNCAT="B":2,1:1)  S IBMGBD=0,IBEOB=0  F  S IBEOB=$O(^IBM(361.1,"B",IBIFN,IBEOB)) Q:'IBEOB  D  .N I  .F I=0,1,2 S IBEOB(I)=$G(^IBM(361.1,IBEOB,I))  .I $P(IBEOB(0),U,4)'=1 Q  ;make sure it's an MRA  .;  .; Handle CMS-1500 Form Type and UB Outpatient:  **.;JRA IB\*2.0\*592 Do the same for Dental J430D as for CMS-1500**  **.;I** IBFRMTYP=2!('IBINPAT) D Q **;JRA IB\*2.0\*592 ';'**  .I IBFRMTYP=2**!(IBFRMTYP=7**!('IBINPAT)) D  **Q  ;JRA IB\*2.0\*592**  ..; calculate Medicare unpaid amount from line-level (outpatient)  ..S LNLVL=0 F  S LNLVL=$O(^IBM(361.1,IBEOB,15,LNLVL)) Q:'LNLVL  D  ;  ...K EOBADJ  ...M EOBADJ=^IBM(361.1,IBEOB,15,LNLVL,1)  ...; Total up the Medicare Contract Adjustment across ALL Service Lines to find  ...; Medicare supplemental Balance Due  ...S IBTOT=$$CALC(.EOBADJ,IBTYPLAN,IBPCE,IBEDT,$G(IBEXF)),IBE=$P(IBTOT,U,2)  ...S IBMGBD=$G(IBMGBD)+$P(IBTOT,U)  .;  .; Handle Inpatient UB Form Type Next: Calculate from Claim level data  .K EOBADJ  .M EOBADJ=^IBM(361.1,IBEOB,10)  .S IBTOT=$$CALC(.EOBADJ,IBTYPLAN,IBPCE,IBEDT,$G(IBEXF)),IBE=$P(IBTOT,U,2)  .S IBMGBD=$G(IBMGBD)+$P(IBTOT,U)  Q IBMGBD\_$G(IBE)  ; CALC(EOBADJ,IBTYPLAN,IBPCE,IBEDT,IBEXF) ; FUNCTION - Calculate Medicare Supplemental Balance due  ; Sums up Amounts on ALL Reason Codes under ALL Group Codes = 'PR' and CO/Reason code=45.  ; If those reason codes have an entry in the effective date mutliple, multiples that  ; reason amount by the % the Type of plan will pay. If no entry, assume 100% payment for PR.  ; any other Group and reason codes would be 0%.   ; Adds up all those sums and returns that value as the total PR&CO the Medicare   ; Supplemental plan will pay.  ;  ; Input EOBADJ = Array of Group Codes & Reason Codes from either the Claim   ; Level (10) or Service Line Level (15) of EOB file (#361.1)  ; IBTYPLAN = ien in TYPE OF PLAN file   ; IBPCE = 2 for PART A, 3 for PART B - REQUIRED  ; IBEDT = effective date of plan rates   ; IBEXF = Excess Flag, set to 1 if NOT to include excess charges in calculation but to  ; return "e" for excess indicator if plan allows excess and there are excess   ; charges. Used by PR column of MRW screen to show PR without excess  ; amounts included in calculation.  ; Output amount that Medicare supplemental plan will pay  ;  N GRPLVL,RSNLVL,RSNAMT,MCA,GRPCD,RSNCD,RSN0,CALC,IBIND  Q:$G(IBPCE)="" ""  S:$G(IBTYPLAN)="" IBTYPLAN=$$TYPLN(IBIFN)  I $G(IBEDT)="" S IBEDT=$$MSEDT(IBIFN,IBTYPLAN) Q:IBEDT="" ""  S (GRPLVL,MCA)=0  F  S GRPLVL=$O(EOBADJ(GRPLVL)) Q:'GRPLVL  D   .S GRPCD=$P($G(EOBADJ(GRPLVL,0)),U)  .; For now they want to calculate all PR but only apply %age calcs to PR-1,2 & 3   .I GRPCD'="PR" Q:'$D(^IBE(355.1,IBTYPLAN,14,"AEDT",IBEDT,GRPCD))  .S RSNLVL=0  .F  S RSNLVL=$O(EOBADJ(GRPLVL,1,RSNLVL)) Q:'RSNLVL  D  ;  ..S RSN0=$G(EOBADJ(GRPLVL,1,RSNLVL,0)),RSNAMT=$P(RSN0,U,2),RSNCD=$P(RSN0,U)  ..I GRPCD="PR",RSNCD="AAA" Q  ; ignore PR-AAA  ..; For now they want to calculate all PR but only apply %age calcs to PR-1,2 & 3  ..I GRPCD="PR","1^2^3"'[RSNCD,'$D(^IBE(355.1,IBTYPLAN,14,"AEDT",IBEDT,GRPCD,RSNCD)) S MCA=MCA+RSNAMT Q  ..Q:'$D(^IBE(355.1,IBTYPLAN,14,"AEDT",IBEDT,GRPCD,RSNCD))  ..; if there is an entry in the effective date multiple for this grp/rsn code use it to calculate amount for PART A and B.  ..; for MRW, don't add up excess charges if IBEXF=1, just send back an "e" indicator to alert user of excess   ..I $G(IBEXF)=1,GRPCD="CO",RSNCD=45,$P($G(^IBE(355.1,IBTYPLAN,14,"AEDT",IBEDT,GRPCD,RSNCD)),U,IBPCE)>0 S IBIND="e" Q  ..S CALC=$P($G(^IBE(355.1,IBTYPLAN,14,"AEDT",IBEDT,GRPCD,RSNCD)),U,IBPCE)/100  ..S MCA=MCA+(RSNAMT\*CALC)  Q MCA\_U\_$G(IBIND)  ; MSEDT(IBIFN,IBTYPLAN) ; does this claim's TYPE OF PLAN have an effective date multiple on or before the  ; claim 'statement covers from' date  ; IBIFN = claim ien - REQUIRED  ; IBTYPLAN = Type of Plan ien  ; returns eff.date calculation multiple to use or null  ; called from SKIP^IBCCCB, BLD^IBCECOB1, TOT^IBCECOB2, CRIT^IBCEMQC, & SECOND^IBCEMSR  ;  ; IB\*2.0\*447: the below quit statement has been added because CBO has decided not to implement  ; these changes with patch 447 after all. Once a long-term maintenance plan for the plan type  ; calculations can be worked out and CBO is ready to implement the special calculations, the  ; below quit statement and these comments should be removed and the type of plan special calculations   ; will immediately take effect. For now, returning a null will allow existing code to bypass   ; the special calculation table in file 355.1 and calculate everything as 100% of Patient Responsibility (PR).  Q ""  ;  N IBSVDT  Q:$G(IBIFN)="" ""  S:$G(IBTYPLAN)="" IBTYPLAN=+$$TYPLN(IBIFN)  S IBSVDT=+$P($G(^DGCR(399,IBIFN,"U")),U)  Q:$D(^IBE(355.1,IBTYPLAN,14,"B",IBSVDT)) IBSVDT  Q $O(^IBE(355.1,IBTYPLAN,14,"B",IBSVDT),-1)  ; TYPLN(IBIFN) ; find type of plan for claim  ; IBIFN = claim ien - REQUIRED  ; returns ien from file 355.1 or null if none found  ;  Q:$G(IBIFN)="" ""  N IBCOBN,IBGRPNO  S IBCOBN=$$COBN^IBCEF(IBIFN)+1 ;find next payer  S IBGRPNO=+$P($G(^DGCR(399,IBIFN,"I"\_IBCOBN)),U,18) ; group plan number  Q $P($G(^IBA(355.3,IBGRPNO,0)),U,9) ; type of plan - IEN  ; | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCEP | | | | | | | | |
| **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** | | | | | | | | | |
| IBCEP ;ALB/TMP - Functions for PROVIDER ID MAINT - INS CO PARAMS ;11-02-00  ;;2.0;INTEGRATED BILLING;\*\*137,232,320,348,349\*\*;21-MAR-94;Build 46  ;;Per VHA Directive 2004-038, this routine should not be modified.  ; EN ; -- main entry point for IBCE PRV INS PARAMS  N IBINS,IBCUINC ; Variable should be available throughout actions  D FULL^VALM1  D EN^VALM("IBCE PRV INS PARAMS")  Q  ; HDR ; -- header code  K VALMHDR  I $G(IBINS) S VALMHDR(1)="INSURANCE CO: "\_$P($G(^DIC(36,+IBINS,0)),U)  Q  ; INIT ; Initialization  N DIR,DIC,DA,X,Y,DTOUT,DUOUT  S DIC(0)="AEMQ",DIC="^DIC(36," D ^DIC  I Y'>0 D  . S VALMQUIT=1  E  D  . S DIR="YA",DIR("A")="DO YOU WANT TO INCLUDE ANY CARE UNIT DETAIL?: ",DIR("?",1)="If you want to see the specific care unit defined for the insurance co",DIR("?")="you should respond yes here"  . W ! D ^DIR K DIR W !  . I $D(DTOUT)!$D(DUOUT) S VALMQUIT=1 Q  . S IBCUINC=(Y=1)  . S IBINS=+Y D BLD(IBINS,IBCUINC)  Q  ; BLD(IBINS,IBCUINC) ; Build display for ins co level provider ID parameters  ; IBINS = ien of ins co (file 36)  ; IBCUINC = flag:  ; = 1 if care unit list should be included or 0 if not  N A,A0,A1,A2,A3,Z0,IB1,IB12,IB4,IBLCT,IBPTYP  S IBLCT=0  S IB4=$G(^DIC(36,IBINS,4))  K ^TMP("IBPRV\_INS\_ID\_PARAMS",$J)  ;  S Z0="Perf Prov Secondary ID Type (1500): "\_$E($$EXPAND^IBTRE(36,4.01,+$P(IB4,U))\_$J("",20),1,20) D SET1(.IBLCT,Z0)  S Z0="Perf Prov Secondary ID Type (UB04): "\_$E($$EXPAND^IBTRE(36,4.02,+$P(IB4,U,2))\_$J("",20),1,20) D SET1(.IBLCT,Z0)  S Z0=$J("",20)\_"Required: "\_$$EXPAND^IBTRE(36,4.03,$P(IB4,U,3)) D SET1(.IBLCT,Z0)  S Z0=$J("",10)\_"Care Unit Name: "\_$$EXPAND^IBTRE(36,4.09,$P(IB4,U,9)) D SET1(.IBLCT,Z0)  S Z0=""  D SET1(.IBLCT,Z0)  ;  I '$D(^IBA(355.96,"D",IBINS)) D  G BLDQ ;No care unit needed  . S Z0=$J("",7)\_"\*\*\* NO CARE UNITS DEFINED FOR THIS INS CO PROVIDER SECONDARY ID \*\*\*" D SET1(.IBLCT,Z0)  ;  S Z0=$J("",17)\_"VALID CARE UNITS FOR THIS INSURANCE COMPANY" D SET1(.IBLCT,Z0),CNTRL^VALM10(IBLCT,18,46,IORVON,IORVOFF)  S A=0  F  S A=$O(^IBA(355.96,"AC",IBINS,A)) Q:'A  S IBPTYP=$P($G(^IBE(355.97,A,0)),U) I IBPTYP'="" D  . S A2=IBPTYP\_U\_A,^TMP("IBPRV\_INS\_ID\_PARAMS\_SORT",$J,A2)=""  . S A0=0 F  S A0=$O(^IBA(355.96,"AC",IBINS,A,A0)) Q:'A0  S A1=$G(^IBA(355.96,A0,0)) D  .. I '$G(IBCUINC) S:'$D(^TMP("IBPRV\_INS\_ID\_PARAMS\_SORT",$J,A2,$P(A1,U,4)\_U\_$P(A1,U,5))) ^($P(A1,U,4)\_U\_$P(A1,U,5))="" Q  .. I $P(A1,U,4)'="",$P(A1,U,5)'="" D  ... S A3=$E($P($G(^IBE(355.95,+A1,0)),U)\_$J("",1,30),1,30)\_U\_$S($P($G(^(0)),U,2)'="":$P(^(0),U,2),1:"<No description available>")  ... I '$D(^TMP("IBPRV\_INS\_ID\_PARAMS\_SORT",$J,A2,$P(A1,U,4)\_U\_$P(A1,U,5),$P(A3,U))) S ^($P(A3,U))=$P(A3,U,2)  . ; records are fully sorted  S A=""  F  S A=$O(^TMP("IBPRV\_INS\_ID\_PARAMS\_SORT",$J,A)) Q:'A  S A2="PROVIDER ID TYPE: "\_$P(A,U),IB1=1 D:'IB1 SET1(.IBLCT,"") D SET1(.IBLCT,A2) S IB12=1 S:$G(IBCUINC) IB1=0 D  . S A0="" F  S A0=$O(^TMP("IBPRV\_INS\_ID\_PARAMS\_SORT",$J,A,A0)) Q:A0=""  D  .. S Z0=$J("",5)\_"FORM TYPE: "\_$E($$EXPAND^IBTRE(355.96,.04,$P(A0,U))\_$J("",25),1,25)\_" CARE TYPE: "\_$E($$EXPAND^IBTRE(355.96,.05,$P(A0,U,2))\_$J("",25),1,25)  .. D:'IB12 SET1(.IBLCT,"") D SET1(.IBLCT,Z0)  .. Q:'$G(IBCUINC)  .. S IB12=0  .. S A1="" F  S A1=$O(^TMP("IBPRV\_INS\_ID\_PARAMS\_SORT",$J,A,A0,A1)) Q:A1=""  S Z0=$J("",10)\_A1\_$G(^(A1)) D SET1(.IBLCT,Z0)  ; BLDQ K ^TMP("IBPRV\_INS\_ID\_PARAMS\_SORT",$J)  S VALMCNT=IBLCT,VALMBG=1  Q  ; SET1(IBLCT,Z0) ;  S IBLCT=IBLCT+1 D SET^VALM10(IBLCT,Z0)  Q  ; EXPND ;  Q  ; HELP ;  Q  ; EXIT ;  K ^TMP("IBPRV\_INS\_ID\_PARAMS",$J)  D CLEAN^VALM10  Q  ; EDIT ; Entrypoint called from IBCSCE to invoke provider id edit functions  Q  ; EDIT1 ; Edit parameters  N IB,IBY,IBCNS,DIE,DR,X,Y  D FULL^VALM1  S IBCNS=IBINS,IBY=12  D MAIN^IBCNSC1  S VALMBCK="R"  Q  ; NETID() ; Returns the ien of the entry in file 355.97 that is designated as the  ; NETWORK ID  N Z S Z=0 F  S Z=$O(^IBE(355.97,Z)) Q:'Z  Q:$P($G(^(Z,1)),U,6)  Q Z  ; EMCID() ; Returns the ien of the entry in file 355.97 that is designated as the  ; EMC ID  N Z S Z=0 F  S Z=$O(^IBE(355.97,Z)) Q:'Z  Q:$P($G(^(Z,1)),U,5)  Q Z  ; UPIN() ; Returns the ien of the entry in file 355.97 that is designated as the  ; UPIN ID  Q +$O(^IBE(355.97,"B","UPIN",0))  ; EDITID(IBCNS) ; Edit provider id's from insurance co enter/edit  ; IBCNS = ien of file 36  Q   ; WCJ 12/30/2005  N X,Y,Z4,DIR  S Z4=$G(^DIC(36,IBCNS,4))  I 'Z4,'$P(Z4,U,2) Q  S DIR("A",1)="USE PROVIDER ID MAINTENANCE TO ENTER/EDIT PROV SECONDARY ID'S FOR THIS CO.",DIR("A")="PRESS RETURN TO CONTINUE: ",DIR(0)="EA" W ! D ^DIR K DIR  Q  ; | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCEP ;ALB/TMP - Functions for PROVIDER ID MAINT - INS CO PARAMS ;11-02-00  ;;2.0;INTEGRATED BILLING;\*\*137,232,320,348,349,**592**\*\*;21-MAR-94;Build 46  ;;Per VHA Directive 2004-038, this routine should not be modified.  ; EN ; -- main entry point for IBCE PRV INS PARAMS  N IBINS,IBCUINC ; Variable should be available throughout actions  D FULL^VALM1  D EN^VALM("IBCE PRV INS PARAMS")  Q  ; HDR ; -- header code  K VALMHDR  I $G(IBINS) S VALMHDR(1)="INSURANCE CO: "\_$P($G(^DIC(36,+IBINS,0)),U)  Q  ; INIT ; Initialization  N DIR,DIC,DA,X,Y,DTOUT,DUOUT  S DIC(0)="AEMQ",DIC="^DIC(36," D ^DIC  I Y'>0 D  . S VALMQUIT=1  E  D  . S DIR="YA",DIR("A")="DO YOU WANT TO INCLUDE ANY CARE UNIT DETAIL?: ",DIR("?",1)="If you want to see the specific care unit defined for the insurance co",DIR("?")="you should respond yes here"  . W ! D ^DIR K DIR W !  . I $D(DTOUT)!$D(DUOUT) S VALMQUIT=1 Q  . S IBCUINC=(Y=1)  . S IBINS=+Y D BLD(IBINS,IBCUINC)  Q  ; BLD(IBINS,IBCUINC) ; Build display for ins co level provider ID parameters  ; IBINS = ien of ins co (file 36)  ; IBCUINC = flag:  ; = 1 if care unit list should be included or 0 if not  N A,A0,A1,A2,A3,Z0,IB1,IB12,IB4,IBLCT,IBPTYP  S IBLCT=0  S IB4=$G(^DIC(36,IBINS,4))  K ^TMP("IBPRV\_INS\_ID\_PARAMS",$J)  ;  S Z0="Perf Prov Secondary ID Type (1500): "\_$E($$EXPAND^IBTRE(36,4.01,+$P(IB4,U))\_$J("",20),1,20) D SET1(.IBLCT,Z0) **;JWS;IB\*2.0\*592; form J430D**  **S Z0="Perf Prov Secondary ID Type (J430D): "\_$E($$EXPAND^IBTRE(36,4.14,+$P(IB4,U,14))\_$J("",20),1,20) D SET1(.IBLCT,Z0)**  S Z0="Perf Prov Secondary ID Type (UB04): "\_$E($$EXPAND^IBTRE(36,4.02,+$P(IB4,U,2))\_$J("",20),1,20) D SET1(.IBLCT,Z0)  S Z0=$J("",20)\_"Required: "\_$$EXPAND^IBTRE(36,4.03,$P(IB4,U,3)) D SET1(.IBLCT,Z0)  S Z0=$J("",10)\_"Care Unit Name: "\_$$EXPAND^IBTRE(36,4.09,$P(IB4,U,9)) D SET1(.IBLCT,Z0)  S Z0=""  D SET1(.IBLCT,Z0)  ;  I '$D(^IBA(355.96,"D",IBINS)) D  G BLDQ ;No care unit needed  . S Z0=$J("",7)\_"\*\*\* NO CARE UNITS DEFINED FOR THIS INS CO PROVIDER SECONDARY ID \*\*\*" D SET1(.IBLCT,Z0)  ;  S Z0=$J("",17)\_"VALID CARE UNITS FOR THIS INSURANCE COMPANY" D SET1(.IBLCT,Z0),CNTRL^VALM10(IBLCT,18,46,IORVON,IORVOFF)  S A=0  F  S A=$O(^IBA(355.96,"AC",IBINS,A)) Q:'A  S IBPTYP=$P($G(^IBE(355.97,A,0)),U) I IBPTYP'="" D  . S A2=IBPTYP\_U\_A,^TMP("IBPRV\_INS\_ID\_PARAMS\_SORT",$J,A2)=""  . S A0=0 F  S A0=$O(^IBA(355.96,"AC",IBINS,A,A0)) Q:'A0  S A1=$G(^IBA(355.96,A0,0)) D  .. I '$G(IBCUINC) S:'$D(^TMP("IBPRV\_INS\_ID\_PARAMS\_SORT",$J,A2,$P(A1,U,4)\_U\_$P(A1,U,5))) ^($P(A1,U,4)\_U\_$P(A1,U,5))="" Q  .. I $P(A1,U,4)'="",$P(A1,U,5)'="" D  ... S A3=$E($P($G(^IBE(355.95,+A1,0)),U)\_$J("",1,30),1,30)\_U\_$S($P($G(^(0)),U,2)'="":$P(^(0),U,2),1:"<No description available>")  ... I '$D(^TMP("IBPRV\_INS\_ID\_PARAMS\_SORT",$J,A2,$P(A1,U,4)\_U\_$P(A1,U,5),$P(A3,U))) S ^($P(A3,U))=$P(A3,U,2)  . ; records are fully sorted  S A=""  F  S A=$O(^TMP("IBPRV\_INS\_ID\_PARAMS\_SORT",$J,A)) Q:'A  S A2="PROVIDER ID TYPE: "\_$P(A,U),IB1=1 D:'IB1 SET1(.IBLCT,"") D SET1(.IBLCT,A2) S IB12=1 S:$G(IBCUINC) IB1=0 D  . S A0="" F  S A0=$O(^TMP("IBPRV\_INS\_ID\_PARAMS\_SORT",$J,A,A0)) Q:A0=""  D  .. S Z0=$J("",5)\_"FORM TYPE: "\_$E($$EXPAND^IBTRE(355.96,.04,$P(A0,U))\_$J("",25),1,25)\_" CARE TYPE: "\_$E($$EXPAND^IBTRE(355.96,.05,$P(A0,U,2))\_$J("",25),1,25)  .. D:'IB12 SET1(.IBLCT,"") D SET1(.IBLCT,Z0)  .. Q:'$G(IBCUINC)  .. S IB12=0  .. S A1="" F  S A1=$O(^TMP("IBPRV\_INS\_ID\_PARAMS\_SORT",$J,A,A0,A1)) Q:A1=""  S Z0=$J("",10)\_A1\_$G(^(A1)) D SET1(.IBLCT,Z0)  ; BLDQ K ^TMP("IBPRV\_INS\_ID\_PARAMS\_SORT",$J)  S VALMCNT=IBLCT,VALMBG=1  Q  ; SET1(IBLCT,Z0) ;  S IBLCT=IBLCT+1 D SET^VALM10(IBLCT,Z0)  Q  ; EXPND ;  Q  ; HELP ;  Q  ; EXIT ;  K ^TMP("IBPRV\_INS\_ID\_PARAMS",$J)  D CLEAN^VALM10  Q  ; EDIT ; Entrypoint called from IBCSCE to invoke provider id edit functions  Q  ; EDIT1 ; Edit parameters  N IB,IBY,IBCNS,DIE,DR,X,Y  D FULL^VALM1  S IBCNS=IBINS,IBY=12  D MAIN^IBCNSC1  S VALMBCK="R"  Q  ; NETID() ; Returns the ien of the entry in file 355.97 that is designated as the  ; NETWORK ID  N Z S Z=0 F  S Z=$O(^IBE(355.97,Z)) Q:'Z  Q:$P($G(^(Z,1)),U,6)  Q Z  ; EMCID() ; Returns the ien of the entry in file 355.97 that is designated as the  ; EMC ID  N Z S Z=0 F  S Z=$O(^IBE(355.97,Z)) Q:'Z  Q:$P($G(^(Z,1)),U,5)  Q Z  ; UPIN() ; Returns the ien of the entry in file 355.97 that is designated as the  ; UPIN ID  Q +$O(^IBE(355.97,"B","UPIN",0))  ; EDITID(IBCNS) ; Edit provider id's from insurance co enter/edit  ; IBCNS = ien of file 36  Q   ; WCJ 12/30/2005  N X,Y,Z4,DIR  S Z4=$G(^DIC(36,IBCNS,4))  I 'Z4,'$P(Z4,U,2) Q  S DIR("A",1)="USE PROVIDER ID MAINTENANCE TO ENTER/EDIT PROV SECONDARY ID'S FOR THIS CO.",DIR("A")="PRESS RETURN TO CONTINUE: ",DIR(0)="EA" W ! D ^DIR K DIR  Q  ; | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCEP0 | | | | | | | | |
| **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** | | | | | | | | | |
| IBCEP0 ;ALB/TMP - Functions for PROVIDER ID MAINTENANCE ;13-DEC-99  ;;2.0;INTEGRATED BILLING;\*\*137,191,239,232,320,348,349,377\*\*;21-MAR-94;Build 23  ;;Per VHA Directive 2004-038, this routine should not be modified.  ; EN ; -- main entry point for IBCE PRV INS ID  N IBINS,IBDSP,IBSORT,IBPRV ; Variables should be available throughout actions  K IBFASTXT  D FULL^VALM1  D EN^VALM("IBCE PRVINS ID")  Q  ; EN1(IBINS) ; Entrypoint from insurance co maintenance  N IBDSP,IBSORT ; Variables should be available throughout actions  D FULL^VALM1  D EN^VALM("IBCE PRVINS ID FROM INS MAINT")  Q  ; HDR ; -- header code  N Z,Z0,Z1,IBCT,IBPPTYP,IBEMCTYP  S IBCT=1  K VALMHDR  I $G(IBINS) D  . N PCF,PCDISP  . S PCF=$P($G(^DIC(36,+IBINS,3)),U,13)  . S PCDISP=$S(PCF="C":"(Child)",PCF="P":"(Parent)",1:"")  . S VALMHDR(1)="Insurance Co: "\_$P($G(^DIC(36,+IBINS,0)),U)\_" "\_PCDISP  . ; Get performing provider id type for insurance co  . S IBPPTYP=$$PPTYP(IBINS)  . ; Get ien of EMC ID from file 355.97  . S IBEMCTYP=+$$EMCID^IBCEP()  . I $G(IBSORT)="ALL"!($G(IBDSP)="I")!($G(IBSORT)=IBPPTYP)!($G(IBSORT)=IBEMCTYP) D  .. ; Look for care unit in either of these id types - if there, report on line 2 of header  .. I $G(IBSORT)=IBPPTYP S IBEMCTYP=0  .. I $G(IBSORT)=IBEMCTYP S IBPPTYP=0  .. F Z0=IBPPTYP\_"P",IBEMCTYP\_"E" S Z1="" F  S Z1=$O(^IBA(355.96,"D",+IBINS,+Z0,Z1)) Q:Z1=""  I Z1'="\*N/A\*" S Z($E(Z0,$L(Z0)))=1 Q  .. I $D(Z("P"))!$D(Z("E")) D  ... S IBCT=IBCT+1  ... S VALMHDR(IBCT)=" "\_$S($D(Z("P")):"PERFORMING PROV ID"\_$S($D(Z("E")):" AND ",1:""),1:"")\_$S($D(Z("E")):"EMC PROV ID",1:"")\_" MAY REQUIRE CARE UNIT"  . I $D(Z("P"))!$D(Z("E")) S IBCT=IBCT+1,VALMHDR(IBCT)=" "  . S IBCT=IBCT+1,VALMHDR(IBCT)=" PROVIDER "\_$S($G(IBDSP)="I":"ID TYPE",1:"NAME ")\_$J("",6)\_"FORM CARE TYPE CARE UNIT ID#"  Q  ; INIT ; Initialization  K ^TMP("IB\_EDITED\_IDS",$J) ; This will be to keep track of ID's edited during this session  D INSID(.IBINS,.IBDSP,.IBSORT)  I $G(IBDSP)="I",$G(IBSORT) S IBPRV=IBSORT  I '$G(IBINS) S VALMQUIT=1  Q  ; INSID(IBINS,IBDSP,IBSORT) ;  N DIC,DIR,DA,X,Y,IBOK,DTOUT,DUOUT  S IBOK=1  I '$G(IBINS) D  . S DIC(0)="AEMQ",DIC="^DIC(36," D ^DIC  . I Y'>0 S IBOK=0 Q  . S IBINS=+Y  I '$G(IBINS) S IBOK=0  I 'IBOK G INSIDQ  ;  S DIR(0)="SA^D:INSURANCE CO DEFAULT IDS;I:INDIVIDUAL PROVIDER IDS FURNISHED BY THE INS CO;A:ALL IDS FURNISHED BY THE INS CO BY PROVIDER TYPE"  S DIR("A")="SELECT DISPLAY CONTENT: ",DIR("B")="A"  S DIR("?",1)="(D) DISPLAY CONTAINS ONLY THOSE IDS ASSIGNED AS DEFAULTS TO THE FACILITY BY",DIR("?",2)=" THE INSURANCE COMPANY"  S DIR("?",3)="(I) DISPLAY CONTAINS ONLY THOSE IDS ASSIGNED TO INDIVIDUAL PROVIDERS BY THE",DIR("?",4)=" INSURANCE COMPANY"  S DIR("?",5)="(A) DISPLAY CONTAINS ALL IDS ASSIGNED BY THE INSURANCE COMPANY FOR ONE OR ALL",DIR("?")=" PROVIDER ID TYPES"  W ! D ^DIR K DIR W !  I $D(DTOUT)!$D(DUOUT)!("DIA"'[Y) S IBOK=0 G INSIDQ  S IBDSP=Y,IBSORT=""  I IBDSP="A"!(IBDSP="I") F  D  Q:'IBOK!(IBSORT'="")  . ;  . I IBDSP="A" D  .. S DIR("A")="Display only IDs with a specific ID Qualifier?: "  .. S DIR("?",1)="Answer Yes to select a specific ID Qualifier by which to display IDs."  .. S DIR("?")="Answer No to display all IDs."  .. Q  . ;  . I IBDSP="I" D  .. S DIR("A")="Display IDs for a specific Provider?: "  .. S DIR("?",1)="Answer Yes to select a specific Provider."  .. S DIR("?")="Answer No to display all Providers."  .. Q  . ;  . S DIR("B")="NO",DIR(0)="YA"  . W ! D ^DIR K DIR W !  . I $D(DTOUT)!$D(DUOUT) S IBOK=0 Q  . I Y'=1 S IBSORT="ALL" Q  . ;  . I IBDSP="A" D  Q  .. S DIC(0)="AEMQ",DIC="^IBE(355.97,",DIC("S")="I $S('$P(^(0),U,2):1,1:$P(^(0),U,2)=3)"  .. S DIC("A")="Select type of ID Qualifier: "  .. D ^DIC K DIC  .. I Y>0 S IBSORT=+Y Q  .. I $D(DTOUT)!$D(DUOUT) S IBOK=0  . ;  . I IBDSP="I" D  Q  .. N DA  .. S DIR(0)="399.0222,.02A",DIR("A")="SELECT PROVIDER: "  .. W ! D ^DIR K DIR W !  .. I Y>0 S IBSORT=Y Q  .. I $D(DTOUT)!$D(DUOUT) S IBOK=0 Q  . S IBOK=0 Q  ;  G:'IBOK INSIDQ  D BLD(IBINS,IBDSP,IBSORT) INSIDQ I 'IBOK S VALMQUIT=1  Q  ; BLD(IBINS,IBDSP,IBSORT) ; Build display for Insurance co level provider ID's  N IB,IBENT,IBLCT,IBCT,IBPRV,IBSRT1,IBSRT2,IBOSRT1,IBOSRT2,CU,FT,PT,CT,Z,Z0  K ^TMP("IBPRV\_INS\_ID",$J),^TMP("IBPRV\_INS\_SORT",$J)  ;  S (IBENT,IBCT,IBLCT)=0  ;  I "DA"[$G(IBDSP) D  . S CU="" F  S CU=$O(^IBA(355.91,"AUNIQ",IBINS,CU)) Q:CU=""  S FT="" F  S FT=$O(^IBA(355.91,"AUNIQ",IBINS,CU,FT)) Q:FT=""  D  .. S CT="" F  S CT=$O(^IBA(355.91,"AUNIQ",IBINS,CU,FT,CT)) Q:CT=""  S PT=0 F  S PT=$S(IBDSP="A"&IBSORT:IBSORT,1:$O(^IBA(355.91,"AUNIQ",IBINS,CU,FT,CT,PT))) Q:'PT  D  Q:IBDSP="A"&IBSORT  ... S Z=0 F  S Z=$O(^IBA(355.91,"AUNIQ",IBINS,CU,FT,CT,PT,Z)) Q:'Z  S IB=$G(^IBA(355.91,Z,0)) S ^TMP("IBPRV\_INS\_SORT",$J,PT,"^<<INS CO DEFAULT>>",FT,CT,CU,Z)=$P(IB,U,7)\_U  ;  I "IA"[$G(IBDSP) D  . S IBPRV=""  . N IB1,IB2  . F  S IBPRV=$O(^IBA(355.9,"AE",IBINS,IBPRV)) Q:'IBPRV  S Z=0 F  S Z=$O(^IBA(355.9,"AE",IBINS,IBPRV,Z)) Q:'Z  S IB=$G(^IBA(355.9,Z,0)) D  .. Q:$P(IB,U,4)=""!($P(IB,U,5)="")!($P(IB,U,6)="")!($P(IB,U,16)="")  .. I IBSORT,$S(IBDSP="I":IBPRV'=IBSORT,1:$P(IB,U,6)'=IBSORT) Q  .. S IB1=$S(IBDSP="A":$P(IB,U,6),1:U\_$$EXPAND^IBTRE(355.9,.01,IBPRV)\_U\_IBPRV)  .. S IB2=$S(IBDSP="I":$P(IB,U,6),1:U\_$$EXPAND^IBTRE(355.9,.01,IBPRV)\_U\_IBPRV)  .. S ^TMP("IBPRV\_INS\_SORT",$J,IB1,IB2,$P(IB,U,4),$P(IB,U,5),$P(IB,U,16),Z)=$P(IB,U,7)\_U\_IBPRV  ;  S IBOSRT1=""  S IBSRT1="" F  S IBSRT1=$O(^TMP("IBPRV\_INS\_SORT",$J,IBSRT1)) Q:IBSRT1=""  D  . S IBSRT2="",IBOSRT2=""  . F  S IBSRT2=$O(^TMP("IBPRV\_INS\_SORT",$J,IBSRT1,IBSRT2)) Q:IBSRT2=""  D  .. I IBOSRT1'=IBSRT1 D  ... I IBOSRT1'="" S IBLCT=IBLCT+1 D SET^VALM10(IBLCT," ",IBCT+1)  ... S IBLCT=IBLCT+1 D SET^VALM10(IBLCT,$S(IBDSP'="I":"ID Qualifier",1:"Provider")\_": "\_$S(IBDSP'="I":$$EXPAND^IBTRE(355.91,.06,IBSRT1),1:$P(IBSRT1,U,2\_$S($P(IBSRT2,U,3)["VA(200":" (VA)",1:"(NON-VA)"))),IBCT+1)  ... S IBOSRT1=IBSRT1  .. ;  .. S FT="" F  S FT=$O(^TMP("IBPRV\_INS\_SORT",$J,IBSRT1,IBSRT2,FT)) Q:FT=""  S CT="" F  S CT=$O(^TMP("IBPRV\_INS\_SORT",$J,IBSRT1,IBSRT2,FT,CT)) Q:CT=""  D  ... S CU="" F  S CU=$O(^TMP("IBPRV\_INS\_SORT",$J,IBSRT1,IBSRT2,FT,CT,CU)) Q:CU=""  S Z=0 F  S Z=$O(^TMP("IBPRV\_INS\_SORT",$J,IBSRT1,IBSRT2,FT,CT,CU,Z)) Q:'Z  S IB=$G(^(Z)) D  .... S IBLCT=IBLCT+1,IBCT=IBCT+1  .... S Z0=$E(IBCT\_$J("",4),1,4)\_" "  .... I IBDSP'="I" S Z0=Z0\_$E($S(IBOSRT2'=IBSRT2:$P(IBSRT2,U,2),1:"")\_$J("",20),1,20)  .... I IBDSP="I" S Z0=Z0\_$E($S(IBOSRT2'=IBSRT2:$$EXPAND^IBTRE(355.9,.06,IBSRT2),1:"")\_$J("",20),1,20)  .... S IBOSRT2=IBSRT2  .... S Z0=Z0\_" "\_$S(FT=1:"UB-04",FT=2:"1500 ",1:"BOTH ")\_" "\_$E($S(CT=3:"RX",CT=1:"INPT",CT=2:"OUTPT",1:"INPT/OUTPT")\_$J("",11),1,11)\_" "\_$E($S(CU'="\*N/A\*":$P($G(^IBA(355.95,+$P($G(^IBA(355.96,+CU,0)),U),0)),U),1:"")\_$J("",15),1,15)  .... D SET^VALM10(IBLCT,Z0\_" "\_$P(IB,U),IBCT)  .... S ^TMP("IBPRV\_INS\_ID",$J,"ZIDX",IBCT)=Z,^(IBCT,"PRV")=$P(IB,U,2)  .... I '$D(^TMP("IBPRV\_INS\_ID",$J,$S(IBDSP="I":"ZXPRV",1:"ZXPTYP"),IBSRT1)) S ^(IBSRT1)=IBLCT-1  K ^TMP("IBPRV\_INS\_SORT",$J)  ;  I IBLCT=0 D  G BLDQ ; No entries found  . D SET^VALM10(1," ")  . S Z=" No "\_$S(IBDSP="D":"default ",1:"")  . S Z=Z\_"ID's found for "\_$S(IBDSP="I":"provider "\_$S(IBSORT:"("\_$$EXPAND^IBTRE(355.9,.01,IBSORT)\_") ",1:"")\_"and ",IBDSP="A":"provider type "\_$S(IBSORT:"("\_$$EXPAND^IBTRE(355.9,.06,IBSORT)\_") ",1:"")\_"and ",1:"")\_"insurance co"  . D SET^VALM10(2,Z)  . S IBLCT=2  ; BLDQ S VALMCNT=IBLCT,VALMBG=1  Q  ; EXPND ;  Q  ; HELP ;  Q  ; EXIT ;  K IBFASTXT  D COPYPROV^IBCEP5A(IBINS)  K ^TMP("IBPRV\_INS\_ID",$J)  D CLEAN^VALM10  Q  ; SEL(IBDA,MANY) ; Select from provider id list  ; IBDA is passed by reference and IBDA(1) returned containing  ; ien's of the provider id records selected (file 355.9).  ; If > 1 entry can be selected, MANY is set to 1  N Z  S IBDA=0  D EN^VALM2($G(XQORNOD(0)),$S($G(MANY):"",1:"S"))  S Z=0 F  S Z=$O(VALMY(Z)) Q:'Z  S IBDA=IBDA+1,IBDA(IBDA)=+$G(^TMP("IBPRV\_INS\_ID",$J,"ZIDX",Z))\_U\_$G(^(Z,"PRV"))  Q  ; ENX(IBINS1) ; Insurance co level defaults for all providers or  ; for all providers by care unit  N DIC,DIE,DR,DA,X,Y,DLAYGO  I '$G(IBINS1) D  G:'$G(IBINS1) ENQ  . S DIC="^IBA(355.91,",DIC(0)="AELMQ",DLAYGO=355.91 D ^DIC  . I Y>0 S IBINS1=+Y  S DIE="^IBA(355.91,",DA=IBINS1,DR=".01;.06;.04;.05;.03;.07" D ^DIE  ; ENQ Q  ; PPTYP(IBINS) ; Returns the ien of the default performing provider type for   ; insurance company IBINS (ien file 36)  Q +$G(^DIC(36,+IBINS,4))  ; SCREEN(WHICH) ; This screen is used the menu protocol to screen out the ID functions if it is a child ins co  Q:'$G(DA) 0  Q:'$G(DA(1)) 0  N FILE,IENS,FIELD,FLAG,TARGET  S FILE=101.01,IENS=DA\_","\_DA(1),FIELD=".01",FLAG="I"  D GETS^DIQ(FILE,IENS,FIELD,FLAG,"TARGET")  Q:'$D(TARGET) 0  N IEN  S IEN=$G(TARGET(FILE,IENS\_",",FIELD,FLAG))  Q:'+IEN 0  S FILE=101,FIELD=1,FLAG="E"  K TARGET  D GETS^DIQ(FILE,IEN,FIELD,FLAG,"TARGET")  Q:'$D(TARGET) 0  I $G(TARGET(FILE,IEN\_",",FIELD,FLAG))'[WHICH Q 1  Q:'$G(IBINS) 0  N PCF  S PCF=$P($G(^DIC(36,+IBINS,3)),U,13)  I PCF="C" Q 0  Q 1 | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCEP0 ;ALB/TMP - Functions for PROVIDER ID MAINTENANCE ;13-DEC-99  ;;2.0;INTEGRATED BILLING;\*\*137,191,239,232,320,348,349,377,**592**\*\*;21-MAR-94;Build 23  ;;Per VHA Directive 2004-038, this routine should not be modified.  ; EN ; -- main entry point for IBCE PRV INS ID  N IBINS,IBDSP,IBSORT,IBPRV ; Variables should be available throughout actions  K IBFASTXT  D FULL^VALM1  D EN^VALM("IBCE PRVINS ID")  Q  ; EN1(IBINS) ; Entrypoint from insurance co maintenance  N IBDSP,IBSORT ; Variables should be available throughout actions  D FULL^VALM1  D EN^VALM("IBCE PRVINS ID FROM INS MAINT")  Q  ; HDR ; -- header code  N Z,Z0,Z1,IBCT,IBPPTYP,IBEMCTYP  S IBCT=1  K VALMHDR  I $G(IBINS) D  . N PCF,PCDISP  . S PCF=$P($G(^DIC(36,+IBINS,3)),U,13)  . S PCDISP=$S(PCF="C":"(Child)",PCF="P":"(Parent)",1:"")  . S VALMHDR(1)="Insurance Co: "\_$P($G(^DIC(36,+IBINS,0)),U)\_" "\_PCDISP  . ; Get performing provider id type for insurance co  . S IBPPTYP=$$PPTYP(IBINS)  . ; Get ien of EMC ID from file 355.97  . S IBEMCTYP=+$$EMCID^IBCEP()  . I $G(IBSORT)="ALL"!($G(IBDSP)="I")!($G(IBSORT)=IBPPTYP)!($G(IBSORT)=IBEMCTYP) D  .. ; Look for care unit in either of these id types - if there, report on line 2 of header  .. I $G(IBSORT)=IBPPTYP S IBEMCTYP=0  .. I $G(IBSORT)=IBEMCTYP S IBPPTYP=0  .. F Z0=IBPPTYP\_"P",IBEMCTYP\_"E" S Z1="" F  S Z1=$O(^IBA(355.96,"D",+IBINS,+Z0,Z1)) Q:Z1=""  I Z1'="\*N/A\*" S Z($E(Z0,$L(Z0)))=1 Q  .. I $D(Z("P"))!$D(Z("E")) D  ... S IBCT=IBCT+1  ... S VALMHDR(IBCT)=" "\_$S($D(Z("P")):"PERFORMING PROV ID"\_$S($D(Z("E")):" AND ",1:""),1:"")\_$S($D(Z("E")):"EMC PROV ID",1:"")\_" MAY REQUIRE CARE UNIT"  . I $D(Z("P"))!$D(Z("E")) S IBCT=IBCT+1,VALMHDR(IBCT)=" "  . S IBCT=IBCT+1,VALMHDR(IBCT)=" PROVIDER "\_$S($G(IBDSP)="I":"ID TYPE",1:"NAME ")\_$J("",6)\_"FORM CARE TYPE CARE UNIT ID#"  Q  ; INIT ; Initialization  K ^TMP("IB\_EDITED\_IDS",$J) ; This will be to keep track of ID's edited during this session  D INSID(.IBINS,.IBDSP,.IBSORT)  I $G(IBDSP)="I",$G(IBSORT) S IBPRV=IBSORT  I '$G(IBINS) S VALMQUIT=1  Q  ; INSID(IBINS,IBDSP,IBSORT) ;  N DIC,DIR,DA,X,Y,IBOK,DTOUT,DUOUT  S IBOK=1  I '$G(IBINS) D  . S DIC(0)="AEMQ",DIC="^DIC(36," D ^DIC  . I Y'>0 S IBOK=0 Q  . S IBINS=+Y  I '$G(IBINS) S IBOK=0  I 'IBOK G INSIDQ  ;  S DIR(0)="SA^D:INSURANCE CO DEFAULT IDS;I:INDIVIDUAL PROVIDER IDS FURNISHED BY THE INS CO;A:ALL IDS FURNISHED BY THE INS CO BY PROVIDER TYPE"  S DIR("A")="SELECT DISPLAY CONTENT: ",DIR("B")="A"  S DIR("?",1)="(D) DISPLAY CONTAINS ONLY THOSE IDS ASSIGNED AS DEFAULTS TO THE FACILITY BY",DIR("?",2)=" THE INSURANCE COMPANY"  S DIR("?",3)="(I) DISPLAY CONTAINS ONLY THOSE IDS ASSIGNED TO INDIVIDUAL PROVIDERS BY THE",DIR("?",4)=" INSURANCE COMPANY"  S DIR("?",5)="(A) DISPLAY CONTAINS ALL IDS ASSIGNED BY THE INSURANCE COMPANY FOR ONE OR ALL",DIR("?")=" PROVIDER ID TYPES"  W ! D ^DIR K DIR W !  I $D(DTOUT)!$D(DUOUT)!("DIA"'[Y) S IBOK=0 G INSIDQ  S IBDSP=Y,IBSORT=""  I IBDSP="A"!(IBDSP="I") F  D  Q:'IBOK!(IBSORT'="")  . ;  . I IBDSP="A" D  .. S DIR("A")="Display only IDs with a specific ID Qualifier?: "  .. S DIR("?",1)="Answer Yes to select a specific ID Qualifier by which to display IDs."  .. S DIR("?")="Answer No to display all IDs."  .. Q  . ;  . I IBDSP="I" D  .. S DIR("A")="Display IDs for a specific Provider?: "  .. S DIR("?",1)="Answer Yes to select a specific Provider."  .. S DIR("?")="Answer No to display all Providers."  .. Q  . ;  . S DIR("B")="NO",DIR(0)="YA"  . W ! D ^DIR K DIR W !  . I $D(DTOUT)!$D(DUOUT) S IBOK=0 Q  . I Y'=1 S IBSORT="ALL" Q  . ;  . I IBDSP="A" D  Q  .. S DIC(0)="AEMQ",DIC="^IBE(355.97,",DIC("S")="I $S('$P(^(0),U,2):1,1:$P(^(0),U,2)=3)"  .. S DIC("A")="Select type of ID Qualifier: "  .. D ^DIC K DIC  .. I Y>0 S IBSORT=+Y Q  .. I $D(DTOUT)!$D(DUOUT) S IBOK=0  . ;  . I IBDSP="I" D  Q  .. N DA  .. S DIR(0)="399.0222,.02A",DIR("A")="SELECT PROVIDER: "  .. W ! D ^DIR K DIR W !  .. I Y>0 S IBSORT=Y Q  .. I $D(DTOUT)!$D(DUOUT) S IBOK=0 Q  . S IBOK=0 Q  ;  G:'IBOK INSIDQ  D BLD(IBINS,IBDSP,IBSORT) INSIDQ I 'IBOK S VALMQUIT=1  Q  ; BLD(IBINS,IBDSP,IBSORT) ; Build display for Insurance co level provider ID's  N IB,IBENT,IBLCT,IBCT,IBPRV,IBSRT1,IBSRT2,IBOSRT1,IBOSRT2,CU,FT,PT,CT,Z,Z0  K ^TMP("IBPRV\_INS\_ID",$J),^TMP("IBPRV\_INS\_SORT",$J)  ;  S (IBENT,IBCT,IBLCT)=0  ;  I "DA"[$G(IBDSP) D  . S CU="" F  S CU=$O(^IBA(355.91,"AUNIQ",IBINS,CU)) Q:CU=""  S FT="" F  S FT=$O(^IBA(355.91,"AUNIQ",IBINS,CU,FT)) Q:FT=""  D  .. S CT="" F  S CT=$O(^IBA(355.91,"AUNIQ",IBINS,CU,FT,CT)) Q:CT=""  S PT=0 F  S PT=$S(IBDSP="A"&IBSORT:IBSORT,1:$O(^IBA(355.91,"AUNIQ",IBINS,CU,FT,CT,PT))) Q:'PT  D  Q:IBDSP="A"&IBSORT  ... S Z=0 F  S Z=$O(^IBA(355.91,"AUNIQ",IBINS,CU,FT,CT,PT,Z)) Q:'Z  S IB=$G(^IBA(355.91,Z,0)) S ^TMP("IBPRV\_INS\_SORT",$J,PT,"^<<INS CO DEFAULT>>",FT,CT,CU,Z)=$P(IB,U,7)\_U  ;  I "IA"[$G(IBDSP) D  . S IBPRV=""  . N IB1,IB2  . F  S IBPRV=$O(^IBA(355.9,"AE",IBINS,IBPRV)) Q:'IBPRV  S Z=0 F  S Z=$O(^IBA(355.9,"AE",IBINS,IBPRV,Z)) Q:'Z  S IB=$G(^IBA(355.9,Z,0)) D  .. Q:$P(IB,U,4)=""!($P(IB,U,5)="")!($P(IB,U,6)="")!($P(IB,U,16)="")  .. I IBSORT,$S(IBDSP="I":IBPRV'=IBSORT,1:$P(IB,U,6)'=IBSORT) Q  .. S IB1=$S(IBDSP="A":$P(IB,U,6),1:U\_$$EXPAND^IBTRE(355.9,.01,IBPRV)\_U\_IBPRV)  .. S IB2=$S(IBDSP="I":$P(IB,U,6),1:U\_$$EXPAND^IBTRE(355.9,.01,IBPRV)\_U\_IBPRV)  .. S ^TMP("IBPRV\_INS\_SORT",$J,IB1,IB2,$P(IB,U,4),$P(IB,U,5),$P(IB,U,16),Z)=$P(IB,U,7)\_U\_IBPRV  ;  S IBOSRT1=""  S IBSRT1="" F  S IBSRT1=$O(^TMP("IBPRV\_INS\_SORT",$J,IBSRT1)) Q:IBSRT1=""  D  . S IBSRT2="",IBOSRT2=""  . F  S IBSRT2=$O(^TMP("IBPRV\_INS\_SORT",$J,IBSRT1,IBSRT2)) Q:IBSRT2=""  D  .. I IBOSRT1'=IBSRT1 D  ... I IBOSRT1'="" S IBLCT=IBLCT+1 D SET^VALM10(IBLCT," ",IBCT+1)  ... S IBLCT=IBLCT+1 D SET^VALM10(IBLCT,$S(IBDSP'="I":"ID Qualifier",1:"Provider")\_": "\_$S(IBDSP'="I":$$EXPAND^IBTRE(355.91,.06,IBSRT1),1:$P(IBSRT1,U,2\_$S($P(IBSRT2,U,3)["VA(200":" (VA)",1:"(NON-VA)"))),IBCT+1)  ... S IBOSRT1=IBSRT1  .. ;  .. S FT="" F  S FT=$O(^TMP("IBPRV\_INS\_SORT",$J,IBSRT1,IBSRT2,FT)) Q:FT=""  S CT="" F  S CT=$O(^TMP("IBPRV\_INS\_SORT",$J,IBSRT1,IBSRT2,FT,CT)) Q:CT=""  D  ... S CU="" F  S CU=$O(^TMP("IBPRV\_INS\_SORT",$J,IBSRT1,IBSRT2,FT,CT,CU)) Q:CU=""  S Z=0 F  S Z=$O(^TMP("IBPRV\_INS\_SORT",$J,IBSRT1,IBSRT2,FT,CT,CU,Z)) Q:'Z  S IB=$G(^(Z)) D  .... S IBLCT=IBLCT+1,IBCT=IBCT+1  .... S Z0=$E(IBCT\_$J("",4),1,4)\_" "  .... I IBDSP'="I" S Z0=Z0\_$E($S(IBOSRT2'=IBSRT2:$P(IBSRT2,U,2),1:"")\_$J("",20),1,20)  .... I IBDSP="I" S Z0=Z0\_$E($S(IBOSRT2'=IBSRT2:$$EXPAND^IBTRE(355.9,.06,IBSRT2),1:"")\_$J("",20),1,20)  .... S IBOSRT2=IBSRT2 **.... ;JRA IB\*2.0\*592 Modify to accomodate Dental form 'J430D'**  .... S Z0=Z0\_" "\_$S(FT=1:"UB-04",FT=2:"1500",**FT=4:"J430D",**1:"ALL ")\_" "\_$E($S(CT=3:"RX",CT=1:"INPT",CT=2:"OUTPT",1:"INPT/OUTPT")\_$J("",11),1,11) **;JWS;JRA IB\*2.0\*592**  .... S Z0=Z0\_" "\_$E($S(CU'="\*N/A\*":$P($G(^IBA(355.95,+$P($G(^IBA(355.96,+CU,0)),U),0)),U),1:"")\_$J("",15),1,15) **;JWS;JRA IB\*2.0\*592**  .... D SET^VALM10(IBLCT,Z0\_" "\_$P(IB,U),IBCT)  .... S ^TMP("IBPRV\_INS\_ID",$J,"ZIDX",IBCT)=Z,^(IBCT,"PRV")=$P(IB,U,2)  .... I '$D(^TMP("IBPRV\_INS\_ID",$J,$S(IBDSP="I":"ZXPRV",1:"ZXPTYP"),IBSRT1)) S ^(IBSRT1)=IBLCT-1  K ^TMP("IBPRV\_INS\_SORT",$J)  ;  I IBLCT=0 D  G BLDQ ; No entries found  . D SET^VALM10(1," ")  . S Z=" No "\_$S(IBDSP="D":"default ",1:"")  . S Z=Z\_"ID's found for "\_$S(IBDSP="I":"provider "\_$S(IBSORT:"("\_$$EXPAND^IBTRE(355.9,.01,IBSORT)\_") ",1:"")\_"and ",IBDSP="A":"provider type "\_$S(IBSORT:"("\_$$EXPAND^IBTRE(355.9,.06,IBSORT)\_") ",1:"")\_"and ",1:"")\_"insurance co"  . D SET^VALM10(2,Z)  . S IBLCT=2  ; BLDQ S VALMCNT=IBLCT,VALMBG=1  Q  ; EXPND ;  Q  ; HELP ;  Q  ; EXIT ;  K IBFASTXT  D COPYPROV^IBCEP5A(IBINS)  K ^TMP("IBPRV\_INS\_ID",$J)  D CLEAN^VALM10  Q  ; SEL(IBDA,MANY) ; Select from provider id list  ; IBDA is passed by reference and IBDA(1) returned containing  ; ien's of the provider id records selected (file 355.9).  ; If > 1 entry can be selected, MANY is set to 1  N Z  S IBDA=0  D EN^VALM2($G(XQORNOD(0)),$S($G(MANY):"",1:"S"))  S Z=0 F  S Z=$O(VALMY(Z)) Q:'Z  S IBDA=IBDA+1,IBDA(IBDA)=+$G(^TMP("IBPRV\_INS\_ID",$J,"ZIDX",Z))\_U\_$G(^(Z,"PRV"))  Q  ; ENX(IBINS1) ; Insurance co level defaults for all providers or  ; for all providers by care unit  N DIC,DIE,DR,DA,X,Y,DLAYGO  I '$G(IBINS1) D  G:'$G(IBINS1) ENQ  . S DIC="^IBA(355.91,",DIC(0)="AELMQ",DLAYGO=355.91 D ^DIC  . I Y>0 S IBINS1=+Y  S DIE="^IBA(355.91,",DA=IBINS1,DR=".01;.06;.04;.05;.03;.07" D ^DIE  ; ENQ Q  ; PPTYP(IBINS) ; Returns the ien of the default performing provider type for   ; insurance company IBINS (ien file 36)  Q +$G(^DIC(36,+IBINS,4))  ; SCREEN(WHICH) ; This screen is used the menu protocol to screen out the ID functions if it is a child ins co  Q:'$G(DA) 0  Q:'$G(DA(1)) 0  N FILE,IENS,FIELD,FLAG,TARGET  S FILE=101.01,IENS=DA\_","\_DA(1),FIELD=".01",FLAG="I"  D GETS^DIQ(FILE,IENS,FIELD,FLAG,"TARGET")  Q:'$D(TARGET) 0  N IEN  S IEN=$G(TARGET(FILE,IENS\_",",FIELD,FLAG))  Q:'+IEN 0  S FILE=101,FIELD=1,FLAG="E"  K TARGET  D GETS^DIQ(FILE,IEN,FIELD,FLAG,"TARGET")  Q:'$D(TARGET) 0  I $G(TARGET(FILE,IEN\_",",FIELD,FLAG))'[WHICH Q 1  Q:'$G(IBINS) 0  N PCF  S PCF=$P($G(^DIC(36,+IBINS,3)),U,13)  I PCF="C" Q 0  Q 1 | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCEP2 | | | | | | | | |
| **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** | | | | | | | | | |
| IBCEP2 ;ALB/TMP - EDI UTILITIES for provider ID ;13-DEC-99  ;;2.0;INTEGRATED BILLING;\*\*137,181,232,280,320,349,432\*\*;21-MAR-94;Build 192  ;;Per VHA Directive 2004-038, this routine should not be modified.  ; DBIA for access to fields 53.2,54.1,54.2 in file 200: 224  ; GETID(IBIFN,IBTYPE,IBPROV,IBSEQ,IBT,IBT1,IBFUNC) ; Extract IBTYPE id for the bill  ; IBIFN = bill ien (file 399)  ; IBTYPE = 2:PERFORMING PROVIDER ID (1 and 3 deleted)  ; IBSEQ = numeric COB sequence of the insurance on bill  ; IBFUNC = 1:REFERRING;2:OPERATING;3:RENDERING;4:ATTENDING;5:SUPERVISING;9:OTHER;  ; Returns IBT = ien of the provider id type^ien of entry^file # for id  ;  S IBT=0  Q:IBTYPE'=2 ""  N IBID,IBPTYP  S IBID=$$IDFIND(IBIFN,"",IBPROV,IBSEQ,1,.IBT,$G(IBFUNC))  I IBID="" S IBT=""  ;  Q IBID  ; IDFIND(IBIFN,IBPTYP,IBPROV,IBSEQ,IBPERF,IBT,IBFUNC) ;Loop thru source levels  ; (if id definition allows) to find correct ID  ; IBIFN = bill ien (file 399)  ; IBPTYP = ien of the provider id type in file 355.97 or if null,  ; the default performing provider ID type for the ins co. in  ; COB sequence IBSEQ will be calculated  ; IBPROV = (variable pointer syntax) provider on bill IBIFN  ; IBSEQ = numeric COB sequence of the bill  ; IBPERF = 1 if the performing provider id is needed  ; IBFUNC = 1:REFERRING;2:OPERATING;3:RENDERING;4:ATTENDING;5:SUPERVISING;9:OTHER;  ; Returns IBT = ptr to file 355.97^entry #^file #  ;  S IBT=+$G(IBPTYP)  Q:'$G(IBPERF)!'$G(IBPROV) ""  N IBSPEC,IBINS,IBINS4,IBSRC,IBUP,IBID,IBALT,IBPROF,Z  I $G(IBSEQ)="" S IBSEQ=+$$COBN^IBCEF(IBIFN) ; Default to current COB seq  S IBINS=+$P($G(^DGCR(399,IBIFN,"I"\_IBSEQ)),U),IBINS4=$G(^DIC(36,+IBINS,4))  S IBPROF=($$FT^IBCEF(IBIFN)=2) S:'IBPROF IBPROF=2  ; form type is CMS-1500 (prof)=1, UB-04 (inst)=2  I $G(IBPTYP)="",$G(IBFUNC)=1,IBPROF=1 S (IBT,IBPTYP)=+$P(IBINS4,U,4) ; Referring Default ID on CMS-1500  I $G(IBPTYP)="" S (IBT,IBPTYP)=+$P(IBINS4,U,IBPROF) ; Def to perf prv typ for form  I 'IBPTYP Q ""  ; No default id type  S IBSPEC=$G(^IBE(355.97,IBPTYP,1)),IBSRC=$P($G(^IBE(355.97,+IBPTYP,0)),U,2),IBSRC=$S('IBSRC:5,1:IBSRC),IBUP=1  S IBALT=0  ;  F  D  Q:'IBUP!($G(IBID)'="") S IBSRC=IBSRC-1 Q:'IBSRC  . ;  . I IBSRC=1,$TR($P(IBSPEC,U,1,3),"^0")'="" D  Q  ; Indiv prov default  .. N IBSTATE  .. I $P(IBSPEC,U,2) D  Q  ; Federal DEA # from field 53.2 file 200  ... S IBID=$P($G(^VA(200,+IBPROV,"PS")),U,2) ; DBIA224  ... S $P(IBT,U,2,3)=(IBPROV\_U\_200)  .. S IBSTATE=+$$CAREST^IBCEP2A(IBIFN)  .. I $P(IBSPEC,U) D  Q  ; State issued DEA # needed  ... Q:'IBSTATE  ... ; Extract the state issuing DEA # from field 54.2 file 200  ... S Z=+$O(^VA(200,+IBPROV,"PS2","B",IBSTATE,0)),IBID=$P($G(^VA(200,+IBPROV,"PS2",Z,0)),U,2) ; DBIA224  ... S $P(IBT,U,2,3)=(+IBPROV\_";"\_Z\_U\_200)  .. I $P(IBSPEC,U,3) D  Q  ; State license # needed  ... Q:'IBSTATE  ... ; Extract the state license # from field 54.1 file 200  ... I IBPROV["VA(200" S Z=+$O(^VA(200,+IBPROV,"PS1","B",IBSTATE,0)),IBID=$P($G(^VA(200,+IBPROV,"PS1",Z,0)),U,2),$P(IBT,U,2,3)=(+IBPROV\_";"\_IBSTATE\_U\_200) ; DBIA224  ... I IBPROV["IBA(355.93" S IBID=$P($G(^IBA(355.93,+IBPROV,0)),U,12),$P(IBT,U,2,3)=(+IBPROV\_U\_355.93)  . ;  . I IBSRC=2,$P(IBSPEC,U,4) D  Q  ; FACILITY FED TAX ID #  .. N IBXDATA  .. D F^IBCEF("N-FEDERAL TAX ID",,,IBIFN)  .. S IBID=IBXDATA,$P(IBT,U,2,3)=(U\_350.9)  . ;  . I IBSRC=1 S IBID=$$SRC1(IBIFN,"\*ALL\*",IBPTYP,IBPROV,.IBT) Q  . ;  . I IBSRC=2 S IBID=$$SRC2(IBPTYP,.IBT) Q  . ;  . I IBSRC=3 S IBID=$$SRC3(IBIFN,IBINS,IBPTYP,.IBT) Q  . ;   . I IBSRC=4 S IBID=$$SRC4(IBIFN,IBINS,IBPTYP,IBPROV,.IBT) Q  . ;  . I IBSRC=5 S IBID=$$SRC5(IBIFN,IBINS,IBPTYP,IBSEQ,.IBT,$G(IBFUNC)) Q  . ;  . I IBSRC=6 S IBID=$$SRC6(IBIFN,IBINS,IBPTYP,IBPROV,IBSEQ,.IBT) Q  ;  Q $G(IBID)  ; GETALL(IBTYPE,IBIFN,IBPROV,IBPID) ; Extract all performing prov id's for a  ; provider (IBPROV - vp format) on bill IBIFN  ; IBTYPE = type of ID to return (see GETID above)  ;  ; Returns array IBPID(COB SEQ #)=id (pass by reference) AND  ; IBPID(COB SEQ #,1)=ien of id type (ptr to 355.97)  ; IBPID = current insurance co's id  ;  N Z,COB,Z1,IBT  S COB=$$COBN^IBCEF(IBIFN)  F Z=1:1:3 Q:'$D(^DGCR(399,IBIFN,"I"\_Z)) S IBPID(Z)=$$GETID(IBTYPE,IBIFN,IBPROV,Z,.IBT),IBPID(Z,1)=IBT I Z=COB S Z1=IBPID(Z)  Q $G(Z1)  ; SRC1(IBIFN,IBINS,IBPTYP,IBPROV,IBT) ; Licensing/gov't issued # - provider specific  ; Parameter definitions for SRC1, SRC3, SRC4, SRC5, SRC6:  ; IBIFN = ien of bill (file 399)  ; IBINS = ien of insurance co (file 36) or \*ALL\* for all insurance  ; (always \*ALL\* for SRC1)  ; IBPTYP = ien of the provider id type in file 355.97  ; IBPROV = (variable pointer syntax) provider on bill IBIFN  ; IBT = returned as type ien^file ien^file #  ;  N IBID,IB,IBRX,IBIDSV  S IBID="",IB=0,IBRX=$$ISRX^IBCEF1(IBIFN),IBIDSV=""  I $G(IBPROV) F  S IB=$O(^IBA(355.9,"AD",IBPTYP,IBPROV,IBINS,IB)) Q:'IB  D  Q:IBID'=""  . S IBID=$$UNIQ1(IBIFN,IBINS,IBPTYP,IBPROV,"",IB)  . I IBRX,$P($G(^IBA(355.9,IB,0)),U,5)'=3 S:IBIDSV="" IBIDSV=IBID S IBID="" ; Save 1st 'match' if no rx specific id  I IBID="",IBIDSV'="" S IBID=IBIDSV  Q IBID  ; SRC2(IB35597,IBT) ; Facility default - all providers  ; IB35597 = ien of the provider id type entry in file 355.97  ; IBT = returned as type ien^file ien^file #  ;  S $P(IBT,U,2,3)=(+IB35597\_U\_355.97)  Q $P($G(^IBE(355.97,+IB35597,0)),U,4)  ; SRC3(IBIFN,IBINS,IBPTYP,IBT) ; Ins co/all providers  ; See SRC1 for parameter definitions  N IB,IBID,IBRX,IBIDSV  S IBID="",IB=0,IBRX=$$ISRX^IBCEF1(IBIFN),IBIDSV=""  F  S IB=$O(^IBA(355.91,"AC",IBINS,IBPTYP,"\*N/A\*",IB)) Q:'IB  D  Q:IBID'=""  . S IBID=$$UNIQ2(IBIFN,IBINS,IBPTYP,"",IB,.IBT)  . I IBRX,$P($G(^IBA(355.91,IB,0)),U,5)'=3 S:IBIDSV="" IBIDSV=IBID S IBID="" ; Save 1st 'match' if no rx specific id  I IBID="",IBIDSV'="" S IBID=IBIDSV  Q IBID  ; SRC4(IBIFN,IBINS,IBPTYP,IBPROV,IBT) ; Insurance co/individual provider  ; See SRC1 for parameter definitions  ;  N IBID,IB,IBRX,IBIDSV  S IBID="",IB=0,IBRX=$$ISRX^IBCEF1(IBIFN),IBIDSV=""  I $G(IBPROV) F  S IB=$O(^IBA(355.9,"AD",IBPTYP,IBPROV,IBINS,IB)) Q:'IB  D  Q:IBID'=""  . S IBID=$$UNIQ1(IBIFN,IBINS,IBPTYP,IBPROV,"",IB,.IBT)  . I IBRX,$P($G(^IBA(355.9,IB,0)),U,5)'=3 S:IBIDSV="" IBIDSV=IBID S IBID="" ; Save 1st 'match' if no rx specific id  I IBID="",IBIDSV'="" S IBID=IBIDSV  Q IBID  ; SRC5(IBIFN,IBINS,IBPTYP,IBSEQ,IBT,IBFUNC) ; Ins co/all providers/care unit  ; See SRC1 for missing parameter definitions  ; IBSEQ = the numeric COB sequence of the insurance on the bill  ;  Q ""  ;DEM;432 - Pieces 9, 10, and 11 were deleted in 2006. So, code doesn't do anything other than return NULL.  N IBP,IBUNIT,IBID,IB,Z,IBIDSV,IBRX  S IBID="",Z=0,IBRX=$$ISRX^IBCEF1(IBIFN),IBIDSV=""  ; DEM;432 - IBLNPRV variable is a flag to indicate if user input  ; is claim level provider or line level provider user input.  ; DEM;432 - Line provider interested in fuction 1 and 3, referring and rendering respectively.  I '$G(IBLNPRV) S IBP=+$O(^DGCR(399,IBIFN,"PRV","B",$S($G(IBFUNC)=1:1,$$FT^IBCEF(IBIFN)=3:4,1:3),0)),IBUNIT=$P($G(^DGCR(399,IBIFN,"PRV",IBP,0)),U,8+IBSEQ)  I $G(IBLNPRV) S IBP=+$O(^DGCR(399,IBIFN,"CP",IBLNPRV("PROCIEN"),"LNPRV","B",$S($G(IBFUNC)=1:1,1:3),0)),IBUNIT=$P($G(^DGCR(399,IBIFN,"CP",IBLNPRV("PROCIEN"),"LNPRV",IBP,0)),U,8+IBSEQ)  I IBUNIT'="" F  S Z=$O(^IBA(355.96,"AC",IBINS,IBPTYP,Z)) Q:'Z  D  Q:IBID'=""  . S IB=0 F  S IB=$O(^IBA(355.91,"ACARE",Z,IB)) Q:'IB  D  Q:IBID'=""  .. S IBID=$$UNIQ2(IBIFN,IBINS,IBPTYP,IBUNIT,IB,.IBT)  .. I IBRX,$P($G(^IBA(355.91,IB,0)),U,5)'=3 S:IBIDSV="" IBIDSV=IBID S IBID="" ; Save 1st 'match' if no rx specific id  I IBID="",IBIDSV'="" S IBID=IBIDSV  Q IBID  ; SRC6(IBIFN,IBINS,IBPTYP,IBPROV,IBSEQ,IBT) ; Ins co/ind provider/care unit  ; See SRC1 for missing parameter definitions  ; IBSEQ = the numeric COB sequence of the insurance on the bill  ;  Q ""  ;DEM;432 - Pieces 9, 10, and 11 were deleted in 2006. So, code doesn't do anything other than return NULL.  N IBUNIT,IBP,IBID,IB  S IBID="",IB=0  I '$G(IBLNPRV) S IBP=+$O(^DGCR(399,"PRV","B",$S($$FT^IBCEF(IBIFN)=3:3,1:4),0)),IBUNIT=$P($G(^DGCR(399,IBIFN,"PRV",IBP,0)),U,8+IBSEQ)  I $G(IBLNPRV) S IBP=+$O(^DGCR(399,IBIFN,"CP",IBLNPRV("PROCIEN"),"LNPRV","B",$S($$FT^IBCEF(IBIFN)=3:3,1:4),0)),IBUNIT=$P($G(^DGCR(399,IBIFN,"CP",IBLNPRV("PROCIEN"),"LNPRV",IBP,0)),U,8+IBSEQ)  I $G(IBPROV),IBUNIT'="" F  S IB=$O(^IBA(355.9,"AD",IBPTYP,IBPROV,IBINS,IB)) Q:'IB  D  Q:IBID'=""  . S IBID=$$UNIQ1(IBIFN,IBINS,IBPTYP,IBPROV,IBUNIT,IB,.IBT)  Q IBID  ; 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  ;  Q $$UNIQ1^IBCEP2A($G(IBIFN),$G(IBINS),$G(IBPTYP),$G(IBPROV),$G(IBUNIT),$G(IBCU),$G(IBT))  ; UNIQ2(IBIFN,IBINS,IBPTYP,IBUNIT,IBCU,IBT) ; Match on most-least specific  ; \*\*\* SEE PARAMETER DEFINITIONS IN IBCEP3 \*\*\*  ;  ; Start in file 355.91 (Specific Insurance)  ;  Q $$UNIQ2^IBCEP2A($G(IBIFN),$G(IBINS),$G(IBPTYP),$G(IBUNIT),$G(IBCU),$G(IBT)) | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCEP2 ;ALB/TMP - EDI UTILITIES for provider ID ;13-DEC-99  ;;2.0;INTEGRATED BILLING;\*\*137,181,232,280,320,349,432,**592\***\*;21-MAR-94;Build 192  ;;Per VHA Directive 2004-038, this routine should not be modified.  ; DBIA for access to fields 53.2,54.1,54.2 in file 200: 224  ; GETID(IBIFN,IBTYPE,IBPROV,IBSEQ,IBT,IBT1,IBFUNC) ; Extract IBTYPE id for the bill  ; IBIFN = bill ien (file 399)  ; IBTYPE = 2:PERFORMING PROVIDER ID (1 and 3 deleted)  ; IBSEQ = numeric COB sequence of the insurance on bill  ; IBFUNC = 1:REFERRING;2:OPERATING;3:RENDERING;4:ATTENDING;5:SUPERVISING; 6:ASSISTANT SURGEON;9:OTHER;  ; Returns IBT = ien of the provider id type^ien of entry^file # for id  ;  S IBT=0  Q:IBTYPE'=2 ""  N IBID,IBPTYP  S IBID=$$IDFIND(IBIFN,"",IBPROV,IBSEQ,1,.IBT,$G(IBFUNC))  I IBID="" S IBT=""  ;  Q IBID  ; IDFIND(IBIFN,IBPTYP,IBPROV,IBSEQ,IBPERF,IBT,IBFUNC) ;Loop thru source levels  ; (if id definition allows) to find correct ID  ; IBIFN = bill ien (file 399)  ; IBPTYP = ien of the provider id type in file 355.97 or if null,  ; the default performing provider ID type for the ins co. in  ; COB sequence IBSEQ will be calculated  ; IBPROV = (variable pointer syntax) provider on bill IBIFN  ; IBSEQ = numeric COB sequence of the bill  ; IBPERF = 1 if the performing provider id is needed  ; IBFUNC = 1:REFERRING;2:OPERATING;3:RENDERING;4:ATTENDING;5:SUPERVISING;9:OTHER;  ; Returns IBT = ptr to file 355.97^entry #^file #  ;  S IBT=+$G(IBPTYP)  Q:'$G(IBPERF)!'$G(IBPROV) ""  N IBSPEC,IBINS,IBINS4,IBSRC,IBUP,IBID,IBALT,IBPROF,Z  I $G(IBSEQ)="" S IBSEQ=+$$COBN^IBCEF(IBIFN) ; Default to current COB seq  S IBINS=+$P($G(^DGCR(399,IBIFN,"I"\_IBSEQ)),U),IBINS4=$G(^DIC(36,+IBINS,4)) **;JRA IB\*2.0\*592 Same logic for Dental Form 7 as for CMS-1500** **;**S IBPROF=($$FT^IBCEF(IBIFN)=2) S:'IBPROF IBPROF=2 ;JRA IB\*2.0\*592 ';'  **N FT S FT=$$FT^IBCEF(IBIFN) ;JRA IB\*2.0\*592 Added 'FT'**  **S IBPROF=(FT=2!(FT=7)) S:'IBPROF IBPROF=2 ;JRA IB\*2.0\*592**  ; form type is CMS-1500 (prof)=1, UB-04 (inst)=2  **;JWS;IB\*2.0\*592;**  **I $G(IBPTYP)="",FT=7,$G(IBFUNC)=1,IBPROF=1 S (IBT,IBPTYP)=+$P(IBINS4,U,15) ;Referring Default ID on J430D**  I $G(IBPTYP)="",$G(IBFUNC)=1,IBPROF=1 S (IBT,IBPTYP)=+$P(IBINS4,U,4) ; Referring Default ID on CMS-1500  I $G(IBPTYP)="" S (IBT,IBPTYP)=+$P(IBINS4,U,IBPROF) ; Def to perf prv typ for form  I 'IBPTYP Q ""  ; No default id type  S IBSPEC=$G(^IBE(355.97,IBPTYP,1)),IBSRC=$P($G(^IBE(355.97,+IBPTYP,0)),U,2),IBSRC=$S('IBSRC:5,1:IBSRC),IBUP=1  S IBALT=0  ;  F  D  Q:'IBUP!($G(IBID)'="") S IBSRC=IBSRC-1 Q:'IBSRC  . ;  . I IBSRC=1,$TR($P(IBSPEC,U,1,3),"^0")'="" D  Q  ; Indiv prov default  .. N IBSTATE  .. I $P(IBSPEC,U,2) D  Q  ; Federal DEA # from field 53.2 file 200  ... S IBID=$P($G(^VA(200,+IBPROV,"PS")),U,2) ; DBIA224  ... S $P(IBT,U,2,3)=(IBPROV\_U\_200)  .. S IBSTATE=+$$CAREST^IBCEP2A(IBIFN)  .. I $P(IBSPEC,U) D  Q  ; State issued DEA # needed  ... Q:'IBSTATE  ... ; Extract the state issuing DEA # from field 54.2 file 200  ... S Z=+$O(^VA(200,+IBPROV,"PS2","B",IBSTATE,0)),IBID=$P($G(^VA(200,+IBPROV,"PS2",Z,0)),U,2) ; DBIA224  ... S $P(IBT,U,2,3)=(+IBPROV\_";"\_Z\_U\_200)  .. I $P(IBSPEC,U,3) D  Q  ; State license # needed  ... Q:'IBSTATE  ... ; Extract the state license # from field 54.1 file 200  ... I IBPROV["VA(200" S Z=+$O(^VA(200,+IBPROV,"PS1","B",IBSTATE,0)),IBID=$P($G(^VA(200,+IBPROV,"PS1",Z,0)),U,2),$P(IBT,U,2,3)=(+IBPROV\_";"\_IBSTATE\_U\_200) ; DBIA224  ... I IBPROV["IBA(355.93" S IBID=$P($G(^IBA(355.93,+IBPROV,0)),U,12),$P(IBT,U,2,3)=(+IBPROV\_U\_355.93)  . ;  . I IBSRC=2,$P(IBSPEC,U,4) D  Q  ; FACILITY FED TAX ID #  .. N IBXDATA  .. D F^IBCEF("N-FEDERAL TAX ID",,,IBIFN)  .. S IBID=IBXDATA,$P(IBT,U,2,3)=(U\_350.9)  . ;  . I IBSRC=1 S IBID=$$SRC1(IBIFN,"\*ALL\*",IBPTYP,IBPROV,.IBT) Q  . ;  . I IBSRC=2 S IBID=$$SRC2(IBPTYP,.IBT) Q  . ;  . I IBSRC=3 S IBID=$$SRC3(IBIFN,IBINS,IBPTYP,.IBT) Q  . ;   . I IBSRC=4 S IBID=$$SRC4(IBIFN,IBINS,IBPTYP,IBPROV,.IBT) Q  . ;  . I IBSRC=5 S IBID=$$SRC5(IBIFN,IBINS,IBPTYP,IBSEQ,.IBT,$G(IBFUNC)) Q  . ;  . I IBSRC=6 S IBID=$$SRC6(IBIFN,IBINS,IBPTYP,IBPROV,IBSEQ,.IBT) Q  ;  Q $G(IBID)  ; GETALL(IBTYPE,IBIFN,IBPROV,IBPID) ; Extract all performing prov id's for a  ; provider (IBPROV - vp format) on bill IBIFN  ; IBTYPE = type of ID to return (see GETID above)  ;  ; Returns array IBPID(COB SEQ #)=id (pass by reference) AND  ; IBPID(COB SEQ #,1)=ien of id type (ptr to 355.97)  ; IBPID = current insurance co's id  ;  N Z,COB,Z1,IBT  S COB=$$COBN^IBCEF(IBIFN)  F Z=1:1:3 Q:'$D(^DGCR(399,IBIFN,"I"\_Z)) S IBPID(Z)=$$GETID(IBTYPE,IBIFN,IBPROV,Z,.IBT),IBPID(Z,1)=IBT I Z=COB S Z1=IBPID(Z)  Q $G(Z1)  ; SRC1(IBIFN,IBINS,IBPTYP,IBPROV,IBT) ; Licensing/gov't issued # - provider specific  ; Parameter definitions for SRC1, SRC3, SRC4, SRC5, SRC6:  ; IBIFN = ien of bill (file 399)  ; IBINS = ien of insurance co (file 36) or \*ALL\* for all insurance  ; (always \*ALL\* for SRC1)  ; IBPTYP = ien of the provider id type in file 355.97  ; IBPROV = (variable pointer syntax) provider on bill IBIFN  ; IBT = returned as type ien^file ien^file #  ;  N IBID,IB,IBRX,IBIDSV  S IBID="",IB=0,IBRX=$$ISRX^IBCEF1(IBIFN),IBIDSV=""  I $G(IBPROV) F  S IB=$O(^IBA(355.9,"AD",IBPTYP,IBPROV,IBINS,IB)) Q:'IB  D  Q:IBID'=""  . S IBID=$$UNIQ1(IBIFN,IBINS,IBPTYP,IBPROV,"",IB)  . I IBRX,$P($G(^IBA(355.9,IB,0)),U,5)'=3 S:IBIDSV="" IBIDSV=IBID S IBID="" ; Save 1st 'match' if no rx specific id  I IBID="",IBIDSV'="" S IBID=IBIDSV  Q IBID  ; SRC2(IB35597,IBT) ; Facility default - all providers  ; IB35597 = ien of the provider id type entry in file 355.97  ; IBT = returned as type ien^file ien^file #  ;  S $P(IBT,U,2,3)=(+IB35597\_U\_355.97)  Q $P($G(^IBE(355.97,+IB35597,0)),U,4)  ; SRC3(IBIFN,IBINS,IBPTYP,IBT) ; Ins co/all providers  ; See SRC1 for parameter definitions  N IB,IBID,IBRX,IBIDSV  S IBID="",IB=0,IBRX=$$ISRX^IBCEF1(IBIFN),IBIDSV=""  F  S IB=$O(^IBA(355.91,"AC",IBINS,IBPTYP,"\*N/A\*",IB)) Q:'IB  D  Q:IBID'=""  . S IBID=$$UNIQ2(IBIFN,IBINS,IBPTYP,"",IB,.IBT)  . I IBRX,$P($G(^IBA(355.91,IB,0)),U,5)'=3 S:IBIDSV="" IBIDSV=IBID S IBID="" ; Save 1st 'match' if no rx specific id  I IBID="",IBIDSV'="" S IBID=IBIDSV  Q IBID  ; SRC4(IBIFN,IBINS,IBPTYP,IBPROV,IBT) ; Insurance co/individual provider  ; See SRC1 for parameter definitions  ;  N IBID,IB,IBRX,IBIDSV  S IBID="",IB=0,IBRX=$$ISRX^IBCEF1(IBIFN),IBIDSV=""  I $G(IBPROV) F  S IB=$O(^IBA(355.9,"AD",IBPTYP,IBPROV,IBINS,IB)) Q:'IB  D  Q:IBID'=""  . S IBID=$$UNIQ1(IBIFN,IBINS,IBPTYP,IBPROV,"",IB,.IBT)  . I IBRX,$P($G(^IBA(355.9,IB,0)),U,5)'=3 S:IBIDSV="" IBIDSV=IBID S IBID="" ; Save 1st 'match' if no rx specific id  I IBID="",IBIDSV'="" S IBID=IBIDSV  Q IBID  ; SRC5(IBIFN,IBINS,IBPTYP,IBSEQ,IBT,IBFUNC) ; Ins co/all providers/care unit  ; See SRC1 for missing parameter definitions  ; IBSEQ = the numeric COB sequence of the insurance on the bill  ;  Q ""  ;DEM;432 - Pieces 9, 10, and 11 were deleted in 2006. So, code doesn't do anything other than return NULL.  N IBP,IBUNIT,IBID,IB,Z,IBIDSV,IBRX  S IBID="",Z=0,IBRX=$$ISRX^IBCEF1(IBIFN),IBIDSV=""  ; DEM;432 - IBLNPRV variable is a flag to indicate if user input  ; is claim level provider or line level provider user input.  ; DEM;432 - Line provider interested in fuction 1 and 3, referring and rendering respectively.  I '$G(IBLNPRV) S IBP=+$O(^DGCR(399,IBIFN,"PRV","B",$S($G(IBFUNC)=1:1,$$FT^IBCEF(IBIFN)=3:4,1:3),0)),IBUNIT=$P($G(^DGCR(399,IBIFN,"PRV",IBP,0)),U,8+IBSEQ)  I $G(IBLNPRV) S IBP=+$O(^DGCR(399,IBIFN,"CP",IBLNPRV("PROCIEN"),"LNPRV","B",$S($G(IBFUNC)=1:1,1:3),0)),IBUNIT=$P($G(^DGCR(399,IBIFN,"CP",IBLNPRV("PROCIEN"),"LNPRV",IBP,0)),U,8+IBSEQ)  I IBUNIT'="" F  S Z=$O(^IBA(355.96,"AC",IBINS,IBPTYP,Z)) Q:'Z  D  Q:IBID'=""  . S IB=0 F  S IB=$O(^IBA(355.91,"ACARE",Z,IB)) Q:'IB  D  Q:IBID'=""  .. S IBID=$$UNIQ2(IBIFN,IBINS,IBPTYP,IBUNIT,IB,.IBT)  .. I IBRX,$P($G(^IBA(355.91,IB,0)),U,5)'=3 S:IBIDSV="" IBIDSV=IBID S IBID="" ; Save 1st 'match' if no rx specific id  I IBID="",IBIDSV'="" S IBID=IBIDSV  Q IBID  ; SRC6(IBIFN,IBINS,IBPTYP,IBPROV,IBSEQ,IBT) ; Ins co/ind provider/care unit  ; See SRC1 for missing parameter definitions  ; IBSEQ = the numeric COB sequence of the insurance on the bill  ;  Q ""  ;DEM;432 - Pieces 9, 10, and 11 were deleted in 2006. So, code doesn't do anything other than return NULL.  N IBUNIT,IBP,IBID,IB  S IBID="",IB=0  I '$G(IBLNPRV) S IBP=+$O(^DGCR(399,"PRV","B",$S($$FT^IBCEF(IBIFN)=3:3,1:4),0)),IBUNIT=$P($G(^DGCR(399,IBIFN,"PRV",IBP,0)),U,8+IBSEQ)  I $G(IBLNPRV) S IBP=+$O(^DGCR(399,IBIFN,"CP",IBLNPRV("PROCIEN"),"LNPRV","B",$S($$FT^IBCEF(IBIFN)=3:3,1:4),0)),IBUNIT=$P($G(^DGCR(399,IBIFN,"CP",IBLNPRV("PROCIEN"),"LNPRV",IBP,0)),U,8+IBSEQ)  I $G(IBPROV),IBUNIT'="" F  S IB=$O(^IBA(355.9,"AD",IBPTYP,IBPROV,IBINS,IB)) Q:'IB  D  Q:IBID'=""  . S IBID=$$UNIQ1(IBIFN,IBINS,IBPTYP,IBPROV,IBUNIT,IB,.IBT)  Q IBID  ; UNIQ1(IBIFN,IBINS,IBPTYP,IBPROV,IBUNIT,IBCU,IBT) ; Match most-least specific  ; \*\*\* SEE PARAMETER DEFINITIONS IN IBCEP3 \*\*\*  ;  ; Start in file 355.9 (Specific Provider)  ; IBPROV = (variable pointer syntax) provider on bill IBIFN  ;  Q $$UNIQ1^IBCEP2A($G(IBIFN),$G(IBINS),$G(IBPTYP),$G(IBPROV),$G(IBUNIT),$G(IBCU),$G(IBT))  ; UNIQ2(IBIFN,IBINS,IBPTYP,IBUNIT,IBCU,IBT) ; Match on most-least specific  ; \*\*\* SEE PARAMETER DEFINITIONS IN IBCEP3 \*\*\*  ;  ; Start in file 355.91 (Specific Insurance)  ;  Q $$UNIQ2^IBCEP2A($G(IBIFN),$G(IBINS),$G(IBPTYP),$G(IBUNIT),$G(IBCU),$G(IBT)) | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCEP2B | | | | | | | | |
| **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** | | | | | | | | | |
| IBCEP2B ;ALB/TMP - EDI UTILITIES for provider ID ;18-MAY-04  ;;2.0;INTEGRATED BILLING;\*\*232,320,400,432\*\*;21-MAR-94;Build 192  ;;Per VHA Directive 2004-038, this routine should not be modified.  ; PROVID(IBIFN,IBPRIEN,IBCOBN,DIPA) ; Provider id entry on billing screen 10, and line level provider input on billing screens 4&5.  ; IBIFN = ien file 399  ; IBPRIEN = ien file 399.0222, or ien file 399.0404.  ; IBCOBN = the COB number of the id being edited  ; DIPA = passed by ref, returned with id data  ; DIPA("EDIT")=-1 if no id editing = 1 if edit id = 2 if stuff id  ; DIPA("PRID")= id to stuff DIPA("PRIDT")= id type to stuff  N PRN0,Z  Q:'$G(^DGCR(399,IBIFN,"I1"))  I $G(IBLNPRV),'$G(IBLNPRV("LNPRVIEN")),'$G(IBLNPRV("PROCIEN")) Q  ; DEM;432 - If line provider user input.  ; DEM;432 - Updated variable PRNO to be equal to line level provider if we are coming from line level provider user input.  S PRN0=$S($G(IBLNPRV):$G(^DGCR(399,IBIFN,"CP",IBLNPRV("PROCIEN"),"LNPRV",IBLNPRV("LNPRVIEN"),0)),1:$G(^DGCR(399,IBIFN,"PRV",IBPRIEN,0)))  S DIPA("EDIT")=1,(DIPA("PRID"),DIPA("PRIDT"))=""  W @IOF  W !,?19,"\*\*\*\* SECONDARY PERFORMING PROVIDER IDs \*\*\*\*"  W !!,$P("PRIMARY^SECONDARY^TERTIARY",U,IBCOBN)\_" INSURANCE CO: "\_$P($G(^DIC(36,+$G(^DGCR(399,IBIFN,"I"\_IBCOBN)),0)),U)  ; DEM;432 - Added line and conditions if line level provider user input.  I '$G(IBLNPRV) W !,"PROVIDER: "\_$$EXTERNAL^DILFD(399.0222,.02,"",$P(PRN0,U,2))\_" ("\_$$EXTERNAL^DILFD(399.0222,.01,"",+PRN0)\_")",!  I $G(IBLNPRV) W !,"Line Level Provider: "\_$$EXTERNAL^DILFD(399.0404,.02,"",$P(PRN0,U,2))\_" ("\_$$EXTERNAL^DILFD(399.0404,.01,"",+PRN0)\_")",!  ;  I $P(PRN0,U,4+IBCOBN)="" K DIPA("PRID"),DIPA("PRIDT") D NEWID(IBIFN,IBPRIEN,IBCOBN,.DIPA) ; No id currently exists for the ins seq/prov  ;  Q  ; NEWID(IBIFN,IBPRIEN,IBCOBN,DIPA) ;  N IBDEF,IBCT,IBNUM,IBINS,IBFRM,IBCAR,IBARR,IBARRS,IB0,IBM,IBQUIT,IBSEL,PRN,PRT,PRN,PRN0,DIR,X,Y,Z,Z0,IBZ,IBZ1,IBTYP,IBREQ,IBREQT,IBTYPN,IBID,IBUSED  S IBREQ=0,IBREQT=""  Q:($G(IBLNPRV))&('$G(IBLNPRV("LNPRVIEN"))&'$G(IBLNPRV("PROCIEN"))) ; DEM;432 - If line provider user input.  ; DEM;432 - Updated variable PRNO to be equal to line level provider if we are coming from line level provider user input.  S PRN0=$S($G(IBLNPRV):$G(^DGCR(399,IBIFN,"CP",IBLNPRV("PROCIEN"),"LNPRV",IBLNPRV("LNPRVIEN"),0)),1:$G(^DGCR(399,IBIFN,"PRV",IBPRIEN,0)))  S Z(IBCOBN)=$S($G(DIPA("I"\_IBCOBN)):$$GETTYP^IBCEP2A(IBIFN,IBCOBN,$P(PRN0,U)),1:"")  S IBINS=+$G(^DGCR(399,IBIFN,"I"\_IBCOBN)),IB0=$S($G(IBLNPRV):$G(^DGCR(399,IBIFN,"CP",IBLNPRV("PROCIEN"),"LNPRV",IBLNPRV("LNPRVIEN"),0)),1:$G(^DGCR(399,IBIFN,"PRV",IBPRIEN,0)))  S IBCAR=$$INPAT^IBCEF(IBIFN),IBCAR=$S('IBCAR:2,1:1)  S IBFRM=$$FT^IBCEF(IBIFN),IBFRM=$S(IBFRM=2:2,1:1)  I $P(Z(IBCOBN),U) D  . W !,"INS. COMPANY'S DEFAULT SECONDARY ID TYPE IS: "\_$$EXTERNAL^DILFD(36,4.01,"",$P(Z(IBCOBN),U)) S IBREQT=+Z(IBCOBN)  . I $P(Z(IBCOBN),U,2) W !,?2," AND IS REQUIRED TO BE ENTERED FOR THIS CLAIM" S IBREQ=1  I $$CUNEED^IBCEP3(IBIFN,IBCOBN) W !,"CARE UNITS ARE DEFINED"\_$S($P($G(^DIC(36,IBINS,4)),U,9)'="":" AS "\_$P(^(4),U,9),1:"")\_" FOR THESE IDs"  D PRACT^IBCEF71(IBINS,IBFRM,IBCAR,$P(IB0,U,2),.IBARR,$P(IB0,U),$S($$COBN^IBCEF(IBIFN)=IBCOBN:"C",1:"O"),355.9,1)  S (IBNUM,IBCT)=0,IBDEF=""  I $O(IBARR(""))="" S IBCT=IBCT+1,DIR("A",IBCT)="NO SECONDARY IDS ARE DEFINED FOR THIS PROV THAT ARE VALID FOR THIS CLAIM"  S IBCT=IBCT+1,DIR("A",IBCT)="SELECT A SECONDARY ID OR ACTION FROM THE LIST BELOW: ",IBCT=IBCT+1,DIR("A",IBCT)=" "  ;  S IBCT=IBCT+1,IBNUM=IBNUM+1,DIR("A",IBCT)=" "\_$E(IBNUM\_$J("",3),1,3)\_" - NO SECONDARY ID NEEDED",IBNUM=IBNUM+1,IBCT=IBCT+1,DIR("A",IBCT)=" "\_$E(IBNUM\_$J("",3),1,3)\_" - ADD AN ID FOR THIS CLAIM ONLY"  I $O(IBARR(""))="" S IBDEF=1,DIPA("EDIT")=$$SELID(.DIR,IBDEF,.IBID,.DIPA,IBNUM) Q  ;  S PRN=$$GETID^IBCEP2(IBIFN,2,$P(PRN0,U,2),IBCOBN,.PRT,,$P(PRN0,U)),IBDEF=""  ;  I PRN'="",PRT D  . N PRT1  . S PRT1=$P($G(^IBE(355.97,+PRT,0)),U)  . I $P($G(^IBE(355.97,+PRT,1)),U,3) S PRT1="ST LIC("\_$P($G(^DIC(5,+$$CAREST^IBCEP2A(IBIFN),0)),U,2)\_")"  . S IBCT=IBCT+1,IBNUM=IBNUM+1  . S DIR("A",IBCT)=" "\_$E(IBNUM\_$J("",3),1,3)\_" - "\_$E("<DEFAULT> "\_PRN\_$J("",29),1,29)\_" "\_$E(PRT1\_$J("",15),1,15)  . S DIR("A",IBCT)=DIR("A",IBCT)\_" "\_$S($P(PRT,U,3)'["355.9":"",$P($G(^IBA(+$P(PRT,U,3),+$P(PRT,U,2),0)),U,3)'="":$$EXTERNAL^DILFD(355.9,.03,"",$P($G(^IBA(+$P(PRT,U,3),+$P(PRT,U,2),0)),U,3)),1:"")  . S IBID(IBNUM)=PRN\_U\_+PRT,IBDEF=IBNUM,IBID(IBNUM,1)=DIR("A",IBCT),IBDEF=IBNUM,IBDEF("IEN")=$P(PRT,U,2,3)  . S IBUSED(PRT,PRN,0)=""  ;  S IBQUIT=0,IBSEL=1  ; Sort ids by id type  S IBZ="" F  S IBZ=$O(IBARR(IBZ)) Q:IBZ=""  S IBZ1="" F  S IBZ1=$O(IBARR(IBZ,IBZ1)) Q:IBZ1=""  D  . S IBTYP=+$P(IBARR(IBZ,IBZ1),U,9)  . I $P(IBARR(IBZ,IBZ1),U,4)]"" Q:$D(IBUSED(IBTYP,$P(IBARR(IBZ,IBZ1),U,4),+$P(IBARR(IBZ,IBZ1),U,7)))  . I $P($G(IBDEF("IEN")),U,2)["355.9",$P(IBARR(IBZ,IBZ1),U,8),$P(IBARR(IBZ,IBZ1),U,8)=+$G(IBDEF("IEN")) Q:$S($P(IBZ1,U)'["INS DEF":$P($G(IBDEF("IEN")),U,2)=355.9,1:$P($G(IBDEF("IEN")),U,2)=355.91)  . S IBARRS(IBTYP,IBZ,IBZ1)=IBARR(IBZ,IBZ1)  . I $P(IBARR(IBZ,IBZ1),U,4)]"" S IBUSED(IBTYP,$P(IBARR(IBZ,IBZ1),U,4),+$P(IBARR(IBZ,IBZ1),U,7))=""  S IBTYP="" F  S IBTYP=$O(IBARRS(IBTYP)) Q:IBTYP=""  S IBZ="" F  S IBZ=$O(IBARRS(IBTYP,IBZ)) Q:IBZ=""  D  Q:IBQUIT  . S IBZ1="" F  S IBZ1=$O(IBARRS(IBTYP,IBZ,IBZ1)) Q:IBZ1=""  S IBCT=IBCT+1,IBNUM=IBNUM+1 D  Q:IBQUIT  .. S Z0=IBARRS(IBTYP,IBZ,IBZ1)  .. S IBARR=$S($P(Z0,U,8)&(IBZ1'["LIC"):$G(^IBA("355.9"\_$S($P(IBZ1,U)'="INS DEF":"",1:1),+$P(Z0,U,8),0)),1:"")  .. S IBTYPN=$S(IBTYP=+$$STLIC^IBCEP8():"ST LIC ("\_$P($G(^DIC(5,+$P(Z0,U,7),0)),U,2)\_")",1:$P($G(^IBE(355.97,IBTYP,0)),U))  .. S DIR("A",IBCT)=" "\_$E(IBNUM\_$J("",3),1,3)\_" - "\_$E($S($P(IBZ1,U)="INS DEF":"<INS DEF> ",1:"")\_$P(Z0,U,4)\_$J("",29),1,29)\_" "\_$E(IBTYPN\_$J("",15),1,15)\_" "\_$S($P(IBARR,U,3):$$EXTERNAL^DILFD(355.9,.03,"",$P(IBARR,U,3)),1:"")  .. S IBID(IBNUM,1)=DIR("A",IBCT),IBID(IBNUM)=$P(Z0,U,4)\_U\_IBTYP  .. I (IBNUM#15)=0 S IBM=$$MORE(.DIR) D  Q:IBQUIT  ... I IBM<0 S IBQUIT=1,IBSEL=0 Q  ; User aborted list  ... I 'IBM S IBQUIT=1 Q  ; User wants to select  ... W ! K DIR S IBCT=1  I 'IBSEL S DIPA("EDIT")=-1  I IBSEL S:IBDEF=""&$G(IBREQ) IBDEF=2 S DIPA("EDIT")=$$SELID(.DIR,IBDEF,.IBID,.DIPA,IBNUM)  Q  ; SELID(DIR,IBDEF,IBID,DIPA,IBNUM) ; Returns the selection from the array of possible IDs/ID actions  N IDACT,IDSEL,X,Y  S IDACT=""  S DIR("B")=$S('$G(IBDEF):1,1:IBDEF),DIR("A",+$O(DIR("A",""),-1)+1)=" "  S DIR(0)="NA^1:"\_IBNUM,DIR("A")="Selection: " W ! D ^DIR K DIR  I $D(DTOUT)!$D(DUOUT)!(Y=1) S IDACT=-1 G SELIDQ  I Y=2 S IDACT=1 G SELIDQ  S IDSEL=Y  S DIR("A",1)="ID SELECTED:",DIR("A",2)=" "\_$G(IBID(+Y,1)),DIR("A")="IS THIS CORRECT?: ",DIR("B")="YES",DIR(0)="YA" W ! D ^DIR K DIR  I Y'=1 S IDACT=-1 G SELIDQ  S DIPA("PRID")=$P(IBID(IDSEL),U),DIPA("PRIDT")=$P(IBID(IDSEL),U,2),IDACT=2  ; SELIDQ Q IDACT  ; MORE(DIR) ;  N DIR,X,Y,DUOUT,DTOUT  S DIR(0)="YA",DIR("A")="MORE?: ",DIR("B")="NO" W ! D ^DIR K DIR("B")  Q $S($D(DTOUT)!$D(DUOUT):-1,1:Y)  ;  ; IBFIDFL = E = Electronic Form Type  ; A = Additional ID's  ; LF - VA Lab/Facility FACID(IBINS,IBFIDFL) ; Enter/edit billing facility ids  ; IBINS = ien of ins co (file 36)  N IBID,Z,Z0,Y  K ^TMP($J,"IBBF\_ID")  W @IOF  D GETBPNUM(IBINS)  K ^TMP("IBCE\_PRVFAC\_MAINT\_INS",$J)  S ^TMP("IBCE\_PRVFAC\_MAINT\_INS",$J)=IBFIDFL\_U\_IBINS\_U\_"1"  D EN^VALM("IBCE PRVFAC MAINT")  K ^TMP("IBCE\_PRVFAC\_MAINT\_INS",$J)  W @IOF  D FULL^VALM1  Q  ; GETBPNUM(IBINS) ;  N Z,Z0,IBID,IBMAIN  S IBMAIN=$$MAIN(),^TMP($J,"IBBF\_ID")=IBMAIN  S IBID=$$BF^IBCU()  S Z=0 F  S Z=$O(^IBA(355.92,"B",IBINS,Z)) Q:'Z  D  . S Z0=$G(^IBA(355.92,Z,0))  . Q:$P(Z0,U,8)'="E"  ; WCJ 1/13/06 There are several ID types in this file   . Q:$P(Z0,U,3)]""  . S ^TMP($J,"IBBF\_ID",$S($P(Z0,U,5)=IBMAIN:0,1:+$P(Z0,U,5)),+$P(Z0,U,4))=$P(Z0,U,7)  . S ^TMP($J,"IBBF\_ID",$S($P(Z0,U,5)=IBMAIN:0,1:+$P(Z0,U,5)),+$P(Z0,U,4),"QUAL")=$P(Z0,U,6)  Q  ; MAIN() ; Returns ien of main division of the database  Q +$$PRIM^VASITE()  ; FACNUM(IBIFN,IBCOB,IBQF) ; Function returns the current division's fac billing  ; prov id for the COB insurance sequence from file 355.92  ; IBIFN = ien file 399  ; IBCOB = # of COB ins seq or if "", current assumed  ; IBQF - 1 if qualifier is to be returned instead of ID  N Z,IBDIV,IBFT,X,BPZ  S X="",IBDIV=0  S:'$G(IBCOB) IBCOB=+$$COBN^IBCEF(IBIFN)  ;  ; IB\*2\*400 - esg - 11/7/08 - Determine the division associated with the billing provider first  S BPZ=+$$B^IBCEF79(IBIFN,IBCOB) ; Inst file pointer as the billing provider for payer seq IBCOB  I BPZ S IBDIV=+$O(^DG(40.8,"AD",BPZ,0)) ; Billing Provider division (may not exist)  ;  I 'IBDIV S IBDIV=+$P($G(^DGCR(399,IBIFN,0)),U,22) ; Division on claim  I 'IBDIV S IBDIV=$$MAIN() ; main division  ;  S IBFT=$$FT^IBCEF(IBIFN),IBFT=$S(IBFT=3:1,1:2)  K ^TMP($J,"IBBF\_ID")  D GETBPNUM(+$P($G(^DGCR(399,IBIFN,"M")),U,IBCOB))  I IBDIV=+$G(^TMP($J,"IBBF\_ID")) S IBDIV=0  I '$G(IBQF) S X=$S($D(^TMP($J,"IBBF\_ID",IBDIV,IBFT)):^(IBFT),1:$G(^TMP($J,"IBBF\_ID",0,IBFT)))  I $G(IBQF) S X=$S($D(^TMP($J,"IBBF\_ID",IBDIV,IBFT,"QUAL")):^("QUAL"),1:$G(^TMP($J,"IBBF\_ID",0,IBFT,"QUAL")))  K ^TMP($J,"IBBF\_ID")  Q X  ; SOP(IBIFN,IBZD) ; Returns X12 current source of pay code for bill ien IBIFN  ; IBZD = the current ins policy type, if known  N IBZ  S IBZ=""  I $G(IBZD)="" D F^IBCEF("N-CURRENT INS POLICY TYPE","IBZD",,IBIFN)  S IBZ=$S($G(IBZD)="":"G2","MAMB16"[IBZD:"1C",IBZD="TV"!(IBZD="MC"):"1D",IBZD="CH":"1H",IBZD="BL":$S($$FT^IBCEF(IBIFN)=2:"1B",1:"1A"),1:"G2")  Q IBZ  ; | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCEP2B ;ALB/TMP - EDI UTILITIES for provider ID ;18-MAY-04  ;;2.0;INTEGRATED BILLING;\*\*232,320,400,432,**592**\*\*;21-MAR-94;Build 192  ;;Per VHA Directive 2004-038, this routine should not be modified.  ; PROVID(IBIFN,IBPRIEN,IBCOBN,DIPA) ; Provider id entry on billing screen 10, and line level provider input on billing screens 4&5.  ; IBIFN = ien file 399  ; IBPRIEN = ien file 399.0222, or ien file 399.0404.  ; IBCOBN = the COB number of the id being edited  ; DIPA = passed by ref, returned with id data  ; DIPA("EDIT")=-1 if no id editing = 1 if edit id = 2 if stuff id  ; DIPA("PRID")= id to stuff DIPA("PRIDT")= id type to stuff  N PRN0,Z  Q:'$G(^DGCR(399,IBIFN,"I1"))  I $G(IBLNPRV),'$G(IBLNPRV("LNPRVIEN")),'$G(IBLNPRV("PROCIEN")) Q  ; DEM;432 - If line provider user input.  ; DEM;432 - Updated variable PRNO to be equal to line level provider if we are coming from line level provider user input.  S PRN0=$S($G(IBLNPRV):$G(^DGCR(399,IBIFN,"CP",IBLNPRV("PROCIEN"),"LNPRV",IBLNPRV("LNPRVIEN"),0)),1:$G(^DGCR(399,IBIFN,"PRV",IBPRIEN,0)))  S DIPA("EDIT")=1,(DIPA("PRID"),DIPA("PRIDT"))=""  W @IOF  W !,?19,"\*\*\*\* SECONDARY PERFORMING PROVIDER IDs \*\*\*\*"  W !!,$P("PRIMARY^SECONDARY^TERTIARY",U,IBCOBN)\_" INSURANCE CO: "\_$P($G(^DIC(36,+$G(^DGCR(399,IBIFN,"I"\_IBCOBN)),0)),U)  ; DEM;432 - Added line and conditions if line level provider user input.  I '$G(IBLNPRV) W !,"PROVIDER: "\_$$EXTERNAL^DILFD(399.0222,.02,"",$P(PRN0,U,2))\_" ("\_$$EXTERNAL^DILFD(399.0222,.01,"",+PRN0)\_")",!  I $G(IBLNPRV) W !,"Line Level Provider: "\_$$EXTERNAL^DILFD(399.0404,.02,"",$P(PRN0,U,2))\_" ("\_$$EXTERNAL^DILFD(399.0404,.01,"",+PRN0)\_")",!  ;  I $P(PRN0,U,4+IBCOBN)="" K DIPA("PRID"),DIPA("PRIDT") D NEWID(IBIFN,IBPRIEN,IBCOBN,.DIPA) ; No id currently exists for the ins seq/prov  ;  Q  ; NEWID(IBIFN,IBPRIEN,IBCOBN,DIPA) ;  N IBDEF,IBCT,IBNUM,IBINS,IBFRM,IBCAR,IBARR,IBARRS,IB0,IBM,IBQUIT,IBSEL,PRN,PRT,PRN,PRN0,DIR,X,Y,Z,Z0,IBZ,IBZ1,IBTYP,IBREQ,IBREQT,IBTYPN,IBID,IBUSED  S IBREQ=0,IBREQT=""  Q:($G(IBLNPRV))&('$G(IBLNPRV("LNPRVIEN"))&'$G(IBLNPRV("PROCIEN"))) ; DEM;432 - If line provider user input.  ; DEM;432 - Updated variable PRNO to be equal to line level provider if we are coming from line level provider user input.  S PRN0=$S($G(IBLNPRV):$G(^DGCR(399,IBIFN,"CP",IBLNPRV("PROCIEN"),"LNPRV",IBLNPRV("LNPRVIEN"),0)),1:$G(^DGCR(399,IBIFN,"PRV",IBPRIEN,0)))  S Z(IBCOBN)=$S($G(DIPA("I"\_IBCOBN)):$$GETTYP^IBCEP2A(IBIFN,IBCOBN,$P(PRN0,U)),1:"")  S IBINS=+$G(^DGCR(399,IBIFN,"I"\_IBCOBN)),IB0=$S($G(IBLNPRV):$G(^DGCR(399,IBIFN,"CP",IBLNPRV("PROCIEN"),"LNPRV",IBLNPRV("LNPRVIEN"),0)),1:$G(^DGCR(399,IBIFN,"PRV",IBPRIEN,0)))  S IBCAR=$$INPAT^IBCEF(IBIFN),IBCAR=$S('IBCAR:2,1:1) **;JRA IB\*2.0\*592 Same logic for Dental Form 7 as for CMS-1500**  **;**S IBFRM=$$FT^IBCEF(IBIFN),IBFRM=$S(IBFRM=2:2,1:1) **;JRA IB\*2.0\*592 ';'**  S IBFRM=$$FT^IBCEF(IBIFN),IBFRM=$S(IBFRM=2:2,**IBFRM=7:4**,1:1) **;JWS;JRA IB\*2.0\*592**  I $P(Z(IBCOBN),U) D  . W !,"INS. COMPANY'S DEFAULT SECONDARY ID TYPE IS: "\_$$EXTERNAL^DILFD(36,4.01,"",$P(Z(IBCOBN),U)) S IBREQT=+Z(IBCOBN)  . I $P(Z(IBCOBN),U,2) W !,?2," AND IS REQUIRED TO BE ENTERED FOR THIS CLAIM" S IBREQ=1  I $$CUNEED^IBCEP3(IBIFN,IBCOBN) W !,"CARE UNITS ARE DEFINED"\_$S($P($G(^DIC(36,IBINS,4)),U,9)'="":" AS "\_$P(^(4),U,9),1:"")\_" FOR THESE IDs"  D PRACT^IBCEF71(IBINS,IBFRM,IBCAR,$P(IB0,U,2),.IBARR,$P(IB0,U),$S($$COBN^IBCEF(IBIFN)=IBCOBN:"C",1:"O"),355.9,1)  S (IBNUM,IBCT)=0,IBDEF=""  I $O(IBARR(""))="" S IBCT=IBCT+1,DIR("A",IBCT)="NO SECONDARY IDS ARE DEFINED FOR THIS PROV THAT ARE VALID FOR THIS CLAIM"  S IBCT=IBCT+1,DIR("A",IBCT)="SELECT A SECONDARY ID OR ACTION FROM THE LIST BELOW: ",IBCT=IBCT+1,DIR("A",IBCT)=" "  ;  S IBCT=IBCT+1,IBNUM=IBNUM+1,DIR("A",IBCT)=" "\_$E(IBNUM\_$J("",3),1,3)\_" - NO SECONDARY ID NEEDED",IBNUM=IBNUM+1,IBCT=IBCT+1,DIR("A",IBCT)=" "\_$E(IBNUM\_$J("",3),1,3)\_" - ADD AN ID FOR THIS CLAIM ONLY"  I $O(IBARR(""))="" S IBDEF=1,DIPA("EDIT")=$$SELID(.DIR,IBDEF,.IBID,.DIPA,IBNUM) Q  ;  S PRN=$$GETID^IBCEP2(IBIFN,2,$P(PRN0,U,2),IBCOBN,.PRT,,$P(PRN0,U)),IBDEF=""  ;  I PRN'="",PRT D  . N PRT1  . S PRT1=$P($G(^IBE(355.97,+PRT,0)),U)  . I $P($G(^IBE(355.97,+PRT,1)),U,3) S PRT1="ST LIC("\_$P($G(^DIC(5,+$$CAREST^IBCEP2A(IBIFN),0)),U,2)\_")"  . S IBCT=IBCT+1,IBNUM=IBNUM+1  . S DIR("A",IBCT)=" "\_$E(IBNUM\_$J("",3),1,3)\_" - "\_$E("<DEFAULT> "\_PRN\_$J("",29),1,29)\_" "\_$E(PRT1\_$J("",15),1,15)  . S DIR("A",IBCT)=DIR("A",IBCT)\_" "\_$S($P(PRT,U,3)'["355.9":"",$P($G(^IBA(+$P(PRT,U,3),+$P(PRT,U,2),0)),U,3)'="":$$EXTERNAL^DILFD(355.9,.03,"",$P($G(^IBA(+$P(PRT,U,3),+$P(PRT,U,2),0)),U,3)),1:"")  . S IBID(IBNUM)=PRN\_U\_+PRT,IBDEF=IBNUM,IBID(IBNUM,1)=DIR("A",IBCT),IBDEF=IBNUM,IBDEF("IEN")=$P(PRT,U,2,3)  . S IBUSED(PRT,PRN,0)=""  ;  S IBQUIT=0,IBSEL=1  ; Sort ids by id type  S IBZ="" F  S IBZ=$O(IBARR(IBZ)) Q:IBZ=""  S IBZ1="" F  S IBZ1=$O(IBARR(IBZ,IBZ1)) Q:IBZ1=""  D  . S IBTYP=+$P(IBARR(IBZ,IBZ1),U,9)  . I $P(IBARR(IBZ,IBZ1),U,4)]"" Q:$D(IBUSED(IBTYP,$P(IBARR(IBZ,IBZ1),U,4),+$P(IBARR(IBZ,IBZ1),U,7)))  . I $P($G(IBDEF("IEN")),U,2)["355.9",$P(IBARR(IBZ,IBZ1),U,8),$P(IBARR(IBZ,IBZ1),U,8)=+$G(IBDEF("IEN")) Q:$S($P(IBZ1,U)'["INS DEF":$P($G(IBDEF("IEN")),U,2)=355.9,1:$P($G(IBDEF("IEN")),U,2)=355.91)  . S IBARRS(IBTYP,IBZ,IBZ1)=IBARR(IBZ,IBZ1)  . I $P(IBARR(IBZ,IBZ1),U,4)]"" S IBUSED(IBTYP,$P(IBARR(IBZ,IBZ1),U,4),+$P(IBARR(IBZ,IBZ1),U,7))=""  S IBTYP="" F  S IBTYP=$O(IBARRS(IBTYP)) Q:IBTYP=""  S IBZ="" F  S IBZ=$O(IBARRS(IBTYP,IBZ)) Q:IBZ=""  D  Q:IBQUIT  . S IBZ1="" F  S IBZ1=$O(IBARRS(IBTYP,IBZ,IBZ1)) Q:IBZ1=""  S IBCT=IBCT+1,IBNUM=IBNUM+1 D  Q:IBQUIT  .. S Z0=IBARRS(IBTYP,IBZ,IBZ1)  .. S IBARR=$S($P(Z0,U,8)&(IBZ1'["LIC"):$G(^IBA("355.9"\_$S($P(IBZ1,U)'="INS DEF":"",1:1),+$P(Z0,U,8),0)),1:"")  .. S IBTYPN=$S(IBTYP=+$$STLIC^IBCEP8():"ST LIC ("\_$P($G(^DIC(5,+$P(Z0,U,7),0)),U,2)\_")",1:$P($G(^IBE(355.97,IBTYP,0)),U))  .. S DIR("A",IBCT)=" "\_$E(IBNUM\_$J("",3),1,3)\_" - "\_$E($S($P(IBZ1,U)="INS DEF":"<INS DEF> ",1:"")\_$P(Z0,U,4)\_$J("",29),1,29)\_" "\_$E(IBTYPN\_$J("",15),1,15)\_" "\_$S($P(IBARR,U,3):$$EXTERNAL^DILFD(355.9,.03,"",$P(IBARR,U,3)),1:"")  .. S IBID(IBNUM,1)=DIR("A",IBCT),IBID(IBNUM)=$P(Z0,U,4)\_U\_IBTYP  .. I (IBNUM#15)=0 S IBM=$$MORE(.DIR) D  Q:IBQUIT  ... I IBM<0 S IBQUIT=1,IBSEL=0 Q  ; User aborted list  ... I 'IBM S IBQUIT=1 Q  ; User wants to select  ... W ! K DIR S IBCT=1  I 'IBSEL S DIPA("EDIT")=-1  I IBSEL S:IBDEF=""&$G(IBREQ) IBDEF=2 S DIPA("EDIT")=$$SELID(.DIR,IBDEF,.IBID,.DIPA,IBNUM)  Q  ; SELID(DIR,IBDEF,IBID,DIPA,IBNUM) ; Returns the selection from the array of possible IDs/ID actions  N IDACT,IDSEL,X,Y  S IDACT=""  S DIR("B")=$S('$G(IBDEF):1,1:IBDEF),DIR("A",+$O(DIR("A",""),-1)+1)=" "  S DIR(0)="NA^1:"\_IBNUM,DIR("A")="Selection: " W ! D ^DIR K DIR  I $D(DTOUT)!$D(DUOUT)!(Y=1) S IDACT=-1 G SELIDQ  I Y=2 S IDACT=1 G SELIDQ  S IDSEL=Y  S DIR("A",1)="ID SELECTED:",DIR("A",2)=" "\_$G(IBID(+Y,1)),DIR("A")="IS THIS CORRECT?: ",DIR("B")="YES",DIR(0)="YA" W ! D ^DIR K DIR  I Y'=1 S IDACT=-1 G SELIDQ  S DIPA("PRID")=$P(IBID(IDSEL),U),DIPA("PRIDT")=$P(IBID(IDSEL),U,2),IDACT=2  ; SELIDQ Q IDACT  ; MORE(DIR) ;  N DIR,X,Y,DUOUT,DTOUT  S DIR(0)="YA",DIR("A")="MORE?: ",DIR("B")="NO" W ! D ^DIR K DIR("B")  Q $S($D(DTOUT)!$D(DUOUT):-1,1:Y)  ;  ; IBFIDFL = E = Electronic Form Type  ; A = Additional ID's  ; LF - VA Lab/Facility FACID(IBINS,IBFIDFL) ; Enter/edit billing facility ids  ; IBINS = ien of ins co (file 36)  N IBID,Z,Z0,Y  K ^TMP($J,"IBBF\_ID")  W @IOF  D GETBPNUM(IBINS)  K ^TMP("IBCE\_PRVFAC\_MAINT\_INS",$J)  S ^TMP("IBCE\_PRVFAC\_MAINT\_INS",$J)=IBFIDFL\_U\_IBINS\_U\_"1"  D EN^VALM("IBCE PRVFAC MAINT")  K ^TMP("IBCE\_PRVFAC\_MAINT\_INS",$J)  W @IOF  D FULL^VALM1  Q  ; GETBPNUM(IBINS) ;  N Z,Z0,IBID,IBMAIN  S IBMAIN=$$MAIN(),^TMP($J,"IBBF\_ID")=IBMAIN  S IBID=$$BF^IBCU()  S Z=0 F  S Z=$O(^IBA(355.92,"B",IBINS,Z)) Q:'Z  D  . S Z0=$G(^IBA(355.92,Z,0))  . Q:$P(Z0,U,8)'="E"  ; WCJ 1/13/06 There are several ID types in this file   . Q:$P(Z0,U,3)]""  . S ^TMP($J,"IBBF\_ID",$S($P(Z0,U,5)=IBMAIN:0,1:+$P(Z0,U,5)),+$P(Z0,U,4))=$P(Z0,U,7)  . S ^TMP($J,"IBBF\_ID",$S($P(Z0,U,5)=IBMAIN:0,1:+$P(Z0,U,5)),+$P(Z0,U,4),"QUAL")=$P(Z0,U,6)  Q  ; MAIN() ; Returns ien of main division of the database  Q +$$PRIM^VASITE()  ; FACNUM(IBIFN,IBCOB,IBQF) ; Function returns the current division's fac billing  ; prov id for the COB insurance sequence from file 355.92  ; IBIFN = ien file 399  ; IBCOB = # of COB ins seq or if "", current assumed  ; IBQF - 1 if qualifier is to be returned instead of ID  N Z,IBDIV,IBFT,X,BPZ  S X="",IBDIV=0  S:'$G(IBCOB) IBCOB=+$$COBN^IBCEF(IBIFN)  ;  ; IB\*2\*400 - esg - 11/7/08 - Determine the division associated with the billing provider first  S BPZ=+$$B^IBCEF79(IBIFN,IBCOB) ; Inst file pointer as the billing provider for payer seq IBCOB  I BPZ S IBDIV=+$O(^DG(40.8,"AD",BPZ,0)) ; Billing Provider division (may not exist)  ;  I 'IBDIV S IBDIV=+$P($G(^DGCR(399,IBIFN,0)),U,22) ; Division on claim  I 'IBDIV S IBDIV=$$MAIN() ; main division **;JWS;IB\*2.0\*592;**  S IBFT=$$FT^IBCEF(IBIFN),IBFT=$S(IBFT=3:1,**IBFT=7:4**,1:2)  K ^TMP($J,"IBBF\_ID")  D GETBPNUM(+$P($G(^DGCR(399,IBIFN,"M")),U,IBCOB))  I IBDIV=+$G(^TMP($J,"IBBF\_ID")) S IBDIV=0  I '$G(IBQF) S X=$S($D(^TMP($J,"IBBF\_ID",IBDIV,IBFT)):^(IBFT),1:$G(^TMP($J,"IBBF\_ID",0,IBFT)))  I $G(IBQF) S X=$S($D(^TMP($J,"IBBF\_ID",IBDIV,IBFT,"QUAL")):^("QUAL"),1:$G(^TMP($J,"IBBF\_ID",0,IBFT,"QUAL")))  K ^TMP($J,"IBBF\_ID")  Q X  ; SOP(IBIFN,IBZD) ; Returns X12 current source of pay code for bill ien IBIFN  ; IBZD = the current ins policy type, if known  N IBZ,IBFT  **;JRA IB\*2.0\*592 Added 'IBFT'**  S IBZ=""  I $G(IBZD)="" D F^IBCEF("N-CURRENT INS POLICY TYPE","IBZD",,IBIFN)  **S IBFT=$$FT^IBCEF(IBIFN) ;JRA IB\*2.0\*592**  **;**S IBZ=$S($G(IBZD)="":"G2","MAMB16"[IBZD:"1C",IBZD="TV"!(IBZD="MC"):"1D",IBZD="CH":"1H",IBZD="BL":$S($$FT^IBCEF(IBIFN)=2:"1B",1:"1A"),1:"G2") **;JRA IB\*2.0\*592 ';'**  S IBZ=$S($G(IBZD)="":"G2","MAMB16"[IBZD:"1C",IBZD="TV"!(IBZD="MC"):"1D",IBZD="CH":"1H",IBZD="BL":$S((IBFT=2**!(IBFT=7)):"**1B",1:"1A"),1:"G2") **;JRA IB\*2.0\*592**  Q IBZ  ; | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCEP3 | | | | | | | | |
| **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** | | | | | | | | | |
| IBCEP3 ;ALB/TMP - EDI UTILITIES for provider ID ;25-SEP-00  ;;2.0;INTEGRATED BILLING;\*\*137,207,232,280,349\*\*;21-MAR-94;Build 46  ;;Per VHA Directive 2004-038, this routine should not be modified.  ; CUNEED(IBIFN,IBSEQ,IBPTYP,IBRET,IBEMC) ; Determine if care unit needed for  ; provider type and insurance company(s) on bill  ; IBIFN = ien of bill (file 399)  ; IBSEQ = specific COB sequence to check or null for check all  ; IBPTYP = the ien of the provider id type in file 355.97 or if null,  ; the default performing provider ID type for the ins co's.  ; IBRET = flag to return insurance ien (0) or file 355.97 ien (1)  ; IBEMC = no longer used  ;  ; Function returns care unit needed flag (0=not needed, 1=needed) ^  ; AND if IBSEQ="": primary ins or 355.97 ien if care unit needed ^  ; secondary ins or 355.97 ien if care unit needed ^  ; tertiary ins or 355.97 ien if care unit needed  ; (these would be '^' pieces 2,3,4)  ; if IBSEQ : IBSEQ seq ins or 355.97 ien if care unit needed  ; (this would be '^' piece 2)  ;  Q:$G(IBEMC) 0  N Q,Z,Z0,Z4,IB,IBCTYP,IBFTYP,IBQ,IBRX,IBPT  S (IBRX,IB)=0  S IBFTYP=$$FT^IBCEF(IBIFN),IBCTYP=$$INPAT^IBCEF(IBIFN,1)  S IBFTYP=$S(IBFTYP=3:1,1:2) S:IBCTYP'=1 IBCTYP=2  I IBCTYP=2 S IBRX=$$ISRX^IBCEF1(IBIFN) ; Outpatient pharmacy  S IBPT=$G(IBPTYP)  ;  S (Z,IBQ)=0  F  D  Q:IBQ  . I $G(IBSEQ) S Z=IBSEQ,IBQ=1 ; Only once for specific COB sequence  . I '$G(IBSEQ) S Z=Z+1,IBPTYP=IBPT I Z>3 S IBQ=1 Q  ; Up to 3 times - all ins  . S Z0=$$INSSEQ^IBCEP1(IBIFN,Z),Z4=$G(^DIC(36,+Z0,4))  . I '$G(IBPTYP) S IBPTYP=+Z4  . I 'Z0!'IBPTYP S:'Z0 IBQ=1 Q  . S Q=+$$CAREUN(Z0,IBPTYP,IBFTYP,IBCTYP,IBRX)  . I Q S $P(IB,U,$S($G(IBSEQ):Z+1,1:2))=$S($G(IBRET):Q,1:Z0)  ;  I $TR(IB,"^0") S $P(IB,U)=1  Q IB  ; CAREUN(IBINS,IBPTYP,IBFTYP,IBCTYP,IBRX) ; Find ien (file 355.96) for care  ; unit for the combination of ins co, prov type, form type and  ; care type  ; IBINS = ien of ins co (file 36)  ; IBPTYP = ien of provider id type (file 355.97)  ; IBFTYP = form type (1=UB,2=1500)  ; IBCTYP = care type (1=inpat,2=outpat)  ; IBRX = 1 if outpat/Rx bill  ;  N IB  S IB=""  ;  I $G(IBRX) D  . N T  . S T=$O(^IBA(355.96,"AD",IBINS,IBFTYP,3,IBPTYP,0))  . I 'T S T=$O(^IBA(355.96,"AD",IBINS,0,3,IBPTYP,0))  . I T S IB=T  ;  I 'IB D  ; Find from most specific to least specific  . I $O(^IBA(355.96,"AD",IBINS,IBFTYP,IBCTYP,IBPTYP,0)) S IB=+$O(^(0)) Q  . I $O(^IBA(355.96,"AD",IBINS,IBFTYP,0,IBPTYP,0)) S IB=+$O(^(0)) Q  . I $O(^IBA(355.96,"AD",IBINS,0,IBCTYP,IBPTYP,0)) S IB=+$O(^(0)) Q  . I $O(^IBA(355.96,"AD",IBINS,0,0,IBPTYP,0)) S IB=+$O(^(0)) Q  ;  Q IB  ; DISP(IBINS,IBTYPE) ; Return the name of the type of care unit needed  ; IBINS = ien of ins co (file 36)  ; IBTYPE = 2:PERFORMING PROVIDER ID  I $G(IBTYPE)'=2 Q ""  Q $P($G(^DIC(36,+IBINS,4)),U,9)  ; DELID(IBIFN,IBSEQ,IBX) ; Delete all provider data specific to an ins co  ; represented by the COB sequence IBSEQ for bill IBIFN  ; IBX = 1 if called from care unit prompt - don't delete value  N IBZ,IBDR,X,Y,Z0,Z1  S IBZ=0  Q:'$G(IBSEQ)!($G(IBSEQ)>3)  F  S IBZ=$O(^DGCR(399,IBIFN,"PRV",IBZ)) Q:'IBZ  S Z0=$G(^(IBZ,0)),Z1=$G(^(1)) D  . ; Delete provider id's  . I $P(Z0,U,4+IBSEQ)'="" S IBDR(399.0222,IBZ\_","\_IBIFN\_",",(4+IBSEQ/100))="@"  . ; Delete provider id types  . I $P(Z0,U,11+IBSEQ)'="" S IBDR(399.0222,IBZ\_","\_IBIFN\_",",(11+IBSEQ/100))="@"  . I $D(IBDR) D FILE^DIE(,"IBDR")  Q  ; SETID(IBIFN,IBSEQ) ; Default provider id for bill IBIFN and ins co for COB  ; sequence IBSEQ  N IBZ,X,Y,IBDR,IBT  S IBZ=0  Q  ; No longer used as of patch 232  ;Q:'$G(IBSEQ)!($G(IBSEQ)>3)  ;F S IBZ=$O(^DGCR(399,IBIFN,"PRV",IBZ)) Q:'IBZ S Z0=$G(^(IBZ,0)),Z1=$G(^(1)) D  ;. ; Update provider id's if no care unit is needed  ;. I $P(Z0,U,2)'="" D  ;.. S Z=$$GETID^IBCEP2(IBIFN,2,$P(Z0,U,2),IBSEQ,.IBT)  ;.. I Z'="",IBT S IBDR(399.0222,IBZ\_","\_IBIFN\_",",(4+IBSEQ/100))=Z,IBDR(399.0222,IBZ\_","\_IBIFN\_",",(11+IBSEQ/100))=+IBT  ;. I $D(IBDR) D FILE^DIE(,"IBDR")  Q  ; ALLID(IBIFN,IBFLD,IBFUNC) ; If form type or care type (I/O/RX) changes,  ; determine new provider id values if possible and update them  ; this includes primary, secondary, tertiary id's  ; IBIFN = ien of claim (file 399)  ; IBFLD = ien of the field being changed when this call is made  ; (.19 = form type .25 = care type)  ; IBFUNC = 1 to add, 2 to delete  N Z,Z0,IBC,IBDR,IBT  S Z=0  F  S Z=$O(^DGCR(399,IBIFN,"PRV",Z)) Q:'Z  S Z0=$G(^(Z,0)) D  . F IBC=5:1:7 I $S(IBFUNC=2:$P(Z0,U,IBC)'="",1:1) S IBDR(399.0222,IBC\_","\_IBIFN\_",",(IBC/100))=$S(IBFUNC=2:"@",1:$$GETID^IBCEP2(IBIFN,2,$P(Z0,U,2),IBC-4,.IBT))  I $D(IBDR) D FILE^DIE(,"IBDR")  Q  ; CUMNT ; Add/edit care unit  N D,DIE,DIC,DIK,DIR,DA,X,Y,IB,IBINS,IBF,IBCT,IBOK,IBPTYP,IBOLD,IBY,IBINS1,IBPTYP1,DUOUT,DTOUT INS F  D  Q:Y'>0  . S DIC="^DIC(36,",DIC(0)="AEMQ" D ^DIC K DIC  . I $D(DUOUT)!$D(DTOUT) S Y=-1 Q  . I Y'>0 S DIR(0)="EA",DIR("A")="Insurance Co is required - press enter to continue: " D ^DIR K DIR Q  . S IBINS=+Y,IBF="A",IBINS1=$P(Y,U,2)  I $O(^IBA(355.96,"D",IBINS,""))'="" D  . W ! S DIR("A")="(A)dd or (E)dit entries?: ",DIR("B")="Add",DIR(0)="SA^A:Add;E:Edit" D ^DIR W ! K DIR  . S IBF=Y  Q:$G(IBF)=""!("AE"'[$G(IBF))  ;  I IBINS>0 D  . I IBF="A" D NEW^IBCEP4A(1)  . I IBF="E" D CHANGE^IBCEP4A(1)  ;  Q  ; DUP(IBDA,IBOLD,IBFUNC) ; Check if the combination of ins co, prov type, care  ; type and form already exists in file 355.96  ; IBDA = ien of entry in file 355.96  ; IBOLD = the 0-node before changes were made - used to reset the fields  N DUP,IB0,DR,X,Y,DIK,DIE,DA  S IB0=$G(^IBA(355.96,IBDA,0)),DUP=0  ;  I $O(^IBA(355.96,"AUNIQ",+$P(IB0,U,3),+IB0,+$P(IB0,U,4),+$P(IB0,U,5),+$P(IB0,U,6),0))'=IBDA!($O(^IBA(355.96,"AUNIQ",+$P(IB0,U,3),+IB0,+$P(IB0,U,4),+$P(IB0,U,5),+$P(IB0,U,6),""),-1)'=IBDA) D  . S DUP=1  . I IBFUNC="E" D  .. S DR=";.01///"\_$P(IBOLD,U)\_";.03///"\_$S($P(IBOLD,U,3)'="":"/"\_$P(IBOLD,U,3),1:"@")\_";.04///"\_$S($P(IBOLD,U,4)'="":"/"\_$P(IBOLD,U,4),1:"@")  .. S DR=DR\_";05///"\_$S($P(IBOLD,U,5)'="":"/"\_$P(IBOLD,U,5),1:"@")\_";.06///"\_$S($P(IBOLD,U,6)'="":"/"\_$P(IBOLD,U,6),1:"@")  .. S DA=IBDA,DIE="^IBA(355.96," D ^DIE  . I IBFUNC="A" D  .. S DA=IBDA,DIK="^IBA(355.96," D ^DIK  Q DUP  ; PROFID(IBIFN,IBSEQ,IBID) ; Return id and type of rendering provider id  ; used for insurance co at COB seq IBSEQ for bill ien IBIFN  ; RETURN VALUES:  ; piece 1:  ; 1 = FEDERAL TAX ID  ; 2 = INSURANCE CO SPECIFIC ID  ; 3 = NETWORK ID  ; "" = not a CMS-1500 bill or no id found  ; piece 2:  ; the id #  N IBTYP,IBXDATA,IBZ  S:'$G(IBSEQ) IBSEQ=+$$COBN^IBCEF(IBXIEN)  S IBTYP=""\_U\_$G(IBID)  G:$$FT^IBCEF(IBIFN)'=2 PROFIDQ  I '$D(IBID) D F^IBCEF("N-ALL ATT/RENDERING PROV ID","IBZ",,IBIFN) S IBID=$$NOPUNCT^IBCEF($P(IBZ,U,IBSEQ+1))  G:IBID="" PROFIDQ  S IBTYP=$S($$NOPUNCT^IBCEF(IBID)=$$NOPUNCT^IBCEF($P($G(^IBE(350.9,1,1)),U,5)):1,$$NETWRK(IBIFN,IBID,IBSEQ):3,1:2)  S IBTYP=IBTYP\_U\_IBID  ; PROFIDQ Q IBTYP  ; NETWRK(IBIFN,IBID,IBSEQ) ; Determine if ID number IBID is the same as the  ; network id for the insurance co  ; IBIFN = bill ien (file 399)  ; IBSEQ = COB seq # of bill  ; Returns 1 if network ID match is found for bill IBIFN, COB seq IBSEQ  N IBINS,IBNET  S IBNET=0  Q IBNET  ; This section needs work \*\*\*\*\*\*\*\*\*  I '$G(IBSEQ) S IBSEQ=+$$COBN^IBCEF(IBXIEN)  S IBINS=+$G(^DGCR(399,IBIFN,"I"\_IBSEQ))  I $P($G(^IBE(355.97,+$$PPTYP^IBCEP0(IBINS),1)),U,6) D  . ; performing provider id type is a network id type  . I $$NOPUNCT^IBCEF($G(IBID))=$$NOPUNCT^IBCEF($$GETID^IBCEP2(IBIFN,3,$$PERFPRV^IBCEP2A(IBIFN),IBSEQ)) S IBNET=1  Q IBNET  ;  ;  ; Parameter definitions for UNIQ1 and UNIQ2 in IBCEP2  ; IBIFN = ien of bill (file 399)  ; IBINS = ien of insurance co (file 36) or \*ALL\* for all insurance  ; IBPTYP = the ien of the provider id type in file 355.97  ; IBUNIT = the value of the specific care unit to use for a match  ; or \*N/A\* if none needed  ; IBCU = the ien of the entry being matched in start file  ; IBT = the second and third pieces are set to the entry ien^file # | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCEP3 ;ALB/TMP - EDI UTILITIES for provider ID ;25-SEP-00  ;;2.0;INTEGRATED BILLING;\*\*137,207,232,280,349,**592\*\*;**21-MAR-94;Build 46  ;;Per VHA Directive 2004-038, this routine should not be modified.  ; CUNEED(IBIFN,IBSEQ,IBPTYP,IBRET,IBEMC) ; Determine if care unit needed for  ; provider type and insurance company(s) on bill  ; IBIFN = ien of bill (file 399)  ; IBSEQ = specific COB sequence to check or null for check all  ; IBPTYP = the ien of the provider id type in file 355.97 or if null,  ; the default performing provider ID type for the ins co's.  ; IBRET = flag to return insurance ien (0) or file 355.97 ien (1)  ; IBEMC = no longer used  ;  ; Function returns care unit needed flag (0=not needed, 1=needed) ^  ; AND if IBSEQ="": primary ins or 355.97 ien if care unit needed ^  ; secondary ins or 355.97 ien if care unit needed ^  ; tertiary ins or 355.97 ien if care unit needed  ; (these would be '^' pieces 2,3,4)  ; if IBSEQ : IBSEQ seq ins or 355.97 ien if care unit needed  ; (this would be '^' piece 2)  ;  Q:$G(IBEMC) 0  N Q,Z,Z0,Z4,IB,IBCTYP,IBFTYP,IBQ,IBRX,IBPT  S (IBRX,IB)=0  S IBFTYP=$$FT^IBCEF(IBIFN),IBCTYP=$$INPAT^IBCEF(IBIFN,1) **;JWS;IB\*2.0\*592**  S IBFTYP=$S(IBFTYP=3:1,**IBFTYP=7:4**,1:2) S:IBCTYP'=1 IBCTYP=2  I IBCTYP=2 S IBRX=$$ISRX^IBCEF1(IBIFN) ; Outpatient pharmacy  S IBPT=$G(IBPTYP)  ;  S (Z,IBQ)=0  F  D  Q:IBQ  . I $G(IBSEQ) S Z=IBSEQ,IBQ=1 ; Only once for specific COB sequence  . I '$G(IBSEQ) S Z=Z+1,IBPTYP=IBPT I Z>3 S IBQ=1 Q  ; Up to 3 times - all ins  . S Z0=$$INSSEQ^IBCEP1(IBIFN,Z),Z4=$G(^DIC(36,+Z0,4))  . I '$G(IBPTYP) S IBPTYP=+Z4  . I 'Z0!'IBPTYP S:'Z0 IBQ=1 Q  . S Q=+$$CAREUN(Z0,IBPTYP,IBFTYP,IBCTYP,IBRX)  . I Q S $P(IB,U,$S($G(IBSEQ):Z+1,1:2))=$S($G(IBRET):Q,1:Z0)  ;  I $TR(IB,"^0") S $P(IB,U)=1  Q IB  ; CAREUN(IBINS,IBPTYP,IBFTYP,IBCTYP,IBRX) ; Find ien (file 355.96) for care  ; unit for the combination of ins co, prov type, form type and  ; care type  ; IBINS = ien of ins co (file 36)  ; IBPTYP = ien of provider id type (file 355.97)  ; IBFTYP = form type (1=UB,2=1500)  ; IBCTYP = care type (1=inpat,2=outpat)  ; IBRX = 1 if outpat/Rx bill  ;  N IB  S IB=""  ;  I $G(IBRX) D  . N T  . S T=$O(^IBA(355.96,"AD",IBINS,IBFTYP,3,IBPTYP,0))  . I 'T S T=$O(^IBA(355.96,"AD",IBINS,0,3,IBPTYP,0))  . I T S IB=T  ;  I 'IB D  ; Find from most specific to least specific  . I $O(^IBA(355.96,"AD",IBINS,IBFTYP,IBCTYP,IBPTYP,0)) S IB=+$O(^(0)) Q  . I $O(^IBA(355.96,"AD",IBINS,IBFTYP,0,IBPTYP,0)) S IB=+$O(^(0)) Q  . I $O(^IBA(355.96,"AD",IBINS,0,IBCTYP,IBPTYP,0)) S IB=+$O(^(0)) Q  . I $O(^IBA(355.96,"AD",IBINS,0,0,IBPTYP,0)) S IB=+$O(^(0)) Q  ;  Q IB  ; DISP(IBINS,IBTYPE) ; Return the name of the type of care unit needed  ; IBINS = ien of ins co (file 36)  ; IBTYPE = 2:PERFORMING PROVIDER ID  I $G(IBTYPE)'=2 Q ""  Q $P($G(^DIC(36,+IBINS,4)),U,9)  ; DELID(IBIFN,IBSEQ,IBX) ; Delete all provider data specific to an ins co  ; represented by the COB sequence IBSEQ for bill IBIFN  ; IBX = 1 if called from care unit prompt - don't delete value  N IBZ,IBDR,X,Y,Z0,Z1  S IBZ=0  Q:'$G(IBSEQ)!($G(IBSEQ)>3)  F  S IBZ=$O(^DGCR(399,IBIFN,"PRV",IBZ)) Q:'IBZ  S Z0=$G(^(IBZ,0)),Z1=$G(^(1)) D  . ; Delete provider id's  . I $P(Z0,U,4+IBSEQ)'="" S IBDR(399.0222,IBZ\_","\_IBIFN\_",",(4+IBSEQ/100))="@"  . ; Delete provider id types  . I $P(Z0,U,11+IBSEQ)'="" S IBDR(399.0222,IBZ\_","\_IBIFN\_",",(11+IBSEQ/100))="@"  . I $D(IBDR) D FILE^DIE(,"IBDR")  Q  ; SETID(IBIFN,IBSEQ) ; Default provider id for bill IBIFN and ins co for COB  ; sequence IBSEQ  N IBZ,X,Y,IBDR,IBT  S IBZ=0  Q  ; No longer used as of patch 232  ;Q:'$G(IBSEQ)!($G(IBSEQ)>3)  ;F S IBZ=$O(^DGCR(399,IBIFN,"PRV",IBZ)) Q:'IBZ S Z0=$G(^(IBZ,0)),Z1=$G(^(1)) D  ;. ; Update provider id's if no care unit is needed  ;. I $P(Z0,U,2)'="" D  ;.. S Z=$$GETID^IBCEP2(IBIFN,2,$P(Z0,U,2),IBSEQ,.IBT)  ;.. I Z'="",IBT S IBDR(399.0222,IBZ\_","\_IBIFN\_",",(4+IBSEQ/100))=Z,IBDR(399.0222,IBZ\_","\_IBIFN\_",",(11+IBSEQ/100))=+IBT  ;. I $D(IBDR) D FILE^DIE(,"IBDR")  Q  ; ALLID(IBIFN,IBFLD,IBFUNC) ; If form type or care type (I/O/RX) changes,  ; determine new provider id values if possible and update them  ; this includes primary, secondary, tertiary id's  ; IBIFN = ien of claim (file 399)  ; IBFLD = ien of the field being changed when this call is made  ; (.19 = form type .25 = care type)  ; IBFUNC = 1 to add, 2 to delete  N Z,Z0,IBC,IBDR,IBT  S Z=0  F  S Z=$O(^DGCR(399,IBIFN,"PRV",Z)) Q:'Z  S Z0=$G(^(Z,0)) D  . F IBC=5:1:7 I $S(IBFUNC=2:$P(Z0,U,IBC)'="",1:1) S IBDR(399.0222,IBC\_","\_IBIFN\_",",(IBC/100))=$S(IBFUNC=2:"@",1:$$GETID^IBCEP2(IBIFN,2,$P(Z0,U,2),IBC-4,.IBT))  I $D(IBDR) D FILE^DIE(,"IBDR")  Q  ; CUMNT ; Add/edit care unit  N D,DIE,DIC,DIK,DIR,DA,X,Y,IB,IBINS,IBF,IBCT,IBOK,IBPTYP,IBOLD,IBY,IBINS1,IBPTYP1,DUOUT,DTOUT INS F  D  Q:Y'>0  . S DIC="^DIC(36,",DIC(0)="AEMQ" D ^DIC K DIC  . I $D(DUOUT)!$D(DTOUT) S Y=-1 Q  . I Y'>0 S DIR(0)="EA",DIR("A")="Insurance Co is required - press enter to continue: " D ^DIR K DIR Q  . S IBINS=+Y,IBF="A",IBINS1=$P(Y,U,2)  I $O(^IBA(355.96,"D",IBINS,""))'="" D  . W ! S DIR("A")="(A)dd or (E)dit entries?: ",DIR("B")="Add",DIR(0)="SA^A:Add;E:Edit" D ^DIR W ! K DIR  . S IBF=Y  Q:$G(IBF)=""!("AE"'[$G(IBF))  ;  I IBINS>0 D  . I IBF="A" D NEW^IBCEP4A(1)  . I IBF="E" D CHANGE^IBCEP4A(1)  ;  Q  ; DUP(IBDA,IBOLD,IBFUNC) ; Check if the combination of ins co, prov type, care  ; type and form already exists in file 355.96  ; IBDA = ien of entry in file 355.96  ; IBOLD = the 0-node before changes were made - used to reset the fields  N DUP,IB0,DR,X,Y,DIK,DIE,DA  S IB0=$G(^IBA(355.96,IBDA,0)),DUP=0  ;  I $O(^IBA(355.96,"AUNIQ",+$P(IB0,U,3),+IB0,+$P(IB0,U,4),+$P(IB0,U,5),+$P(IB0,U,6),0))'=IBDA!($O(^IBA(355.96,"AUNIQ",+$P(IB0,U,3),+IB0,+$P(IB0,U,4),+$P(IB0,U,5),+$P(IB0,U,6),""),-1)'=IBDA) D  . S DUP=1  . I IBFUNC="E" D  .. S DR=";.01///"\_$P(IBOLD,U)\_";.03///"\_$S($P(IBOLD,U,3)'="":"/"\_$P(IBOLD,U,3),1:"@")\_";.04///"\_$S($P(IBOLD,U,4)'="":"/"\_$P(IBOLD,U,4),1:"@")  .. S DR=DR\_";05///"\_$S($P(IBOLD,U,5)'="":"/"\_$P(IBOLD,U,5),1:"@")\_";.06///"\_$S($P(IBOLD,U,6)'="":"/"\_$P(IBOLD,U,6),1:"@")  .. S DA=IBDA,DIE="^IBA(355.96," D ^DIE  . I IBFUNC="A" D  .. S DA=IBDA,DIK="^IBA(355.96," D ^DIK  Q DUP  ; PROFID(IBIFN,IBSEQ,IBID) ; Return id and type of rendering provider id  ; used for insurance co at COB seq IBSEQ for bill ien IBIFN  ; RETURN VALUES:  ; piece 1:  ; 1 = FEDERAL TAX ID  ; 2 = INSURANCE CO SPECIFIC ID  ; 3 = NETWORK ID  ; "" = not a CMS-1500 bill or no id found  ; piece 2:  ; the id #  N IBTYP,IBXDATA,IBZ  S:'$G(IBSEQ) IBSEQ=+$$COBN^IBCEF(IBXIEN)  S IBTYP=""\_U\_$G(IBID)  **;JWS;IB\*2.0\*592**  I $$FT^IBCEF(IBIFN)'=2**,$$FT^IBCEF(IBIFN)'=7** G PROFIDQ  I '$D(IBID) D F^IBCEF("N-ALL ATT/RENDERING PROV ID","IBZ",,IBIFN) S IBID=$$NOPUNCT^IBCEF($P(IBZ,U,IBSEQ+1))  G:IBID="" PROFIDQ  S IBTYP=$S($$NOPUNCT^IBCEF(IBID)=$$NOPUNCT^IBCEF($P($G(^IBE(350.9,1,1)),U,5)):1,$$NETWRK(IBIFN,IBID,IBSEQ):3,1:2)  S IBTYP=IBTYP\_U\_IBID  ; PROFIDQ Q IBTYP  ; NETWRK(IBIFN,IBID,IBSEQ) ; Determine if ID number IBID is the same as the  ; network id for the insurance co  ; IBIFN = bill ien (file 399)  ; IBSEQ = COB seq # of bill  ; Returns 1 if network ID match is found for bill IBIFN, COB seq IBSEQ  N IBINS,IBNET  S IBNET=0  Q IBNET  ; This section needs work \*\*\*\*\*\*\*\*\*  I '$G(IBSEQ) S IBSEQ=+$$COBN^IBCEF(IBXIEN)  S IBINS=+$G(^DGCR(399,IBIFN,"I"\_IBSEQ))  I $P($G(^IBE(355.97,+$$PPTYP^IBCEP0(IBINS),1)),U,6) D  . ; performing provider id type is a network id type  . I $$NOPUNCT^IBCEF($G(IBID))=$$NOPUNCT^IBCEF($$GETID^IBCEP2(IBIFN,3,$$PERFPRV^IBCEP2A(IBIFN),IBSEQ)) S IBNET=1  Q IBNET  ;  ;  ; Parameter definitions for UNIQ1 and UNIQ2 in IBCEP2  ; IBIFN = ien of bill (file 399)  ; IBINS = ien of insurance co (file 36) or \*ALL\* for all insurance  ; IBPTYP = the ien of the provider id type in file 355.97  ; IBUNIT = the value of the specific care unit to use for a match  ; or \*N/A\* if none needed  ; IBCU = the ien of the entry being matched in start file  ; IBT = the second and third pieces are set to the entry ien^file # | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCEP4 | | | | | | | | |
| **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** | | | | | | | | | |
| IBCEP4 ;ALB/TMP - EDI UTILITIES for provider ID ;29-SEP-00  ;;2.0;INTEGRATED BILLING;\*\*137,320,348,349,377\*\*;21-MAR-94;Build 23  ;;Per VHA Directive 2004-038, this routine should not be modified.  ; EN ; -- main entry point  N IBINS,IBALL,IB95  D ENX  Q  ; EN1(IBINS) ; -- Entry point from provider number maintenence  N IBPRV,IBALL,IB95  S VALMBCK="R"  D ENX  Q  ; ENX ; Common call to list template for dual entry points  N IBSLEV,DIR,Y  K IBFASTXT  D FULL^VALM1  S DIR(0)="SA^1:Performing Provider Care Units;2:Billing Provider Care Units"  S DIR("A")="Enter Type of Care Unit: ",DIR("B")=$P($P(DIR(0),":",2),";",1)  W ! D ^DIR K DIR W !  I Y'>0 Q  S IBSLEV=+Y  I IBSLEV=2 D EN^VALM("IBCE 2ND PRVID CARE UNIT MAINT") Q  D EN^VALM("IBCE PRVCARE UNIT MAINT")  Q  ; HDR ; -- header  K VALMHDR  S VALMHDR(1)=" "  S VALMHDR(2)="Insurance Co: "\_$S('$G(IBALL)&$G(IBINS):$P($G(^DIC(36,+IBINS,0)),U),1:"ALL")  Q  ; INIT ; -- init variables, list array  N Z,IB,IBLCT,IBENT,IBNM,IB0,Z0,Z1,IBQ,DIR,Y,X  I $G(IBINS) S Y=IBINS ; For entrypoint from provider number maintenance  ;  I '$G(IBINS) D  . S DIR(0)="PA^DIC(36,:AEMQ",DIR("A")="Select INSURANCE CO: ",DIR("?")="Select an INSURANCE CO to display its care units"  . D ^DIR K DIR  . I $D(DTOUT)!$D(DUOUT) S Y=-2 Q  . I Y>0 S IBINS=+Y Q  ;  I Y'=-2 D  . D BLD  E  D  . S VALMQUIT=1  Q  ; BLD ; Bld display - IBINS must = ien of file 36  K ^TMP("IBPRV\_CU",$J)  ;  I $G(IBSLEV)=2 Q  ;  S (IBENT,IBLCT)=0,IBNM=""  F  S IBNM=$O(^IBA(355.95,"C",IBINS,IBNM)) Q:IBNM=""  S Z=0 F  S Z=$O(^IBA(355.95,"C",IBINS,IBNM,Z)) Q:'Z  S IB=$G(^IBA(355.95,Z,0)) I IB'="",$P(IB,U,4)="" D  . S IBLCT=IBLCT+1,IBENT=IBENT+1  . I '$D(^IBA(355.96,"AUNIQ",IBINS,Z)) D SET^VALM10(IBLCT,$E(IBENT\_" ",1,4)\_$E($P(IB,U)\_$J("",30),1,30)\_" "\_$E($P(IB,U,2)\_$J("",20),1,20)\_" (NO COMBINATIONS FOUND)",IBENT) Q  . D SET^VALM10(IBLCT,$E(IBENT\_" ",1,4)\_$E($P(IB,U)\_$J("",30),1,30)\_" "\_$E($P(IB,U,2)\_$J("",20),1,20),IBENT)  . S ^TMP("IBPRV\_CU",$J,"ZIDX",IBENT)=Z  . S Z0=0 F  S Z0=$O(^IBA(355.96,"AE",Z,Z0)) Q:'Z0  S Z1=0 F  S Z1=$O(^IBA(355.96,"AE",Z,Z0,Z1)) Q:'Z1  S IB0=$G(^IBA(355.96,Z1,0)) I IB0'="" D  .. S IBLCT=IBLCT+1  .. S IBQ=$J("",28)\_"o "\_$E($$EXPAND^IBTRE(355.96,.06,+$P(IB0,U,6))\_$J("",20),1,20)  .. S IBQ=IBQ\_" "\_$E($P("Both form types^UB-04 Only^CMS-1500 Only",U,$P(IB0,U,4)+1)\_$J("",15),1,15)\_" "\_$E($P("Inpt/Outpt^Inpt Only^Outpt Only^RX Only",U,+$P(IB0,U,5)+1)\_$J("",10),1,10)  .. D SET^VALM10(IBLCT,IBQ,IBENT)  ;  I 'IBLCT D SET^VALM10(1,"No CARE UNITs Found"\_$S('$G(IBINS):"",1:" for Insurance Co")) S IBLCT=1  S VALMCNT=IBLCT,VALMBG=1  Q  ; HELP ; -- help  ;  I $G(IBSLEV)=2 Q  ;  S X="?" D DISP^XQORM1 W !!  Q  ; EXIT ; -- exit  D CLEAN^VALM10  K ^TMP("IBPRV\_CU",$J),IBINS,IBALL  Q  ; EXPND ;  Q  ; SEL(IBDA,MANY) ; Select from care unit list  ; IBDA is passed by reference and IBDA(1) returned containing  ; ien's of the care unit selected (file 355.95).  ; If > 1 entry can be selected, MANY is set to 1  N Z  S IBDA=0  D EN^VALM2($G(XQORNOD(0)),$S($G(MANY):"",1:"S"))  S Z=0 F  S Z=$O(VALMY(Z)) Q:'Z  S IBDA=IBDA+1,IBDA(IBDA)=+$G(^TMP("IBPRV\_CU",$J,"ZIDX",Z))  Q  ; DISP(IBVAR,IBINS,IBPTYP,IBFT,IBCT,START,END) ; Set up display array for  ; provider id  N Z  S START=$S($G(START):START,1:1)  S (Z,END)=$G(START)  S @IBVAR@(START)="INSURANCE: "\_$S(IBINS:$P($G(^DIC(36,+IBINS,0)),U),1:"ALL INSURANCE")  S @IBVAR@(START+1)="PROV TYPE: "\_$$EXPAND^IBTRE(355.96,.06,IBPTYP)  S @IBVAR@(START+2)="FORM TYPE: "\_$$EXPAND^IBTRE(355.96,.04,IBFT)  S @IBVAR@(START+3)="CARE TYPE: "\_$$EXPAND^IBTRE(355.96,.05,IBCT)  S END=$G(START)+3  Q  ; CAREUOK(IBIFN,IBCU,IBTYPE,IBSEQ) ; Returns 1 if care unit is appropriate   ; for bill based on provider type, care type, bill type and insurance co  ; IBIFN = ien of bill (file 399)  ; IBCU = the ien of the care unit (file 355.96)  ; IBTYPE = type of ID being checked (1=performing, 2=EMC)  ; IBSEQ = the COB seq being checked (1-3)  N Z,IBOK,IBINS,IBCT,IBFT,IBPTYP,IBRX  S IBOK=0  S IBINS=+$$FINDINS^IBCEF1(IBIFN,+IBSEQ),IBFT=$S($$FT^IBCEF(IBIFN)=2:2,1:1)  S IBPTYP=+$S(IBTYPE=1:$$PPTYP^IBCEP0(IBINS),1:$$EMCID^IBCEP())  S IBRX=$$ISRX^IBCEF1(IBIFN)  S IBCT=$S('IBRX:$S($$INPAT^IBCEF(IBIFN,1):1,1:2),1:3)  ;Check from most general to most specific  I $D(^IBA(355.96,"AD",IBINS,0,0,IBPTYP,IBCU)) S IBOK=1 G CAREOKQ  I 'IBRX,$D(^IBA(355.96,"AD",IBINS,IBFT,0,IBPTYP,IBCU)) S IBOK=1 G CAREOKQ  I $D(^IBA(355.96,"AD",IBINS,0,IBCT,IBPTYP,IBCU)) S IBOK=1 G CAREOKQ  I $D(^IBA(355.96,"AD",IBINS,IBFT,IBCT,IBPTYP,IBCU)) S IBOK=1 G CAREOKQ  ; CAREOKQ Q IBOK  ; | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCEP4 ;ALB/TMP - EDI UTILITIES for provider ID ;29-SEP-00  ;;2.0;INTEGRATED BILLING;\*\*137,320,348,349,377,**592**\*\*;21-MAR-94;Build 23  ;;Per VHA Directive 2004-038, this routine should not be modified.  ; EN ; -- main entry point  N IBINS,IBALL,IB95  D ENX  Q  ; EN1(IBINS) ; -- Entry point from provider number maintenence  N IBPRV,IBALL,IB95  S VALMBCK="R"  D ENX  Q  ; ENX ; Common call to list template for dual entry points  N IBSLEV,DIR,Y  K IBFASTXT  D FULL^VALM1  S DIR(0)="SA^1:Performing Provider Care Units;2:Billing Provider Care Units"  S DIR("A")="Enter Type of Care Unit: ",DIR("B")=$P($P(DIR(0),":",2),";",1)  W ! D ^DIR K DIR W !  I Y'>0 Q  S IBSLEV=+Y  I IBSLEV=2 D EN^VALM("IBCE 2ND PRVID CARE UNIT MAINT") Q  D EN^VALM("IBCE PRVCARE UNIT MAINT")  Q  ; HDR ; -- header  K VALMHDR  S VALMHDR(1)=" "  S VALMHDR(2)="Insurance Co: "\_$S('$G(IBALL)&$G(IBINS):$P($G(^DIC(36,+IBINS,0)),U),1:"ALL")  Q  ; INIT ; -- init variables, list array  N Z,IB,IBLCT,IBENT,IBNM,IB0,Z0,Z1,IBQ,DIR,Y,X  I $G(IBINS) S Y=IBINS ; For entrypoint from provider number maintenance  ;  I '$G(IBINS) D  . S DIR(0)="PA^DIC(36,:AEMQ",DIR("A")="Select INSURANCE CO: ",DIR("?")="Select an INSURANCE CO to display its care units"  . D ^DIR K DIR  . I $D(DTOUT)!$D(DUOUT) S Y=-2 Q  . I Y>0 S IBINS=+Y Q  ;  I Y'=-2 D  . D BLD  E  D  . S VALMQUIT=1  Q  ; BLD ; Bld display - IBINS must = ien of file 36  K ^TMP("IBPRV\_CU",$J)  ;  I $G(IBSLEV)=2 Q  ;  S (IBENT,IBLCT)=0,IBNM=""  F  S IBNM=$O(^IBA(355.95,"C",IBINS,IBNM)) Q:IBNM=""  S Z=0 F  S Z=$O(^IBA(355.95,"C",IBINS,IBNM,Z)) Q:'Z  S IB=$G(^IBA(355.95,Z,0)) I IB'="",$P(IB,U,4)="" D  . S IBLCT=IBLCT+1,IBENT=IBENT+1  . I '$D(^IBA(355.96,"AUNIQ",IBINS,Z)) D SET^VALM10(IBLCT,$E(IBENT\_" ",1,4)\_$E($P(IB,U)\_$J("",30),1,30)\_" "\_$E($P(IB,U,2)\_$J("",20),1,20)\_" (NO COMBINATIONS FOUND)",IBENT) Q  . D SET^VALM10(IBLCT,$E(IBENT\_" ",1,4)\_$E($P(IB,U)\_$J("",30),1,30)\_" "\_$E($P(IB,U,2)\_$J("",20),1,20),IBENT)  . S ^TMP("IBPRV\_CU",$J,"ZIDX",IBENT)=Z  . S Z0=0 F  S Z0=$O(^IBA(355.96,"AE",Z,Z0)) Q:'Z0  S Z1=0 F  S Z1=$O(^IBA(355.96,"AE",Z,Z0,Z1)) Q:'Z1  S IB0=$G(^IBA(355.96,Z1,0)) I IB0'="" D  .. S IBLCT=IBLCT+1  .. S IBQ=$J("",28)\_"o "\_$E($$EXPAND^IBTRE(355.96,.06,+$P(IB0,U,6))\_$J("",20),1,20)  **.. ;JRA IB\*2.0\*592 Modify to accommodate Dental Form J430D**  **.. ;**S IBQ=IBQ\_" "\_$E($P("Both form types^UB-04 Only^CMS-1500 Only",U,$P(IB0,U,4)+1)\_$J("",15),1,15)\_" "\_$E($P("Inpt/Outpt^Inpt Only^Outpt Only^RX Only",U,+$P(IB0,U,5)+1)\_$J("",10),1,10) **;JRA IB\*2.0\*592 ';'**  .. S IBQ=IBQ\_" "\_$E($P("**All** Form Types^UB-04 Only^CMS-1500 Only^^**J430D** Only",U,$P(IB0,U,4)+1)\_$J("",15),1,15)\_" "\_$E($P("Inpt/Outpt^Inpt Only^Outpt Only^RX Only",U,+$P(IB0,U,5)+1)\_$J("",10),1,10) **;JRA IB\*2.0\*592**  .. D SET^VALM10(IBLCT,IBQ,IBENT)  ;  I 'IBLCT D SET^VALM10(1,"No CARE UNITs Found"\_$S('$G(IBINS):"",1:" for Insurance Co")) S IBLCT=1  S VALMCNT=IBLCT,VALMBG=1  Q  ; HELP ; -- help  ;  I $G(IBSLEV)=2 Q  ;  S X="?" D DISP^XQORM1 W !!  Q  ; EXIT ; -- exit  D CLEAN^VALM10  K ^TMP("IBPRV\_CU",$J),IBINS,IBALL  Q  ; EXPND ;  Q  ; SEL(IBDA,MANY) ; Select from care unit list  ; IBDA is passed by reference and IBDA(1) returned containing  ; ien's of the care unit selected (file 355.95).  ; If > 1 entry can be selected, MANY is set to 1  N Z  S IBDA=0  D EN^VALM2($G(XQORNOD(0)),$S($G(MANY):"",1:"S"))  S Z=0 F  S Z=$O(VALMY(Z)) Q:'Z  S IBDA=IBDA+1,IBDA(IBDA)=+$G(^TMP("IBPRV\_CU",$J,"ZIDX",Z))  Q  ; DISP(IBVAR,IBINS,IBPTYP,IBFT,IBCT,START,END) ; Set up display array for  ; provider id  N Z  S START=$S($G(START):START,1:1)  S (Z,END)=$G(START)  S @IBVAR@(START)="INSURANCE: "\_$S(IBINS:$P($G(^DIC(36,+IBINS,0)),U),1:"ALL INSURANCE")  S @IBVAR@(START+1)="PROV TYPE: "\_$$EXPAND^IBTRE(355.96,.06,IBPTYP)  S @IBVAR@(START+2)="FORM TYPE: "\_$$EXPAND^IBTRE(355.96,.04,IBFT)  S @IBVAR@(START+3)="CARE TYPE: "\_$$EXPAND^IBTRE(355.96,.05,IBCT)  S END=$G(START)+3  Q  ; CAREUOK(IBIFN,IBCU,IBTYPE,IBSEQ) ; Returns 1 if care unit is appropriate   ; for bill based on provider type, care type, bill type and insurance co  ; IBIFN = ien of bill (file 399)  ; IBCU = the ien of the care unit (file 355.96)  ; IBTYPE = type of ID being checked (1=performing, 2=EMC)  ; IBSEQ = the COB seq being checked (1-3)  N Z,IBOK,IBINS,IBCT,IBFT,IBPTYP,IBRX  S IBOK=0  S IBINS=+$$FINDINS^IBCEF1(IBIFN,+IBSEQ),IBFT=$S($$FT^IBCEF(IBIFN)=2:2,1:1)  S IBPTYP=+$S(IBTYPE=1:$$PPTYP^IBCEP0(IBINS),1:$$EMCID^IBCEP())  S IBRX=$$ISRX^IBCEF1(IBIFN)  S IBCT=$S('IBRX:$S($$INPAT^IBCEF(IBIFN,1):1,1:2),1:3)  ;Check from most general to most specific  I $D(^IBA(355.96,"AD",IBINS,0,0,IBPTYP,IBCU)) S IBOK=1 G CAREOKQ  I 'IBRX,$D(^IBA(355.96,"AD",IBINS,IBFT,0,IBPTYP,IBCU)) S IBOK=1 G CAREOKQ  I $D(^IBA(355.96,"AD",IBINS,0,IBCT,IBPTYP,IBCU)) S IBOK=1 G CAREOKQ  I $D(^IBA(355.96,"AD",IBINS,IBFT,IBCT,IBPTYP,IBCU)) S IBOK=1 G CAREOKQ  ; CAREOKQ Q IBOK  ; | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCEP5 | | | | | | | | |
| **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** | | | | | | | | | |
| IBCEP5 ;ALB/TMP - EDI UTILITIES for provider ID ;29-SEP-00  ;;2.0;INTEGRATED BILLING;\*\*137,232,320,348,349,377\*\*;21-MAR-94;Build 23  ;;Per VHA Directive 2004-038, this routine should not be modified.  ; EN ; -- main entry point for IBCE PRV MAINT  N IBPRV,IBINS EN1 ; Entrypoint for non-VA provider ID maintenance hook  N IBSLEV,DIR,Y,X,IBPRMPT,IBNVAFL,IBIF  K IBFASTXT  S IBIF="" I $G(IBPRV) S IBIF=$$GET1^DIQ(355.93,IBPRV,.02,"I")  D FULL^VALM1  S IBPRMPT=$S(IBIF=1:"LAB OR FACILITY",1:"PROVIDER")  S DIR(0)="SA^1:"\_IBPRMPT\_"'S OWN IDS;2:"\_IBPRMPT\_" IDS FURNISHED BY AN INSURANCE COMPANY"  S DIR("A")="SELECT SOURCE OF ID: ",DIR("B")=$P($P(DIR(0),":",2),";")  W ! D ^DIR K DIR W !  I Y'>0 Q  S IBSLEV=+Y  D EN^VALM("IBCE PRVPRV MAINT")  Q  ; HDR ; -- header code  N IBC,Z,IBIF  S IBIF="" I $G(IBNPRV) S IBIF=$$GET1^DIQ(355.93,IBNPRV,.02,"I")  K VALMHDR  S IBC=1  S IBPRMPT=$S(IBIF=1:"Lab or Facility",1:"Performing Provider")  S Z="\*\* "\_$S($G(IBSLEV)=1:IBPRMPT\_"'s Own IDs (No Specific Insurance Co)",1:IBPRMPT\_" IDs from Insurance Co")\_" \*\*"  S VALMHDR(IBC)=$J("",80-$L(Z)\2)\_Z,IBC=IBC+1  I $G(IBPRV),'+IBIF S VALMHDR(IBC)="PROVIDER : "\_$$EXPAND^IBTRE(355.9,.01,IBPRV)\_$S(IBPRV["VA(200":" (VA PROVIDER)",1:" (NON-VA PROVIDER)"),IBC=IBC+1  I $G(IBPRV),+IBIF S VALMHDR(IBC)="Provider: "\_$$EXPAND^IBTRE(355.9,.01,IBPRV)\_$S(IBIF=1:"(Non-VA Lab or Facility)",1:""),IBC=IBC+1  I $G(IBINS) D  . N PCF,PCDISP  . S PCF=$P($G(^DIC(36,+IBINS,3)),"^",13)  . S PCDISP=$S($G(IBSLEV)'=2!($G(IBPRV)'["VA(200,"):"",PCF="C":"(Child)",PCF="P":"(Parent)",1:"")  . S VALMHDR(IBC)=$S(IBIF:"Insurance Co: ",1:"INSURANCE CO: ")\_$P($G(^DIC(36,+IBINS,0)),U)\_" "\_PCDISP  Q  ; INIT ; -- init variables and list array  N IBFILE,DIR,DIC,Y,X,DTOUT,DUOUT,IBIF,AGAIN  ;  K ^TMP("IB\_EDITED\_IDS",$J) ; This will be to keep track of ID's edited during this session  S IBIF="" I $G(IBNPRV) S IBIF=$$GET1^DIQ(355.93,IBNPRV,.02,"I")  ;  ; Removing Care Unit under certain conditions  ; This list is used for multiple purposes and not all have Care Units Associated with them  ; Also, a different protocol menu is used with these  ; IBNPRV is a non VA provider  ; IBIF = 1 means this is a group or facility, not an individual.  ;   I $G(IBNPRV),$G(IBIF)=1 D  . S VALM("TITLE")="Secondary Provider ID"  . K VALMDDF("CAREUNIT")  . I VALMCAP["Care Unit" S VALMCAP=$P(VALMCAP,"Care Unit")\_" "\_$P(VALMCAP,"Care Unit",2)  . K VALM("PROTOCOL")  . S Y=$$FIND1^DIC(101,,,"IBCE PRVNVA LOF MAINT")  . I Y S VALM("PROTOCOL")=+Y\_";ORD(101,"  ;  I $G(IBPRV) S IBFILE="IBA(355.93,",IBPRV=+IBPRV\_";"\_IBFILE  I '$G(IBPRV) D  G:$G(VALMQUIT) INITQ  . S DIR(0)="SAO^V:VA PROVIDER;N:NON-VA PROVIDER",DIR("A")="(V)A or (N)on-VA provider: ",DIR("B")="V"  . D ^DIR K DIR  . I "NV"'[Y!(Y="") S VALMQUIT=1 Q  . S IBFILE=$S(Y="V":"VA(200,",1:"IBA(355.93,")  . S DIC=U\_IBFILE,DIC(0)="AEMQ"\_$S(IBFILE["355.93":"L",1:"")  . S DIC("A")="Select "\_$S(IBFILE["355.93":"NON-",1:"")\_"V.A. PROVIDER NAME: "  . S:IBFILE["355.93" DIC("DR")=".02////2;.03;.04"  . F  D  I $G(IBPRV)!$G(VALMQUIT) K DIC Q  .. D ^DIC  .. I $D(DTOUT)!$D(DUOUT) S VALMQUIT=1 Q  .. I Y'>0 W !,\*7,"This is a required response. Enter '^' to exit" Q  .. S IBPRV=+Y\_";"\_IBFILE  ; AGAIN I $G(IBSLEV)=2 D  G:$G(AGAIN) AGAIN G:$G(VALMQUIT) INITQ  . S AGAIN=0  . S DIR(0)="PA^DIC(36,:AEMQ",DIR("A")="Select INSURANCE CO: ",DIR("?",1)="Select an INSURANCE CO to display its provider ID's"  . D ^DIR K DIR  . I $D(DTOUT)!$D(DUOUT) S VALMQUIT=1 Q  . S IBINS=$S(Y>0:+Y,1:"NO")  . I $G(IBPRV)'["VA(200," Q    ; Only VA providers  . I $P($G(^DIC(36,+IBINS,3)),"^",13)="C" D  S AGAIN=1 Q  .. W !,\*7,"This is a Child Insurance Company. Editing IDs is not permitted."  ;  E  D  . S IBINS="NO"  D BLD INITQ Q  ; BLD ; Build initial display  ; Assumes IBPRV = the variable ptr for prov id file (355.9)  ; IBINS = the ien of the ins co or if null, ALL is assumed  ; IBSLEV = 1 to display only provider default ids  ; = 2 to display all provider/insurance co ids  N IB,IBLCT,IBCT,CT,PT,CU,INS,FT,Z,IBENT,IB1,IBIF  ;  S IBIF="" I $G(IBPRV)[355.93 S IBIF=$$GET1^DIQ(355.93,+IBPRV,.02,"I")  ;  K ^TMP("IBPRV\_",$J),^TMP("IBPRV\_SORT",$J)  K Z0  S (IBENT,IBCT,IBLCT)=0,INS="",IB1=1  F  S INS=$S($G(IBINS):IBINS,IBSLEV=1:"\*ALL\*",1:$O(^IBA(355.9,"AUNIQ",IBPRV,INS))) Q:$S(INS="":1,$G(IBINS)!(IBSLEV=1):$D(CU),1:0) S CU="",IB1=0 F  S CU=$O(^IBA(355.9,"AUNIQ",IBPRV,INS,CU)) Q:CU=""  D  . S FT="" F  S FT=$O(^IBA(355.9,"AUNIQ",IBPRV,INS,CU,FT)) Q:FT=""  S CT="" F  S CT=$O(^IBA(355.9,"AUNIQ",IBPRV,INS,CU,FT,CT)) Q:CT=""  S PT=0 F  S PT=$O(^IBA(355.9,"AUNIQ",IBPRV,INS,CU,FT,CT,PT)) Q:'PT  D  .. S Z=0 F  S Z=$O(^IBA(355.9,"AUNIQ",IBPRV,INS,CU,FT,CT,PT,Z)) Q:'Z  S IB=$G(^IBA(355.9,Z,0)) D  ... S ^TMP("IBPRV\_SORT",$J,$S(INS:$P($G(^DIC(36,+INS,0)),U)\_" ",1:" ALL"),PT,FT,CT,CU,Z)=$P(IB,U,7)  ;  I IBSLEV=1,IBPRV["IBA(355.93",$P($G(^IBA(355.93,+IBPRV,0)),U,12)'="" S ^TMP("IBPRV\_SORT",$J," ALL",+$$STLIC^IBCEP8(),0,0,"\*N/A\*",0)=$P(^IBA(355.93,+IBPRV,0),U,12)  S INS="" F  S INS=$O(^TMP("IBPRV\_SORT",$J,INS)) Q:INS=""  D  . I '$G(IBINS),'IBIF D:IBLCT SET^VALM10(IBLCT+1," ",IBCT) S IBLCT=$S(IBLCT:IBLCT+2,1:1) D SET^VALM10(IBLCT,"INSURANCE CO: "\_$S($E(INS)=" ":"ALL INSURANCE",1:INS),$S(IBCT:IBCT,1:1))  . S PT=""  . F  S PT=$O(^TMP("IBPRV\_SORT",$J,INS,PT)) Q:PT=""  S FT="" F  S FT=$O(^TMP("IBPRV\_SORT",$J,INS,PT,FT)) Q:FT=""  S CT="" F  S CT=$O(^TMP("IBPRV\_SORT",$J,INS,PT,FT,CT)) Q:CT=""  D  .. S CU="" F  S CU=$O(^TMP("IBPRV\_SORT",$J,INS,PT,FT,CT,CU)) Q:CU=""  S Z="" F  S Z=$O(^TMP("IBPRV\_SORT",$J,INS,PT,FT,CT,CU,Z)) Q:Z=""  S IB=$G(^(Z)) D  ... S IBLCT=IBLCT+1,IBCT=IBCT+1  ... S Z0=$E(IBCT\_" ",1,4)\_" "\_$E($$EXPAND^IBTRE(355.9,.06,PT)\_$S(PT=$$STLIC^IBCEP8():"("\_$P($G(^DIC(5,+$P($G(^IBA(355.93,+IBPRV,0)),U,7),0)),U,2)\_")",1:"")\_$J("",20),1,20)\_" "\_$S(FT=1:"UB-04",FT=2:"1500 ",1:"BOTH ")  ... S Z0=Z0\_" "\_$E($S(CT=3:"RX",CT=1:"INPT",CT=2:"OUTPT",1:"INPT/OUTPT")\_$J("",11),1,11)  ... S Z0=Z0\_" "\_$E($S(CU'="\*N/A\*":$P($G(^IBA(355.95,+$G(^IBA(355.96,CU,0)),0)),U),1:"")\_$J("",15),1,15) I Z0["MEDICINE" X "\*"  ... D SET^VALM10(IBLCT,Z0\_" "\_IB,IBCT)  ... S ^TMP("IBPRV\_",$J,"ZIDX",IBCT)=$S(Z'=0:Z,1:"LIC^"\_IBPRV)  I IBSLEV=1,IBPRV["VA(200" D  . N IBP  . S IBP=+IBPRV  . Q:'$$GETLIC^IBCEP5D(.IBP)  . I IBCT S IBLCT=IBLCT+1 D SET^VALM10(IBLCT," ",IBCT)  . S Z=0 F  S Z=$O(IBP(Z)) Q:'Z  D  .. S IBLCT=IBLCT+1,IBCT=IBCT+1  .. D SET^VALM10(IBLCT,$E(IBCT\_" ",1,4)\_$E($P($G(^DIC(5,+Z,0)),U,2)\_" STATE LICENSE #"\_$J("",20),1,20)\_$J("",39)\_IBP(Z),IBCT)  .. S ^TMP("IBPRV\_",$J,"ZIDX",IBCT)="LIC^"\_+IBPRV  K ^TMP("IBPRV\_SORT",$J)  ;  I IBLCT=0 D  G BLDQ ; No entries for ins co selected  . D SET^VALM10(1," ")  . D SET^VALM10(2," No ID's found for provider "\_$S('$G(IBINS):"",1:"and selected insurance co"))  . S IBLCT=2  ; BLDQ K VALMCNT,VALMBG  S VALMCNT=IBLCT,VALMBG=1  Q  ; HELP ; -- help code  S X="?" D DISP^XQORM1 W !!  Q  ; EXIT ; -- exit code  D COPYPROV^IBCEP5A(IBINS)  K IBPRV  D CLEAN^VALM10  K ^TMP("IBPRV\_",$J),^TMP("IBPRV\_SORT",$J),IBINS,IBALL  Q  ; EXPND ; -- expand code  Q  ; SEL(IBDA,MANY) ; Select from provider id list  ; IBDA is passed by reference and IBDA(1) returned containing  ; ien's of the provider id records selected (file 355.9).  ; If > 1 entry can be selected, MANY is set to 1  N Z  S IBDA=0  D EN^VALM2($G(XQORNOD(0)),$S($G(MANY):"",1:"S"))  S Z=0 F  S Z=$O(VALMY(Z)) Q:'Z  S IBDA=IBDA+1,IBDA(IBDA)=$G(^TMP("IBPRV\_",$J,"ZIDX",Z))  Q  ; | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCEP5 ;ALB/TMP - EDI UTILITIES for provider ID ;29-SEP-00  ;;2.0;INTEGRATED BILLING;\*\*137,232,320,348,349,377,**592**\*\*;21-MAR-94;Build 23  ;;Per VHA Directive 2004-038, this routine should not be modified.  ; EN ; -- main entry point for IBCE PRV MAINT  N IBPRV,IBINS EN1 ; Entrypoint for non-VA provider ID maintenance hook  N IBSLEV,DIR,Y,X,IBPRMPT,IBNVAFL,IBIF  K IBFASTXT  S IBIF="" I $G(IBPRV) S IBIF=$$GET1^DIQ(355.93,IBPRV,.02,"I")  D FULL^VALM1  S IBPRMPT=$S(IBIF=1:"LAB OR FACILITY",1:"PROVIDER")  S DIR(0)="SA^1:"\_IBPRMPT\_"'S OWN IDS;2:"\_IBPRMPT\_" IDS FURNISHED BY AN INSURANCE COMPANY"  S DIR("A")="SELECT SOURCE OF ID: ",DIR("B")=$P($P(DIR(0),":",2),";")  W ! D ^DIR K DIR W !  I Y'>0 Q  S IBSLEV=+Y  D EN^VALM("IBCE PRVPRV MAINT")  Q  ; HDR ; -- header code  N IBC,Z,IBIF  S IBIF="" I $G(IBNPRV) S IBIF=$$GET1^DIQ(355.93,IBNPRV,.02,"I")  K VALMHDR  S IBC=1  S IBPRMPT=$S(IBIF=1:"Lab or Facility",1:"Performing Provider")  S Z="\*\* "\_$S($G(IBSLEV)=1:IBPRMPT\_"'s Own IDs (No Specific Insurance Co)",1:IBPRMPT\_" IDs from Insurance Co")\_" \*\*"  S VALMHDR(IBC)=$J("",80-$L(Z)\2)\_Z,IBC=IBC+1  I $G(IBPRV),'+IBIF S VALMHDR(IBC)="PROVIDER : "\_$$EXPAND^IBTRE(355.9,.01,IBPRV)\_$S(IBPRV["VA(200":" (VA PROVIDER)",1:" (NON-VA PROVIDER)"),IBC=IBC+1  I $G(IBPRV),+IBIF S VALMHDR(IBC)="Provider: "\_$$EXPAND^IBTRE(355.9,.01,IBPRV)\_$S(IBIF=1:"(Non-VA Lab or Facility)",1:""),IBC=IBC+1  I $G(IBINS) D  . N PCF,PCDISP  . S PCF=$P($G(^DIC(36,+IBINS,3)),"^",13)  . S PCDISP=$S($G(IBSLEV)'=2!($G(IBPRV)'["VA(200,"):"",PCF="C":"(Child)",PCF="P":"(Parent)",1:"")  . S VALMHDR(IBC)=$S(IBIF:"Insurance Co: ",1:"INSURANCE CO: ")\_$P($G(^DIC(36,+IBINS,0)),U)\_" "\_PCDISP  Q  ; INIT ; -- init variables and list array  N IBFILE,DIR,DIC,Y,X,DTOUT,DUOUT,IBIF,AGAIN  ;  K ^TMP("IB\_EDITED\_IDS",$J) ; This will be to keep track of ID's edited during this session  S IBIF="" I $G(IBNPRV) S IBIF=$$GET1^DIQ(355.93,IBNPRV,.02,"I")  ;  ; Removing Care Unit under certain conditions  ; This list is used for multiple purposes and not all have Care Units Associated with them  ; Also, a different protocol menu is used with these  ; IBNPRV is a non VA provider  ; IBIF = 1 means this is a group or facility, not an individual.  ;   I $G(IBNPRV),$G(IBIF)=1 D  . S VALM("TITLE")="Secondary Provider ID"  . K VALMDDF("CAREUNIT")  . I VALMCAP["Care Unit" S VALMCAP=$P(VALMCAP,"Care Unit")\_" "\_$P(VALMCAP,"Care Unit",2)  . K VALM("PROTOCOL")  . S Y=$$FIND1^DIC(101,,,"IBCE PRVNVA LOF MAINT")  . I Y S VALM("PROTOCOL")=+Y\_";ORD(101,"  ;  I $G(IBPRV) S IBFILE="IBA(355.93,",IBPRV=+IBPRV\_";"\_IBFILE  I '$G(IBPRV) D  G:$G(VALMQUIT) INITQ  . S DIR(0)="SAO^V:VA PROVIDER;N:NON-VA PROVIDER",DIR("A")="(V)A or (N)on-VA provider: ",DIR("B")="V"  . D ^DIR K DIR  . I "NV"'[Y!(Y="") S VALMQUIT=1 Q  . S IBFILE=$S(Y="V":"VA(200,",1:"IBA(355.93,")  . S DIC=U\_IBFILE,DIC(0)="AEMQ"\_$S(IBFILE["355.93":"L",1:"")  . S DIC("A")="Select "\_$S(IBFILE["355.93":"NON-",1:"")\_"V.A. PROVIDER NAME: "  . S:IBFILE["355.93" DIC("DR")=".02////2;.03;.04"  . F  D  I $G(IBPRV)!$G(VALMQUIT) K DIC Q  .. D ^DIC  .. I $D(DTOUT)!$D(DUOUT) S VALMQUIT=1 Q  .. I Y'>0 W !,\*7,"This is a required response. Enter '^' to exit" Q  .. S IBPRV=+Y\_";"\_IBFILE  ; AGAIN I $G(IBSLEV)=2 D  G:$G(AGAIN) AGAIN G:$G(VALMQUIT) INITQ  . S AGAIN=0  . S DIR(0)="PA^DIC(36,:AEMQ",DIR("A")="Select INSURANCE CO: ",DIR("?",1)="Select an INSURANCE CO to display its provider ID's"  . D ^DIR K DIR  . I $D(DTOUT)!$D(DUOUT) S VALMQUIT=1 Q  . S IBINS=$S(Y>0:+Y,1:"NO")  . I $G(IBPRV)'["VA(200," Q    ; Only VA providers  . I $P($G(^DIC(36,+IBINS,3)),"^",13)="C" D  S AGAIN=1 Q  .. W !,\*7,"This is a Child Insurance Company. Editing IDs is not permitted."  ;  E  D  . S IBINS="NO"  D BLD INITQ Q  ; BLD ; Build initial display  ; Assumes IBPRV = the variable ptr for prov id file (355.9)  ; IBINS = the ien of the ins co or if null, ALL is assumed  ; IBSLEV = 1 to display only provider default ids  ; = 2 to display all provider/insurance co ids  N IB,IBLCT,IBCT,CT,PT,CU,INS,FT,Z,IBENT,IB1,IBIF,FORM,CAREUNT,CARETYP  **;JRA IB\*2.0\*592 Added: FORM,CAREUNT,CARETYP**  ;  S IBIF="" I $G(IBPRV)[355.93 S IBIF=$$GET1^DIQ(355.93,+IBPRV,.02,"I")  ;  K ^TMP("IBPRV\_",$J),^TMP("IBPRV\_SORT",$J)  K Z0  S (IBENT,IBCT,IBLCT)=0,INS="",IB1=1  F  S INS=$S($G(IBINS):IBINS,IBSLEV=1:"\*ALL\*",1:$O(^IBA(355.9,"AUNIQ",IBPRV,INS))) Q:$S(INS="":1,$G(IBINS)!(IBSLEV=1):$D(CU),1:0) S CU="",IB1=0 F  S CU=$O(^IBA(355.9,"AUNIQ",IBPRV,INS,CU)) Q:CU=""  D  . S FT="" F  S FT=$O(^IBA(355.9,"AUNIQ",IBPRV,INS,CU,FT)) Q:FT=""  S CT="" F  S CT=$O(^IBA(355.9,"AUNIQ",IBPRV,INS,CU,FT,CT)) Q:CT=""  S PT=0 F  S PT=$O(^IBA(355.9,"AUNIQ",IBPRV,INS,CU,FT,CT,PT)) Q:'PT  D  .. S Z=0 F  S Z=$O(^IBA(355.9,"AUNIQ",IBPRV,INS,CU,FT,CT,PT,Z)) Q:'Z  S IB=$G(^IBA(355.9,Z,0)) D  ... S ^TMP("IBPRV\_SORT",$J,$S(INS:$P($G(^DIC(36,+INS,0)),U)\_" ",1:" ALL"),PT,FT,CT,CU,Z)=$P(IB,U,7)  ;  I IBSLEV=1,IBPRV["IBA(355.93",$P($G(^IBA(355.93,+IBPRV,0)),U,12)'="" S ^TMP("IBPRV\_SORT",$J," ALL",+$$STLIC^IBCEP8(),0,0,"\*N/A\*",0)=$P(^IBA(355.93,+IBPRV,0),U,12)  S INS="" F  S INS=$O(^TMP("IBPRV\_SORT",$J,INS)) Q:INS=""  D  . I '$G(IBINS),'IBIF D:IBLCT SET^VALM10(IBLCT+1," ",IBCT) S IBLCT=$S(IBLCT:IBLCT+2,1:1) D SET^VALM10(IBLCT,"INSURANCE CO: "\_$S($E(INS)=" ":"ALL INSURANCE",1:INS),$S(IBCT:IBCT,1:1))  . S PT=""  . F  S PT=$O(^TMP("IBPRV\_SORT",$J,INS,PT)) Q:PT=""  S FT="" F  S FT=$O(^TMP("IBPRV\_SORT",$J,INS,PT,FT)) Q:FT=""  S CT="" F  S CT=$O(^TMP("IBPRV\_SORT",$J,INS,PT,FT,CT)) Q:CT=""  D  .. S CU="" F  S CU=$O(^TMP("IBPRV\_SORT",$J,INS,PT,FT,CT,CU)) Q:CU=""  S Z="" F  S Z=$O(^TMP("IBPRV\_SORT",$J,INS,PT,FT,CT,CU,Z)) Q:Z=""  S IB=$G(^(Z)) D  ... S IBLCT=IBLCT+1,IBCT=IBCT+1 **... ;JRA IB\*2.0\*592 Modify to accommodate Dental Form 7 (FT=4)**  ... **;**S Z0=$E(IBCT\_" ",1,4)\_" "\_$E($$EXPAND^IBTRE(355.9,.06,PT)\_$S(PT=$$STLIC^IBCEP8():"("\_$P($G(^DIC(5,+$P($G(^IBA(355.93,+IBPRV,0)),U,7),0)),U,2)\_")",1:"")\_$J("",20),1,20)\_" "\_$S(FT=1:"UB-04",FT=2:"1500 ",1:"BOTH ") **;JRA IB\*2.0\*592 ';'**  **... S FORM=$S(FT=1:"UB-04",FT=2:"CMS-1500",FT=4:"J430D",1:"ALL") ;JRA IB\*2.0\*592**  **... S Z0=$E(IBCT\_" ",1,4)\_" "\_$E($$EXPAND^IBTRE(355.9,.06,PT)\_$S(PT=$$STLIC^IBCEP8():"("\_$P($G(^DIC(5,+$P($G(^IBA(355.93,+IBPRV,0)),U,7),0)),U,2)\_")",1:"")\_$J("",20),1,20)\_" "\_FORM  ;JRA IB\*2.0\*592**  **... ;**S Z0=Z0\_" "\_$E($S(CT=3:"RX",CT=1:"INPT",CT=2:"OUTPT",1:"INPT/OUTPT")\_$J("",11),1,11) **;JRA IB\*2.0\*592 ';'**  **... S CARETYP=$E($S(CT=3:"RX",CT=1:"INPT",CT=2:"OUTPT",1:"INPT/OUTPT"),1,10) ;JRA IB\*2.0\*592  ... S Z0=Z0\_$J("",11-$L(FORM))\_CARETYP  ;JRA IB\*2.0\*592**  **... ;S** Z0=Z0\_" "\_$E($S(CU'="\*N/A\*":$P($G(^IBA(355.95,+$G(^IBA(355.96,CU,0)),0)),U),1:"")\_$J("",15),1,15)\_"|" I Z0["MEDICINE" X "\*" **;JRA IB\*2.0\*592 ';'**  ... **S CAREUNT=$E($S(CU'="\*N/A\*":$P($G(^IBA(355.95,+$G(^IBA(355.96,CU,0)),0)),U),1:""),1,12) ;JRA IB\*2.0\*592  ... S CAREUNT=CAREUNT\_$J("",12-$L(CAREUNT)+1) ;JRA IB\*2.0\*592  ... S Z0=Z0\_($J("",(12-$L(CARETYP)+1))) ;JRA IB\*2.0\*592  ... S Z0=Z0\_CAREUNT  ;JRA IB\*2.0\*592  ... ;D SET^VALM10(IBLCT,Z0\_" "\_IB,IBCT) ;JRA IB\*2.0\*592 ';'  ... D SET^VALM10(IBLCT,Z0\_IB,IBCT) ;JRA IB\*2.0\*592**  ... S ^TMP("IBPRV\_",$J,"ZIDX",IBCT)=$S(Z'=0:Z,1:"LIC^"\_IBPRV)  I IBSLEV=1,IBPRV["VA(200" D  . N IBP  . S IBP=+IBPRV  . Q:'$$GETLIC^IBCEP5D(.IBP)  . I IBCT S IBLCT=IBLCT+1 D SET^VALM10(IBLCT," ",IBCT)  . S Z=0 F  S Z=$O(IBP(Z)) Q:'Z  D  .. S IBLCT=IBLCT+1,IBCT=IBCT+1  .. D SET^VALM10(IBLCT,$E(IBCT\_" ",1,4)\_$E($P($G(^DIC(5,+Z,0)),U,2)\_" STATE LICENSE #"\_$J("",20),1,20)\_$J("",39)\_IBP(Z),IBCT)  .. S ^TMP("IBPRV\_",$J,"ZIDX",IBCT)="LIC^"\_+IBPRV  K ^TMP("IBPRV\_SORT",$J)  ;  I IBLCT=0 D  G BLDQ ; No entries for ins co selected  . D SET^VALM10(1," ")  . D SET^VALM10(2," No ID's found for provider "\_$S('$G(IBINS):"",1:"and selected insurance co"))  . S IBLCT=2  ; BLDQ K VALMCNT,VALMBG  S VALMCNT=IBLCT,VALMBG=1  Q  ; HELP ; -- help code  S X="?" D DISP^XQORM1 W !!  Q  ; EXIT ; -- exit code  D COPYPROV^IBCEP5A(IBINS)  K IBPRV  D CLEAN^VALM10  K ^TMP("IBPRV\_",$J),^TMP("IBPRV\_SORT",$J),IBINS,IBALL  Q  ; EXPND ; -- expand code  Q  ; SEL(IBDA,MANY) ; Select from provider id list  ; IBDA is passed by reference and IBDA(1) returned containing  ; ien's of the provider id records selected (file 355.9).  ; If > 1 entry can be selected, MANY is set to 1  N Z  S IBDA=0  D EN^VALM2($G(XQORNOD(0)),$S($G(MANY):"",1:"S"))  S Z=0 F  S Z=$O(VALMY(Z)) Q:'Z  S IBDA=IBDA+1,IBDA(IBDA)=$G(^TMP("IBPRV\_",$J,"ZIDX",Z))  Q  ; | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCEP5B | | | | | | | | |
| **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** | | | | | | | | | |
| IBCEP5B ;ALB/TMP - EDI UTILITIES for prov ID ;29-SEP-00  ;;2.0;INTEGRATED BILLING;\*\*137,239,232,320,348,349\*\*;21-MAR-94;Build 46  ;;Per VHA Directive 2004-038, this routine should not be modified.  ; NEWID(IBFILE,IBINS,IBPRV,IBPTYP,IBIEN,IBF) ; Generic add prov id  ; at both prov (file 355.9) and ins co levels (355.91)  ; IBFILE = 355.9 or 355.91 - the file being edited  ; IBINS = ien of ins co (36) or \*ALL\* for all ins co  ; IBPRV = vp ien of billing prov  ; IBPTYP = ien of prov type (file 355.97)  ; IBIEN = ien of entry being added (req'd)  ; IBF = 1 if deleting from ins-related options, "" from prov-related  N DIC,DIR,X,Y,Z,DA,DR,DIE,DO,DD,DLAYGO,DTOUT,DUOUT,IBQ,IBCUND,IB3559,IB35591,Q,IBDR,IBID,AFT  S IB35591(.03)="",IBPTYP=$G(IBPTYP)  F Z=.04,.05,.03 D  G:Z="" NEWQ  . I $S(Z'=.03:1,1:$S('$G(IBINS):0,1:$G(IBCUND))) D  Q:Z=""  .. N DA  .. I Z'=.03 S DIR(0)=IBFILE\_","\_Z  .. I Z=.03 D  ... S DIR(0)="PAO^355.95:AEMQ"  ... S DIR("S")="I $O(^IBA(355.96,""AUNIQ"","\_IBINS\_",Y,"\_$G(IB3559(.04))\_","\_$G(IB3559(.05))\_","\_IBPTYP\_",0))!($O(^IBA(355.96,""AUNIQ"","\_IBINS\_",Y,"\_$G(IB3559(.04))\_",0,"\_IBPTYP\_",0)))"  ... S DIR("S")=DIR("S")\_"!($O(^IBA(355.96,""AUNIQ"","\_IBINS\_",Y,0,"\_$G(IB3559(.05))\_","\_IBPTYP\_",0)))!($O(^IBA(355.96,""AUNIQ"","\_IBINS\_",Y,0,0,"\_IBPTYP\_",0)))"  ... S DIR("?",1)="Care unit describes areas of service and is assigned by the payer, if",DIR("?")=" applicable. Use Care Unit Maintenance to add or modify care units."  .. ;  .. I Z=.04,IBPRV["355.93",$$GET1^DIQ(355.93,+IBPRV,.02,"I")=1 D  ... I $$GET1^DIQ(355.97,IBPTYP,.03,"I")="EI" S $P(DIR(0),U,3)="K:Y'=1 X",DIR("?")="Provider ID Qualifier selected only allows institutional (UB type) forms" Q  ... I $$GET1^DIQ(355.97,IBPTYP,.03,"I")="TJ" S $P(DIR(0),U,3)="K:Y'=2 X",DIR("?")="Provider ID Qualifier selected only allows professional (CMS-1500) forms" Q  ... N AFT  ... S AFT=$$GET1^DIQ(355.97,IBPTYP,.07,"I") ; get allowable form type for this Provider ID Type  ... I AFT="B" S $P(DIR(0),U,3)="K:"".0.1.2.""'[("".""\_Y\_""."") X",DIR("?")="Provider ID Qualifier selected allows institutional, professional or both" Q  ... I AFT="P" S $P(DIR(0),U,3)="K:Y'=2 X",DIR("?")="Provider ID Qualifier selected only allows professional (CMS-1500) forms" Q  ... I AFT="I" S $P(DIR(0),U,3)="K:Y'=1 X",DIR("?")="Provider ID Qualifier selected only allows institutional (UB type) forms" Q  .. ;  .. S DA=0  .. I Z=.04,$P($G(^IBE(355.97,+IBPTYP,0)),U,3)="1A" D SETDIR(.DIR)  .. D ^DIR K DIR  .. I $D(DTOUT)!$D(DUOUT) S Z="" K IB3559,IB35591 Q  .. S IB3559(Z)=$S(Z'=.03:$P(Y,U),1:$S($P(Y,U)>0:$P(Y,U),1:"\*N/A\*"))  . I Z=.05 D  .. S IBCUND=$$CAREUN^IBCEP3(IBINS,IBPTYP,IB3559(.04),IB3559(.05),IB3559(.05)=3)  .. S:'IBCUND!($G(IB3559(.03))=0) IB3559(.03)="\*N/A\*"  .. I '$G(IBINS) S IBINS="\*ALL\*"  . I Z=.03 D CAREUN^IBCEP5C  ;  I $D(IB3559) D  . N Q,Z2,Z3,Z4,Z5,Z6,IBLAST,IBOK,DIR,Y,X  . S IBLAST=0  . D DISP^IBCEP4("Q",IBINS,IBPTYP,IB3559(.04),IB3559(.05),1)  . W !!,"THE FOLLOWING WAS CHOSEN:"  . S Q=0 F  S Q=$O(Q(Q)) Q:'Q  W !,?3,Q(Q)  . I IBCUND W !,?3,"CARE UNIT: "\_$$EXPAND^IBTRE(355.96,.01,IB3559(.03))  . S Z2=IBINS,Z3=IB35591(.03),Z4=IB3559(.04),Z5=IB3559(.05),Z6=IBPTYP  . S IBOK=1  . ; If both forms, chk for specific  . I 'Z4 S IBOK=$$COMBOK^IBCEP5C(IBFILE,IBPRV\_U\_4\_U\_Z2\_U\_Z3\_U\_Z4\_U\_Z5\_U\_Z6,1,$G(IBFILE)=355.91)  . ; If specific form, chk for all  . I IBOK,Z4 S IBOK=$$COMBOK^IBCEP5C(IBFILE,IBPRV\_U\_4\_U\_Z2\_U\_Z3\_U\_Z4\_U\_Z5\_U\_Z6,0,$G(IBFILE)=355.91)  . ; If both care types, chk for specific  . I IBOK,'Z5 S IBOK=$$COMBOK^IBCEP5C(IBFILE,IBPRV\_U\_5\_U\_Z2\_U\_Z3\_U\_Z4\_U\_Z5\_U\_Z6,1,$G(IBFILE)=355.91)  . ; If specific care type, chk for all  . I IBOK,Z5 S IBOK=$$COMBOK^IBCEP5C(IBFILE,IBPRV\_U\_5\_U\_Z2\_U\_Z3\_U\_Z4\_U\_Z5\_U\_Z6,0,$G(IBFILE)=355.91)  . I 'IBOK K IB3559,IB35591  . I IBOK D  .. S DIR(0)=IBFILE\_",.07"  .. W ! D ^DIR K DIR  .. S IBID=Y  .. I $D(DTOUT)!$D(DUOUT) K IB3559,IB35591 S IBOK=0 Q  .. S IBDR=$S(IBFILE=355.9:$S($G(IBINS):".02////"\_IBINS\_";",1:""),1:"")\_$S($G(IBCUND):".03////"\_$S(IB35591(.03):IB35591(.03),1:"\*N/A\*")\_";",1:"")\_".04////"\_IB3559(.04)\_";.05////"\_IB3559(.05)\_";.06////"\_IBPTYP\_$S(IBID'="":";.07////"\_IBID,1:"")  .. ;  .. I $G(IBIEN) D  ... S DR=IBDR,DA=IBIEN,DIE="^IBA("\_IBFILE\_","  ... D ^DIE  ... I $D(Y) K IB3559,IB35591 S IBOK=0  ; NEWQ ;  I '$D(IB3559),$G(IBIEN) D  Q  . N DIR,DIK,DA,X,Y  . S DA=IBIEN,DIK="^IBA("\_IBFILE\_"," D ^DIK  . S DIR(0)="EA",DIR("A",1)=$S('$G(IBOK):"",1:"PROBLEM ENCOUNTERED FILING THE RECORD - ")\_"RECORD NOT ADDED",DIR("A")="PRESS ENTER to continue " W ! D ^DIR K DIR  ;  ; Save this for Copy ID actions  I $G(IBIEN) D  . I IBFILE=355.91!(IBFILE=355.9&($P($G(^IBA(IBFILE,IBIEN,0)),U)["VA(200,")) D  .. N NEXTONE S NEXTONE=$$NEXTONE^IBCEP5A()  .. S ^TMP("IB\_EDITED\_IDS",$J,NEXTONE)=IBIEN\_U\_"ADD"\_U\_IBFILE  .. S ^TMP("IB\_EDITED\_IDS",$J,NEXTONE,0)=$G(^IBA(IBFILE,IBIEN,0))  Q  ; CHG(IBFILE,IBDA) ; Generic call - edit prov id  ; IBFILE = 355.9 or 355.91 (file being edited)  ; IBDA = ien in file  ;  N DIR,DIE,DA,DR,IBCUCHK,IBOK,IB0,IBOLD,X,Y,Z  F Z=1:1:3 L +^IBA(IBFILE,IBDA):5 Q:$T  W !,"Attempting to lock record"  I '$T D  G CHGQ  . W !,"RECORD LOCKED BY ANOTHER USER - TRY AGAIN LATER"  . D ENTER(.DIR)  . W ! D ^DIR K DIR W !  S (IB0,IBOLD)=$G(^IBA(IBFILE,IBDA,0))  G:IB0="" CHGQ  F Z=.04,.05,.06,.03 S IBOK=$$EDIT(IBFILE,Z,IB0,IBOLD,IBDA,0) S:IBOK="\*ALL\*" IBOK="" Q:$P(IBOK,U,2) S $P(IB0,U,Z\*100)=$P(IBOK,U)  I $P(IBOK,U,2) S DIR(0)="EA",DIR("A")="NO CHANGES MADE, PRESS ENTER TO CONTINUE: " W ! D ^DIR K DIR W ! G CHGQ  S IBOK=$$EDIT(IBFILE,.07,IB0,IBOLD,IBDA,1)  I '$P(IBOK,U,2) S $P(IB0,U,7)=$P(IBOK,U)  I $P(IBOK,U,2)!(IB0=IBOLD) S DIR(0)="EA",DIR("A")="NO CHANGES MADE, PRESS ENTER TO CONTINUE: " W ! D ^DIR K DIR W ! G CHGQ  S IBCUCHK=$$CUCHK^IBCEP5C(IBDA,IB0) G:IBCUCHK CHGQ  S DR=""  F Z=2,4:1:7,3 I $P(IB0,U,Z)'=$P(IBOLD,U,Z) S DR=DR\_$S(DR'="":";",1:"")\_(Z/100)\_"///"\_$S($P(IB0,U,Z)'="@":"/",1:"")\_$P(IB0,U,Z)  I DR'="" D  . I IBFILE=355.91!(IBFILE=355.9&($P(IB0,U)["VA(200,")) D  .. N NEXTONE  .. S NEXTONE=$$NEXTONE^IBCEP5A()  .. S ^TMP("IB\_EDITED\_IDS",$J,NEXTONE)=IBDA\_U\_"MOD"\_U\_IBFILE\_U\_IBDA  .. S ^TMP("IB\_EDITED\_IDS",$J,NEXTONE,"OLD0")=IBOLD  .. S ^TMP("IB\_EDITED\_IDS",$J,NEXTONE,0)=IB0  . S DIE="^IBA("\_IBFILE\_",",DA=IBDA D ^DIE CHGQ L -^IBA(IBFILE,IBDA)  Q  ; DEL(IBFILE,IBDA,IBF) ; Delete prov specific ID's  ; IBFILE = 355.9 or 355.91 for the file  ; IBDA = ien of entry in file IBFILE  ; IBF = 1 if deleting from ins co-related options, ""  ; from prov-related options  D DEL^IBCEP5C(IBFILE,IBDA,$G(IBF))  Q  ; EDIT(IBFILE,IBFLD,IB0,IBOLD,IBIEN,IBCK1) ; Generic edit flds  Q $$EDIT^IBCEP5D($G(IBFILE),$G(IBFLD),$G(IB0),$G(IBOLD),$G(IBIEN),$G(IBCK1))  ; SETDIR(DIR) ; Sets dir for BLUE CROSS only UB-04 form type  S DIR("B")="UB-04",$P(DIR(0),U,3)="K:Y'=1 X",DIR("?")="ONLY UB-04 FORM TYPE IS VALID FOR BLUE CROSS ID"  Q  ; ENTER(DIR) ;  S DIR(0)="EA",DIR("A")="PRESS ENTER TO CONTINUE: "  Q | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCEP5B ;ALB/TMP - EDI UTILITIES for prov ID ;29-SEP-00  ;;2.0;INTEGRATED BILLING;\*\*137,239,232,320,348,349,**592\***\*;21-MAR-94;Build 46  ;;Per VHA Directive 2004-038, this routine should not be modified.  ; NEWID(IBFILE,IBINS,IBPRV,IBPTYP,IBIEN,IBF) ; Generic add prov id  ; at both prov (file 355.9) and ins co levels (355.91)  ; IBFILE = 355.9 or 355.91 - the file being edited  ; IBINS = ien of ins co (36) or \*ALL\* for all ins co  ; IBPRV = vp ien of billing prov  ; IBPTYP = ien of prov type (file 355.97)  ; IBIEN = ien of entry being added (req'd)  ; IBF = 1 if deleting from ins-related options, "" from prov-related  N DIC,DIR,X,Y,Z,DA,DR,DIE,DO,DD,DLAYGO,DTOUT,DUOUT,IBQ,IBCUND,IB3559,IB35591,Q,IBDR,IBID,AFT  S IB35591(.03)="",IBPTYP=$G(IBPTYP)  F Z=.04,.05,.03 D  G:Z="" NEWQ  . I $S(Z'=.03:1,1:$S('$G(IBINS):0,1:$G(IBCUND))) D  Q:Z=""  .. N DA  .. I Z'=.03 S DIR(0)=IBFILE\_","\_Z  .. I Z=.03 D  ... S DIR(0)="PAO^355.95:AEMQ"  ... S DIR("S")="I $O(^IBA(355.96,""AUNIQ"","\_IBINS\_",Y,"\_$G(IB3559(.04))\_","\_$G(IB3559(.05))\_","\_IBPTYP\_",0))!($O(^IBA(355.96,""AUNIQ"","\_IBINS\_",Y,"\_$G(IB3559(.04))\_",0,"\_IBPTYP\_",0)))"  ... S DIR("S")=DIR("S")\_"!($O(^IBA(355.96,""AUNIQ"","\_IBINS\_",Y,0,"\_$G(IB3559(.05))\_","\_IBPTYP\_",0)))!($O(^IBA(355.96,""AUNIQ"","\_IBINS\_",Y,0,0,"\_IBPTYP\_",0)))"  ... S DIR("?",1)="Care unit describes areas of service and is assigned by the payer, if",DIR("?")=" applicable. Use Care Unit Maintenance to add or modify care units."  .. ;  .. I Z=.04,IBPRV["355.93",$$GET1^DIQ(355.93,+IBPRV,.02,"I")=1 D  ... I $$GET1^DIQ(355.97,IBPTYP,.03,"I")="EI" S $P(DIR(0),U,3)="K:Y'=1 X",DIR("?")="Provider ID Qualifier selected only allows institutional (UB type) forms" Q  **... ;JRA IB\*2.0\*592 Modify to accommodate Dental Form J430D (same logic as CMS-1500)**  **... ;**I $$GET1^DIQ(355.97,IBPTYP,.03,"I")="TJ" S $P(DIR(0),U,3)="K:Y'=2 X",DIR("?")="Provider ID Qualifier selected only allows professional (CMS-1500) forms" Q **;JRA IB\*2.0\*592 ';'**  ... I $$GET1^DIQ(355.97,IBPTYP,.03,"I")="TJ" S $P(DIR(0),U,3)="K:Y'=2 X",DIR("?")="Provider ID Qualifier selected only allows professional (CMS-1500 **& J430D**) forms" Q  **;JRA IB\*2.0\*592**  ... N AFT  ... S AFT=$$GET1^DIQ(355.97,IBPTYP,.07,"I") ; get allowable form type for this Provider ID Type  ... I AFT="B" S $P(DIR(0),U,3)="K:"".0.1.2.""'[("".""\_Y\_""."") X",DIR("?")="Provider ID Qualifier selected allows institutional, professional or both" Q  **... ;JRA IB\*2.0\*592 Modify to accommodate Dental Form J430D (same logic as CMS-1500)**  **... ;**I AFT="P" S $P(DIR(0),U,3)="K:Y'=2 X",DIR("?")="Provider ID Qualifier selected only allows professional (CMS-1500) forms" Q **;JRA IB\*2.0\*592 ';'**  ... I AFT="P" S $P(DIR(0),U,3)="K:Y'=2 X",DIR("?")="Provider ID Qualifier selected only allows professional (CMS-1500 **& J430D**) forms" Q  **;JRA IB\*2.0\*592**  ... I AFT="I" S $P(DIR(0),U,3)="K:Y'=1 X",DIR("?")="Provider ID Qualifier selected only allows institutional (UB type) forms" Q  .. ;  .. S DA=0  .. I Z=.04,$P($G(^IBE(355.97,+IBPTYP,0)),U,3)="1A" D SETDIR(.DIR)  .. D ^DIR K DIR  .. I $D(DTOUT)!$D(DUOUT) S Z="" K IB3559,IB35591 Q  .. S IB3559(Z)=$S(Z'=.03:$P(Y,U),1:$S($P(Y,U)>0:$P(Y,U),1:"\*N/A\*"))  . I Z=.05 D  .. S IBCUND=$$CAREUN^IBCEP3(IBINS,IBPTYP,IB3559(.04),IB3559(.05),IB3559(.05)=3)  .. S:'IBCUND!($G(IB3559(.03))=0) IB3559(.03)="\*N/A\*"  .. I '$G(IBINS) S IBINS="\*ALL\*"  . I Z=.03 D CAREUN^IBCEP5C  ;  I $D(IB3559) D  . N Q,Z2,Z3,Z4,Z5,Z6,IBLAST,IBOK,DIR,Y,X  . S IBLAST=0  . D DISP^IBCEP4("Q",IBINS,IBPTYP,IB3559(.04),IB3559(.05),1)  . W !!,"THE FOLLOWING WAS CHOSEN:"  . S Q=0 F  S Q=$O(Q(Q)) Q:'Q  W !,?3,Q(Q)  . I IBCUND W !,?3,"CARE UNIT: "\_$$EXPAND^IBTRE(355.96,.01,IB3559(.03))  . S Z2=IBINS,Z3=IB35591(.03),Z4=IB3559(.04),Z5=IB3559(.05),Z6=IBPTYP  . S IBOK=1  . ; If both forms, chk for specific  . I 'Z4 S IBOK=$$COMBOK^IBCEP5C(IBFILE,IBPRV\_U\_4\_U\_Z2\_U\_Z3\_U\_Z4\_U\_Z5\_U\_Z6,1,$G(IBFILE)=355.91)  . ; If specific form, chk for all  . I IBOK,Z4 S IBOK=$$COMBOK^IBCEP5C(IBFILE,IBPRV\_U\_4\_U\_Z2\_U\_Z3\_U\_Z4\_U\_Z5\_U\_Z6,0,$G(IBFILE)=355.91)  . ; If both care types, chk for specific  . I IBOK,'Z5 S IBOK=$$COMBOK^IBCEP5C(IBFILE,IBPRV\_U\_5\_U\_Z2\_U\_Z3\_U\_Z4\_U\_Z5\_U\_Z6,1,$G(IBFILE)=355.91)  . ; If specific care type, chk for all  . I IBOK,Z5 S IBOK=$$COMBOK^IBCEP5C(IBFILE,IBPRV\_U\_5\_U\_Z2\_U\_Z3\_U\_Z4\_U\_Z5\_U\_Z6,0,$G(IBFILE)=355.91)  . I 'IBOK K IB3559,IB35591  . I IBOK D  .. S DIR(0)=IBFILE\_",.07"  .. W ! D ^DIR K DIR  .. S IBID=Y  .. I $D(DTOUT)!$D(DUOUT) K IB3559,IB35591 S IBOK=0 Q  .. S IBDR=$S(IBFILE=355.9:$S($G(IBINS):".02////"\_IBINS\_";",1:""),1:"")\_$S($G(IBCUND):".03////"\_$S(IB35591(.03):IB35591(.03),1:"\*N/A\*")\_";",1:"")\_".04////"\_IB3559(.04)\_";.05////"\_IB3559(.05)\_";.06////"\_IBPTYP\_$S(IBID'="":";.07////"\_IBID,1:"")  .. ;  .. I $G(IBIEN) D  ... S DR=IBDR,DA=IBIEN,DIE="^IBA("\_IBFILE\_","  ... D ^DIE  ... I $D(Y) K IB3559,IB35591 S IBOK=0  ; NEWQ ;  I '$D(IB3559),$G(IBIEN) D  Q  . N DIR,DIK,DA,X,Y  . S DA=IBIEN,DIK="^IBA("\_IBFILE\_"," D ^DIK  . S DIR(0)="EA",DIR("A",1)=$S('$G(IBOK):"",1:"PROBLEM ENCOUNTERED FILING THE RECORD - ")\_"RECORD NOT ADDED",DIR("A")="PRESS ENTER to continue " W ! D ^DIR K DIR  ;  ; Save this for Copy ID actions  I $G(IBIEN) D  . I IBFILE=355.91!(IBFILE=355.9&($P($G(^IBA(IBFILE,IBIEN,0)),U)["VA(200,")) D  .. N NEXTONE S NEXTONE=$$NEXTONE^IBCEP5A()  .. S ^TMP("IB\_EDITED\_IDS",$J,NEXTONE)=IBIEN\_U\_"ADD"\_U\_IBFILE  .. S ^TMP("IB\_EDITED\_IDS",$J,NEXTONE,0)=$G(^IBA(IBFILE,IBIEN,0))  Q  ; CHG(IBFILE,IBDA) ; Generic call - edit prov id  ; IBFILE = 355.9 or 355.91 (file being edited)  ; IBDA = ien in file  ;  N DIR,DIE,DA,DR,IBCUCHK,IBOK,IB0,IBOLD,X,Y,Z  F Z=1:1:3 L +^IBA(IBFILE,IBDA):5 Q:$T  W !,"Attempting to lock record"  I '$T D  G CHGQ  . W !,"RECORD LOCKED BY ANOTHER USER - TRY AGAIN LATER"  . D ENTER(.DIR)  . W ! D ^DIR K DIR W !  S (IB0,IBOLD)=$G(^IBA(IBFILE,IBDA,0))  G:IB0="" CHGQ  F Z=.04,.05,.06,.03 S IBOK=$$EDIT(IBFILE,Z,IB0,IBOLD,IBDA,0) S:IBOK="\*ALL\*" IBOK="" Q:$P(IBOK,U,2) S $P(IB0,U,Z\*100)=$P(IBOK,U)  I $P(IBOK,U,2) S DIR(0)="EA",DIR("A")="NO CHANGES MADE, PRESS ENTER TO CONTINUE: " W ! D ^DIR K DIR W ! G CHGQ  S IBOK=$$EDIT(IBFILE,.07,IB0,IBOLD,IBDA,1)  I '$P(IBOK,U,2) S $P(IB0,U,7)=$P(IBOK,U)  I $P(IBOK,U,2)!(IB0=IBOLD) S DIR(0)="EA",DIR("A")="NO CHANGES MADE, PRESS ENTER TO CONTINUE: " W ! D ^DIR K DIR W ! G CHGQ  S IBCUCHK=$$CUCHK^IBCEP5C(IBDA,IB0) G:IBCUCHK CHGQ  S DR=""  F Z=2,4:1:7,3 I $P(IB0,U,Z)'=$P(IBOLD,U,Z) S DR=DR\_$S(DR'="":";",1:"")\_(Z/100)\_"///"\_$S($P(IB0,U,Z)'="@":"/",1:"")\_$P(IB0,U,Z)  I DR'="" D  . I IBFILE=355.91!(IBFILE=355.9&($P(IB0,U)["VA(200,")) D  .. N NEXTONE  .. S NEXTONE=$$NEXTONE^IBCEP5A()  .. S ^TMP("IB\_EDITED\_IDS",$J,NEXTONE)=IBDA\_U\_"MOD"\_U\_IBFILE\_U\_IBDA  .. S ^TMP("IB\_EDITED\_IDS",$J,NEXTONE,"OLD0")=IBOLD  .. S ^TMP("IB\_EDITED\_IDS",$J,NEXTONE,0)=IB0  . S DIE="^IBA("\_IBFILE\_",",DA=IBDA D ^DIE CHGQ L -^IBA(IBFILE,IBDA)  Q  ; DEL(IBFILE,IBDA,IBF) ; Delete prov specific ID's  ; IBFILE = 355.9 or 355.91 for the file  ; IBDA = ien of entry in file IBFILE  ; IBF = 1 if deleting from ins co-related options, ""  ; from prov-related options  D DEL^IBCEP5C(IBFILE,IBDA,$G(IBF))  Q  ; EDIT(IBFILE,IBFLD,IB0,IBOLD,IBIEN,IBCK1) ; Generic edit flds  Q $$EDIT^IBCEP5D($G(IBFILE),$G(IBFLD),$G(IB0),$G(IBOLD),$G(IBIEN),$G(IBCK1))  ; SETDIR(DIR) ; Sets dir for BLUE CROSS only UB-04 form type  S DIR("B")="UB-04",$P(DIR(0),U,3)="K:Y'=1 X",DIR("?")="ONLY UB-04 FORM TYPE IS VALID FOR BLUE CROSS ID"  Q  ; ENTER(DIR) ;  S DIR(0)="EA",DIR("A")="PRESS ENTER TO CONTINUE: "  Q | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCEP5C | | | | | | | | |
| **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** | | | | | | | | | |
| IBCEP5C ;ALB/TMP - EDI UTILITIES for provider ID ;02-NOV-00  ;;2.0;INTEGRATED BILLING;\*\*137,239,232,320,348,349\*\*;21-MAR-94;Build 46  ;;Per VHA Directive 2004-038, this routine should not be modified.  ; COMBOK(IBFILE,IBDAT,IBALL,IBF) ; Generic ask if conflict, should id rec still  ; be added?  ; IBFILE = 355.9 or 355.91 for the file being edited  ; IBDAT = var ptr prov ien (355.9) ^ pc to check ^  ; ins co ien or \*ALL\* ^ care unit or \*N/A\* ^  ; form type code ^ care type code ^ prov id type ptr  ; IBALL = flag:  ; 0 = Individual entry selected - check for existing ALL entry  ; 1 = 'ALL' selected - check for existing individual ones  ; IBF = 1 if deleting from ins co-related options, ""  ; from provider-related options  ; Returns 1 if ok to continue, 0 if not  ;  N X,Y,Q,DIR,Z,IBD,IBDD,IBOK,IBSPEC  S IBALL=$G(IBALL),IBOK=1  S IBD=+$P(IBDAT,U,2),IBDD=$S(IBD=4:5,1:4)  F Z=2:1:6 D  . I IBD'=Z,$P(IBDAT,U,Z+1)'="" S Z(Z)=$P(IBDAT,U,Z+1) Q  . I IBD=Z S IBD(Z)=$P(IBDAT,U,Z+1)  K IBSPEC  I IBALL D  ; Check for specific  . N X0,X1  . S X1=0  . F  S X1=$O(^IBA(IBFILE,"AC",$S(IBFILE=355.9:Z(6),1:Z(2)),$S(IBFILE=355.9:Z(2),1:Z(6)),$S(IBFILE=355.9:$P(IBDAT,U),1:Z(3)),X1)) Q:'X1  S X0=$G(^IBA(IBFILE,X1,0)) I $S(IBFILE=355.9:$P(X0,U,3)=Z(3),1:1) D  .. I $P(X0,U,IBD)'=IBD(IBD),"12"[$P(X0,U,IBD),($P(X0,U,IBDD)=Z(IBDD)!($P(X0,U,IBDD)=0)!(Z(IBDD)=0&(IBD(IBD)=0))) S X1($P(X0,U,IBD))=X1 Q  .. I IBD(IBD)=0,Z(IBDD)=0 S X1(0)=X1  . S X0=0 F  S X0=$O(X1(X0)) Q:X0=""  D  .. S IBSPEC=$S($G(IBSPEC)'="":IBSPEC\_" ",1:"")\_$P($S(IBD=4:"UB-04^CMS-1500",1:"INPT^OUTPT"),U,X0)\_" ONLY"  . I $D(X1(0)) S IBSPEC=$S($G(IBSPEC)'="":IBSPEC\_" ",1:"")\_$S(IBD=4:"BOTH UB-04 and CMS-1500 form type AND BOTH INPT and OUTPT care type",1:"BOTH INPT and OUTPT care type AND BOTH UB-04 and CMS-1500 form type")  . ;  I 'IBALL D  . N X0,X1  . S X1=0  . F  S X1=$O(^IBA(IBFILE,"AC",$S(IBFILE=355.9:Z(6),1:Z(2)),$S(IBFILE=355.9:Z(2),1:Z(6)),$S(IBFILE=355.9:$P(IBDAT,U),1:Z(3)),X1)) Q:'X1  D  .. S X0=$G(^IBA(IBFILE,X1,0))  .. I $S(IBFILE=355.9:$P(X0,U,16)=Z(3),1:1),$P(X0,U,IBD)=0,$S($P(X0,U,IBDD)=Z(IBDD):1,1:$P(X0,U,IBDD)=0) S IBSPEC=""  ;  I $D(IBSPEC) D  . N X0,X1,TEXT,IBWHAT  . S IBWHAT=$S(IBFILE=355.9:$S($G(IBF):"INS CO AND PROVIDER",1:"PROVIDER"),1:"INSURANCE CO")  . S X0=$S($D(IBD(4)):"UB-04^CMS-1500",1:"INPT^OUTPT")  . S X1=$S($D(IBD(4)):"FORM TYPE",1:"CARE TYPE")  . S DIR(0)="YA"  . S TEXT(1)="WARNING ... POTENTIAL CONFLICT DETECTED!!"  . S TEXT(2)=" YOUR NEW COMBINATION APPLIES TO "\_$S(IBALL:"BOTH "\_$S(IBD=4:"FORM ",1:"INPT AND OUTPT CARE ")\_"TYPES",1:"ONLY "\_$P(X0,U,IBD(IBD))\_" "\_X1)  . S TEXT(3)=" THIS SAME COMBINATION ALREADY EXISTS FOR THE "\_IBWHAT\_" & "\_$S('IBALL:"ALL "\_X1\_"S",1:"SPECIFIC "\_X1\_"(S):")  . S:IBSPEC'="" TEXT(4)=$J("",4)\_IBSPEC  . S TEXT($S($D(TEXT(4)):5,1:4))=" "  . S DIR("A")="ARE YOU SURE YOU STILL WANT TO ADD THIS RECORD?: "  . S DIR("?",1)=" "  . S DIR("?",2)="This combination appears to be conflicting with one(s) already on file."  . S DIR("?",3)="It has already been defined for the "\_$$LOW^XLFSTR(IBWHAT)\_" for "\_$S(IBALL:"at least 1 specific ",1:"ALL ")\_$S(IBD=4:"form",1:"care")\_" type"\_$S(IBALL:".",1:"s.")  . S DIR("?")="Respond NO to reject this conflicting record or YES to continue on to add it in spite of the apparent conflict.",DIR("B")="NO"  . W !! F Q=1:1 Q:'$D(TEXT(Q)) W TEXT(Q),!  . D ^DIR K DIR W !  . S IBOK=(Y=1)  Q IBOK  ; CAREUN ;Called from NEWID^IBCEP5B to check for existing record combination  N DIR  I IBFILE'=355.9 D  . S IB35591(.03)=IB3559(.03)  . I "0"[IB35591(.03) S IB35591(.03)="\*N/A\*"  . I IB35591(.03)'="\*N/A\*" S IB35591(.03)=$O(^IBA(355.96,"AUNIQ",IBINS,IB3559(.03),IB3559(.04),IB3559(.05),IBPTYP,"")) I 'IB35591(.03) D  .. S IB35591(.03)=$O(^IBA(355.96,"AUNIQ",IBINS,IB3559(.03),IB3559(.04),0,IBPTYP,"")) I 'IB35591(.03) D  ... S IB35591(.03)=$O(^IBA(355.96,"AUNIQ",IBINS,IB3559(.03),0,IB3559(.05),IBPTYP,"")) I 'IB35591(.03) D  .... S IB35591(.03)=$O(^IBA(355.96,"AUNIQ",IBINS,IB3559(.03),0,0,IBPTYP,""))  . I $D(^IBA(355.91,"AUNIQ",IBINS,IB35591(.03),IB3559(.04),IB3559(.05),IBPTYP)) D  Q  .. S DIR(0)="EA",DIR("A",1)="This record already exists - NOT ADDED",DIR("A")="PRESS the ENTER key to continue" W ! D ^DIR K DIR,IB3559,IB35591 W !  I IBFILE=355.9 D  . S IB35591(.03)=IB3559(.03)  . I "0"[IB35591(.03) S IB35591(.03)="\*N/A\*"  . I IB35591(.03)'="\*N/A\*" S IB35591(.03)=$O(^IBA(355.96,"AUNIQ",IBINS,IB3559(.03),IB3559(.04),IB3559(.05),IBPTYP,"")) I 'IB35591(.03) D  .. S IB35591(.03)=$O(^IBA(355.96,"AUNIQ",IBINS,IB3559(.03),IB3559(.04),0,IBPTYP,"")) I 'IB35591(.03) D  ... S IB35591(.03)=$O(^IBA(355.96,"AUNIQ",IBINS,IB3559(.03),0,IB3559(.05),IBPTYP,"")) I 'IB35591(.03) D  .... S IB35591(.03)=$O(^IBA(355.96,"AUNIQ",IBINS,IB3559(.03),0,0,IBPTYP,""))  . I $D(^IBA(355.9,"AUNIQ",IBPRV,IBINS,IB35591(.03),IB3559(.04),IB3559(.05),IBPTYP)) D  Q  .. S DIR(0)="EA",DIR("A",1)="This record already exists - NOT ADDED",DIR("A")="PRESS the ENTER key to continue" W ! D ^DIR K DIR,IB3559,IB35591 W !  Q  ; DEL(IBFILE,IBDA,IBF) ; Delete prov specific ID's  ; IBFILE = 355.9 or 355.91 for the file  ; IBDA = ien of entry in file IBFILE  ; IBF = 1 if deleting from ins co-related options, ""  ; from prov-related options  N IB0,IBLAST,IBX,DIK,DA,DIR,X,Y,Z  F Z=1:1:3 L +^IBA(IBFILE,IBDA):5 Q:$T  I '$T D  G DELQ  . W !,"RECORD IS LOCKED BY ANOTHER USER - TRY AGAIN LATER"  . D ENTER^IBCEP5B(.DIR)  . W ! D ^DIR K DIR W !  S IB0=$G(^IBA(IBFILE,IBDA,0))  S IBX=0  S IBX=IBX+1,DIR("A",IBX)=" PROVIDER: "\_$S(IBFILE=355.9:$$EXPAND^IBTRE(355.9,.01,$P(IB0,U)),1:"\*ALL\*")  D DISP^IBCEP4("DIR(""A"")",$P(IB0,U,$S(IBFILE=355.9:2,1:1)),$P(IB0,U,6),$P(IB0,U,4),$P(IB0,U,5),IBX+1,.IBLAST)  I $P(IB0,U,3)'="" S DIR("A",IBLAST+1)="CARE UNIT: "\_$$EXPAND^IBTRE(355.91,.03,$P(IB0,U,3))  S DIR("A",IBLAST+2)=" PROV ID: "\_$P(IB0,U,7),DIR("A",IBLAST+3)=" "  S DIR("A")="OK TO DELETE THIS "\_$S($G(IBF):"INSURANCE COMPANY ",1:"")\_"PROVIDER ID RECORD?: ",DIR("B")="NO"  S DIR(0)="YA"  W ! D ^DIR K DIR W !  I Y'=1 G DELQ  I IBDA>0 D  . I IBFILE=355.91!(IBFILE=355.9&($P($G(^IBA(IBFILE,IBDA,0)),U)["VA(200,")) D  .. N NEXTONE S NEXTONE=$$NEXTONE^IBCEP5A()  .. S ^TMP("IB\_EDITED\_IDS",$J,NEXTONE)=IBDA\_U\_"DEL"\_U\_IBFILE\_U\_IBDA  .. S ^TMP("IB\_EDITED\_IDS",$J,NEXTONE,0)=$G(^IBA(IBFILE,IBDA,0))  . S DA=IBDA,DIK="^IBA("\_IBFILE\_"," D ^DIK DELQ L -^IBA(IBFILE,IBDA)  Q  ; CUCHK(IBDA,IB0) ;Called from CHG^IBCEP5B to check for existing combination  ; during edit   ; IBDA = the ien of the record being edited  ; IB0 = Proposed changed 0 node of the entry in the file  ; FUNCTION RETURNS 0 if no duplicate found, 1 if record already exists  N Z,IBCUCHK,DIR,X,Y  S IBCUCHK=0  I IBFILE=355.91 S Z=+$O(^IBA(355.91,"AUNIQ",$P(IB0,U,1),$S($P(IB0,U,3)="@":"\*N/A\*",$P(IB0,U,3):$P(IB0,U,3),1:$P(IB0,U,10)),$P(IB0,U,4),$P(IB0,U,5),$P(IB0,U,6),0)) I Z,Z'=IBDA S IBCUCHK=1  I IBFILE=355.9 D  . N X,X1  . S X=$S($P(IB0,U,2):$P(IB0,U,2),1:$P(IB0,U,15)) S:X="" X="\*ALL\*"  . S X1=$S($P(IB0,U,3):$P(IB0,U,3),$P(IB0,U,3)="@":"",1:$P(IB0,U,16)) S:X1="" X1="\*N/A\*"  . S Z=+$O(^IBA(355.9,"AUNIQ",$P(IB0,U,1),X,X1,$P(IB0,U,4),$P(IB0,U,5),$P(IB0,U,6),0)) I Z,Z'=IBDA S IBCUCHK=1  I IBCUCHK D  . S DIR(0)="EA",DIR("A",1)="This combination already exists - RECORD NOT CHANGED",DIR("A")="PRESS the ENTER key to continue" W ! D ^DIR K DIR W !  Q IBCUCHK  ; | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCEP5C ;ALB/TMP - EDI UTILITIES for provider ID ;02-NOV-00  ;;2.0;INTEGRATED BILLING;\*\*137,239,232,320,348,349,**592**\*\*;21-MAR-94;Build 46  ;;Per VHA Directive 2004-038, this routine should not be modified.  ; COMBOK(IBFILE,IBDAT,IBALL,IBF) ; Generic ask if conflict, should id rec still  ; be added?  ; IBFILE = 355.9 or 355.91 for the file being edited  ; IBDAT = var ptr prov ien (355.9) ^ pc to check ^  ; ins co ien or \*ALL\* ^ care unit or \*N/A\* ^  ; form type code ^ care type code ^ prov id type ptr  ; IBALL = flag:  ; 0 = Individual entry selected - check for existing ALL entry  ; 1 = 'ALL' selected - check for existing individual ones  ; IBF = 1 if deleting from ins co-related options, ""  ; from provider-related options  ; Returns 1 if ok to continue, 0 if not  ;  N X,Y,Q,DIR,Z,IBD,IBDD,IBOK,IBSPEC  S IBALL=$G(IBALL),IBOK=1  S IBD=+$P(IBDAT,U,2),IBDD=$S(IBD=4:5,1:4)  F Z=2:1:6 D  . I IBD'=Z,$P(IBDAT,U,Z+1)'="" S Z(Z)=$P(IBDAT,U,Z+1) Q  . I IBD=Z S IBD(Z)=$P(IBDAT,U,Z+1)  K IBSPEC  I IBALL D  ; Check for specific  . N X0,X1  . S X1=0  . F  S X1=$O(^IBA(IBFILE,"AC",$S(IBFILE=355.9:Z(6),1:Z(2)),$S(IBFILE=355.9:Z(2),1:Z(6)),$S(IBFILE=355.9:$P(IBDAT,U),1:Z(3)),X1)) Q:'X1  S X0=$G(^IBA(IBFILE,X1,0)) I $S(IBFILE=355.9:$P(X0,U,3)=Z(3),1:1) D  .. I $P(X0,U,IBD)'=IBD(IBD),"12"[$P(X0,U,IBD),($P(X0,U,IBDD)=Z(IBDD)!($P(X0,U,IBDD)=0)!(Z(IBDD)=0&(IBD(IBD)=0))) S X1($P(X0,U,IBD))=X1 Q  .. I IBD(IBD)=0,Z(IBDD)=0 S X1(0)=X1  . S X0=0 F  S X0=$O(X1(X0)) Q:X0=""  D  **.. ;JRA IB\*2.0\*592 Modify for Dental form J340D (treat the same as the CMS-1500)**  **.. ;**S IBSPEC=$S($G(IBSPEC)'="":IBSPEC\_" ",1:"")\_$P($S(IBD=4:"UB-04^CMS-1500",1:"INPT^OUTPT"),U,X0)\_" ONLY" **;JRA IB\*2.0\*592 ';'**  .. S IBSPEC=$S($G(IBSPEC)'="":IBSPEC\_" ",1:"")\_$P($S(IBD=4:"UB-04^CMS-1500**/J430D**",1:"INPT^OUTPT"),U,X0)\_" ONLY"  **;JRA IB\*2.0\*592**  **. ;I** $D(X1(0)) S IBSPEC=$S($G(IBSPEC)'="":IBSPEC\_" ",1:"")\_$S(IBD=4:"BOTH UB-04 and CMS-1500 form type AND BOTH INPT and OUTPT care type",1:"BOTH INPT and OUTPT care type AND BOTH UB-04 and CMS-1500 form type") **;JRA IB\*2.0\*592 ';'** . I $D(X1(0)) S IBSPEC=$S($G(IBSPEC)'="":IBSPEC\_" ",1:"")\_$S(IBD=4:"**UB-04, CMS-1500 and J430D form types AND BOTH INPT and OUTPT care type",1:"BOTH INPT and OUTPT care type AND UB-04, CMS-1500 and J430D form types**") ;**JRA IB\*2.0\*592**  . ;  I 'IBALL D  . N X0,X1  . S X1=0  . F  S X1=$O(^IBA(IBFILE,"AC",$S(IBFILE=355.9:Z(6),1:Z(2)),$S(IBFILE=355.9:Z(2),1:Z(6)),$S(IBFILE=355.9:$P(IBDAT,U),1:Z(3)),X1)) Q:'X1  D  .. S X0=$G(^IBA(IBFILE,X1,0))  .. I $S(IBFILE=355.9:$P(X0,U,16)=Z(3),1:1),$P(X0,U,IBD)=0,$S($P(X0,U,IBDD)=Z(IBDD):1,1:$P(X0,U,IBDD)=0) S IBSPEC=""  ;  I $D(IBSPEC) D  . N X0,X1,TEXT,IBWHAT  . S IBWHAT=$S(IBFILE=355.9:$S($G(IBF):"INS CO AND PROVIDER",1:"PROVIDER"),1:"INSURANCE CO")  **. ;JRA IB\*2.0\*592 Modify for Dental form J340D (treat the same as the CMS-1500)**  **. ;S** X0=$S($D(IBD(4)):"UB-04^CMS-1500",1:"INPT^OUTPT") **;JRA IB\*2.0\*592 ';'**  . S X0=$S($D(IBD(4)):"UB-04^CMS-1500**/J430D**",1:"INPT^OUTPT") **;JRA IB\*2.0\*592**  . S X1=$S($D(IBD(4)):"FORM TYPE",1:"CARE TYPE")  . S DIR(0)="YA"  . S TEXT(1)="WARNING ... POTENTIAL CONFLICT DETECTED!!"  . S TEXT(2)=" YOUR NEW COMBINATION APPLIES TO "\_$S(IBALL:"BOTH "\_$S(IBD=4:"FORM ",1:"INPT AND OUTPT CARE ")\_"TYPES",1:"ONLY "\_$P(X0,U,IBD(IBD))\_" "\_X1)  . S TEXT(3)=" THIS SAME COMBINATION ALREADY EXISTS FOR THE "\_IBWHAT\_" & "\_$S('IBALL:"ALL "\_X1\_"S",1:"SPECIFIC "\_X1\_"(S):")  . S:IBSPEC'="" TEXT(4)=$J("",4)\_IBSPEC  . S TEXT($S($D(TEXT(4)):5,1:4))=" "  . S DIR("A")="ARE YOU SURE YOU STILL WANT TO ADD THIS RECORD?: "  . S DIR("?",1)=" "  . S DIR("?",2)="This combination appears to be conflicting with one(s) already on file."  . S DIR("?",3)="It has already been defined for the "\_$$LOW^XLFSTR(IBWHAT)\_" for "\_$S(IBALL:"at least 1 specific ",1:"ALL ")\_$S(IBD=4:"form",1:"care")\_" type"\_$S(IBALL:".",1:"s.")  . S DIR("?")="Respond NO to reject this conflicting record or YES to continue on to add it in spite of the apparent conflict.",DIR("B")="NO"  . W !! F Q=1:1 Q:'$D(TEXT(Q)) W TEXT(Q),!  . D ^DIR K DIR W !  . S IBOK=(Y=1)  Q IBOK  ; CAREUN ;Called from NEWID^IBCEP5B to check for existing record combination  N DIR  I IBFILE'=355.9 D  . S IB35591(.03)=IB3559(.03)  . I "0"[IB35591(.03) S IB35591(.03)="\*N/A\*"  . I IB35591(.03)'="\*N/A\*" S IB35591(.03)=$O(^IBA(355.96,"AUNIQ",IBINS,IB3559(.03),IB3559(.04),IB3559(.05),IBPTYP,"")) I 'IB35591(.03) D  .. S IB35591(.03)=$O(^IBA(355.96,"AUNIQ",IBINS,IB3559(.03),IB3559(.04),0,IBPTYP,"")) I 'IB35591(.03) D  ... S IB35591(.03)=$O(^IBA(355.96,"AUNIQ",IBINS,IB3559(.03),0,IB3559(.05),IBPTYP,"")) I 'IB35591(.03) D  .... S IB35591(.03)=$O(^IBA(355.96,"AUNIQ",IBINS,IB3559(.03),0,0,IBPTYP,""))  . I $D(^IBA(355.91,"AUNIQ",IBINS,IB35591(.03),IB3559(.04),IB3559(.05),IBPTYP)) D  Q  .. S DIR(0)="EA",DIR("A",1)="This record already exists - NOT ADDED",DIR("A")="PRESS the ENTER key to continue" W ! D ^DIR K DIR,IB3559,IB35591 W !  I IBFILE=355.9 D  . S IB35591(.03)=IB3559(.03)  . I "0"[IB35591(.03) S IB35591(.03)="\*N/A\*"  . I IB35591(.03)'="\*N/A\*" S IB35591(.03)=$O(^IBA(355.96,"AUNIQ",IBINS,IB3559(.03),IB3559(.04),IB3559(.05),IBPTYP,"")) I 'IB35591(.03) D  .. S IB35591(.03)=$O(^IBA(355.96,"AUNIQ",IBINS,IB3559(.03),IB3559(.04),0,IBPTYP,"")) I 'IB35591(.03) D  ... S IB35591(.03)=$O(^IBA(355.96,"AUNIQ",IBINS,IB3559(.03),0,IB3559(.05),IBPTYP,"")) I 'IB35591(.03) D  .... S IB35591(.03)=$O(^IBA(355.96,"AUNIQ",IBINS,IB3559(.03),0,0,IBPTYP,""))  . I $D(^IBA(355.9,"AUNIQ",IBPRV,IBINS,IB35591(.03),IB3559(.04),IB3559(.05),IBPTYP)) D  Q  .. S DIR(0)="EA",DIR("A",1)="This record already exists - NOT ADDED",DIR("A")="PRESS the ENTER key to continue" W ! D ^DIR K DIR,IB3559,IB35591 W !  Q  ; DEL(IBFILE,IBDA,IBF) ; Delete prov specific ID's  ; IBFILE = 355.9 or 355.91 for the file  ; IBDA = ien of entry in file IBFILE  ; IBF = 1 if deleting from ins co-related options, ""  ; from prov-related options  N IB0,IBLAST,IBX,DIK,DA,DIR,X,Y,Z  F Z=1:1:3 L +^IBA(IBFILE,IBDA):5 Q:$T  I '$T D  G DELQ  . W !,"RECORD IS LOCKED BY ANOTHER USER - TRY AGAIN LATER"  . D ENTER^IBCEP5B(.DIR)  . W ! D ^DIR K DIR W !  S IB0=$G(^IBA(IBFILE,IBDA,0))  S IBX=0  S IBX=IBX+1,DIR("A",IBX)=" PROVIDER: "\_$S(IBFILE=355.9:$$EXPAND^IBTRE(355.9,.01,$P(IB0,U)),1:"\*ALL\*")  D DISP^IBCEP4("DIR(""A"")",$P(IB0,U,$S(IBFILE=355.9:2,1:1)),$P(IB0,U,6),$P(IB0,U,4),$P(IB0,U,5),IBX+1,.IBLAST)  I $P(IB0,U,3)'="" S DIR("A",IBLAST+1)="CARE UNIT: "\_$$EXPAND^IBTRE(355.91,.03,$P(IB0,U,3))  S DIR("A",IBLAST+2)=" PROV ID: "\_$P(IB0,U,7),DIR("A",IBLAST+3)=" "  S DIR("A")="OK TO DELETE THIS "\_$S($G(IBF):"INSURANCE COMPANY ",1:"")\_"PROVIDER ID RECORD?: ",DIR("B")="NO"  S DIR(0)="YA"  W ! D ^DIR K DIR W !  I Y'=1 G DELQ  I IBDA>0 D  . I IBFILE=355.91!(IBFILE=355.9&($P($G(^IBA(IBFILE,IBDA,0)),U)["VA(200,")) D  .. N NEXTONE S NEXTONE=$$NEXTONE^IBCEP5A()  .. S ^TMP("IB\_EDITED\_IDS",$J,NEXTONE)=IBDA\_U\_"DEL"\_U\_IBFILE\_U\_IBDA  .. S ^TMP("IB\_EDITED\_IDS",$J,NEXTONE,0)=$G(^IBA(IBFILE,IBDA,0))  . S DA=IBDA,DIK="^IBA("\_IBFILE\_"," D ^DIK DELQ L -^IBA(IBFILE,IBDA)  Q  ; CUCHK(IBDA,IB0) ;Called from CHG^IBCEP5B to check for existing combination  ; during edit   ; IBDA = the ien of the record being edited  ; IB0 = Proposed changed 0 node of the entry in the file  ; FUNCTION RETURNS 0 if no duplicate found, 1 if record already exists  N Z,IBCUCHK,DIR,X,Y  S IBCUCHK=0  I IBFILE=355.91 S Z=+$O(^IBA(355.91,"AUNIQ",$P(IB0,U,1),$S($P(IB0,U,3)="@":"\*N/A\*",$P(IB0,U,3):$P(IB0,U,3),1:$P(IB0,U,10)),$P(IB0,U,4),$P(IB0,U,5),$P(IB0,U,6),0)) I Z,Z'=IBDA S IBCUCHK=1  I IBFILE=355.9 D  . N X,X1  . S X=$S($P(IB0,U,2):$P(IB0,U,2),1:$P(IB0,U,15)) S:X="" X="\*ALL\*"  . S X1=$S($P(IB0,U,3):$P(IB0,U,3),$P(IB0,U,3)="@":"",1:$P(IB0,U,16)) S:X1="" X1="\*N/A\*"  . S Z=+$O(^IBA(355.9,"AUNIQ",$P(IB0,U,1),X,X1,$P(IB0,U,4),$P(IB0,U,5),$P(IB0,U,6),0)) I Z,Z'=IBDA S IBCUCHK=1  I IBCUCHK D  . S DIR(0)="EA",DIR("A",1)="This combination already exists - RECORD NOT CHANGED",DIR("A")="PRESS the ENTER key to continue" W ! D ^DIR K DIR W !  Q IBCUCHK  ; | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCEP7 | | | | | | | | |
| **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** | | | | | | | | | |
| IBCEP7 ;ALB/TMP - Functions for fac level PROVIDER ID MAINT ;11-07-00  ;;2.0;INTEGRATED BILLING;\*\*137,232,320,348,349\*\*;21-MAR-94;Build 46  ;;Per VHA Directive 2004-038, this routine should not be modified.  ; HDR ; -- hdr code  I '$D(^TMP("IBCE\_PRVFAC\_MAINT",$J)) D INIT  N IBINS,PCF,PCDISP,IBPARAM,IBEFTFL  K VALMHDR  S IBPARAM=$G(^TMP("IBCE\_PRVFAC\_MAINT\_INS",$J))  S IBEFTFL=$P(IBPARAM,U) ; Electronic Form type flag  S IBINS=+$P(IBPARAM,U,2) ; Insurance co  S PCF=$P($G(^DIC(36,+IBINS,3)),U,13),PCDISP=$S(PCF="P":"(Parent)",1:"")  S VALMHDR(1)="Insurance Co: "\_$P($G(^DIC(36,+IBINS,0)),U)\_PCDISP  S VALMHDR(1)=VALMHDR(1)\_$S(IBEFTFL="E":" Billing Provider Secondary IDs",IBEFTFL="A":" Additional Billing Provider Sec. IDs",IBEFTFL="LF":" VA-Lab/Facility Secondary IDs",1:"")  I IBEFTFL="LF" S VALMHDR(2)="VA-Lab/Facility Primary ID: Federal Tax ID"  Q  ; INIT ; Initialize  N IBCT,IBD,Z,Z0,Z00,Z1,IBS,IBX,IBDIV,IBEFTFL,IBINS,IBPARAM,IBLCT,IBCU  K ^TMP("IBCE\_PRVFAC\_MAINT",$J)  S (IBLCT,IBCT)=0  S IBPARAM=$G(^TMP("IBCE\_PRVFAC\_MAINT\_INS",$J))  S IBEFTFL=$P(IBPARAM,U) ; Electronic Form type flag  S IBINS=+$P(IBPARAM,U,2) ; Insurance co  ;  I IBEFTFL="A" D  . K VALM("PROTOCOL")  . S Y=$$FIND1^DIC(101,,,"IBCE PRVFAC ADDIDS MAINT")  . I Y S VALM("PROTOCOL")=+Y\_";ORD(101,"  ;  I IBEFTFL="LF" D  . S VALM("TITLE")="VA-Lab/Facility IDs"  . K VALM("PROTOCOL")  . S Y=$$FIND1^DIC(101,,,"IBCE PRVFAC VALF MAINT")  . I Y S VALM("PROTOCOL")=+Y\_";ORD(101,"  ;  ; Compile the appropriate list of IDs  S Z=0 F  S Z=$O(^IBA(355.92,"B",IBINS,Z)) Q:'Z  D  . S Z0=$G(^IBA(355.92,Z,0))  . Q:'$P(Z0,U,6)!($P(Z0,U,7)="") ; Quit if no provider id or id type  . Q:'($P(Z0,U,8)=IBEFTFL)  . ;Q:$S($P(IBPARAM,U,3)=1:'$P($G(^IBE(355.97,+$P(Z0,U,6),1)),U,9),1:$P($G(^IBE(355.97,+$P(Z0,U,6),1)),U,9))  . S Z1=$G(^IBE(355.97,+$P(Z0,U,6),0))  . S IBS(+$P(Z0,U,5),+$P(Z0,U,3),+$P(Z1,U,2)\_";"\_Z,$P(Z1,U))=+$P(Z0,U,6)\_U\_$P(Z0,U,7)\_U\_Z  ;  S IBD="" F  S IBD=$O(IBS(IBD)) Q:IBD=""  D  . D:IBCT SET1(.IBLCT," ",IBCT+1)  . D SET1(.IBLCT,"Division: "\_$$DIV(IBD),IBCT+1)  . S IBCU="" F  S IBCU=$O(IBS(IBD,IBCU)) Q:IBCU=""  D  .. I IBCU D SET1(.IBLCT," Care Unit: "\_$$EXTERNAL^DILFD(355.92,.03,"",IBCU),IBCT+1)  .. S Z="" F  S Z=$O(IBS(IBD,IBCU,Z),-1) Q:Z=""  D  ... S Z0="" F  S Z0=$O(IBS(IBD,IBCU,Z,Z0)) Q:Z0=""  S IBX=IBS(IBD,IBCU,Z,Z0) D  .... S IBCT=IBCT+1  .... I $P(Z,";",2) D  Q  ..... S Z00=$G(^IBA(355.92,+$P(Z,";",2),0))  ..... S Z1=$E(IBCT\_$J("",3),1,3)\_" "\_$E(Z0\_$J("",25),1,25)\_" "\_$E($S($P(IBX,U,2)'="":$P(IBX,U,2),1:$$IDNUM^IBCEP7A(+IBX))\_$J("",15),1,15)\_" "\_$P("BOTH^UB04^1500^RX",U,$P(Z00,U,4)+1)  ..... D SET1(.IBLCT,Z1,IBCT)  ..... S ^TMP("IBCE\_PRVFAC\_MAINT",$J,"ZIDX",IBCT)=+$P(Z,";",2)  ;  I 'IBLCT D  . D SET1(1," ")  . N TEXT  . I IBEFTFL="E" S TEXT="No Billing Provider Secondary IDs found"  . I IBEFTFL="A" S TEXT="No Billing Provider Additional IDs found"  . I IBEFTFL="LF" S TEXT="No VA Lab/Facility IDs found"  . D SET1(2,TEXT)  . S IBLCT=2  S VALMBG=1,VALMCNT=IBLCT  Q  ; SET1(IBLCT,TEXT,IBCT) ;  S IBLCT=IBLCT+1 D SET^VALM10(IBLCT,TEXT,$G(IBCT))  Q  ; DIV(IBD) ; Returns 'ALL/DEFAULT' or div NAME whose ien=IBD  N MAIN  I IBD Q $$EXTERNAL^DILFD(355.92,.05,"",IBD)  S MAIN=$$MAIN^IBCEP2B()  S MAIN=$$EXTERNAL^DILFD(355.92,.05,"",MAIN)  S MAIN=MAIN\_"/Default for All Divisions"  Q MAIN  ; EDIT1 ;  N IBFUNC,IBINS,IBDA,Z,DIR,X,Y,DTOUT,DUOUT,DP,IBPARAM,IBEFTFL  D FULL^VALM1  S IBPARAM=$G(^TMP("IBCE\_PRVFAC\_MAINT\_INS",$J))  S IBEFTFL=$P(IBPARAM,U) ; Electronic Form type flag  S IBINS=+$P(IBPARAM,U,2) ; Insurance co  S IBFUNC="E"  D SEL  I $G(IBDA) S Z=$$EDITFAC(IBDA,IBFUNC,IBEFTFL) I Z D INIT  ; EDIT1Q S VALMBCK="R"  Q EXPND ;  Q HELP ;  Q EXIT ;  N IBPARAM,IBEFTFL  S IBPARAM=$G(^TMP("IBCE\_PRVFAC\_MAINT\_INS",$J))  S IBEFTFL=$P(IBPARAM,U) ; Electronic Form type flag  I IBEFTFL="A" D COPYPROV^IBCEP5A(0)  ;  S (IBLCT,IBCT)=0  K ^TMP("IBCE\_PRVFAC\_MAINT",$J),^TMP("IBCE\_PRVFAC\_MAINT\_INS",$J)  D CLEAN^VALM10  Q SEL ;  N Z  K IBDA  D FULL^VALM1,EN^VALM2($G(XQORNOD(0)),"OS")  S Z=+$O(VALMY(0)) Q:'Z  ; fac/ins co default  S IBDA=$G(^TMP("IBCE\_PRVFAC\_MAINT",$J,"ZIDX",Z))  Q  ; EDITFAC(IBDA,IBFUNC,IBEFTFL) ; edits ins co facility id (355.92), entry IBDA  N IBRBLD,Z,Z0,DIK,DIE,DP,DA,DR,DIR,X,Y,IBDA0,IBDIV,IBITYP,IBFORM,IBCAREUN,NEXTONE  S IBRBLD=0 S:$G(IBDA) IBDA0=$G(^IBA(355.92,+IBDA,0))  ; "E"diting 355.92 entry  I IBFUNC="E" D  . S Z0=$TR(IBDA0,U)  . Q:'$$FACFLDS^IBCEP7C(IBDA,IBINS,.IBITYP,.IBFORM,.IBDIV,"E",.IBCAREUN,IBEFTFL)  . S DIE="^IBA(355.92,",DA=IBDA  . S DR=".03////"\_$S($G(IBCAREUN)]""&($G(IBCAREUN)'="\*N/A\*"):IBCAREUN,1:"")\_";.04////"\_IBFORM\_$S(IBDIV:";.05////"\_IBDIV,1:"")\_";.06////"\_IBITYP\_";"  . S DR=DR\_".07"\_$S(IBEFTFL="E"!(IBEFTFL="A"):"Billing Provider Secondary ID",1:"VA Lab or Facility Secondary ID")  . I IBEFTFL="A" D  .. S NEXTONE=$$NEXTONE()  .. S ^TMP("IB\_EDITED\_IDS",$J,NEXTONE)=IBDA\_U\_"MOD"\_U\_355.92  .. S ^TMP("IB\_EDITED\_IDS",$J,NEXTONE,"OLD0")=^IBA(355.92,IBDA,0)  . D ^DIE  . I IBEFTFL="A" S ^TMP("IB\_EDITED\_IDS",$J,NEXTONE,0)=^IBA(355.92,IBDA,0)  . I $TR($G(^IBA(355.92,IBDA,0)),U)'=Z0 S IBRBLD=1  ;  ; "D"eleting 355.92 entry  I IBFUNC="D" D  . W !!," Insurance Co: ",$P($G(^DIC(36,+IBDA0,0)),U)  . W !," Division: ",$$DIV($P(IBDA0,U,5))  . W:$P(IBDA0,U,3)]"" !," Care Unit: ",$$EXTERNAL^DILFD(355.92,.03,"",$P(IBDA0,U,3))  . W !," ID Qualifier: ",$$EXTERNAL^DILFD(355.92,.06,"",$P(IBDA0,U,6))  . W !," Form Type: ",$$EXTERNAL^DILFD(355.92,.04,"",$P(IBDA0,U,4))  . W !," ID: ",$P(IBDA0,U,7),!  . S DIR(0)="YA",DIR("A")="ARE YOU SURE YOU WANT TO DELETE THIS ID RECORD?: ",DIR("B")="NO" D ^DIR K DIR  . S DIR("A")="NOTHING DELETED - PRESS RETURN TO CONTINUE: "  . I Y=1 D  .. S DIK="^IBA(355.92,",DA=IBDA  .. D ^DIK  .. I IBEFTFL="A" D  ... N NEXTONE  ... S NEXTONE=$$NEXTONE()  ... S ^TMP("IB\_EDITED\_IDS",$J,NEXTONE)=IBDA\_U\_"DEL"\_U\_355.92  ... S ^TMP("IB\_EDITED\_IDS",$J,NEXTONE,0)=IBDA0  .. S DIR("A")="ID DELETED - PRESS RETURN TO CONTINUE: ",IBRBLD=1  .. S DIR(0)="EA" W ! D ^DIR K DIR  ;  Q IBRBLD  ; FACID(Y) ;  N Z,Z1,Z2  S Z=U\_$P($G(^IBE(355.97,+Y,0)),U,3)\_U,Z1=$$SUB2^IBCEF73(1),Z2=$$SUB2^IBCEF73(2)  I Z1[Z!(Z2[Z) Q 1  Q 0  ; ADD1 ;  N IBFUNC,IBINS,IBDA,Z,DIR,X,Y,DTOUT,DUOUT,DP,IBPARAM,IBEFTFL,IBINS  D FULL^VALM1  ;  S IBPARAM=$G(^TMP("IBCE\_PRVFAC\_MAINT\_INS",$J))  S IBEFTFL=$P(IBPARAM,U) ; Electronic Form type flag  S IBINS=+$P(IBPARAM,U,2) ; Insurance co ;  ;  S Z=$$ADDFAC^IBCEP7A(IBINS,IBEFTFL) I Z D INIT  ; ADD1Q S VALMBCK="R"  Q  ; DEL1 ;  N IBFUNC,IBINS,IBDA,Z,DIR,X,Y,DTOUT,DUOUT,DP,IBPARAM,IBEFTDL,IBINS  D FULL^VALM1  ;   S IBPARAM=$G(^TMP("IBCE\_PRVFAC\_MAINT\_INS",$J))  S IBEFTFL=$P(IBPARAM,U) ; Electronic Form type flag  S IBINS=+$P(IBPARAM,U,2) ; Insurance co  ;  S IBFUNC="D"  D SEL  I $G(IBDA) S Z=$$EDITFAC(IBDA,IBFUNC,IBEFTFL) I Z D INIT  ; DEL1Q S VALMBCK="R"  Q  ;  ; Get the next number so that the edits can be replicated in order for other providers/insurance companies NEXTONE() ;  Q $O(^TMP("IB\_EDITED\_IDS",$J,""),-1)+1 | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCEP7 ;ALB/TMP - Functions for fac level PROVIDER ID MAINT ;11-07-00  ;;2.0;INTEGRATED BILLING;\*\*137,232,320,348,349,**592**\*\*;21-MAR-94;Build 46  ;;Per VHA Directive 2004-038, this routine should not be modified.  ; HDR ; -- hdr code  I '$D(^TMP("IBCE\_PRVFAC\_MAINT",$J)) D INIT  N IBINS,PCF,PCDISP,IBPARAM,IBEFTFL  K VALMHDR  S IBPARAM=$G(^TMP("IBCE\_PRVFAC\_MAINT\_INS",$J))  S IBEFTFL=$P(IBPARAM,U) ; Electronic Form type flag  S IBINS=+$P(IBPARAM,U,2) ; Insurance co  S PCF=$P($G(^DIC(36,+IBINS,3)),U,13),PCDISP=$S(PCF="P":"(Parent)",1:"")  S VALMHDR(1)="Insurance Co: "\_$P($G(^DIC(36,+IBINS,0)),U)\_PCDISP  S VALMHDR(1)=VALMHDR(1)\_$S(IBEFTFL="E":" Billing Provider Secondary IDs",IBEFTFL="A":" Additional Billing Provider Sec. IDs",IBEFTFL="LF":" VA-Lab/Facility Secondary IDs",1:"")  I IBEFTFL="LF" S VALMHDR(2)="VA-Lab/Facility Primary ID: Federal Tax ID"  Q  ; INIT ; Initialize  N IBCT,IBD,Z,Z0,Z00,Z1,IBS,IBX,IBDIV,IBEFTFL,IBINS,IBPARAM,IBLCT,IBCU  K ^TMP("IBCE\_PRVFAC\_MAINT",$J)  S (IBLCT,IBCT)=0  S IBPARAM=$G(^TMP("IBCE\_PRVFAC\_MAINT\_INS",$J))  S IBEFTFL=$P(IBPARAM,U) ; Electronic Form type flag  S IBINS=+$P(IBPARAM,U,2) ; Insurance co  ;  I IBEFTFL="A" D  . K VALM("PROTOCOL")  . S Y=$$FIND1^DIC(101,,,"IBCE PRVFAC ADDIDS MAINT")  . I Y S VALM("PROTOCOL")=+Y\_";ORD(101,"  ;  I IBEFTFL="LF" D  . S VALM("TITLE")="VA-Lab/Facility IDs"  . K VALM("PROTOCOL")  . S Y=$$FIND1^DIC(101,,,"IBCE PRVFAC VALF MAINT")  . I Y S VALM("PROTOCOL")=+Y\_";ORD(101,"  ;  ; Compile the appropriate list of IDs  S Z=0 F  S Z=$O(^IBA(355.92,"B",IBINS,Z)) Q:'Z  D  . S Z0=$G(^IBA(355.92,Z,0))  . Q:'$P(Z0,U,6)!($P(Z0,U,7)="") ; Quit if no provider id or id type  . Q:'($P(Z0,U,8)=IBEFTFL)  . ;Q:$S($P(IBPARAM,U,3)=1:'$P($G(^IBE(355.97,+$P(Z0,U,6),1)),U,9),1:$P($G(^IBE(355.97,+$P(Z0,U,6),1)),U,9))  . S Z1=$G(^IBE(355.97,+$P(Z0,U,6),0))  . S IBS(+$P(Z0,U,5),+$P(Z0,U,3),+$P(Z1,U,2)\_";"\_Z,$P(Z1,U))=+$P(Z0,U,6)\_U\_$P(Z0,U,7)\_U\_Z  ;  S IBD="" F  S IBD=$O(IBS(IBD)) Q:IBD=""  D  . D:IBCT SET1(.IBLCT," ",IBCT+1)  . D SET1(.IBLCT,"Division: "\_$$DIV(IBD),IBCT+1)  . S IBCU="" F  S IBCU=$O(IBS(IBD,IBCU)) Q:IBCU=""  D  .. I IBCU D SET1(.IBLCT," Care Unit: "\_$$EXTERNAL^DILFD(355.92,.03,"",IBCU),IBCT+1)  .. S Z="" F  S Z=$O(IBS(IBD,IBCU,Z),-1) Q:Z=""  D  ... S Z0="" F  S Z0=$O(IBS(IBD,IBCU,Z,Z0)) Q:Z0=""  S IBX=IBS(IBD,IBCU,Z,Z0) D  .... S IBCT=IBCT+1  .... I $P(Z,";",2) D  Q  ..... S Z00=$G(^IBA(355.92,+$P(Z,";",2),0))  **..... ;JWS;IB\*2.0\*592;Dental form #7 J430D: changed BOTH to ALL, add J430D**  ..... S Z1=$E(IBCT\_$J("",3),1,3)\_" "\_$E(Z0\_$J("",25),1,25)\_" "\_$E($S($P(IBX,U,2)'="":$P(IBX,U,2),1:$$IDNUM^IBCEP7A(+IBX))\_$J("",15),1,15)\_" "\_$P("ALL^UB04^1500^RX^**J430D**",U,$P(Z00,U,4)+1)  ..... D SET1(.IBLCT,Z1,IBCT)  ..... S ^TMP("IBCE\_PRVFAC\_MAINT",$J,"ZIDX",IBCT)=+$P(Z,";",2)  ;  I 'IBLCT D  . D SET1(1," ")  . N TEXT  . I IBEFTFL="E" S TEXT="No Billing Provider Secondary IDs found"  . I IBEFTFL="A" S TEXT="No Billing Provider Additional IDs found"  . I IBEFTFL="LF" S TEXT="No VA Lab/Facility IDs found"  . D SET1(2,TEXT)  . S IBLCT=2  S VALMBG=1,VALMCNT=IBLCT  Q  ; SET1(IBLCT,TEXT,IBCT) ;  S IBLCT=IBLCT+1 D SET^VALM10(IBLCT,TEXT,$G(IBCT))  Q  ; DIV(IBD) ; Returns 'ALL/DEFAULT' or div NAME whose ien=IBD  N MAIN  I IBD Q $$EXTERNAL^DILFD(355.92,.05,"",IBD)  S MAIN=$$MAIN^IBCEP2B()  S MAIN=$$EXTERNAL^DILFD(355.92,.05,"",MAIN)  S MAIN=MAIN\_"/Default for All Divisions"  Q MAIN  ; EDIT1 ;  N IBFUNC,IBINS,IBDA,Z,DIR,X,Y,DTOUT,DUOUT,DP,IBPARAM,IBEFTFL  D FULL^VALM1  S IBPARAM=$G(^TMP("IBCE\_PRVFAC\_MAINT\_INS",$J))  S IBEFTFL=$P(IBPARAM,U) ; Electronic Form type flag  S IBINS=+$P(IBPARAM,U,2) ; Insurance co  S IBFUNC="E"  D SEL  I $G(IBDA) S Z=$$EDITFAC(IBDA,IBFUNC,IBEFTFL) I Z D INIT  ; EDIT1Q S VALMBCK="R"  Q EXPND ;  Q HELP ;  Q EXIT ;  N IBPARAM,IBEFTFL  S IBPARAM=$G(^TMP("IBCE\_PRVFAC\_MAINT\_INS",$J))  S IBEFTFL=$P(IBPARAM,U) ; Electronic Form type flag  I IBEFTFL="A" D COPYPROV^IBCEP5A(0)  ;  S (IBLCT,IBCT)=0  K ^TMP("IBCE\_PRVFAC\_MAINT",$J),^TMP("IBCE\_PRVFAC\_MAINT\_INS",$J)  D CLEAN^VALM10  Q SEL ;  N Z  K IBDA  D FULL^VALM1,EN^VALM2($G(XQORNOD(0)),"OS")  S Z=+$O(VALMY(0)) Q:'Z  ; fac/ins co default  S IBDA=$G(^TMP("IBCE\_PRVFAC\_MAINT",$J,"ZIDX",Z))  Q  ; EDITFAC(IBDA,IBFUNC,IBEFTFL) ; edits ins co facility id (355.92), entry IBDA  N IBRBLD,Z,Z0,DIK,DIE,DP,DA,DR,DIR,X,Y,IBDA0,IBDIV,IBITYP,IBFORM,IBCAREUN,NEXTONE  S IBRBLD=0 S:$G(IBDA) IBDA0=$G(^IBA(355.92,+IBDA,0))  ; "E"diting 355.92 entry  I IBFUNC="E" D  . S Z0=$TR(IBDA0,U)  . Q:'$$FACFLDS^IBCEP7C(IBDA,IBINS,.IBITYP,.IBFORM,.IBDIV,"E",.IBCAREUN,IBEFTFL)  . S DIE="^IBA(355.92,",DA=IBDA  . S DR=".03////"\_$S($G(IBCAREUN)]""&($G(IBCAREUN)'="\*N/A\*"):IBCAREUN,1:"")\_";.04////"\_IBFORM\_$S(IBDIV:";.05////"\_IBDIV,1:"")\_";.06////"\_IBITYP\_";"  . S DR=DR\_".07"\_$S(IBEFTFL="E"!(IBEFTFL="A"):"Billing Provider Secondary ID",1:"VA Lab or Facility Secondary ID")  . I IBEFTFL="A" D  .. S NEXTONE=$$NEXTONE()  .. S ^TMP("IB\_EDITED\_IDS",$J,NEXTONE)=IBDA\_U\_"MOD"\_U\_355.92  .. S ^TMP("IB\_EDITED\_IDS",$J,NEXTONE,"OLD0")=^IBA(355.92,IBDA,0)  . D ^DIE  . I IBEFTFL="A" S ^TMP("IB\_EDITED\_IDS",$J,NEXTONE,0)=^IBA(355.92,IBDA,0)  . I $TR($G(^IBA(355.92,IBDA,0)),U)'=Z0 S IBRBLD=1  ;  ; "D"eleting 355.92 entry  I IBFUNC="D" D  . W !!," Insurance Co: ",$P($G(^DIC(36,+IBDA0,0)),U)  . W !," Division: ",$$DIV($P(IBDA0,U,5))  . W:$P(IBDA0,U,3)]"" !," Care Unit: ",$$EXTERNAL^DILFD(355.92,.03,"",$P(IBDA0,U,3))  . W !," ID Qualifier: ",$$EXTERNAL^DILFD(355.92,.06,"",$P(IBDA0,U,6))  . W !," Form Type: ",$$EXTERNAL^DILFD(355.92,.04,"",$P(IBDA0,U,4))  . W !," ID: ",$P(IBDA0,U,7),!  . S DIR(0)="YA",DIR("A")="ARE YOU SURE YOU WANT TO DELETE THIS ID RECORD?: ",DIR("B")="NO" D ^DIR K DIR  . S DIR("A")="NOTHING DELETED - PRESS RETURN TO CONTINUE: "  . I Y=1 D  .. S DIK="^IBA(355.92,",DA=IBDA  .. D ^DIK  .. I IBEFTFL="A" D  ... N NEXTONE  ... S NEXTONE=$$NEXTONE()  ... S ^TMP("IB\_EDITED\_IDS",$J,NEXTONE)=IBDA\_U\_"DEL"\_U\_355.92  ... S ^TMP("IB\_EDITED\_IDS",$J,NEXTONE,0)=IBDA0  .. S DIR("A")="ID DELETED - PRESS RETURN TO CONTINUE: ",IBRBLD=1  .. S DIR(0)="EA" W ! D ^DIR K DIR  ;  Q IBRBLD  ; FACID(Y) ;  N Z,Z1,Z2  S Z=U\_$P($G(^IBE(355.97,+Y,0)),U,3)\_U,Z1=$$SUB2^IBCEF73(1),Z2=$$SUB2^IBCEF73(2)  I Z1[Z!(Z2[Z) Q 1  Q 0  ; ADD1 ;  N IBFUNC,IBINS,IBDA,Z,DIR,X,Y,DTOUT,DUOUT,DP,IBPARAM,IBEFTFL,IBINS  D FULL^VALM1  ;  S IBPARAM=$G(^TMP("IBCE\_PRVFAC\_MAINT\_INS",$J))  S IBEFTFL=$P(IBPARAM,U) ; Electronic Form type flag  S IBINS=+$P(IBPARAM,U,2) ; Insurance co ;  ;  S Z=$$ADDFAC^IBCEP7A(IBINS,IBEFTFL) I Z D INIT  ; ADD1Q S VALMBCK="R"  Q  ; DEL1 ;  N IBFUNC,IBINS,IBDA,Z,DIR,X,Y,DTOUT,DUOUT,DP,IBPARAM,IBEFTDL,IBINS  D FULL^VALM1  ;   S IBPARAM=$G(^TMP("IBCE\_PRVFAC\_MAINT\_INS",$J))  S IBEFTFL=$P(IBPARAM,U) ; Electronic Form type flag  S IBINS=+$P(IBPARAM,U,2) ; Insurance co  ;  S IBFUNC="D"  D SEL  I $G(IBDA) S Z=$$EDITFAC(IBDA,IBFUNC,IBEFTFL) I Z D INIT  ; DEL1Q S VALMBCK="R"  Q  ;  ; Get the next number so that the edits can be replicated in order for other providers/insurance companies NEXTONE() ;  Q $O(^TMP("IB\_EDITED\_IDS",$J,""),-1)+1 | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCEP7B | | | | | | | | |
| **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** | | | | | | | | | |
| IBCEP7B ;ALB/TMP - Functions for PROVIDER ID ;1-16-05  ;;2.0;INTEGRATED BILLING;\*\*320,348,349\*\*;16-JAN-2005;Build 46  ;;Per VHA Directive 2004-038, this routine should not be modified.  Q  ; GETID(CLAIM,COB) ;  N DIR,X,Y,DTOUT,DUOUT,WHICH,ID,IBMAIN,IBDIV,DIC,IBINS,DA,DIC,Z,Z0,IBCU,OK,IBCU  ;  S ID=""  S IBINS=$P($G(^DGCR(399,CLAIM,"I"\_COB)),U)  I IBINS="" Q ID  ;  ; Make sure they have careunits IDS defined for this insurance company before we bother asking  S OK=0  S Z=0 F  S Z=$O(^IBA(355.92,"B",IBINS,Z)) Q:'Z  D  Q:OK  . S Z0=$G(^IBA(355.92,Z,0))  . Q:$P(Z0,U,8)'="E"  . Q:$P(Z0,U,3)=""  . S OK=1  I 'OK Q ID  ;  S WHICH=$S(COB=1:"Primary",COB=2:"Secondary",1:"Tertiary")  S DIR("A")="Define "\_WHICH\_" Payer ID by Care Unit? "  S DIR("B")="No"  S DIR(0)="YA"  S DIR("?",1)="Enter No to select "\_WHICH\_" Provider # by Division."  S DIR("?")="Enter Yes to select "\_WHICH\_" Provider # for a specific Care Unit."  D ^DIR  I Y'=1 Q ID  ;  ; Get the Division  S IBMAIN=$$MAIN^IBCEP2B()  S IBDIV=$$EXTERNAL^DILFD(399,.22,"",$P($G(^DGCR(399,CLAIM,0)),U,22))  S DIR("A")="Division: ",DIR(0)="355.92,.05AOr"  ; Default Division  S DIR("B")=$S(IBDIV]"":IBDIV,1:IBMAIN)  D ^DIR K DIR  S IBDIV=+$S(Y>0:+Y,1:0)  I Y<0 Q ID  ;  ; Get the Care Unit  S DIC("A")="Care Unit: "  S DIC("W")="W "" "",$P(^(0),U,2)"  S DIC=355.95,DIC("S")="I $P(^(0),U,3)=+$G(IBINS),$P(^(0),U,4)=+$G(IBDIV)",DIC(0)="AEMQ"  D ^DIC  I Y<0 Q ID  S IBCU=+Y  ;  ; Compile the appropriate list of IDs  S Z=0 F  S Z=$O(^IBA(355.92,"B",IBINS,Z)) Q:'Z  D  Q:ID]""  . S Z0=$G(^IBA(355.92,Z,0))  . Q:$P(Z0,U,8)'="E"  . Q:$P(Z0,U,3)'=IBCU  . S ID=$P(Z0,U,7)\_U\_$P(Z0,U,6)  Q ID  ;  ; See if the insurance company flag is set to send the ATT/REND ID as the Billing Provider ATTREND(CLAIM,COB) ;  N ID,IBINS  S ID=""  S IBINS=$P($G(^DGCR(399,CLAIM,"I"\_COB)),U)  I IBINS="" Q 0  ;  I $$FT^IBCEF(CLAIM)=2,$$GET1^DIQ(36,IBINS,4.06,"I") Q 1 ; 1500  I $$FT^IBCEF(CLAIM)=3,$$GET1^DIQ(36,IBINS,4.08,"I") Q 1 ; ub  Q 0  ;  ; Get a list of the plan types that supress Billing Provider Secondary IDs for this Insurance Co  ; and see if the current plan type is one of them. SUPPPT(CLAIM,COB) ;  N IBINS,SUPPFL  S SUPPFL=0  S IBINS=$P($G(^DGCR(399,CLAIM,"I"\_COB)),U)  I IBINS="" Q SUPPFL  ;  I $D(^DIC(36,IBINS,13)) D  . N PLAN,PLANTYPE  . S PLAN=$P($G(^DGCR(399,CLAIM,"I"\_COB)),U,18) Q:'PLAN  . S PLANTYPE=$P($G(^IBA(355.3,PLAN,0)),U,15) Q:PLANTYPE=""  . Q:'$D(^DIC(36,IBINS,13,"B",PLANTYPE))  . S SUPPFL=1  Q SUPPFL | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCEP7B ;ALB/TMP - Functions for PROVIDER ID ;1-16-05  ;;2.0;INTEGRATED BILLING;\*\*320,348,349,**592**\*\*;16-JAN-2005;Build 46  ;;Per VHA Directive 2004-038, this routine should not be modified.  Q  ; GETID(CLAIM,COB) ;  N DIR,X,Y,DTOUT,DUOUT,WHICH,ID,IBMAIN,IBDIV,DIC,IBINS,DA,DIC,Z,Z0,IBCU,OK,IBCU  ;  S ID=""  S IBINS=$P($G(^DGCR(399,CLAIM,"I"\_COB)),U)  I IBINS="" Q ID  ;  ; Make sure they have careunits IDS defined for this insurance company before we bother asking  S OK=0  S Z=0 F  S Z=$O(^IBA(355.92,"B",IBINS,Z)) Q:'Z  D  Q:OK  . S Z0=$G(^IBA(355.92,Z,0))  . Q:$P(Z0,U,8)'="E"  . Q:$P(Z0,U,3)=""  . S OK=1  I 'OK Q ID  ;  S WHICH=$S(COB=1:"Primary",COB=2:"Secondary",1:"Tertiary")  S DIR("A")="Define "\_WHICH\_" Payer ID by Care Unit? "  S DIR("B")="No"  S DIR(0)="YA"  S DIR("?",1)="Enter No to select "\_WHICH\_" Provider # by Division."  S DIR("?")="Enter Yes to select "\_WHICH\_" Provider # for a specific Care Unit."  D ^DIR  I Y'=1 Q ID  ;  ; Get the Division  S IBMAIN=$$MAIN^IBCEP2B()  S IBDIV=$$EXTERNAL^DILFD(399,.22,"",$P($G(^DGCR(399,CLAIM,0)),U,22))  S DIR("A")="Division: ",DIR(0)="355.92,.05AOr"  ; Default Division  S DIR("B")=$S(IBDIV]"":IBDIV,1:IBMAIN)  D ^DIR K DIR  S IBDIV=+$S(Y>0:+Y,1:0)  I Y<0 Q ID  ;  ; Get the Care Unit  S DIC("A")="Care Unit: "  S DIC("W")="W "" "",$P(^(0),U,2)"  S DIC=355.95,DIC("S")="I $P(^(0),U,3)=+$G(IBINS),$P(^(0),U,4)=+$G(IBDIV)",DIC(0)="AEMQ"  D ^DIC  I Y<0 Q ID  S IBCU=+Y  ;  ; Compile the appropriate list of IDs  S Z=0 F  S Z=$O(^IBA(355.92,"B",IBINS,Z)) Q:'Z  D  Q:ID]""  . S Z0=$G(^IBA(355.92,Z,0))  . Q:$P(Z0,U,8)'="E"  . Q:$P(Z0,U,3)'=IBCU  . S ID=$P(Z0,U,7)\_U\_$P(Z0,U,6)  Q ID  ;  ; See if the insurance company flag is set to send the ATT/REND ID as the Billing Provider ATTREND(CLAIM,COB) ;  N ID,IBINS  S ID=""  S IBINS=$P($G(^DGCR(399,CLAIM,"I"\_COB)),U)  I IBINS="" Q 0  I $$FT^IBCEF(CLAIM)=2,$$GET1^DIQ(36,IBINS,4.06,"I") Q 1 ; 1500 **;JWS;IB\*2.0\*592;Dental form #7 J430D**  **I $$FT^IBCEF(CLAIM)=7,$$GET1^DIQ(36,IBINS,4.16,"I") Q 1 ;J430D**  I $$FT^IBCEF(CLAIM)=3,$$GET1^DIQ(36,IBINS,4.08,"I") Q 1 ; ub  Q 0  ;  ; Get a list of the plan types that supress Billing Provider Secondary IDs for this Insurance Co  ; and see if the current plan type is one of them. SUPPPT(CLAIM,COB) ;  N IBINS,SUPPFL  S SUPPFL=0  S IBINS=$P($G(^DGCR(399,CLAIM,"I"\_COB)),U)  I IBINS="" Q SUPPFL  ;  I $D(^DIC(36,IBINS,13)) D  . N PLAN,PLANTYPE  . S PLAN=$P($G(^DGCR(399,CLAIM,"I"\_COB)),U,18) Q:'PLAN  . S PLANTYPE=$P($G(^IBA(355.3,PLAN,0)),U,15) Q:PLANTYPE=""  . Q:'$D(^DIC(36,IBINS,13,"B",PLANTYPE))  . S SUPPFL=1  Q SUPPFL | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCEP7C | | | | | | | | |
| **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** | | | | | | | | | |
| IBCEP7C ;ALB/TMP - Functions for fac level PROVIDER ID MAINT ;11-07-00  ;;2.0;INTEGRATED BILLING;\*\*137,232,320,348,349\*\*;21-MAR-94;Build 46  ;;Per VHA Directive 2004-038, this routine should not be modified.  ;  G AWAY AWAY Q  ;  ; IBDA - IEN for file 355.92  ; IBFUNC = "A"dd or "E"dit FACFLDS(IBDA,IBINS,IBITYP,IBFORM,IBDIV,IBFUNC,IBCAREUN,IBEFTFL) ; Chk for dups on fac id fld combos  ;  N IB,IBOK,DIC,DIR,X,Y,DTOUT,DUOUT,Z,Z0,DIE,DA,IBMAIN,IBQUIT,IBPARAM,IBCUF,IBDA0,IBCNTADD,I,IBLIMIT  ;  S IBOK=0,IBDA0=""  I $G(IBDA) S IBDA0=$G(^IBA(355.92,IBDA,0))  S IBPARAM=$G(^TMP("IBCE\_PRVFAC\_MAINT\_INS",$J))  S IBCUF=$S($P(IBDA0,U,3)]"":1,1:0) ; Care Unit Flag  ;  I IBEFTFL="E",IBFUNC="A" D  G:$D(DTOUT)!$D(DUOUT) FLDSQ  . K DIR  . S DIR("A")="Define Billing Provider Secondary IDs by Care Units? "  . S DIR("B")="No"  . S DIR(0)="YAO"  . S DIR("?",1)="Enter No to define a Billing Provider Secondary ID for the Division."  . S DIR("?",2)="Enter Yes to define a Billing Provider Secondary ID for a specific Care Unit."  . S DIR("?",3)="If no Care Unit is entered on Billing Screen 3, the Billing Provider"  . S DIR("?")="Secondary ID defined for the Division will be transmitted in the claim."  . D ^DIR  . S IBCUF=$G(Y) ; Care Unit Flag  ;  ; Get the Division  S IBMAIN=$$MAIN^IBCEP2B()  S IBDIV=0  I IBEFTFL="E"!(IBEFTFL="LF") D  G:$D(DTOUT)!$D(DUOUT) FLDSQ  . K DIR  . S (IBQUIT,IBOK)=0,DA=$G(IBDA)  . S DIR("A")="Division: ",DIR(0)="355.92,.05AOr"  . ; Default Division - Main if adding or Existing if editing  . I IBFUNC="E" S DIR("B")=$P($$DIV^IBCEP7($P(IBDA0,U,5)),"/")  . I IBFUNC="A" S DIR("B")=$P($$EXTERNAL^DILFD(355.92,.05,"",IBMAIN),"/")  . D ^DIR K DIR  . Q:$D(DTOUT)!$D(DUOUT)  . S IBDIV=+$S(Y>0:+Y,1:0)  ;  ; See if there are any Care Units  S IBCAREUN="\*N/A\*"  I IBEFTFL="E",IBCUF D  . N TAR  . D LIST^DIC(355.95,,.01,,,,,,"I $P(^(0),U,3)=+$G(IBINS),$P(^(0),U,4)=+$G(IBDIV)",,"TAR")  . Q:+$G(TAR("DILIST",0))  . S IBCUF=0  . W !!,"There are no Care Units defined for this Division.",!  ;  ; Get the Care Unit  I IBEFTFL="E",IBCUF D  I Y<1 G FLDSQ  . K DIC  . S DIC("A")="Care Unit: "  . I IBFUNC="E" D  ; default only if editing  .. Q:IBDIV'=$P(IBDA0,U,5) ; don't default if division has changed  .. S DIC("B")=$$EXTERNAL^DILFD(355.92,.03,"",$P(IBDA0,U,3))  . S DIC=355.95,DIC("S")="I $P(^(0),U,3)=+$G(IBINS),$P(^(0),U,4)=+$G(IBDIV)",DIC(0)="AEMQ"  . D ^DIC  . I Y>0 S IBCAREUN=+Y  ;  ; Think this is done for sorting purposes. Makes the main division first  I IBDIV=IBMAIN S IBDIV=0  ;  ; Get the Provider ID Type  K DIR  S IBQUIT=0  I $P(IBPARAM,U,3)'=1 D  . S DIR("?")="Can NOT be State LIC # or Billing Facility Primary"  . S DIR("A")="ID Qualifier: "  . S DIR(0)="355.92,.06A^^K:'$$FACID^IBCEP7(+Y)!$P($G(^IBE(355.97,+Y,1)),U,9)!($P($G(^(0)),U,3)=""0B"") X"  . W ! D ^DIR K DIR  . I $D(DTOUT)!$D(DUOUT) S IBQUIT=1  E  D  G:$D(DTOUT)!$D(DUOUT) FLDSQ  . S DIR("A")="ID Qualifier: "    ;,DIR(0)="355.92,.06Ar"  . S DIR(0)="PAr^355.97:AEMQ"  . S DIR("?")="Enter a Qualifier to indentify the type of ID number you are entering."  . ; Default Type of ID - Electronic Plan Type if adding or Existing if editing  . N PITIEN S PITIEN=$S(IBFUNC="A"&(IBEFTFL="E"):$$BF^IBCU(),IBFUNC="E":$P(IBDA0,U,6),1:"")  . I PITIEN]"" S DIR("B")=$P($G(^IBE(355.97,PITIEN,0)),U)  . I IBEFTFL="E" D  .. S DIR("?",1)=" The current default ID Qualifier is based upon the Electronic Plan Type."  .. S DIR("?",2)=" You may change the ID Qualifier and the change will apply to all Plan"  .. S DIR("?")=" Types."  .. S DIR("S")="I ($P($G(^(0)),U,3)=$P($G(^IBE(355.97,PITIEN,0)),U,3))!$$BPS^IBCEPU(Y)"  . I IBEFTFL="A" S DIR("S")="I $$BPS^IBCEPU(Y)"  . I IBEFTFL="LF" S DIR("S")="I $$LFINS^IBCEPU(Y)"  . D ^DIR K DIR  G:IBQUIT FLDSQ  S IBITYP=$P(Y,U)  ;  ; Get Form Type  K DIR  S DIR("A")="Form Type: "  S DIR(0)=$S(IBEFTFL="LF":"SA^0:BOTH;1:UB-04;2:CMS-1500",1:"SA^1:UB-04;2:CMS-1500")  ;  I $G(IBDA) S DIR("B")=$S(+$P($G(^IBA(355.92,IBDA,0)),U,4)=0:"BOTH",1:$P("UB-04^CMS-1500",U,+$P($G(^IBA(355.92,IBDA,0)),U,4)))  ;  D ^DIR K DIR  G:$D(DTOUT)!$D(DUOUT) FLDSQ  S IBFORM=$P(Y,U)  ;  ; Set up array of exisiting IDs by form type, divison, and care units to avoid duplications  S Z=0 F  S Z=$O(^IBA(355.92,"B",IBINS,Z)) Q:'Z  D  . S Z0=$G(^IBA(355.92,Z,0))  . I '(IBFUNC="E"&(Z=IBDA)) D  .. I IBEFTFL="LF",$P(Z0,U,8)'="LF" Q   ; If lab/facility ID, it only needs to be unique among lab/facility IDs  .. I IBEFTFL'="LF",$P(Z0,U,8)="LF" Q   ; If not lab/facility ID, it must be unigue for the others (secondary and additional)  .. I IBEFTFL="A",$P(Z0,U,8)="E" Q  .. I $P(Z0,U,8)="E",IBEFTFL'="A" S IB("\*N/A\*",$P(Z0,U,4),+$P(Z0,U,5),$S($P(Z0,U,3)]"":$P(Z0,U,3),1:"\*N/A\*"))=Z  .. S IB($P(Z0,U,6),$P(Z0,U,4),+$P(Z0,U,5),$S($P(Z0,U,3)]"":$P(Z0,U,3),1:"\*N/A\*"))=Z  . ;  . ; count them  . I IBFUNC="A",$P(Z0,U,8)=IBEFTFL,IBDIV=$P(Z0,U,5)!(IBDIV=0&($P(Z0,U,5)="")) D  .. I ".1.2."[("."\_$P(Z0,U,4)\_".") S IBCNTADD($P(Z0,U,4))=$G(IBCNTADD($P(Z0,U,4)))+1 Q  .. N I  .. F I=1,2 S IBCNTADD(I)=$G(IBCNTADD(I))+1  ; Check for duplications  S IBOK=1  ; Don't check if nothing is being changed. The ID itself can be changed after return to calling program.  I IBFUNC="E" S Z0=$G(^IBA(355.92,IBDA,0)) I $P(Z0,U,3)=IBCAREUN!($P(Z0,U,3)=""&(IBCAREUN="\*N/A\*")),IBFORM=$P(Z0,U,4),IBDIV=$P(Z0,U,5),IBITYP=$P(Z0,U,6) G FLDSQ  I $G(IB($S(IBEFTFL="E":"\*N/A\*",1:IBITYP),IBFORM,IBDIV,IBCAREUN)) D  . N Z,ZPC8 S Z=$G(IB($S(IBEFTFL="E":"\*N/A\*",1:IBITYP),IBFORM,IBDIV,IBCAREUN))  . S ZPC8=""  . I +Z S ZPC8=$P($G(^IBA(355.92,Z,0)),U,8)  . S IBOK="0^DUPLICATE"\_U\_ZPC8  I IBOK,IBFORM=0,$S($D(IB($S(IBEFTFL="E":"\*N/A\*",1:IBITYP),1,IBDIV,IBCAREUN))!$D(IB($S(IBEFTFL="E":"\*N/A\*",1:IBITYP),2,IBDIV,IBCAREUN)):1,1:0) S IBOK="0^FORM^SPECIFIC"  I IBOK,IBFORM'=0,IBFORM'=3,$S($D(IB($S(IBEFTFL="E":"\*N/A\*",1:IBITYP),0,IBDIV,IBCAREUN)):1,1:0) S IBOK="0^FORM^BOTH"  ;  S IBLIMIT=$S(IBEFTFL="A":6,IBEFTFL="LF":5,1:"")  I IBOK,IBFUNC="A",IBEFTFL'="E" D  . I ".1.2."[("."\_IBFORM\_".") D  Q  .. I $G(IBCNTADD(IBFORM))>(IBLIMIT-1) S IBOK="0^LIMIT"  . N I  . I IBFORM=0 F I=1,2 I $G(IBCNTADD(I))>IBLIMIT S IBOK="0^LIMIT" Q  ;  I 'IBOK D  . I $P(IBOK,U,2)="DUPLICATE" D  Q  .. S DIR("A",1)="This ID combination is already defined",DIR("A",2)=""  .. ; under "\_$S($P(IBOK,U,3)="A":" Additonal IDs",$P(IBOK,U,3)="E":"Billing Provider Secondary ID",1:"VA Lab/Facility IDs")\_$S(IBFUNC="A":" - try editing it instead",1:""),DIR("A",2)=" "  . ;  . I $P(IBOK,U,2)="BOTH" D  Q  .. S DIR("A",1)="An ID combination for both form types already exists. Delete this one",DIR("A",2)="before defining and form specific IDs"\_$S(IBDIV:" for this division"),DIR("A",4)=" "  . ;  . I $P(IBOK,U,2)="FORM" D  Q  .. I $P(IBOK,U,3)="BOTH" S DIR("A",1)="This ID already exists for both form types - Delete it to enter this ID for",DIR("A",2)=" a specific form type",DIR("A",3)=" " Q  .. S DIR("A",1)="This ID already exists for a specific form type - Delete specific form type",DIR("A",2)=" ID(s) before entering one for both form types",DIR("A",3)=" "  . ;   . I $P(IBOK,U,2)="LIMIT" D  Q  .. S DIR("A",1)="Limit is "\_IBLIMIT\_" IDs for each form type",DIR("A",2)=" "  .. I IBEFTFL="A" D  ... S DIR("A",1)="A maximum of 6 Additional Billing Provider Sec IDs can be entered for each Form"  ... S DIR("A",2)="Type. Before you can add another ID, you must delete an existing ID."  ... S DIR("A",3)=" "  ;  I 'IBOK S DIR(0)="EA",DIR("A")="PRESS RETURN TO CONTINUE: " W ! D ^DIR K DIR  ; FLDSQ Q +IBOK | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCEP7C ;ALB/TMP - Functions for fac level PROVIDER ID MAINT ;11-07-00  ;;2.0;INTEGRATED BILLING;\*\*137,232,320,348,349,**592\*\*;**21-MAR-94;Build 46  ;;Per VHA Directive 2004-038, this routine should not be modified.  ;  G AWAY AWAY Q  ;  ; IBDA - IEN for file 355.92  ; IBFUNC = "A"dd or "E"dit FACFLDS(IBDA,IBINS,IBITYP,IBFORM,IBDIV,IBFUNC,IBCAREUN,IBEFTFL) ; Chk for dups on fac id fld combos  ;  N IB,IBOK,DIC,DIR,X,Y,DTOUT,DUOUT,Z,Z0,DIE,DA,IBMAIN,IBQUIT,IBPARAM,IBCUF,IBDA0,IBCNTADD,I,IBLIMIT  ;  S IBOK=0,IBDA0=""  I $G(IBDA) S IBDA0=$G(^IBA(355.92,IBDA,0))  S IBPARAM=$G(^TMP("IBCE\_PRVFAC\_MAINT\_INS",$J))  S IBCUF=$S($P(IBDA0,U,3)]"":1,1:0) ; Care Unit Flag  ;  I IBEFTFL="E",IBFUNC="A" D  G:$D(DTOUT)!$D(DUOUT) FLDSQ  . K DIR  . S DIR("A")="Define Billing Provider Secondary IDs by Care Units? "  . S DIR("B")="No"  . S DIR(0)="YAO"  . S DIR("?",1)="Enter No to define a Billing Provider Secondary ID for the Division."  . S DIR("?",2)="Enter Yes to define a Billing Provider Secondary ID for a specific Care Unit."  . S DIR("?",3)="If no Care Unit is entered on Billing Screen 3, the Billing Provider"  . S DIR("?")="Secondary ID defined for the Division will be transmitted in the claim."  . D ^DIR  . S IBCUF=$G(Y) ; Care Unit Flag  ;  ; Get the Division  S IBMAIN=$$MAIN^IBCEP2B()  S IBDIV=0  I IBEFTFL="E"!(IBEFTFL="LF") D  G:$D(DTOUT)!$D(DUOUT) FLDSQ  . K DIR  . S (IBQUIT,IBOK)=0,DA=$G(IBDA)  . S DIR("A")="Division: ",DIR(0)="355.92,.05AOr"  . ; Default Division - Main if adding or Existing if editing  . I IBFUNC="E" S DIR("B")=$P($$DIV^IBCEP7($P(IBDA0,U,5)),"/")  . I IBFUNC="A" S DIR("B")=$P($$EXTERNAL^DILFD(355.92,.05,"",IBMAIN),"/")  . D ^DIR K DIR  . Q:$D(DTOUT)!$D(DUOUT)  . S IBDIV=+$S(Y>0:+Y,1:0)  ;  ; See if there are any Care Units  S IBCAREUN="\*N/A\*"  I IBEFTFL="E",IBCUF D  . N TAR  . D LIST^DIC(355.95,,.01,,,,,,"I $P(^(0),U,3)=+$G(IBINS),$P(^(0),U,4)=+$G(IBDIV)",,"TAR")  . Q:+$G(TAR("DILIST",0))  . S IBCUF=0  . W !!,"There are no Care Units defined for this Division.",!  ;  ; Get the Care Unit  I IBEFTFL="E",IBCUF D  I Y<1 G FLDSQ  . K DIC  . S DIC("A")="Care Unit: "  . I IBFUNC="E" D  ; default only if editing  .. Q:IBDIV'=$P(IBDA0,U,5) ; don't default if division has changed  .. S DIC("B")=$$EXTERNAL^DILFD(355.92,.03,"",$P(IBDA0,U,3))  . S DIC=355.95,DIC("S")="I $P(^(0),U,3)=+$G(IBINS),$P(^(0),U,4)=+$G(IBDIV)",DIC(0)="AEMQ"  . D ^DIC  . I Y>0 S IBCAREUN=+Y  ;  ; Think this is done for sorting purposes. Makes the main division first  I IBDIV=IBMAIN S IBDIV=0  ;  ; Get the Provider ID Type  K DIR  S IBQUIT=0  I $P(IBPARAM,U,3)'=1 D  . S DIR("?")="Can NOT be State LIC # or Billing Facility Primary"  . S DIR("A")="ID Qualifier: "  . S DIR(0)="355.92,.06A^^K:'$$FACID^IBCEP7(+Y)!$P($G(^IBE(355.97,+Y,1)),U,9)!($P($G(^(0)),U,3)=""0B"") X"  . W ! D ^DIR K DIR  . I $D(DTOUT)!$D(DUOUT) S IBQUIT=1  E  D  G:$D(DTOUT)!$D(DUOUT) FLDSQ  . S DIR("A")="ID Qualifier: "    ;,DIR(0)="355.92,.06Ar"  . S DIR(0)="PAr^355.97:AEMQ"  . S DIR("?")="Enter a Qualifier to indentify the type of ID number you are entering."  . ; Default Type of ID - Electronic Plan Type if adding or Existing if editing  . N PITIEN S PITIEN=$S(IBFUNC="A"&(IBEFTFL="E"):$$BF^IBCU(),IBFUNC="E":$P(IBDA0,U,6),1:"")  . I PITIEN]"" S DIR("B")=$P($G(^IBE(355.97,PITIEN,0)),U)  . I IBEFTFL="E" D  .. S DIR("?",1)=" The current default ID Qualifier is based upon the Electronic Plan Type."  .. S DIR("?",2)=" You may change the ID Qualifier and the change will apply to all Plan"  .. S DIR("?")=" Types."  .. S DIR("S")="I ($P($G(^(0)),U,3)=$P($G(^IBE(355.97,PITIEN,0)),U,3))!$$BPS^IBCEPU(Y)"  . I IBEFTFL="A" S DIR("S")="I $$BPS^IBCEPU(Y)"  . I IBEFTFL="LF" S DIR("S")="I $$LFINS^IBCEPU(Y)"  . D ^DIR K DIR  G:IBQUIT FLDSQ  S IBITYP=$P(Y,U)  ;  ; Get Form Type  K DIR  S DIR("A")="Form Type: " **;JWS;IB\*2.0\*592;Dental form #7 J430D**  S DIR(0)=$S(IBEFTFL="LF":"SA^0:ALL;1:UB-04;2:CMS-1500;**4:J430D**",1:"SA^1:UB-04;2:CMS-1500;**4:J430D**")  I $G(IBDA) S DIR("B")=$S(+$P($G(^IBA(355.92,IBDA,0)),U,4)=0:"ALL",1:$P("UB-04^CMS-1500^^**J430D**",U,+$P($G(^IBA(355.92,IBDA,0)),U,4)))  **; end ;JWS;IB\*2.0\*592;Dental form #7 J430D**  D ^DIR K DIR  G:$D(DTOUT)!$D(DUOUT) FLDSQ  S IBFORM=$P(Y,U)  ;  ; Set up array of exisiting IDs by form type, divison, and care units to avoid duplications  S Z=0 F  S Z=$O(^IBA(355.92,"B",IBINS,Z)) Q:'Z  D  . S Z0=$G(^IBA(355.92,Z,0))  . I '(IBFUNC="E"&(Z=IBDA)) D  .. I IBEFTFL="LF",$P(Z0,U,8)'="LF" Q   ; If lab/facility ID, it only needs to be unique among lab/facility IDs  .. I IBEFTFL'="LF",$P(Z0,U,8)="LF" Q   ; If not lab/facility ID, it must be unigue for the others (secondary and additional)  .. I IBEFTFL="A",$P(Z0,U,8)="E" Q  .. I $P(Z0,U,8)="E",IBEFTFL'="A" S IB("\*N/A\*",$P(Z0,U,4),+$P(Z0,U,5),$S($P(Z0,U,3)]"":$P(Z0,U,3),1:"\*N/A\*"))=Z  .. S IB($P(Z0,U,6),$P(Z0,U,4),+$P(Z0,U,5),$S($P(Z0,U,3)]"":$P(Z0,U,3),1:"\*N/A\*"))=Z  . ;  . ; count them  . I IBFUNC="A",$P(Z0,U,8)=IBEFTFL,IBDIV=$P(Z0,U,5)!(IBDIV=0&($P(Z0,U,5)="")) D  .. I ".1.2."[("."\_$P(Z0,U,4)\_".") S IBCNTADD($P(Z0,U,4))=$G(IBCNTADD($P(Z0,U,4)))+1 Q  .. N I  .. F I=1,2 S IBCNTADD(I)=$G(IBCNTADD(I))+1  ; Check for duplications  S IBOK=1  ; Don't check if nothing is being changed. The ID itself can be changed after return to calling program.  I IBFUNC="E" S Z0=$G(^IBA(355.92,IBDA,0)) I $P(Z0,U,3)=IBCAREUN!($P(Z0,U,3)=""&(IBCAREUN="\*N/A\*")),IBFORM=$P(Z0,U,4),IBDIV=$P(Z0,U,5),IBITYP=$P(Z0,U,6) G FLDSQ  I $G(IB($S(IBEFTFL="E":"\*N/A\*",1:IBITYP),IBFORM,IBDIV,IBCAREUN)) D  . N Z,ZPC8 S Z=$G(IB($S(IBEFTFL="E":"\*N/A\*",1:IBITYP),IBFORM,IBDIV,IBCAREUN))  . S ZPC8=""  . I +Z S ZPC8=$P($G(^IBA(355.92,Z,0)),U,8)  . S IBOK="0^DUPLICATE"\_U\_ZPC8  I IBOK,IBFORM=0,$S($D(IB($S(IBEFTFL="E":"\*N/A\*",1:IBITYP),1,IBDIV,IBCAREUN))!$D(IB($S(IBEFTFL="E":"\*N/A\*",1:IBITYP),2,IBDIV,IBCAREUN)):1,1:0) S IBOK="0^FORM^SPECIFIC"  **;JWS;IB\*2.0\*592;Dental form #7 J430D changed BOTH to ALL**  I IBOK,IBFORM'=0,IBFORM'=3,$S($D(IB($S(IBEFTFL="E":"\*N/A\*",1:IBITYP),0,IBDIV,IBCAREUN)):1,1:0) S IBOK="0^FORM^**ALL**"  ;  S IBLIMIT=$S(IBEFTFL="A":6,IBEFTFL="LF":5,1:"")  I IBOK,IBFUNC="A",IBEFTFL'="E" D  . I ".1.2."[("."\_IBFORM\_".") D  Q  .. I $G(IBCNTADD(IBFORM))>(IBLIMIT-1) S IBOK="0^LIMIT"  . N I  . I IBFORM=0 F I=1,2 I $G(IBCNTADD(I))>IBLIMIT S IBOK="0^LIMIT" Q  ;  I 'IBOK D  . I $P(IBOK,U,2)="DUPLICATE" D  Q  .. S DIR("A",1)="This ID combination is already defined",DIR("A",2)=""  .. ; under "\_$S($P(IBOK,U,3)="A":" Additonal IDs",$P(IBOK,U,3)="E":"Billing Provider Secondary ID",1:"VA Lab/Facility IDs")\_$S(IBFUNC="A":" - try editing it instead",1:""),DIR("A",2)=" "  . ;  **. ;JWS;IB\*2.0\*592;Dental form #7 J430D changed to ALL from BOTH**  . I $P(IBOK,U,2)="**ALL"** D  Q  .. **;JWS;IB\*2.0\*592;Dental form #7 J430D changed to 'all' from 'both'**  .. S DIR("A",1)="An ID combination for **all** form types already exists. Delete this one",DIR("A",2)="before defining a form specific ID"\_$S(IBDIV:" for this division"),DIR("A",4)=" "  . ;  . I $P(IBOK,U,2)="FORM" D  Q  **.. ;JWS;IB\*2.0\*592;Dental form #7 J430D changed to ALL from BOTH**  .. I $P(IBOK,U,3)="ALL" S DIR("A",1)="This ID already exists for **all** form types - Delete it to enter this ID for",DIR("A",2)=" a specific form type",DIR("A",3)=" " Q  .. S DIR("A",1)="This ID already exists for a specific form type - Delete specific form type",DIR("A",2)=" ID(s) before entering one for **all** form types",DIR("A",3)=" "  . ;   . I $P(IBOK,U,2)="LIMIT" D  Q  .. S DIR("A",1)="Limit is "\_IBLIMIT\_" IDs for each form type",DIR("A",2)=" "  .. I IBEFTFL="A" D  ... S DIR("A",1)="A maximum of 6 Additional Billing Provider Sec IDs can be entered for each Form"  ... S DIR("A",2)="Type. Before you can add another ID, you must delete an existing ID."  ... S DIR("A",3)=" "  ;  I 'IBOK S DIR(0)="EA",DIR("A")="PRESS RETURN TO CONTINUE: " W ! D ^DIR K DIR  ; FLDSQ Q +IBOK | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCEP8A | | | | | | | | |
| **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** | | | | | | | | | |
| IBCEP8A ;ALB/ESG - Functions for provider ID maint ;12/27/2005  ;;2.0;INTEGRATED BILLING;\*\*320,349\*\*;21-MAR-94;Build 46  ;;Per VHA Directive 2004-038, this routine should not be modified.  ;  Q  ; CLIA(IBIFN) ; Default CLIA# for claim  NEW CLIA,NONVA,DIV,INST  S CLIA="",IBIFN=+$G(IBIFN)  S NONVA=+$P($G(^DGCR(399,IBIFN,"U2")),U,10) ; non-VA facility ptr  I NONVA S CLIA=$$CLIANVA^IBCEP8(IBIFN) G CLIAX  ;  ; retrieve the default VA clia# based on claim data  S DIV=+$P($G(^DGCR(399,IBIFN,0)),U,22) ; claim's division  I 'DIV G CLIAX  S INST=+$P($G(^DG(40.8,DIV,0)),U,7) ; inst file pointer  I 'INST G CLIAX  S CLIA=$$ID^XUAF4("CLIA",INST) ; API for clia# CLIAX ;  Q CLIA  ; LAB(IBIFN) ; Function determines if LAB type of service is on claim  ; Claim must be a CMS-1500 claim form type  NEW LAB,LN,IBXDATA  S LAB=0  I $$FT^IBCEF(IBIFN)'=2 G LABX    ;cms-1500 form types only  D F^IBCEF("N-HCFA 1500 SERVICES (PRINT)",,,IBIFN)  S LN=0  F  S LN=$O(IBXDATA(LN)) Q:'LN  I $P(IBXDATA(LN),U,4)=5 S LAB=1 Q LABX ;  Q LAB  ; CLIAREQ(IBIFN) ; Function determines if the CLIA# is required for claim  ; Return value=1 Yes, the CLIA# is required; otherwise 0.  NEW REQ S REQ=0  I $$FT^IBCEF(IBIFN)'=2 G CLIAREQX        ; cms-1500 claim  I '$$LAB(IBIFN) G CLIAREQX               ; lab type of service  ;  ; this is required for VA facility  I '$P($G(^DGCR(399,IBIFN,"U2")),U,10) S REQ=1 G CLIAREQX  ;  ; for non-VA facility, further check non-VA care type  ; Codes 1 and 3 are specifically Non-Lab  I '$F(".1.3.","."\_$P($G(^DGCR(399,IBIFN,"U2")),U,11)\_".") S REQ=1 CLIAREQX ;  Q REQ  ; MAMMO(IBIFN,IBMC) ; Function to determine the default mammography certification  ; number for the claim  ; Array IBMC is returned if passed by reference  ; IBMC = # of associated mammo#'s  ; IBMC(n) = [1] coding system or "" for Non-VA Facilities  ; [2] mammo cert#  NEW MAMMO,NONVA,INST,CODSYS,IBMCID,CDSYS  S MAMMO="",IBIFN=+$G(IBIFN),IBMC=0  S NONVA=+$P($G(^DGCR(399,IBIFN,"U2")),U,10) ; non-VA facility ptr  I NONVA D  G MAMMOX  . S MAMMO=$P($G(^IBA(355.93,NONVA,0)),U,15) Q:MAMMO=""  . S IBMC=1,IBMC(1)=""\_U\_MAMMO  . Q  ;  ; retrieve the default VA mammo# based on claim data  S INST=+$$SITE^VASITE() ; inst file pointer  I 'INST G MAMMOX  ;  ; Kernel API from XU\*8\*394 to get a list of coding systems  D LCDSYS^XUAF4(.CDSYS)  S CODSYS="MAMMO"  F  S CODSYS=$O(CDSYS(CODSYS)) Q:$E(CODSYS,1,5)'="MAMMO"  D  . S IBMCID=$$ID^XUAF4(CODSYS,INST) Q:IBMCID=""  . S IBMC=IBMC+1  . S IBMC(IBMC)=$P(CODSYS,"-",2)\_U\_IBMCID  . I $P(CODSYS,"-",2)="FDA" S MAMMO=IBMCID    ; FDA is default ID#  . Q  I IBMC,MAMMO="" S MAMMO=$P(IBMC(1),U,2) MAMMOX ;  Q MAMMO  ; MAMMODP(IBIFN) ; Procedure to display a listing of default mammo cert#'s  ; Used during input template on screen 8 for CMS-1500 claims  NEW IBMC,IBZ  I $$MAMMO(IBIFN,.IBMC)  I 'IBMC W !!?3,"No default mammography certification numbers on file.",! G MAMMODPX  W !!?3,"The Mammography Certification #" W:IBMC>1 "'s"  W " defined for this " W:$P($G(^DGCR(399,IBIFN,"U2")),U,10) "non-"  W "VA facility " W:IBMC>1 "are:" W:IBMC'>1 "is:"  S IBZ=0  F  S IBZ=$O(IBMC(IBZ)) Q:'IBZ  W !?7,$P(IBMC(IBZ),U,2),?21,$P(IBMC(IBZ),U,1)  W !?3,"If you enter a different number it will be sent with this claim only."  I $P($G(^DGCR(399,IBIFN,"U2")),U,10) W !?3,"To change the defined Mammography Certification #, use Prov ID Maint."  W ! MAMMODPX ;  Q  ; XRAY(IBIFN) ; Function determines if X-RAY type of service is on claim  ; Claim must be a CMS-1500 claim form type  NEW XRAY,LN,IBXDATA  S XRAY=0  I $$FT^IBCEF(IBIFN)'=2 G XRAYX    ;cms-1500 form types only  D F^IBCEF("N-HCFA 1500 SERVICES (PRINT)",,,IBIFN)  S LN=0  F  S LN=$O(IBXDATA(LN)) Q:'LN  I $P(IBXDATA(LN),U,4)=4 S XRAY=1 Q XRAYX ;  Q XRAY  ; EIN(IBIFN) ; Function to return the EIN/tax ID for either the VA facility  ; or the non-VA facility. Used for SUB-9.  NEW ID,IBU2,NONVA  S ID="",IBU2=$G(^DGCR(399,IBIFN,"U2"))  S NONVA=+$P(IBU2,U,10) ; non-VA facility ptr  I NONVA D  G EINX  . S ID=$P($G(^IBA(355.93,NONVA,0)),U,9) ; ID# from file 355.93  . ;  . ; if not defined in file 355.93, then use legacy field# 234 in file  . ; 399 - non-va care id#. See NONVAID^IBCEF72.  . I ID="",$P(IBU2,U,12)'="" S ID=$P(IBU2,U,12)  . Q  ;  ; VA facility  S ID=$P($G(^IBE(350.9,1,1)),U,5) ; Federal tax id from site params EINX ;  Q ID  ; BOX324(IBIFN,IBXSAVE,IBXDATA) ; Procedure which further defines and formats  ; form 1500, box 32, line 4.  ; \*\*\* THIS IS NOT USED FOR THE NEW CMS-1500 CLAIM FORM \*\*\*  ; This is either the facility Tax ID or it is the mammography  ; certification number.  ; Input: IBIFN, IBXSAVE array (pass by ref), IBXDATA (pass by ref)  ; Output: IBXDATA (pass by ref)  ;  NEW IBZ  ;  ; retrieve the mammo# if it exists into variable IBZ  D F^IBCEF("N-MAMMOGRAPHY CERT#","IBZ",,IBIFN)  ;  ; If the claim is for the main VAMC and there is no mammo# then print  ; nothing here. See 364.7 iens# 348, 319, 327 for similar  I '$G(IBXSAVE("REMOTE")),IBZ="" KILL IBXDATA G BOX32X  ;  ; If the mammo# exists, then display that  I IBZ'="" S IBXDATA="Mammography Cert# "\_IBZ G BOX32X  ;  ; Otherwise, display the facility tax id  S IBXDATA="FAC. ID:"\_$G(IBXDATA) BOX32X ;  KILL IBXSAVE("OFAC"),IBXSAVE("REMOTE") ; cleanup  Q  ; SUB1OK(IBIFN) ; This function determines if the claim meets the criteria  ; for being eligible to output a SUB1 segment which is for professional  ; purchased services. Must be CMS-1500, non-VA facility, and Fee Basis.  ;  NEW OK,IBU2  S OK=0,IBU2=$G(^DGCR(399,IBIFN,"U2"))  ;  I $$FT^IBCEF(IBIFN)'=2 G SX                      ; must be cms-1500  I '$P(IBU2,U,10) G SX                            ; must be non-VA fac  I '$F(".1.2.","."\_$P(IBU2,U,11)\_".") G SX        ; must be FEE services  ;  S OK=1 ; all checks passed, OK for SUB1 output SX ;  Q OK  ; | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCEP8A ;ALB/ESG - Functions for provider ID maint ;12/27/2005  ;;2.0;INTEGRATED BILLING;\*\*320,349,**592**\*\*;21-MAR-94;Build 46  ;;Per VHA Directive 2004-038, this routine should not be modified.  ;  Q  ; CLIA(IBIFN) ; Default CLIA# for claim  NEW CLIA,NONVA,DIV,INST  S CLIA="",IBIFN=+$G(IBIFN)  S NONVA=+$P($G(^DGCR(399,IBIFN,"U2")),U,10) ; non-VA facility ptr  I NONVA S CLIA=$$CLIANVA^IBCEP8(IBIFN) G CLIAX  ;  ; retrieve the default VA clia# based on claim data  S DIV=+$P($G(^DGCR(399,IBIFN,0)),U,22) ; claim's division  I 'DIV G CLIAX  S INST=+$P($G(^DG(40.8,DIV,0)),U,7) ; inst file pointer  I 'INST G CLIAX  S CLIA=$$ID^XUAF4("CLIA",INST) ; API for clia# CLIAX ;  Q CLIA  ; LAB(IBIFN) ; Function determines if LAB type of service is on claim  ; Claim must be a CMS-1500 claim form type  N LAB,LN,IBXDATA  S LAB=0 **;JWS;IB\*2.0\*592;Dental form #7 J430D**  I $$FT^IBCEF(IBIFN)'=2**,$$FT^IBCEF(IBIFN)'=7** G LABX  ;cms-1500 and **Dental J430D** form types only  D F^IBCEF("N-HCFA 1500 SERVICES (PRINT)",,,IBIFN)  S LN=0  F  S LN=$O(IBXDATA(LN)) Q:'LN  I $P(IBXDATA(LN),U,4)=5 S LAB=1 Q LABX ;  Q LAB  ; CLIAREQ(IBIFN) ; Function determines if the CLIA# is required for claim  ; Return value=1 Yes, the CLIA# is required; otherwise 0.  N REQ S REQ=0  **;JWS;IB\*2.0\*592;Dental form #7 J430D**  I $$FT^IBCEF(IBIFN)'=2**,$$FT^IBCEF(IBIFN)'=7** G CLIAREQX  ; cms-1500 **and Dental J430D**  I '$$LAB(IBIFN) G CLIAREQX  ; lab type of service  ;  ; this is required for VA facility  I '$P($G(^DGCR(399,IBIFN,"U2")),U,10) S REQ=1 G CLIAREQX  ;  ; for non-VA facility, further check non-VA care type  ; Codes 1 and 3 are specifically Non-Lab  I '$F(".1.3.","."\_$P($G(^DGCR(399,IBIFN,"U2")),U,11)\_".") S REQ=1 CLIAREQX ;  Q REQ  ; MAMMO(IBIFN,IBMC) ; Function to determine the default mammography certification  ; number for the claim  ; Array IBMC is returned if passed by reference  ; IBMC = # of associated mammo#'s  ; IBMC(n) = [1] coding system or "" for Non-VA Facilities  ; [2] mammo cert#  NEW MAMMO,NONVA,INST,CODSYS,IBMCID,CDSYS  S MAMMO="",IBIFN=+$G(IBIFN),IBMC=0  S NONVA=+$P($G(^DGCR(399,IBIFN,"U2")),U,10) ; non-VA facility ptr  I NONVA D  G MAMMOX  . S MAMMO=$P($G(^IBA(355.93,NONVA,0)),U,15) Q:MAMMO=""  . S IBMC=1,IBMC(1)=""\_U\_MAMMO  . Q  ;  ; retrieve the default VA mammo# based on claim data  S INST=+$$SITE^VASITE() ; inst file pointer  I 'INST G MAMMOX  ;  ; Kernel API from XU\*8\*394 to get a list of coding systems  D LCDSYS^XUAF4(.CDSYS)  S CODSYS="MAMMO"  F  S CODSYS=$O(CDSYS(CODSYS)) Q:$E(CODSYS,1,5)'="MAMMO"  D  . S IBMCID=$$ID^XUAF4(CODSYS,INST) Q:IBMCID=""  . S IBMC=IBMC+1  . S IBMC(IBMC)=$P(CODSYS,"-",2)\_U\_IBMCID  . I $P(CODSYS,"-",2)="FDA" S MAMMO=IBMCID    ; FDA is default ID#  . Q  I IBMC,MAMMO="" S MAMMO=$P(IBMC(1),U,2) MAMMOX ;  Q MAMMO  ; MAMMODP(IBIFN) ; Procedure to display a listing of default mammo cert#'s  ; Used during input template on screen 8 for CMS-1500 claims  NEW IBMC,IBZ  I $$MAMMO(IBIFN,.IBMC)  I 'IBMC W !!?3,"No default mammography certification numbers on file.",! G MAMMODPX  W !!?3,"The Mammography Certification #" W:IBMC>1 "'s"  W " defined for this " W:$P($G(^DGCR(399,IBIFN,"U2")),U,10) "non-"  W "VA facility " W:IBMC>1 "are:" W:IBMC'>1 "is:"  S IBZ=0  F  S IBZ=$O(IBMC(IBZ)) Q:'IBZ  W !?7,$P(IBMC(IBZ),U,2),?21,$P(IBMC(IBZ),U,1)  W !?3,"If you enter a different number it will be sent with this claim only."  I $P($G(^DGCR(399,IBIFN,"U2")),U,10) W !?3,"To change the defined Mammography Certification #, use Prov ID Maint."  W ! MAMMODPX ;  Q  ; XRAY(IBIFN) ; Function determines if X-RAY type of service is on claim  ; Claim must be a CMS-1500 claim form type  NEW XRAY,LN,IBXDATA  S XRAY=0  **;JWS;IB\*2.0\*592;Dental form #7 J430D**  I $$FT^IBCEF(IBIFN)'=2**,$$FT^IBCEF(IBIFN)'=7** G XRAYX  ;cms-1500 and **Dental J430D** form types only  D F^IBCEF("N-HCFA 1500 SERVICES (PRINT)",,,IBIFN)  S LN=0  F  S LN=$O(IBXDATA(LN)) Q:'LN  I $P(IBXDATA(LN),U,4)=4 S XRAY=1 Q XRAYX ;  Q XRAY  ; EIN(IBIFN) ; Function to return the EIN/tax ID for either the VA facility  ; or the non-VA facility. Used for SUB-9.  NEW ID,IBU2,NONVA  S ID="",IBU2=$G(^DGCR(399,IBIFN,"U2"))  S NONVA=+$P(IBU2,U,10) ; non-VA facility ptr  I NONVA D  G EINX  . S ID=$P($G(^IBA(355.93,NONVA,0)),U,9) ; ID# from file 355.93  . ;  . ; if not defined in file 355.93, then use legacy field# 234 in file  . ; 399 - non-va care id#. See NONVAID^IBCEF72.  . I ID="",$P(IBU2,U,12)'="" S ID=$P(IBU2,U,12)  . Q  ;  ; VA facility  S ID=$P($G(^IBE(350.9,1,1)),U,5) ; Federal tax id from site params EINX ;  Q ID  ; BOX324(IBIFN,IBXSAVE,IBXDATA) ; Procedure which further defines and formats  ; form 1500, box 32, line 4.  ; \*\*\* THIS IS NOT USED FOR THE NEW CMS-1500 CLAIM FORM \*\*\*  ; This is either the facility Tax ID or it is the mammography  ; certification number.  ; Input: IBIFN, IBXSAVE array (pass by ref), IBXDATA (pass by ref)  ; Output: IBXDATA (pass by ref)  ;  NEW IBZ  ;  ; retrieve the mammo# if it exists into variable IBZ  D F^IBCEF("N-MAMMOGRAPHY CERT#","IBZ",,IBIFN)  ;  ; If the claim is for the main VAMC and there is no mammo# then print  ; nothing here. See 364.7 iens# 348, 319, 327 for similar  I '$G(IBXSAVE("REMOTE")),IBZ="" KILL IBXDATA G BOX32X  ;  ; If the mammo# exists, then display that  I IBZ'="" S IBXDATA="Mammography Cert# "\_IBZ G BOX32X  ;  ; Otherwise, display the facility tax id  S IBXDATA="FAC. ID:"\_$G(IBXDATA) BOX32X ;  KILL IBXSAVE("OFAC"),IBXSAVE("REMOTE") ; cleanup  Q  ; SUB1OK(IBIFN) ; This function determines if the claim meets the criteria  ; for being eligible to output a SUB1 segment which is for professional  ; purchased services. Must be CMS-1500, non-VA facility, and Fee Basis.  ;  NEW OK,IBU2  S OK=0,IBU2=$G(^DGCR(399,IBIFN,"U2"))  ;  **;JWS;IB\*2.0\*592;Dental form #7 J430D**  I $$FT^IBCEF(IBIFN)'=2**,$$FT^IBCEF(IBIFN)'=7** G SX  ; must be cms-1500 **or Dental J430D**  I '$P(IBU2,U,10) G SX                            ; must be non-VA fac  I '$F(".1.2.","."\_$P(IBU2,U,11)\_".") G SX        ; must be FEE services  ;  S OK=1 ; all checks passed, OK for SUB1 output SX ;  Q OK  ; | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCEP9 | | | | | | | | |
| **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** | | | | | | | | | |
| IBCEP9 ;ALB/TMP - MASS UPDATE OF PROVIDER ID FROM FILE OR MANUAL ;08-NOV-00  ;;2.0;INTEGRATED BILLING;\*\*137,200,320,348,349\*\*;21-MAR-94;Build 46  ;;Per VHA Directive 2004-038, this routine should not be modified.  ; EN ; Get parameters and mass input provider id by ins co  N A,DA,DIC,DIE,DIK,DIR,DR,POP,Q,Q0,X,Y,Y3,Z,Z0  N IBCND,IBCU,IBCT,IBDELIM,IBFILE,IBFILEN,IBFILEP,IBFORMAT  N IBFT,IBINFILE,IBINS,IBL,IBN,IBOK,IBOPEN,IBPOS,IBPT,IBQUIT  N IBQUIT1,IBQUOTES,IBRA,IBS,IBSA,IBSTART,IBSRC,IBVERIFY,IBVNAME  K ^TMP("IBPID\_IN",$J),^TMP("IBPID-ERR",$J),^TMP("IBPID",$J)  S IBQUIT=0 1 ; Select INSURANCE COMPANY NAME:  G:IBQUIT ENQ  S IBQUIT1=0  S DIC("S")="I $P($G(^DIC(36,+Y,3)),U,13)'=""C"""  S DIC(0)="AEMQ",DIC="^DIC(36," D ^DIC  I Y'>0 G ENQ  S IBINS=+Y  S IBQUIT=$$LOCK^IBCEP9B(IBINS)  I IBQUIT,$G(IBINS) D  G 1  . D UNLOCK^IBCEP9B(IBINS)  . S IBINS="",IBQUIT=0  . W !!,"Unable to lock all associated insurance companies.",!,"Please try again later.",!!  ; 2 ; get data source  S IBQUIT1=0  S DIR(0)="SA^M:Manual Entry;F:Entry from file"  S DIR("A")="PROVIDER ID DATA SOURCE: ",DIR("B")="Manual Entry"  S Y=$$DIR(.DIR,.IBQUIT,.IBQUIT1)  I Y=""!("FM"'[Y)!IBQUIT1 D UNLOCK^IBCEP9B(IBINS) G 1  S IBSRC=Y,IBVERIFY=0  S IBVERIFY=(Y="M")  I 'IBVERIFY D  G:IBQUIT ENQ G:IBQUIT 2  . S DIR(0)="YA",DIR("A")="DO YOU WANT TO VIEW/VERIFY EACH ENTRY BEFORE IT GETS UPDATED?: "  . S Y=$$DIR(.DIR,.IBQUIT,.IBQUIT1)  . I Y=1 S IBVERIFY=1  ;  G:IBSRC="M" 4 21 ; get parameters for file type  G:IBQUIT ENQ  S IBQUIT1=0  S DIR(0)="SA^D:DELIMITED;F:FIXED LENGTH",DIR("B")="D",DIR("A")="SELECT FILE FORMAT: "  S Y=$$DIR(.DIR,.IBQUIT,.IBQUIT1)  I IBQUIT1 G 2  S IBPOS=Y  I IBPOS="D" D  G:IBQUIT1 21  . S DIR(0)="FA^1:1",DIR("B")=",",DIR("A")="DELIMITER CHARACTER: "  . S Y=$$DIR(.DIR,.IBQUIT,.IBQUIT1)  . Q:IBQUIT1  . S $P(IBPOS,U,2)=Y  . S DIR(0)="YA",DIR("B")="NO",DIR("A")="ARE QUOTES WITHIN A FIELD DOUBLE QUOTED?: "  . S Y=$$DIR(.DIR,.IBQUIT,.IBQUIT1,,,1)  . Q:IBQUIT1  . S $P(IBPOS,U,3)=Y 3 ; select external file name  G:IBQUIT ENQ  S IBQUIT1=0  G:IBSRC="M" 2  S DIR(0)="FA^1:60"  S DIR("A")="FILE NAME PATH: ",DIR("B")=$$PWD^%ZISH  S Y=$$DIR(.DIR,.IBQUIT,.IBQUIT1)  G:IBQUIT1 2  S IBFILEP=$P(Y,U)  S DIR(0)="FA^1:60"  S DIR("A")="FILE NAME: "  S IBSA("\*")=""  S DIR("?")="^S Y3=$$LIST^%ZISH(IBFILEP,""IBSA"",""IBRA"") I Y3=1 S Y3="""" F S Y3=$O(IBRA(Y3)) Q:Y3="""" W !,Y3"  S Y=$$DIR(.DIR,.IBQUIT,.IBQUIT1,,,1)  G:IBQUIT1 2  S IBFILEN=$P(Y,U)  K ^TMP($J),IBRA,Y3  N Y S Y=$$FTG^%ZISH(IBFILEP,IBFILEN,$NA(^TMP($J,1)),2)  I Y=0 W !,"FILE ",IBFILEP,IBFILEN," COULD NOT BE FOUND OR COULD NOT BE OPENED",! G 3  S IBFILE=IO 4 ; select Provider ID Type  G:IBQUIT ENQ  S IBQUIT1=0  S DIR(0)="355.9,.06"  I IBSRC="M" S Z=$P($G(^IBE(355.97,+$$PPTYP^IBCEP0(IBINS),0)),U) S:Z'="" DIR("B")=Z  S Y=$$DIR(.DIR,.IBQUIT,.IBQUIT1)  G:Y=""!IBQUIT1 3  S IBPTYP=$P(Y,U) 5 ; select Forms Type  G:IBQUIT ENQ  S IBQUIT1=0  S DIR(0)="355.9,.04r",DIR("B")="BOTH UB-04 AND CMS-1500 FORMS"  S Y=$$DIR(.DIR,.IBQUIT,.IBQUIT1)  G:IBQUIT1 4  I Y=""!("012"'[Y) G 5  S IBFT=$P(Y,U) 6 ; select Bill Care Type  G:IBQUIT ENQ  S IBQUIT1=0  S DIR(0)="355.9,.05r",DIR("B")="BOTH INPATIENT AND OUTPATIENT"  S Y=$$DIR(.DIR,.IBQUIT,.IBQUIT1)  G:IBQUIT1 5  I Y=""!("0123"'[$P(Y,U)) G 6  S IBCT=$P(Y,U)  ;  S IBCND=$$CAREUN^IBCEP3(IBINS,IBPTYP,IBFT,IBCT,IBCT=3) 7 ; get Care Unit  G:IBQUIT ENQ  S IBQUIT1=0  I IBCND D  G:IBQUIT1 6  . S DIR(0)="355.9,.03O"  . S Y=$$DIR(.DIR,.IBQUIT,.IBQUIT1)  . Q:IBQUIT1  . S IBCU=$P(Y,U)  . I IBCU="" W !!,$J("",22),"\*\*\*\*\* WARNING \*\*\*\*\*",!," YOU WILL NEED TO MANUALLY ENTER THE CARE UNIT FOR EACH PROVIDER",!!  ;  ; Manual entry to get providers from VistA  I IBSRC="M" D MANUAL^IBCEP9B G:IBQUIT1 6  ; For 'OTHER' files ask position/length or delimiter/piece for data  I IBSRC="F" D  I IBQUIT1 G:'IBCND 6 G 7  . F Z="PROV. SSN^SSN^15^1","PROV. NAME^NAM^30","PROV. 1500 ID^PROF\_ID^15","PROV. UB-04 ID^INST\_ID^15" D  Q:IBQUIT1  .. I $P(IBPOS,U)'="D" D  ... N X  ... I IBFT=0!(IBFT=1) Q:Z["PROF\_ID"  I Z["INST\_ID" S $P(Z,U)="PROV. ID"  ... I IBFT=2 Q:Z["INST\_ID"  ... S DIR("A")="START POSITION OF "\_$P(Z,U)\_" FIELD: "  ... S DIR(0)="NA"\_$S($P(Z,U,4)!($P(Z,U)["PROV. ID")!($P(Z,U)["\_ID"):"",1:"O")\_"^1:250"  ... W ! S X=$$DIR1^IBCEP9B(.DIR,Z,.IBQUIT,.IBQUIT1)  ... Q:IBQUIT1  ... I X>0 D  .... S IBPOS($P(Z,U,2))=X  .... S DIR("A")="LENGTH OF "\_$P(Z,U)\_" FIELD: "  .... S DIR(0)="NA"\_$S($P(Z,U,3):"^1:"\_$P(Z,U,3),1:"")  .... S X=$$DIR1^IBCEP9B(.DIR,Z,.IBQUIT,.IBQUIT1)  .... Q:IBQUIT1  .... S $P(IBPOS($P(Z,U,2)),U,2)=IBPOS($P(Z,U,2))+X-1  .. ;  .. I $P(IBPOS,U)="D" D  ... I IBFT=0!(IBFT=1) Q:Z["PROF\_ID"  I Z["INST\_ID" S $P(Z,U)="PROV. ID"  ... I IBFT=2 Q:Z["INST\_ID"  ... W ! S DIR("A")="STARTING '"\_$P(IBPOS,U,2)\_"' PIECE # OF "\_$P(Z,U)\_" FIELD: "  ... S DIR(0)="NA"\_$S($P(Z,U,4)!($P(Z,U)["PROV. ID")!($P(Z,U)["\_ID"):"",1:"O")  ... S X=$$DIR1^IBCEP9B(.DIR,Z,.IBQUIT,.IBQUIT1)  ... Q:IBQUIT1  ... I X>0 D  .... S (DIR("B"),IBPOS($P(Z,U,2)))=X  .... S DIR("A")="ENDING '"\_$P(IBPOS,U,2)\_"' PIECE # OF "\_$P(Z,U)\_" FIELD: "  .... S DIR(0)="NA"\_$S($P(Z,U,4):"",1:"O")\_U\_(IBPOS($P(Z,U,2)))\_":99"  .... S DIR("?")="JUST PRESS THE ENTER KEY IF THIS FIELD IS CONTAINED IN ONLY 1 PIECE"  .... S Y=$$DIR1^IBCEP9B(.DIR,Z,.IBQUIT,.IBQUIT1)  .... Q:IBQUIT1  .... W ! I Y>0,Y'=IBPOS($P(Z,U,2)) S $P(IBPOS($P(Z,U,2)),U,2)=Y  .. ;  . Q:IBQUIT1  . D READFILE^IBCEP9B  . ; P1 ;  S Z="" F  S Z=$O(^TMP("IBPID\_IN",$J,Z)) Q:Z=""  S Z0=0 F  S Z0=$O(^TMP("IBPID\_IN",$J,Z,Z0)) Q:'Z0  S Q=$G(^(Z0)) D  G:IBQUIT ENQ  . ;  . I IBSRC="M" D  Q  .. D DISP^IBCEP9B(Q,0,IBINS,IBPTYP,IBFT,IBCT,$G(IBCU),,IBSRC)  .. ; Manually add IDs  .. S IBN=$$DUP(+Z0\_";VA(200,",IBINS,$S($G(IBCU)'="":IBCU,1:"\*N/A\*"),IBFT,IBCT,IBPTYP)  .. I 'IBN D  Q:IBQUIT!(IBN'>0)  ... S IBN=$$ADDID^IBCEP9B(Z0,IBINS,$G(IBCU),IBFT,IBCT,IBPTYP,,.IBQUIT)  .. S DIE="^IBA(355.9,",DR=".07",DA=+IBN D ^DIE  .. I $D(Y)!($P($G(^IBA(355.9,+IBN,0)),U,7)="") D  ... I $P(IBN,U,3) S DA=+IBN,DIK="^IBA(355.9," D ^DIK  ... S DIR(0)="YA",DIR("B")="NO",DIR("A")="DO YOU WANT TO STOP ENTERING PROVIDER IDs?: "  ... S Y=$$DIR(.DIR,.IBQUIT,.IBQUIT1,,1,1)  ... I Y=1 S IBQUIT=1  .. S IBID=$P($G(^IBA(355.9,+IBN,0)),U,7)  .. S:$L(IBID) ^TMP("IBPID\_IN",$J,U,Z0,"INST\_ID")=IBID  .. I IBID="" K ^TMP("IBPID\_IN",$J,U,Z0)  .. I IBQUIT=1 F  S Z0=$O(^TMP("IBPID\_IN",$J,U,Z0)) Q:Z0=""  K ^TMP("IBPID\_IN",$J,U,Z0) ; user wants to stop, remove all remaining names from list  . ;  . S IBOK=1  . N IBX,IBID  . M IBX=^TMP("IBPID\_IN",$J,Z,Z0)  . I IBSRC="F" S IBID=$S(IBFT=0!(IBFT=1):$G(IBX("INST\_ID")),1:$G(IBX("PROF\_ID")))  . I $G(IBVERIFY) D  ; Display record, ask OK to file id's  .. D DISP^IBCEP9B(Q,0,IBINS,IBPTYP,IBFT,IBCT,$G(IBCU),,IBSRC)  .. W !,"PROVIDER ID: ",IBID  .. S DIR("A")="OK TO FILE THIS ID FOR THIS PROVIDER?: ",DIR(0)="YA",DIR("B")="NO"  .. S Y=$$DIR(.DIR,,,,1,1)  .. I Y'=1 D  Q  ; Send to error array  ... S IBOK=0  ... S ^TMP("IBPID-ERR",$J,2,$P(IBX,U),$P(IBX,U,2)\_" ","PROV ID")=IBID  ... S ^TMP("IBPID\_IN",$J,U,Z0,0)="NO PRINT"  ... N Z1  ... S Z1="" F  S Z1=$O(IBX(Z1)) Q:Z1=""  I $G(IBX(Z1))'="",Z1'["\_ID" S ^TMP("IBPID-ERR",$J,2,$P(IBX,U),$P(IBX,U,2)\_" ",Z1)=IBX(Z1)  . I IBOK D  ; Add/update the record  .. I IBSRC="F" D  ... I IBID'="" D  .... S IBN=$$ADDID^IBCEP9B(+Z0,IBINS,$G(IBCU),IBFT,IBCT,IBPTYP,,.IBQUIT)  .... I IBQUIT D:IBN>0 Q  ..... S DA=+IBN,DIK="^IBA(355.9," D ^DIK  .... I IBN>0 S DIE="^IBA(355.9,",DA=+IBN,DR=".07////"\_IBID D ^DIE  .. ;  ; ENQ ; Print report, exit  I $G(IBINS) D  . D COPY^IBCEPCID(IBINS)  . D UNLOCK^IBCEP9B(IBINS)  ;  I ($D(^TMP("IBPID-ERR",$J)))!($D(^TMP("IBPID\_IN",$J))) D  . N %ZIS,ZTSAVE,ZTRTN,ZTDESC,IBDUZ  . S IBDUZ=$G(DUZ)  . S %ZIS="QM" D ^%ZIS Q:POP  . I $D(IO("Q")) K IO("Q") D  D ^%ZTLOAD K ZTSK D HOME^%ZIS Q  .. S ZTRTN="PRTERR^IBCEP9B",ZTSAVE("^TMP(""IBPID-ERR"",$J,")=""  .. S ZTSAVE("^TMP(""IBPID\_IN"",$J,")="",ZTSAVE("IB\*")=""  .. S ZTDESC="IB - PROVIDER ID BATCH UPDATE ERROR LOG"  . U IO  . D PRTERR^IBCEP9B  K ^TMP("IBPID\_IN",$J),^TMP("IBPID-ERR",$J),^TMP("IBPID",$J)  U IO(0)  Q  ; DUP(IBPRV,IBINS,IBCU,IBFT,IBCT,IBPTYP) ; Check if provider id record already exists in file 355.9  Q +$O(^IBA(355.9,"AUNIQ",IBPRV,IBINS,IBCU,IBFT,IBCT,IBPTYP,0))  ; ERREOF ; Traps EOF error on file read for non-DSM systems  N IBERROR S IBERROR=$$EC^%ZOSV  I IBERROR["ENDOFFILE" D CLOSE(.IBOPEN) G ENQ  D ^%ZTER  Q  ; CLOSE(IBOPEN) ; Close file  D CLOSE^%ZISH("IBINFILE") S IBOPEN=0  Q  ; DIR(DIR,IBQUIT,IBQUIT1,X,IBW1,IBW2) ; Standard call to ^DIR  ; Inputs DIR array  ; Returns IBQUIT,IBQUIT1,X if passed by reference  ; AND  ; FUNCTION returns the value of Y  ; IBW1 = 1 if initial write ! should be done  ; IBW2 = 1 if last write ! should be done  N DIROUT,DTOUT,DUOUT,DA  W:$G(IBW1) ! D ^DIR K DIR W:$G(IBW2) !  S (IBQUIT,IBQUIT1)=0  S DIR("?")="Enter '^' to back up one prompt or '^^' to exit the option"  I $D(DIROUT) S (IBQUIT,IBQUIT1)=1  I $D(DTOUT)!$D(DUOUT) S IBQUIT1=1  Q Y  ; ERR ; Error list  ;; INVALID OR MISSING SSN - NO PROVIDER MATCH FOUND  ;; NO UPDATE PER USER REQUEST  ;; | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCEP9 ;ALB/TMP - MASS UPDATE OF PROVIDER ID FROM FILE OR MANUAL ;08-NOV-00  ;;2.0;INTEGRATED BILLING;\*\*137,200,320,348,349**,592**\*\*;21-MAR-94;Build 46  ;;Per VHA Directive 2004-038, this routine should not be modified.  ; EN ; Get parameters and mass input provider id by ins co  N A,DA,DIC,DIE,DIK,DIR,DR,POP,Q,Q0,X,Y,Y3,Z,Z0  N IBCND,IBCU,IBCT,IBDELIM,IBFILE,IBFILEN,IBFILEP,IBFORMAT  N IBFT,IBINFILE,IBINS,IBL,IBN,IBOK,IBOPEN,IBPOS,IBPT,IBQUIT  N IBQUIT1,IBQUOTES,IBRA,IBS,IBSA,IBSTART,IBSRC,IBVERIFY,IBVNAME  K ^TMP("IBPID\_IN",$J),^TMP("IBPID-ERR",$J),^TMP("IBPID",$J)  S IBQUIT=0 1 ; Select INSURANCE COMPANY NAME:  G:IBQUIT ENQ  S IBQUIT1=0  S DIC("S")="I $P($G(^DIC(36,+Y,3)),U,13)'=""C"""  S DIC(0)="AEMQ",DIC="^DIC(36," D ^DIC  I Y'>0 G ENQ  S IBINS=+Y  S IBQUIT=$$LOCK^IBCEP9B(IBINS)  I IBQUIT,$G(IBINS) D  G 1  . D UNLOCK^IBCEP9B(IBINS)  . S IBINS="",IBQUIT=0  . W !!,"Unable to lock all associated insurance companies.",!,"Please try again later.",!!  ; 2 ; get data source  S IBQUIT1=0  S DIR(0)="SA^M:Manual Entry;F:Entry from file"  S DIR("A")="PROVIDER ID DATA SOURCE: ",DIR("B")="Manual Entry"  S Y=$$DIR(.DIR,.IBQUIT,.IBQUIT1)  I Y=""!("FM"'[Y)!IBQUIT1 D UNLOCK^IBCEP9B(IBINS) G 1  S IBSRC=Y,IBVERIFY=0  S IBVERIFY=(Y="M")  I 'IBVERIFY D  G:IBQUIT ENQ G:IBQUIT 2  . S DIR(0)="YA",DIR("A")="DO YOU WANT TO VIEW/VERIFY EACH ENTRY BEFORE IT GETS UPDATED?: "  . S Y=$$DIR(.DIR,.IBQUIT,.IBQUIT1)  . I Y=1 S IBVERIFY=1  ;  G:IBSRC="M" 4 21 ; get parameters for file type  G:IBQUIT ENQ  S IBQUIT1=0  S DIR(0)="SA^D:DELIMITED;F:FIXED LENGTH",DIR("B")="D",DIR("A")="SELECT FILE FORMAT: "  S Y=$$DIR(.DIR,.IBQUIT,.IBQUIT1)  I IBQUIT1 G 2  S IBPOS=Y  I IBPOS="D" D  G:IBQUIT1 21  . S DIR(0)="FA^1:1",DIR("B")=",",DIR("A")="DELIMITER CHARACTER: "  . S Y=$$DIR(.DIR,.IBQUIT,.IBQUIT1)  . Q:IBQUIT1  . S $P(IBPOS,U,2)=Y  . S DIR(0)="YA",DIR("B")="NO",DIR("A")="ARE QUOTES WITHIN A FIELD DOUBLE QUOTED?: "  . S Y=$$DIR(.DIR,.IBQUIT,.IBQUIT1,,,1)  . Q:IBQUIT1  . S $P(IBPOS,U,3)=Y 3 ; select external file name  G:IBQUIT ENQ  S IBQUIT1=0  G:IBSRC="M" 2  S DIR(0)="FA^1:60"  S DIR("A")="FILE NAME PATH: ",DIR("B")=$$PWD^%ZISH  S Y=$$DIR(.DIR,.IBQUIT,.IBQUIT1)  G:IBQUIT1 2  S IBFILEP=$P(Y,U)  S DIR(0)="FA^1:60"  S DIR("A")="FILE NAME: "  S IBSA("\*")=""  S DIR("?")="^S Y3=$$LIST^%ZISH(IBFILEP,""IBSA"",""IBRA"") I Y3=1 S Y3="""" F S Y3=$O(IBRA(Y3)) Q:Y3="""" W !,Y3"  S Y=$$DIR(.DIR,.IBQUIT,.IBQUIT1,,,1)  G:IBQUIT1 2  S IBFILEN=$P(Y,U)  K ^TMP($J),IBRA,Y3  N Y S Y=$$FTG^%ZISH(IBFILEP,IBFILEN,$NA(^TMP($J,1)),2)  I Y=0 W !,"FILE ",IBFILEP,IBFILEN," COULD NOT BE FOUND OR COULD NOT BE OPENED",! G 3  S IBFILE=IO 4 ; select Provider ID Type  G:IBQUIT ENQ  S IBQUIT1=0  S DIR(0)="355.9,.06"  I IBSRC="M" S Z=$P($G(^IBE(355.97,+$$PPTYP^IBCEP0(IBINS),0)),U) S:Z'="" DIR("B")=Z  S Y=$$DIR(.DIR,.IBQUIT,.IBQUIT1)  G:Y=""!IBQUIT1 3  S IBPTYP=$P(Y,U) 5 ; select Forms Type  G:IBQUIT ENQ  S IBQUIT1=0  **;JWS;IB\*2.0\*592 US1108 - Dental EDI 837D / form J430D**  S DIR(0)="355.9,.04r",DIR("B")="UB-04, CMS-1500 **and J430D** FORMS"  S Y=$$DIR(.DIR,.IBQUIT,.IBQUIT1)  G:IBQUIT1 4  I Y=""!("012"'[Y) G 5  S IBFT=$P(Y,U) 6 ; select Bill Care Type  G:IBQUIT ENQ  S IBQUIT1=0  S DIR(0)="355.9,.05r",DIR("B")="BOTH INPATIENT AND OUTPATIENT"  S Y=$$DIR(.DIR,.IBQUIT,.IBQUIT1)  G:IBQUIT1 5  I Y=""!("0123"'[$P(Y,U)) G 6  S IBCT=$P(Y,U)  ;  S IBCND=$$CAREUN^IBCEP3(IBINS,IBPTYP,IBFT,IBCT,IBCT=3) 7 ; get Care Unit  G:IBQUIT ENQ  S IBQUIT1=0  I IBCND D  G:IBQUIT1 6  . S DIR(0)="355.9,.03O"  . S Y=$$DIR(.DIR,.IBQUIT,.IBQUIT1)  . Q:IBQUIT1  . S IBCU=$P(Y,U)  . I IBCU="" W !!,$J("",22),"\*\*\*\*\* WARNING \*\*\*\*\*",!," YOU WILL NEED TO MANUALLY ENTER THE CARE UNIT FOR EACH PROVIDER",!!  ;  ; Manual entry to get providers from VistA  I IBSRC="M" D MANUAL^IBCEP9B G:IBQUIT1 6  ; For 'OTHER' files ask position/length or delimiter/piece for data  I IBSRC="F" D  I IBQUIT1 G:'IBCND 6 G 7  . F Z="PROV. SSN^SSN^15^1","PROV. NAME^NAM^30","PROV. 1500 ID^PROF\_ID^15","PROV. UB-04 ID^INST\_ID^15" D  Q:IBQUIT1  .. I $P(IBPOS,U)'="D" D  ... N X  ... I IBFT=0!(IBFT=1) Q:Z["PROF\_ID"  I Z["INST\_ID" S $P(Z,U)="PROV. ID"  ... I IBFT=2 Q:Z["INST\_ID"  ... S DIR("A")="START POSITION OF "\_$P(Z,U)\_" FIELD: "  ... S DIR(0)="NA"\_$S($P(Z,U,4)!($P(Z,U)["PROV. ID")!($P(Z,U)["\_ID"):"",1:"O")\_"^1:250"  ... W ! S X=$$DIR1^IBCEP9B(.DIR,Z,.IBQUIT,.IBQUIT1)  ... Q:IBQUIT1  ... I X>0 D  .... S IBPOS($P(Z,U,2))=X  .... S DIR("A")="LENGTH OF "\_$P(Z,U)\_" FIELD: "  .... S DIR(0)="NA"\_$S($P(Z,U,3):"^1:"\_$P(Z,U,3),1:"")  .... S X=$$DIR1^IBCEP9B(.DIR,Z,.IBQUIT,.IBQUIT1)  .... Q:IBQUIT1  .... S $P(IBPOS($P(Z,U,2)),U,2)=IBPOS($P(Z,U,2))+X-1  .. ;  .. I $P(IBPOS,U)="D" D  ... I IBFT=0!(IBFT=1) Q:Z["PROF\_ID"  I Z["INST\_ID" S $P(Z,U)="PROV. ID"  ... I IBFT=2 Q:Z["INST\_ID"  ... W ! S DIR("A")="STARTING '"\_$P(IBPOS,U,2)\_"' PIECE # OF "\_$P(Z,U)\_" FIELD: "  ... S DIR(0)="NA"\_$S($P(Z,U,4)!($P(Z,U)["PROV. ID")!($P(Z,U)["\_ID"):"",1:"O")  ... S X=$$DIR1^IBCEP9B(.DIR,Z,.IBQUIT,.IBQUIT1)  ... Q:IBQUIT1  ... I X>0 D  .... S (DIR("B"),IBPOS($P(Z,U,2)))=X  .... S DIR("A")="ENDING '"\_$P(IBPOS,U,2)\_"' PIECE # OF "\_$P(Z,U)\_" FIELD: "  .... S DIR(0)="NA"\_$S($P(Z,U,4):"",1:"O")\_U\_(IBPOS($P(Z,U,2)))\_":99"  .... S DIR("?")="JUST PRESS THE ENTER KEY IF THIS FIELD IS CONTAINED IN ONLY 1 PIECE"  .... S Y=$$DIR1^IBCEP9B(.DIR,Z,.IBQUIT,.IBQUIT1)  .... Q:IBQUIT1  .... W ! I Y>0,Y'=IBPOS($P(Z,U,2)) S $P(IBPOS($P(Z,U,2)),U,2)=Y  .. ;  . Q:IBQUIT1  . D READFILE^IBCEP9B  . ; P1 ;  S Z="" F  S Z=$O(^TMP("IBPID\_IN",$J,Z)) Q:Z=""  S Z0=0 F  S Z0=$O(^TMP("IBPID\_IN",$J,Z,Z0)) Q:'Z0  S Q=$G(^(Z0)) D  G:IBQUIT ENQ  . ;  . I IBSRC="M" D  Q  .. D DISP^IBCEP9B(Q,0,IBINS,IBPTYP,IBFT,IBCT,$G(IBCU),,IBSRC)  .. ; Manually add IDs  .. S IBN=$$DUP(+Z0\_";VA(200,",IBINS,$S($G(IBCU)'="":IBCU,1:"\*N/A\*"),IBFT,IBCT,IBPTYP)  .. I 'IBN D  Q:IBQUIT!(IBN'>0)  ... S IBN=$$ADDID^IBCEP9B(Z0,IBINS,$G(IBCU),IBFT,IBCT,IBPTYP,,.IBQUIT)  .. S DIE="^IBA(355.9,",DR=".07",DA=+IBN D ^DIE  .. I $D(Y)!($P($G(^IBA(355.9,+IBN,0)),U,7)="") D  ... I $P(IBN,U,3) S DA=+IBN,DIK="^IBA(355.9," D ^DIK  ... S DIR(0)="YA",DIR("B")="NO",DIR("A")="DO YOU WANT TO STOP ENTERING PROVIDER IDs?: "  ... S Y=$$DIR(.DIR,.IBQUIT,.IBQUIT1,,1,1)  ... I Y=1 S IBQUIT=1  .. S IBID=$P($G(^IBA(355.9,+IBN,0)),U,7)  .. S:$L(IBID) ^TMP("IBPID\_IN",$J,U,Z0,"INST\_ID")=IBID  .. I IBID="" K ^TMP("IBPID\_IN",$J,U,Z0)  .. I IBQUIT=1 F  S Z0=$O(^TMP("IBPID\_IN",$J,U,Z0)) Q:Z0=""  K ^TMP("IBPID\_IN",$J,U,Z0) ; user wants to stop, remove all remaining names from list  . ;  . S IBOK=1  . N IBX,IBID  . M IBX=^TMP("IBPID\_IN",$J,Z,Z0)  . I IBSRC="F" S IBID=$S(IBFT=0!(IBFT=1):$G(IBX("INST\_ID")),1:$G(IBX("PROF\_ID")))  . I $G(IBVERIFY) D  ; Display record, ask OK to file id's  .. D DISP^IBCEP9B(Q,0,IBINS,IBPTYP,IBFT,IBCT,$G(IBCU),,IBSRC)  .. W !,"PROVIDER ID: ",IBID  .. S DIR("A")="OK TO FILE THIS ID FOR THIS PROVIDER?: ",DIR(0)="YA",DIR("B")="NO"  .. S Y=$$DIR(.DIR,,,,1,1)  .. I Y'=1 D  Q  ; Send to error array  ... S IBOK=0  ... S ^TMP("IBPID-ERR",$J,2,$P(IBX,U),$P(IBX,U,2)\_" ","PROV ID")=IBID  ... S ^TMP("IBPID\_IN",$J,U,Z0,0)="NO PRINT"  ... N Z1  ... S Z1="" F  S Z1=$O(IBX(Z1)) Q:Z1=""  I $G(IBX(Z1))'="",Z1'["\_ID" S ^TMP("IBPID-ERR",$J,2,$P(IBX,U),$P(IBX,U,2)\_" ",Z1)=IBX(Z1)  . I IBOK D  ; Add/update the record  .. I IBSRC="F" D  ... I IBID'="" D  .... S IBN=$$ADDID^IBCEP9B(+Z0,IBINS,$G(IBCU),IBFT,IBCT,IBPTYP,,.IBQUIT)  .... I IBQUIT D:IBN>0 Q  ..... S DA=+IBN,DIK="^IBA(355.9," D ^DIK  .... I IBN>0 S DIE="^IBA(355.9,",DA=+IBN,DR=".07////"\_IBID D ^DIE  .. ;  ; ENQ ; Print report, exit  I $G(IBINS) D  . D COPY^IBCEPCID(IBINS)  . D UNLOCK^IBCEP9B(IBINS)  ;  I ($D(^TMP("IBPID-ERR",$J)))!($D(^TMP("IBPID\_IN",$J))) D  . N %ZIS,ZTSAVE,ZTRTN,ZTDESC,IBDUZ  . S IBDUZ=$G(DUZ)  . S %ZIS="QM" D ^%ZIS Q:POP  . I $D(IO("Q")) K IO("Q") D  D ^%ZTLOAD K ZTSK D HOME^%ZIS Q  .. S ZTRTN="PRTERR^IBCEP9B",ZTSAVE("^TMP(""IBPID-ERR"",$J,")=""  .. S ZTSAVE("^TMP(""IBPID\_IN"",$J,")="",ZTSAVE("IB\*")=""  .. S ZTDESC="IB - PROVIDER ID BATCH UPDATE ERROR LOG"  . U IO  . D PRTERR^IBCEP9B  K ^TMP("IBPID\_IN",$J),^TMP("IBPID-ERR",$J),^TMP("IBPID",$J)  U IO(0)  Q  ; DUP(IBPRV,IBINS,IBCU,IBFT,IBCT,IBPTYP) ; Check if provider id record already exists in file 355.9  Q +$O(^IBA(355.9,"AUNIQ",IBPRV,IBINS,IBCU,IBFT,IBCT,IBPTYP,0))  ; ERREOF ; Traps EOF error on file read for non-DSM systems  N IBERROR S IBERROR=$$EC^%ZOSV  I IBERROR["ENDOFFILE" D CLOSE(.IBOPEN) G ENQ  D ^%ZTER  Q  ; CLOSE(IBOPEN) ; Close file  D CLOSE^%ZISH("IBINFILE") S IBOPEN=0  Q  ; DIR(DIR,IBQUIT,IBQUIT1,X,IBW1,IBW2) ; Standard call to ^DIR  ; Inputs DIR array  ; Returns IBQUIT,IBQUIT1,X if passed by reference  ; AND  ; FUNCTION returns the value of Y  ; IBW1 = 1 if initial write ! should be done  ; IBW2 = 1 if last write ! should be done  N DIROUT,DTOUT,DUOUT,DA  W:$G(IBW1) ! D ^DIR K DIR W:$G(IBW2) !  S (IBQUIT,IBQUIT1)=0  S DIR("?")="Enter '^' to back up one prompt or '^^' to exit the option"  I $D(DIROUT) S (IBQUIT,IBQUIT1)=1  I $D(DTOUT)!$D(DUOUT) S IBQUIT1=1  Q Y  ; ERR ; Error list  ;; INVALID OR MISSING SSN - NO PROVIDER MATCH FOUND  ;; NO UPDATE PER USER REQUEST  ;; | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCEPTC | | | | | | | | |
| **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** | | | | | | | | | |
| IBCEPTC ;ALB/TMK - EDI PREVIOUSLY TRANSMITTED CLAIMS ; 4/12/05 11:15am  ;;2.0;INTEGRATED BILLING;\*\*296,320,348,349,547\*\*;21-MAR-94;Build 119  ;;Per VA Directive 6402, this routine should not be modified.  ; EN ; Main entrypoint  ; IBDT1,IBDT2 = last transmit date range to use  ; IBSORT = primary sort criteria to use B=BATCH #,I=INS CO NAME  ; IBFORM = form type to limit selection to U=UB-04,C=CMS-1500,B=BOTH  ; IBCRIT = the additional sort criteria needed  ; IBPTCCAN = whether or not to include cancelled claims  ; IBRCBFPC = whether or not to include force print @ clearinghouse  ; ^TMP("IB\_PREV\_CLAIM\_INS",$J) = 1 for specific ins co/null for all  ; ^($J,1,ien)="" for ien of each ins co selected  ; ^($J,2,payer ID,ien)="" if selected  ; IBREP = format output should be put in R=report,S=Listman  ;  N DIR,DIC,X,Y,Z,Z0,Z1,IBHOW,IBACT,IBCT,IBREP,IBCRIT,IBDT1,IBDT2,IBLOC  N IBFORM,IBOK,IBQUIT,IBSORT,IBY,DTOUT,DUOUT,%ZIS,ZTSAVE,ZTRTN,ZTDESC  N POP,IBPAYER,EDI,INST,PROF,IBPTCCAN,DIROUT,DIRUT,DTOUT,DUOUT,IBRCBFPC  ;  W !!,"\*\*\* Please Note \*\*\*"  W ?20,"2 '^' are needed to abort this option (^^)"  W !?20,"1 '^' brings you back to the previous prompt (^)"  W !  ; IB\*2.0\*547 add new prompt for locally printed vs. transmitted claims  S DIR(0)="SA^P:Printed;T:Transmitted",DIR("A")="Run report for (P)rinted or (T)ransmitted claims?: ",DIR("B")="Transmitted"  D ^DIR K DIR  I $D(DTOUT)!$D(DUOUT) G ENQ  ; Set a flag here to indicate user wants locally printed claims and use that to control how the rest of the prompts act.  S IBLOC=$S(Y="T":"",1:1)  ; Q1 ;  W !  ;S DIR(0)="SA^C:Claim;B:Batch;L:List",DIR("A")="Select By: (C)laim, (B)atch or see a (L)ist to pick from?: ",DIR("B")="List"  S DIR(0)="SA^C:Claim;"\_$S(IBLOC:"",1:"B:Batch;")\_"L:List",DIR("A")="Select By: (C)laim"\_$S(IBLOC:"",1:", (B)atch")\_" or see a (L)ist to pick from?: ",DIR("B")="List"  D ^DIR K DIR  I $D(DTOUT)!$D(DUOUT) G ENQ  S IBHOW=Y  I IBLOC=1 W !,"Previously printed claims to a payer that does not accept EDI are omitted."  I IBHOW="L" G Q1A  ;  S IBQUIT=0,IBCT=0  K ^TMP($J,IBHOW)  F  D  Q:IBQUIT  .;I IBHOW="C" S DIR("A")="Select a"\_$S(IBCT:"nother",1:"")\_" Claim: ",DIR(0)="PA^364:AEMQZ",DIR("S")="I '$P(^(0),U,7),'$O(^IBA(364,""B"",+^(0),Y))"  . I IBHOW="C",IBLOC="" S DIR("A")="Select a"\_$S(IBCT:"nother",1:"")\_" Claim: ",DIR(0)="PA^364:AEMQZ",DIR("S")="I '$P(^(0),U,7),'$O(^IBA(364,""B"",+^(0),Y))"  . I IBHOW="C",IBLOC=1 S DIR("A")="Select a"\_$S(IBCT:"nother",1:"")\_" Locally Printed Claim: ",DIR(0)="PA^399:AEMQZ",DIR("S")="I '$D(^IBA(364,""B"",Y)),$$INSOK^IBCEF4(+$$CURR^IBCEF2(Y))"  . I IBHOW="B" S DIR("A")="Select a"\_$S(IBCT:"nother",1:"")\_" Batch: ",DIR(0)="PA^IBA(364.1,:AEMQ^W "" "",$P(^(0),U,3),"" Claims""",DIR("S")="I '$P(^(0),U,14)"  . S DIR("?")="^D SELDSP^IBCEPTC(IBHOW)"  . S:IBCT $P(DIR(0),U)=$P(DIR(0),U)\_"O" ; Optional prompt after one is selected  . D ^DIR K DIR  . I Y'>0 S IBQUIT=$S(X="^":2,X="^^":3,1:1) Q  . S IBY=$S(IBHOW="C":+Y,1:""),Y=$S(IBHOW="C":+Y(0),1:Y) S:IBLOC=1 Y=IBY  . I '$D(^TMP($J,IBHOW,+Y)) S IBCT=IBCT+1,^TMP($J,IBHOW,+Y)=IBY  ;  G:IBQUIT=3 ENQ  G:IBQUIT=2!'$O(^TMP($J,IBHOW,0)) Q1  S Z=0  I IBHOW="C" F  S Z=$O(^TMP($J,"C",Z)) Q:'Z  S ^TMP("IB\_PREV\_CLAIM\_SELECT",$J,Z,0)=^TMP($J,"C",Z)  I IBHOW="B" S (Z,IBCT)=0 F  S Z=$O(^TMP($J,"B",Z)) Q:'Z  D  . S Z0=0 F  S Z0=$O(^IBA(364,"C",Z,Z0)) Q:'Z0  S Z1=+$G(^IBA(364,Z0,0)) I Z1,'$D(^TMP("IB\_PREV\_CLAIM\_SELECT",$J,Z1,0)) S ^(0)=Z0,IBCT=IBCT+1  S ^TMP("IB\_PREV\_CLAIM\_SELECT",$J)=IBCT  D RESUB^IBCEPTC3  G ENQ  ; Q1A K ^TMP("IB\_PREV\_CLAIM\_INS",$J)  S DIR(0)="SA^A:All Payers;S:Selected Payers"  S DIR("A")="Run for (A)ll Payers or (S)elected Payers?: " S DIR("B")="Selected Payers"  W !!,"PAYER SELECTION:" D ^DIR K DIR  I X="^^" G ENQ  I $D(DTOUT)!$D(DUOUT) G Q1  ;  I Y="A" S ^TMP("IB\_PREV\_CLAIM\_INS",$J)="" G Q2  ;  ; esg - 11/21/05 - patch 320 question  W !  S DIR(0)="Y",DIR("A")=" Include all payers with the same electronic Payer ID",DIR("B")="Yes" D ^DIR K DIR  I $D(DIROUT) G ENQ  I $D(DIRUT) G Q1A  S IBPAYER=Y  W !  ;  S ^TMP("IB\_PREV\_CLAIM\_INS",$J)=1  S IBQUIT=0  F  D  Q:IBQUIT  . ; IB\*2.0\*547 allow lookup by EDI#'s using new cross-ref  . ;S DIC(0)="AEMQ",DIC=36,DIC("A")=" Select Insurance Company: "  . S DIC(0)="AEMQn",DIC=36,DIC("A")=" Select Insurance Company: "  . I $O(^TMP("IB\_PREV\_CLAIM\_INS",$J,1,"")) S DIC("A")=" Select Another Insurance Company: "  . S DIC("W")="D INSLIST^IBCEMCA(Y)"  . ;D ^DIC K DIC ; lookup  . N D S D="B^AEI^AEP" D MIX^DIC1 K DIC,D  . I X="^^" S IBQUIT=2 Q          ; user entered "^^"  . I +Y'>0 S IBQUIT=1 Q           ; user is done  . W !  . S ^TMP("IB\_PREV\_CLAIM\_INS",$J,1,+Y)=""  . I 'IBPAYER Q  . S EDI=$$UP^XLFSTR($G(^DIC(36,+Y,3)))  . S PROF=$P(EDI,U,2),INST=$P(EDI,U,4)  . I PROF'="",PROF'["PRNT" S ^TMP("IB\_PREV\_CLAIM\_INS",$J,2,PROF,+Y)=""  . I INST'="",INST'["PRNT" S ^TMP("IB\_PREV\_CLAIM\_INS",$J,2,INST,+Y)=""  . Q  ;  I IBQUIT=2 G ENQ  ;  I '$O(^TMP("IB\_PREV\_CLAIM\_INS",$J,1,0)) D  G Q1A  . W \*7,!!?3,"No payers have been selected. Please try again."  . Q  ; Q2 S DIR(0)="SA^C:CMS-1500;U:UB-04;B:Both",DIR("B")="Both"  S DIR("A")="Run for (U)B-04, (C)MS-1500 or (B)oth: "  W !!,"BILL FORM TYPE SELECTION:" D ^DIR K DIR  I X="^^" G ENQ  I $D(DTOUT)!$D(DUOUT) G Q1A  S IBFORM=Y  ; Q3 S DIR(0)="DA^0:9999999:EPX",DIR("A")="Start with Date "\_$S(IBLOC:"First Printed: ",1:"Last Transmitted: ")  ;S DIR("?",1)="This is the earliest date on which a batch that you want to include on this",DIR("?",2)=" report was last transmitted. You may choose a maximum date range of 90 days.",DIR("?")=" "  S DIR("?",1)="This is the earliest date on which a batch that you want to include on this",DIR("?",2)=" report was "\_$S(IBLOC=1:"first printed",1:"last transmitted")\_". You may choose a maximum date range of 90 days.",DIR("?")=" "  ;W !!,"LAST BATCH TRANSMIT DATE RANGE SELECTION:" D ^DIR K DIR  W !!,$S(IBLOC:"FIRST PRINT",1:"LAST BATCH TRANSMIT")\_" DATE RANGE SELECTION:" D ^DIR K DIR  I X="^^" G ENQ  I $D(DTOUT)!$D(DUOUT) G Q2  S IBDT1=Y  S IBDT2=$$FMADD^XLFDT(IBDT1,90) I IBDT2>DT S IBDT2=DT  S DIR("?",1)="This is the latest date on which a batch that you want to include on this",DIR("?",2)=" report was "\_$S(IBLOC:"first printed",1:"last transmitted")\_". You may choose a maximum date range of 90 days.",DIR("?")=" "  S DIR("B")=$$FMTE^XLFDT(IBDT2,2),DIR(0)="DA^"\_IBDT1\_":"\_IBDT2\_":EPX"  S DIR("A")="Go to Date "\_$S(IBLOC:"First Printed",1:"Last Transmitted")\_":("\_$$FMTE^XLFDT(IBDT1,2)\_"-"\_$$FMTE^XLFDT(IBDT2,2)\_"): " D ^DIR K DIR  I X="^^" G ENQ  I $D(DTOUT)!$D(DUOUT) G Q3  S IBDT2=Y  ; Q4 ; Additional selection criteria  S DIR(0)="SAO^1:MRA Secondary Only;2:Primary Claims Only;3:Secondary Claims Only;4:Claims Previously Printed at Clearinghouse"  S DIR("A",1)="ADDITIONAL SELECTION CRITERIA:",DIR("A",2)=" ",DIR("A",3)="1 - MRA Secondary Only",DIR("A",4)="2 - Primary Claims Only",DIR("A",5)="3 - Secondary Claims Only"  S DIR("A",6)=$S(IBLOC:"",1:"4 - Claims Sent to Print at Clearinghouse Only"),DIR("A",7)=" ",DIR("A")="Select Additional Limiting Criteria (optional): "  S DIR("?")="Select one of the listed criteria to further limit the claims to include"  W ! D ^DIR K DIR  I X="^^" G ENQ  I $D(DTOUT)!$D(DUOUT) G Q3  S IBCRIT=Y  ; Q41 ; Ask user if they want to include cancelled claims  S DIR(0)="Y",DIR("B")="No",DIR("A")="Would you like to include cancelled claims"  W ! D ^DIR K DIR  I X="^^" G ENQ  I $D(DIRUT) G Q4  S IBPTCCAN=Y  ; IB\*2.0\*547 skip next 2 questions if looking for locally printed claims  I IBLOC S IBSORT=2,IBRCBFPC=0 G Q6  ; Q42 ; Include claims that are forced to print at clearinghouse?  S DIR(0)="Y",DIR("B")="No",DIR("A")="Would you like to include claims Forced to Print at the Clearinghouse"  W ! D ^DIR K DIR  I X="^^" G ENQ  I $D(DIRUT) G Q41  S IBRCBFPC=Y  ; Q5 S DIR("L",1)="Select one of the following: ",DIR("L",2)=" ",DIR("L",3)=$J("",10)\_"1 Batch By Last Transmitted Date (Claims within a Batch)",DIR("L",4)=$J("",10)\_"2 Current Payer (Insurance Company)"  S DIR("L",5)=" "  S DIR(0)="SA^1:Batch By Last Transmitted Date (Claims within a Batch);2:Current Payer (Insurance Company)",DIR("B")="Current Payer"  S DIR("A")="Sort By: "  W ! D ^DIR K DIR  I X="^^" G ENQ  I $D(DTOUT)!$D(DUOUT) G Q42  S IBSORT=Y  ; Q6 S DIR(0)="SA^R:Report;S:Screen List"  S DIR("A")="Do you want a (R)eport or a (S)creen List format?: "  S DIR("B")="Screen List"  W ! D ^DIR K DIR  I X="^^" G ENQ  I $D(DTOUT)!$D(DUOUT) G Q5  S IBREP=Y  ; IB \*2.0\*547 call new SUB-routine for locally printed claims (not in file 364)  I IBREP="S",IBLOC D LOC^IBCEPTC0 G ENQ  ;  I IBREP="S",'IBLOC D LIST^IBCEPTC0 G ENQ  ; Q7 ; Select device  F  S IBACT=0 D DEVSEL(.IBACT) Q:IBACT  I IBACT=99 G ENQ  U IO  ; IB \*2.0\*547 call new SUB-routine for locally printed claims (not in file 364)  D:'IBLOC LIST^IBCEPTC0  D:IBLOC LOC^IBCEPTC0  ; ENQ K ^TMP("IB\_PREV\_CLAIM\_INS",$J),^TMP("IB\_PREV\_CLAIM\_SELECT",$J)  Q  ; DEVSEL(IBACT) ;  N DIR,POP,X,Y,ZTRTN,ZTSAVE  W !!,"You will need a 132 column printer for this report!"  S %ZIS="QM" D ^%ZIS I POP S IBACT=99 G DEVSELQ  I $G(IOM),IOM<132 S IBOK=1 D  I 'IBOK S IBACT=0 G DEVSELQ  . S DIR(0)="YA",DIR("A",1)="This report requires output to a 132 column device.",DIR("A",2)="The device you have chosen is only set for "\_IOM\_".",DIR("A")="Are you sure you want to continue?: ",DIR("B")="No"  . W ! D ^DIR K DIR  . I Y'=1 S IBOK=0 W !  I $D(IO("Q")) D  S IBACT=99 G DEVSELQ  . K IO("Q")  . S ZTRTN="LIST^IBCEPTC0",ZTSAVE("IBCRIT(")="",ZTSAVE("IB\*")="",ZTSAVE("^TMP(""IB\_PREV\_CLAIM\_INS"",$J)")="",ZTSAVE("^TMP(""IB\_PREV\_CLAIM\_INS"",$J,")="",ZTDESC="IB - Previously Transmitted Claims Report"  . D ^%ZTLOAD K ZTSK D HOME^%ZIS  S IBACT=1 DEVSELQ Q  ; SELDSP(IBHOW) ; Display list of selected claims/batches  ; IBHOW = "C" for claims "B" for batches  N Z,DIR,CT,QUIT  I '$O(^TMP($J,IBHOW,0)) Q  S (CT,QUIT)=0  W !!,$S(IBHOW="C":"Claims",1:"Batches")," Already Selected:"  S Z=0 F  S Z=$O(^TMP($J,IBHOW,Z)) Q:'Z!QUIT  S Z0=$G(^(Z)) D  Q:QUIT  . I IBHOW="C" W !,?3,$P($G(^DGCR(399,Z,0)),U) Q  . W !,?3,$P($G(^IBA(364.1,Z,0)),U)," ",$P(^(0),U,3)," Claims"  . S CT=CT+1  . I '(CT#10),$O(^TMP($J,IBHOW,Z)) S DIR("A")="Press return for more or '^' to exit ",DIR(0)="EA" W ! D ^DIR K DIR I $D(DTOUT)!$D(DUOUT) S QUIT=1  W !  Q  ; | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCEPTC ;ALB/TMK - EDI PREVIOUSLY TRANSMITTED CLAIMS ; 4/12/05 11:15am  ;;2.0;INTEGRATED BILLING;\*\*296,320,348,349,547,**592**\*\*;21-MAR-94;Build 119  ;;Per VA Directive 6402, this routine should not be modified.  ; EN ; Main entrypoint  ; IBDT1,IBDT2 = last transmit date range to use  ; IBSORT = primary sort criteria to use B=BATCH #,I=INS CO NAME  ; IBFORM = form type to limit selection to U=UB-04,C=CMS-1500,B=BOTH  ; IBCRIT = the additional sort criteria needed  ; IBPTCCAN = whether or not to include cancelled claims  ; IBRCBFPC = whether or not to include force print @ clearinghouse  ; ^TMP("IB\_PREV\_CLAIM\_INS",$J) = 1 for specific ins co/null for all  ; ^($J,1,ien)="" for ien of each ins co selected  ; ^($J,2,payer ID,ien)="" if selected  ; IBREP = format output should be put in R=report,S=Listman  ;  N DIR,DIC,X,Y,Z,Z0,Z1,IBHOW,IBACT,IBCT,IBREP,IBCRIT,IBDT1,IBDT2,IBLOC  N IBFORM,IBOK,IBQUIT,IBSORT,IBY,DTOUT,DUOUT,%ZIS,ZTSAVE,ZTRTN,ZTDESC  N POP,IBPAYER,EDI,INST,PROF,IBPTCCAN,DIROUT,DIRUT,DTOUT,DUOUT,IBRCBFPC  ;  W !!,"\*\*\* Please Note \*\*\*"  W ?20,"2 '^' are needed to abort this option (^^)"  W !?20,"1 '^' brings you back to the previous prompt (^)"  W !  ; IB\*2.0\*547 add new prompt for locally printed vs. transmitted claims  S DIR(0)="SA^P:Printed;T:Transmitted",DIR("A")="Run report for (P)rinted or (T)ransmitted claims?: ",DIR("B")="Transmitted"  D ^DIR K DIR  I $D(DTOUT)!$D(DUOUT) G ENQ  ; Set a flag here to indicate user wants locally printed claims and use that to control how the rest of the prompts act.  S IBLOC=$S(Y="T":"",1:1)  ; Q1 ;  W !  ;S DIR(0)="SA^C:Claim;B:Batch;L:List",DIR("A")="Select By: (C)laim, (B)atch or see a (L)ist to pick from?: ",DIR("B")="List"  S DIR(0)="SA^C:Claim;"\_$S(IBLOC:"",1:"B:Batch;")\_"L:List",DIR("A")="Select By: (C)laim"\_$S(IBLOC:"",1:", (B)atch")\_" or see a (L)ist to pick from?: ",DIR("B")="List"  D ^DIR K DIR  I $D(DTOUT)!$D(DUOUT) G ENQ  S IBHOW=Y  I IBLOC=1 W !,"Previously printed claims to a payer that does not accept EDI are omitted."  I IBHOW="L" G Q1A  ;  S IBQUIT=0,IBCT=0  K ^TMP($J,IBHOW)  F  D  Q:IBQUIT  .;I IBHOW="C" S DIR("A")="Select a"\_$S(IBCT:"nother",1:"")\_" Claim: ",DIR(0)="PA^364:AEMQZ",DIR("S")="I '$P(^(0),U,7),'$O(^IBA(364,""B"",+^(0),Y))"  . I IBHOW="C",IBLOC="" S DIR("A")="Select a"\_$S(IBCT:"nother",1:"")\_" Claim: ",DIR(0)="PA^364:AEMQZ",DIR("S")="I '$P(^(0),U,7),'$O(^IBA(364,""B"",+^(0),Y))"  . I IBHOW="C",IBLOC=1 S DIR("A")="Select a"\_$S(IBCT:"nother",1:"")\_" Locally Printed Claim: ",DIR(0)="PA^399:AEMQZ",DIR("S")="I '$D(^IBA(364,""B"",Y)),$$INSOK^IBCEF4(+$$CURR^IBCEF2(Y))"  . I IBHOW="B" S DIR("A")="Select a"\_$S(IBCT:"nother",1:"")\_" Batch: ",DIR(0)="PA^IBA(364.1,:AEMQ^W "" "",$P(^(0),U,3),"" Claims""",DIR("S")="I '$P(^(0),U,14)"  . S DIR("?")="^D SELDSP^IBCEPTC(IBHOW)"  . S:IBCT $P(DIR(0),U)=$P(DIR(0),U)\_"O" ; Optional prompt after one is selected  . D ^DIR K DIR  . I Y'>0 S IBQUIT=$S(X="^":2,X="^^":3,1:1) Q  . S IBY=$S(IBHOW="C":+Y,1:""),Y=$S(IBHOW="C":+Y(0),1:Y) S:IBLOC=1 Y=IBY  . I '$D(^TMP($J,IBHOW,+Y)) S IBCT=IBCT+1,^TMP($J,IBHOW,+Y)=IBY  ;  G:IBQUIT=3 ENQ  G:IBQUIT=2!'$O(^TMP($J,IBHOW,0)) Q1  S Z=0  I IBHOW="C" F  S Z=$O(^TMP($J,"C",Z)) Q:'Z  S ^TMP("IB\_PREV\_CLAIM\_SELECT",$J,Z,0)=^TMP($J,"C",Z)  I IBHOW="B" S (Z,IBCT)=0 F  S Z=$O(^TMP($J,"B",Z)) Q:'Z  D  . S Z0=0 F  S Z0=$O(^IBA(364,"C",Z,Z0)) Q:'Z0  S Z1=+$G(^IBA(364,Z0,0)) I Z1,'$D(^TMP("IB\_PREV\_CLAIM\_SELECT",$J,Z1,0)) S ^(0)=Z0,IBCT=IBCT+1  S ^TMP("IB\_PREV\_CLAIM\_SELECT",$J)=IBCT  D RESUB^IBCEPTC3  G ENQ  ; Q1A K ^TMP("IB\_PREV\_CLAIM\_INS",$J)  S DIR(0)="SA^A:All Payers;S:Selected Payers"  S DIR("A")="Run for (A)ll Payers or (S)elected Payers?: " S DIR("B")="Selected Payers"  W !!,"PAYER SELECTION:" D ^DIR K DIR  I X="^^" G ENQ  I $D(DTOUT)!$D(DUOUT) G Q1  ;  I Y="A" S ^TMP("IB\_PREV\_CLAIM\_INS",$J)="" G Q2  ;  ; esg - 11/21/05 - patch 320 question  W !  S DIR(0)="Y",DIR("A")=" Include all payers with the same electronic Payer ID",DIR("B")="Yes" D ^DIR K DIR  I $D(DIROUT) G ENQ  I $D(DIRUT) G Q1A  S IBPAYER=Y  W !  ;  S ^TMP("IB\_PREV\_CLAIM\_INS",$J)=1  S IBQUIT=0  F  D  Q:IBQUIT  . ; IB\*2.0\*547 allow lookup by EDI#'s using new cross-ref  . ;S DIC(0)="AEMQ",DIC=36,DIC("A")=" Select Insurance Company: "  . S DIC(0)="AEMQn",DIC=36,DIC("A")=" Select Insurance Company: "  . I $O(^TMP("IB\_PREV\_CLAIM\_INS",$J,1,"")) S DIC("A")=" Select Another Insurance Company: "  . S DIC("W")="D INSLIST^IBCEMCA(Y)"  . ;D ^DIC K DIC ; lookup  . N D S D="B^AEI^AEP" D MIX^DIC1 K DIC,D  . I X="^^" S IBQUIT=2 Q          ; user entered "^^"  . I +Y'>0 S IBQUIT=1 Q           ; user is done  . W !  . S ^TMP("IB\_PREV\_CLAIM\_INS",$J,1,+Y)=""  . I 'IBPAYER Q  . S EDI=$$UP^XLFSTR($G(^DIC(36,+Y,3)))  . S PROF=$P(EDI,U,2),INST=$P(EDI,U,4)  . I PROF'="",PROF'["PRNT" S ^TMP("IB\_PREV\_CLAIM\_INS",$J,2,PROF,+Y)=""  . I INST'="",INST'["PRNT" S ^TMP("IB\_PREV\_CLAIM\_INS",$J,2,INST,+Y)=""  . Q  ;  I IBQUIT=2 G ENQ  ;  I '$O(^TMP("IB\_PREV\_CLAIM\_INS",$J,1,0)) D  G Q1A  . W \*7,!!?3,"No payers have been selected. Please try again."  . Q  ; Q2 **;; JWS;IB\*2.0\*592 US1108 - Dental EDI 837D / form J430D**  S DIR(0)="SA^C:CMS-1500;U:UB-04;**D:J430D**;A:All",DIR("B")="All"  S DIR("A")="Run for (U)B-04, (C)MS-1500**, (D)Dental J430D** or (A)ll: "  W !!,"BILL FORM TYPE SELECTION:" D ^DIR K DIR  I X="^^" G ENQ  I $D(DTOUT)!$D(DUOUT) G Q1A  S IBFORM=Y  ; Q3 S DIR(0)="DA^0:9999999:EPX",DIR("A")="Start with Date "\_$S(IBLOC:"First Printed: ",1:"Last Transmitted: ")  ;S DIR("?",1)="This is the earliest date on which a batch that you want to include on this",DIR("?",2)=" report was last transmitted. You may choose a maximum date range of 90 days.",DIR("?")=" "  S DIR("?",1)="This is the earliest date on which a batch that you want to include on this",DIR("?",2)=" report was "\_$S(IBLOC=1:"first printed",1:"last transmitted")\_". You may choose a maximum date range of 90 days.",DIR("?")=" "  ;W !!,"LAST BATCH TRANSMIT DATE RANGE SELECTION:" D ^DIR K DIR  W !!,$S(IBLOC:"FIRST PRINT",1:"LAST BATCH TRANSMIT")\_" DATE RANGE SELECTION:" D ^DIR K DIR  I X="^^" G ENQ  I $D(DTOUT)!$D(DUOUT) G Q2  S IBDT1=Y  S IBDT2=$$FMADD^XLFDT(IBDT1,90) I IBDT2>DT S IBDT2=DT  S DIR("?",1)="This is the latest date on which a batch that you want to include on this",DIR("?",2)=" report was "\_$S(IBLOC:"first printed",1:"last transmitted")\_". You may choose a maximum date range of 90 days.",DIR("?")=" "  S DIR("B")=$$FMTE^XLFDT(IBDT2,2),DIR(0)="DA^"\_IBDT1\_":"\_IBDT2\_":EPX"  S DIR("A")="Go to Date "\_$S(IBLOC:"First Printed",1:"Last Transmitted")\_":("\_$$FMTE^XLFDT(IBDT1,2)\_"-"\_$$FMTE^XLFDT(IBDT2,2)\_"): " D ^DIR K DIR  I X="^^" G ENQ  I $D(DTOUT)!$D(DUOUT) G Q3  S IBDT2=Y  ; Q4 ; Additional selection criteria  S DIR(0)="SAO^1:MRA Secondary Only;2:Primary Claims Only;3:Secondary Claims Only;4:Claims Previously Printed at Clearinghouse"  S DIR("A",1)="ADDITIONAL SELECTION CRITERIA:",DIR("A",2)=" ",DIR("A",3)="1 - MRA Secondary Only",DIR("A",4)="2 - Primary Claims Only",DIR("A",5)="3 - Secondary Claims Only"  S DIR("A",6)=$S(IBLOC:"",1:"4 - Claims Sent to Print at Clearinghouse Only"),DIR("A",7)=" ",DIR("A")="Select Additional Limiting Criteria (optional): "  S DIR("?")="Select one of the listed criteria to further limit the claims to include"  W ! D ^DIR K DIR  I X="^^" G ENQ  I $D(DTOUT)!$D(DUOUT) G Q3  S IBCRIT=Y  ; Q41 ; Ask user if they want to include cancelled claims  S DIR(0)="Y",DIR("B")="No",DIR("A")="Would you like to include cancelled claims"  W ! D ^DIR K DIR  I X="^^" G ENQ  I $D(DIRUT) G Q4  S IBPTCCAN=Y  ; IB\*2.0\*547 skip next 2 questions if looking for locally printed claims  I IBLOC S IBSORT=2,IBRCBFPC=0 G Q6  ; Q42 ; Include claims that are forced to print at clearinghouse?  S DIR(0)="Y",DIR("B")="No",DIR("A")="Would you like to include claims Forced to Print at the Clearinghouse"  W ! D ^DIR K DIR  I X="^^" G ENQ  I $D(DIRUT) G Q41  S IBRCBFPC=Y  ; Q5 S DIR("L",1)="Select one of the following: ",DIR("L",2)=" ",DIR("L",3)=$J("",10)\_"1 Batch By Last Transmitted Date (Claims within a Batch)",DIR("L",4)=$J("",10)\_"2 Current Payer (Insurance Company)"  S DIR("L",5)=" "  S DIR(0)="SA^1:Batch By Last Transmitted Date (Claims within a Batch);2:Current Payer (Insurance Company)",DIR("B")="Current Payer"  S DIR("A")="Sort By: "  W ! D ^DIR K DIR  I X="^^" G ENQ  I $D(DTOUT)!$D(DUOUT) G Q42  S IBSORT=Y  ; Q6 S DIR(0)="SA^R:Report;S:Screen List"  S DIR("A")="Do you want a (R)eport or a (S)creen List format?: "  S DIR("B")="Screen List"  W ! D ^DIR K DIR  I X="^^" G ENQ  I $D(DTOUT)!$D(DUOUT) G Q5  S IBREP=Y  ; IB \*2.0\*547 call new SUB-routine for locally printed claims (not in file 364)  I IBREP="S",IBLOC D LOC^IBCEPTC0 G ENQ  ;  I IBREP="S",'IBLOC D LIST^IBCEPTC0 G ENQ  ; Q7 ; Select device  F  S IBACT=0 D DEVSEL(.IBACT) Q:IBACT  I IBACT=99 G ENQ  U IO  ; IB \*2.0\*547 call new SUB-routine for locally printed claims (not in file 364)  D:'IBLOC LIST^IBCEPTC0  D:IBLOC LOC^IBCEPTC0  ; ENQ K ^TMP("IB\_PREV\_CLAIM\_INS",$J),^TMP("IB\_PREV\_CLAIM\_SELECT",$J)  Q  ; DEVSEL(IBACT) ;  N DIR,POP,X,Y,ZTRTN,ZTSAVE  W !!,"You will need a 132 column printer for this report!"  S %ZIS="QM" D ^%ZIS I POP S IBACT=99 G DEVSELQ  I $G(IOM),IOM<132 S IBOK=1 D  I 'IBOK S IBACT=0 G DEVSELQ  . S DIR(0)="YA",DIR("A",1)="This report requires output to a 132 column device.",DIR("A",2)="The device you have chosen is only set for "\_IOM\_".",DIR("A")="Are you sure you want to continue?: ",DIR("B")="No"  . W ! D ^DIR K DIR  . I Y'=1 S IBOK=0 W !  I $D(IO("Q")) D  S IBACT=99 G DEVSELQ  . K IO("Q")  . S ZTRTN="LIST^IBCEPTC0",ZTSAVE("IBCRIT(")="",ZTSAVE("IB\*")="",ZTSAVE("^TMP(""IB\_PREV\_CLAIM\_INS"",$J)")="",ZTSAVE("^TMP(""IB\_PREV\_CLAIM\_INS"",$J,")="",ZTDESC="IB - Previously Transmitted Claims Report"  . D ^%ZTLOAD K ZTSK D HOME^%ZIS  S IBACT=1 DEVSELQ Q  ; SELDSP(IBHOW) ; Display list of selected claims/batches  ; IBHOW = "C" for claims "B" for batches  N Z,DIR,CT,QUIT  I '$O(^TMP($J,IBHOW,0)) Q  S (CT,QUIT)=0  W !!,$S(IBHOW="C":"Claims",1:"Batches")," Already Selected:"  S Z=0 F  S Z=$O(^TMP($J,IBHOW,Z)) Q:'Z!QUIT  S Z0=$G(^(Z)) D  Q:QUIT  . I IBHOW="C" W !,?3,$P($G(^DGCR(399,Z,0)),U) Q  . W !,?3,$P($G(^IBA(364.1,Z,0)),U)," ",$P(^(0),U,3)," Claims"  . S CT=CT+1  . I '(CT#10),$O(^TMP($J,IBHOW,Z)) S DIR("A")="Press return for more or '^' to exit ",DIR(0)="EA" W ! D ^DIR K DIR I $D(DTOUT)!$D(DUOUT) S QUIT=1  W !  Q  ; | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCEPTC0 | | | | | | | | |
| **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** | | | | | | | | | |
| IBCEPTC0 ;ALB/ESG - EDI PREVIOUSLY TRANSMITTED CLAIMS CONT ; 12/19/05  ;;2.0;INTEGRATED BILLING;\*\*320,348,547\*\*;21-MAR-94;Build 119  ;;Per VA Directive 6402, this routine should not be modified.  ;  Q  ; LIST ; Queued report format entrypoint  ; variables pre-defined: IBREP,IBSORT,IBFORM,IBDT1,IBDT2,  ; IBCRIT,IBPTCCAN,IBRCBFPC  ; ^TMP("IB\_PREV\_CLAIM\_INS,$J) global  K ^TMP("IB\_PREV\_CLAIM",$J)  N IBBDA,IBBDA0,IBCURI,IBDA,IBDT,IBFT,IBIFN,IBS1,IBS2,IBDTX  N INCLUDE,EDI,PROF,INST,IB0,IBZ1,DATA,IB364,CURSEQ,IBZ,IBZDAT  I IBREP="R" N IBPAGE,IBSTOP,IBHDRDT S (IBPAGE,IBSTOP)=0  ;  ; evaluate claim transmission data from files 364.1 and 364  S IBDT=IBDT1-.1  F  S IBDT=$O(^IBA(364.1,"ALT",IBDT)) Q:'IBDT!((IBDT\1)>IBDT2) S IBBDA=0 F  S IBBDA=$O(^IBA(364.1,"ALT",IBDT,IBBDA)) Q:'IBBDA  D  . S IBDTX=IBDT\1  . S IBDA=0 F  S IBDA=$O(^IBA(364,"C",IBBDA,IBDA)) Q:'IBDA  D  .. D STORE(IBDA,IBBDA,IBDTX,$P($G(^IBA(364,IBDA,0)),U,7)+1)  .. Q  . Q  ;  ; evaluate the test transmissions from file 361.4 (SRS 3.2.10.3)  S IBDT=IBDT1-.1  F  S IBDT=$O(^IBM(361.4,"ALT",IBDT)) Q:'IBDT!(IBDT>IBDT2) S IBIFN=0 F  S IBIFN=$O(^IBM(361.4,"ALT",IBDT,IBIFN)) Q:'IBIFN  S IBZ1=0 F  S IBZ1=$O(^IBM(361.4,IBIFN,1,IBZ1)) Q:'IBZ1  D  . S DATA=$G(^IBM(361.4,IBIFN,1,IBZ1,0)) Q:DATA=""  . S IBDTX=$P(DATA,U,1)\1 ; transmit date  . Q:IBDTX<IBDT1             ; too early  . Q:IBDTX>IBDT2             ; too late  . S IBBDA=+$P(DATA,U,2) ; batch ien  . Q:'IBBDA  . ;  . ; attempt to find the corresponding entry in file 364 for this one  . S IB364="",CURSEQ=$TR(+$P(DATA,U,4),"123","PST")  . S IBZ=" " F  S IBZ=$O(^IBA(364,"B",IBIFN,IBZ),-1) Q:'IBZ  D  Q:IB364  .. S IBZDAT=$G(^IBA(364,IBZ,0))  .. I $P(IBZDAT,U,8)'=CURSEQ Q      ; no match on payer sequence  .. I $F(".X.P.","."\_$P(IBZDAT,U,3)\_".") Q    ; transmission status must be farther than this  .. S IB364=IBZ Q  .. Q  . ;  . I 'IB364 Q      ; need to have an entry in file 364 to proceed  . ;  . D STORE(IB364,IBBDA,IBDTX,3)  . Q  ;  I IBREP="R" D RPT^IBCEPTC1(IBSORT,IBDT1,IBDT2) G END  ; Output report  ;  D EN^VALM("IBCE VIEW PREV TRANS"\_IBSORT) ; List Manager  ; END K ^TMP("IB\_PREV\_CLAIM",$J),^TMP("IB\_PREV\_CLAIM\_INS",$J)  Q  ; LOC ; new sub-routine for locally printed claims (use LIST & STORE tags as a guide)  ; Use the existing AP x-ref to narrow down the list of claims by date, then checks for existence in file 364 (EDI TRANSMIT BILL).  ; If a claim is NOT in file 364, it is a printed-only claim  ; variables pre-defined: IBREP,IBSORT,IBFORM,IBDT1,IBDT2,  ; IBCRIT,IBPTCCAN,IBRCBFPC  ; ^TMP("IB\_PREV\_CLAIM\_INS,$J) global  K ^TMP("IB\_PREV\_CLAIM",$J)  N IBBDA,IBBDA0,IBCURI,IBDA,IBDT,IBFT,IBIFN,IBS1,IBS2,IBDTX  N INCLUDE,EDI,PROF,INST,IB0,IBZ1,DATA,IB364,CURSEQ,IBZ,IBZDAT  I IBREP="R" N IBPAGE,IBSTOP,IBHDRDT S (IBPAGE,IBSTOP)=0  S IBDT=IBDT1-.1  F  S IBDT=$O(^DGCR(399,"AP",IBDT)) Q:'IBDT!(IBDT>IBDT2) S IBIFN=0 F  S IBIFN=$O(^DGCR(399,"AP",IBDT,IBIFN)) Q:'IBIFN  D  .; if it's in the transmit file it is not a printed claim  .Q:$D(^IBA(364,"B",IBIFN))  .S IB0=$G(^DGCR(399,IBIFN,0))  .S IBFT=$$FT^IBCEF(IBIFN) ; form type of claim  .I IBFORM'="B",$S(IBFT=3:IBFORM="C",IBFT=2:IBFORM="U",1:1) Q  .S IBCURI=$$CURR^IBCEF2(IBIFN) I 'IBCURI Q   ; current ins ien  .S EDI=$$UP^XLFSTR($G(^DIC(36,IBCURI,3))) ; 3 node EDI data  .; do not include claims where the ins.co. still cannot transmit electronically  .Q:+$P(EDI,U)=0  .S PROF=$P(EDI,U,2),INST=$P(EDI,U,4) ; payer IDs  .;  .; screen for user selected insurance companies/payers  .I +$G(^TMP("IB\_PREV\_CLAIM\_INS",$J)) D  I 'INCLUDE Q  ..S INCLUDE=0  ..I $D(^TMP("IB\_PREV\_CLAIM\_INS",$J,1,IBCURI)) S INCLUDE=1 Q  ..I '$D(^TMP("IB\_PREV\_CLAIM\_INS",$J,2)) Q  ..I PROF'="",$D(^TMP("IB\_PREV\_CLAIM\_INS",$J,2,PROF)) S INCLUDE=1 Q  ..I INST'="",$D(^TMP("IB\_PREV\_CLAIM\_INS",$J,2,INST)) S INCLUDE=1 Q  ..Q  .;  .I IBCRIT=1,'$$MRASEC^IBCEF4(IBIFN) Q  .I IBCRIT=2,($$COBN^IBCEF(IBIFN)>1) Q  .I IBCRIT=3,($$COBN^IBCEF(IBIFN)=1) Q  .I IBCRIT=4,'$P($G(^DGCR(399,IBIFN,"TX")),U,7) Q  .;  .; skip cancelled claims conditionally  .I $P(IB0,U,13)=7,'IBPTCCAN Q  .;  .S IBS1=$P($G(^DIC(36,+IBCURI,0)),U)\_U\_+IBCURI,IBS2=IBDT  .;  .; Meets all selection criteria - extract to sort global  .S:IBS1="" IBS1=" " S:IBS2="" IBS2=" "  .I '$D(^TMP("IB\_PREV\_CLAIM",$J,IBS1)) S ^TMP("IB\_PREV\_CLAIM",$J,IBS1)=IBIFN  .S ^TMP("IB\_PREV\_CLAIM",$J,IBS1,IBS2,IBIFN)=3 ; 3 = test transmission  ;  I IBREP="R" D RPT^IBCEPTC1(IBSORT,IBDT1,IBDT2) G END  ; Output report  ;  D EN^VALM("IBCE VIEW LOC PRINT") ; List Manager, new one for sort =2  ;  D END  Q  ; STORE(IB364,IBBDA,IBDTX,IBTYP) ; Check and store transmission data  ; Parameters  ; IB364 - ien to file 364 (claim transmission ien)  ; IBBDA - ien to file 364.1 (batch ien)  ; IBDTX - fm transmit date (no time) (either from 364.1 or 361.41)  ; IBTYP - 1 = transmission data from file 364 (field .07 is live)  ; 2 = transmission data from file 364 (field .07 is test)  ; 3 = transmission data from file 361.41 (test always)  ; Note:  ; Variables IBFORM, IBCRIT, IBPTCCAN, IBRCBFPC, and IBSORT are  ; assumed to exist here in this procedure.  ;  NEW IBIFN,IB0,IBFT,IBCURI,INCLUDE,EDI,PROF,INST,IBBDA0,IBS1,IBS2  ;  S IBIFN=+$G(^IBA(364,IB364,0))  S IB0=$G(^DGCR(399,IBIFN,0))  S IBFT=$$FT^IBCEF(IBIFN) ; form type of claim  I IBFORM'="B",$S(IBFT=3:IBFORM="C",IBFT=2:IBFORM="U",1:1) G STOREX  S IBCURI=$$CURR^IBCEF2(IBIFN) I 'IBCURI G STOREX   ; current ins ien  S EDI=$$UP^XLFSTR($G(^DIC(36,IBCURI,3))) ; 3 node EDI data  S PROF=$P(EDI,U,2),INST=$P(EDI,U,4) ; payer IDs  ;  ; screen for user selected insurance companies/payers  I +$G(^TMP("IB\_PREV\_CLAIM\_INS",$J)) D  I 'INCLUDE G STOREX  . S INCLUDE=0  . I $D(^TMP("IB\_PREV\_CLAIM\_INS",$J,1,IBCURI)) S INCLUDE=1 Q  . I '$D(^TMP("IB\_PREV\_CLAIM\_INS",$J,2)) Q  . I PROF'="",$D(^TMP("IB\_PREV\_CLAIM\_INS",$J,2,PROF)) S INCLUDE=1 Q  . I INST'="",$D(^TMP("IB\_PREV\_CLAIM\_INS",$J,2,INST)) S INCLUDE=1 Q  . Q  ;  I IBCRIT=1,'$$MRASEC^IBCEF4(IBIFN) G STOREX  I IBCRIT=2,($$COBN^IBCEF(IBIFN)>1) G STOREX  I IBCRIT=3,($$COBN^IBCEF(IBIFN)=1) G STOREX  I IBCRIT=4,'$P($G(^DGCR(399,IBIFN,"TX")),U,7) G STOREX  ;  ; skip cancelled claims conditionally  I $P(IB0,U,13)=7,'IBPTCCAN G STOREX  ;  ; skip claims forced to print at clearinghouse (claim check)  I $P($G(^DGCR(399,IBIFN,"TX")),U,8)=2,'IBRCBFPC G STOREX  ;  ; skip claims forced to print at clearinghouse (payer check)  I IBFT=2,PROF["PRNT",'IBRCBFPC G STOREX    ; 1500, prof payer ID  I IBFT=3,INST["PRNT",'IBRCBFPC G STOREX    ; ub, inst payer ID  ;  S IBBDA0=$G(^IBA(364.1,+IBBDA,0)) ; 0 node of batch  ;  S IBS1=$S(IBSORT=1:(99999999-IBDTX)\_U\_$P(IBBDA0,U)\_U\_$P(IBBDA0,U,14)\_U\_+$P(IBBDA0,U,5),1:$P($G(^DIC(36,+IBCURI,0)),U)\_U\_+IBCURI)  S IBS2=$S(IBSORT=1:$P(IB0,U,1),1:99999999-IBDTX)  ;  ; Meets all selection criteria - extract to sort global  S:IBS1="" IBS1=" " S:IBS2="" IBS2=" "  I '$D(^TMP("IB\_PREV\_CLAIM",$J,IBS1)) S ^TMP("IB\_PREV\_CLAIM",$J,IBS1)=$S(IBSORT=1:$$FMTE^XLFDT(IBDTX,"1"),1:IBIFN)  S ^TMP("IB\_PREV\_CLAIM",$J,IBS1,IBS2,IB364)=IBTYP  ; STOREX ;  Q  ; | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCEPTC0 ;ALB/ESG - EDI PREVIOUSLY TRANSMITTED CLAIMS CONT ; 12/19/05  ;;2.0;INTEGRATED BILLING;\*\*320,348,547,**592**\*\*;21-MAR-94;Build 119  ;;Per VA Directive 6402, this routine should not be modified.  ;  Q  ; LIST ; Queued report format entrypoint  ; variables pre-defined: IBREP,IBSORT,IBFORM,IBDT1,IBDT2,  ; IBCRIT,IBPTCCAN,IBRCBFPC  ; ^TMP("IB\_PREV\_CLAIM\_INS,$J) global  K ^TMP("IB\_PREV\_CLAIM",$J)  N IBBDA,IBBDA0,IBCURI,IBDA,IBDT,IBFT,IBIFN,IBS1,IBS2,IBDTX  N INCLUDE,EDI,PROF,INST,IB0,IBZ1,DATA,IB364,CURSEQ,IBZ,IBZDAT  I IBREP="R" N IBPAGE,IBSTOP,IBHDRDT S (IBPAGE,IBSTOP)=0  ;  ; evaluate claim transmission data from files 364.1 and 364  S IBDT=IBDT1-.1  F  S IBDT=$O(^IBA(364.1,"ALT",IBDT)) Q:'IBDT!((IBDT\1)>IBDT2) S IBBDA=0 F  S IBBDA=$O(^IBA(364.1,"ALT",IBDT,IBBDA)) Q:'IBBDA  D  . S IBDTX=IBDT\1  . S IBDA=0 F  S IBDA=$O(^IBA(364,"C",IBBDA,IBDA)) Q:'IBDA  D  .. D STORE(IBDA,IBBDA,IBDTX,$P($G(^IBA(364,IBDA,0)),U,7)+1)  .. Q  . Q  ;  ; evaluate the test transmissions from file 361.4 (SRS 3.2.10.3)  S IBDT=IBDT1-.1  F  S IBDT=$O(^IBM(361.4,"ALT",IBDT)) Q:'IBDT!(IBDT>IBDT2) S IBIFN=0 F  S IBIFN=$O(^IBM(361.4,"ALT",IBDT,IBIFN)) Q:'IBIFN  S IBZ1=0 F  S IBZ1=$O(^IBM(361.4,IBIFN,1,IBZ1)) Q:'IBZ1  D  . S DATA=$G(^IBM(361.4,IBIFN,1,IBZ1,0)) Q:DATA=""  . S IBDTX=$P(DATA,U,1)\1 ; transmit date  . Q:IBDTX<IBDT1             ; too early  . Q:IBDTX>IBDT2             ; too late  . S IBBDA=+$P(DATA,U,2) ; batch ien  . Q:'IBBDA  . ;  . ; attempt to find the corresponding entry in file 364 for this one  . S IB364="",CURSEQ=$TR(+$P(DATA,U,4),"123","PST")  . S IBZ=" " F  S IBZ=$O(^IBA(364,"B",IBIFN,IBZ),-1) Q:'IBZ  D  Q:IB364  .. S IBZDAT=$G(^IBA(364,IBZ,0))  .. I $P(IBZDAT,U,8)'=CURSEQ Q      ; no match on payer sequence  .. I $F(".X.P.","."\_$P(IBZDAT,U,3)\_".") Q    ; transmission status must be farther than this  .. S IB364=IBZ Q  .. Q  . ;  . I 'IB364 Q      ; need to have an entry in file 364 to proceed  . ;  . D STORE(IB364,IBBDA,IBDTX,3)  . Q  ;  I IBREP="R" D RPT^IBCEPTC1(IBSORT,IBDT1,IBDT2) G END  ; Output report  ;  D EN^VALM("IBCE VIEW PREV TRANS"\_IBSORT) ; List Manager  ; END K ^TMP("IB\_PREV\_CLAIM",$J),^TMP("IB\_PREV\_CLAIM\_INS",$J)  Q  ; LOC ; new sub-routine for locally printed claims (use LIST & STORE tags as a guide)  ; Use the existing AP x-ref to narrow down the list of claims by date, then checks for existence in file 364 (EDI TRANSMIT BILL).  ; If a claim is NOT in file 364, it is a printed-only claim  ; variables pre-defined: IBREP,IBSORT,IBFORM,IBDT1,IBDT2,  ; IBCRIT,IBPTCCAN,IBRCBFPC  ; ^TMP("IB\_PREV\_CLAIM\_INS,$J) global  K ^TMP("IB\_PREV\_CLAIM",$J)  N IBBDA,IBBDA0,IBCURI,IBDA,IBDT,IBFT,IBIFN,IBS1,IBS2,IBDTX  N INCLUDE,EDI,PROF,INST,IB0,IBZ1,DATA,IB364,CURSEQ,IBZ,IBZDAT  I IBREP="R" N IBPAGE,IBSTOP,IBHDRDT S (IBPAGE,IBSTOP)=0  S IBDT=IBDT1-.1  F  S IBDT=$O(^DGCR(399,"AP",IBDT)) Q:'IBDT!(IBDT>IBDT2) S IBIFN=0 F  S IBIFN=$O(^DGCR(399,"AP",IBDT,IBIFN)) Q:'IBIFN  D  .; if it's in the transmit file it is not a printed claim  .Q:$D(^IBA(364,"B",IBIFN))  .S IB0=$G(^DGCR(399,IBIFN,0))  .S IBFT=$$FT^IBCEF(IBIFN) ; form type of claim **.;JWS;IB\*2.0\*592 US1108 - Dental EDI 837D / form J430D**  .I IBFORM'="A",$S(IBFT=3:IBFORM='"U",IBFT=2:IBFORM'="C",**IBFT=7:IBFORM'="D",**1:1) Q  .S IBCURI=$$CURR^IBCEF2(IBIFN) I 'IBCURI Q   ; current ins ien  .S EDI=$$UP^XLFSTR($G(^DIC(36,IBCURI,3))) ; 3 node EDI data  .; do not include claims where the ins.co. still cannot transmit electronically  .Q:+$P(EDI,U)=0  .S PROF=$P(EDI,U,2),INST=$P(EDI,U,4) ; payer IDs  .;  .; screen for user selected insurance companies/payers  .I +$G(^TMP("IB\_PREV\_CLAIM\_INS",$J)) D  I 'INCLUDE Q  ..S INCLUDE=0  ..I $D(^TMP("IB\_PREV\_CLAIM\_INS",$J,1,IBCURI)) S INCLUDE=1 Q  ..I '$D(^TMP("IB\_PREV\_CLAIM\_INS",$J,2)) Q  ..I PROF'="",$D(^TMP("IB\_PREV\_CLAIM\_INS",$J,2,PROF)) S INCLUDE=1 Q  ..I INST'="",$D(^TMP("IB\_PREV\_CLAIM\_INS",$J,2,INST)) S INCLUDE=1 Q  ..Q  .;  .I IBCRIT=1,'$$MRASEC^IBCEF4(IBIFN) Q  .I IBCRIT=2,($$COBN^IBCEF(IBIFN)>1) Q  .I IBCRIT=3,($$COBN^IBCEF(IBIFN)=1) Q  .I IBCRIT=4,'$P($G(^DGCR(399,IBIFN,"TX")),U,7) Q  .;  .; skip cancelled claims conditionally  .I $P(IB0,U,13)=7,'IBPTCCAN Q  .;  .S IBS1=$P($G(^DIC(36,+IBCURI,0)),U)\_U\_+IBCURI,IBS2=IBDT  .;  .; Meets all selection criteria - extract to sort global  .S:IBS1="" IBS1=" " S:IBS2="" IBS2=" "  .I '$D(^TMP("IB\_PREV\_CLAIM",$J,IBS1)) S ^TMP("IB\_PREV\_CLAIM",$J,IBS1)=IBIFN  .S ^TMP("IB\_PREV\_CLAIM",$J,IBS1,IBS2,IBIFN)=3 ; 3 = test transmission  ;  I IBREP="R" D RPT^IBCEPTC1(IBSORT,IBDT1,IBDT2) G END  ; Output report  ;  D EN^VALM("IBCE VIEW LOC PRINT") ; List Manager, new one for sort =2  ;  D END  Q  ; STORE(IB364,IBBDA,IBDTX,IBTYP) ; Check and store transmission data  ; Parameters  ; IB364 - ien to file 364 (claim transmission ien)  ; IBBDA - ien to file 364.1 (batch ien)  ; IBDTX - fm transmit date (no time) (either from 364.1 or 361.41)  ; IBTYP - 1 = transmission data from file 364 (field .07 is live)  ; 2 = transmission data from file 364 (field .07 is test)  ; 3 = transmission data from file 361.41 (test always)  ; Note:  ; Variables IBFORM, IBCRIT, IBPTCCAN, IBRCBFPC, and IBSORT are  ; assumed to exist here in this procedure.  ;  NEW IBIFN,IB0,IBFT,IBCURI,INCLUDE,EDI,PROF,INST,IBBDA0,IBS1,IBS2  ;  S IBIFN=+$G(^IBA(364,IB364,0))  S IB0=$G(^DGCR(399,IBIFN,0))  S IBFT=$$FT^IBCEF(IBIFN) ; form type of claim **;JWS;IB\*2.0\*592 US1108 - Dental EDI 837D / form J430D**  I IBFORM'="A",$S(IBFT=3:IBFORM'="U",IBFT=2:IBFORM'="C",**IBFT=7:IBFORM'="D",**1:1) G STOREX  S IBCURI=$$CURR^IBCEF2(IBIFN) I 'IBCURI G STOREX   ; current ins ien  S EDI=$$UP^XLFSTR($G(^DIC(36,IBCURI,3))) ; 3 node EDI data  S PROF=$P(EDI,U,2),INST=$P(EDI,U,4) ; payer IDs  ;  ; screen for user selected insurance companies/payers  I +$G(^TMP("IB\_PREV\_CLAIM\_INS",$J)) D  I 'INCLUDE G STOREX  . S INCLUDE=0  . I $D(^TMP("IB\_PREV\_CLAIM\_INS",$J,1,IBCURI)) S INCLUDE=1 Q  . I '$D(^TMP("IB\_PREV\_CLAIM\_INS",$J,2)) Q  . I PROF'="",$D(^TMP("IB\_PREV\_CLAIM\_INS",$J,2,PROF)) S INCLUDE=1 Q  . I INST'="",$D(^TMP("IB\_PREV\_CLAIM\_INS",$J,2,INST)) S INCLUDE=1 Q  . Q  ;  I IBCRIT=1,'$$MRASEC^IBCEF4(IBIFN) G STOREX  I IBCRIT=2,($$COBN^IBCEF(IBIFN)>1) G STOREX  I IBCRIT=3,($$COBN^IBCEF(IBIFN)=1) G STOREX  I IBCRIT=4,'$P($G(^DGCR(399,IBIFN,"TX")),U,7) G STOREX  ;  ; skip cancelled claims conditionally  I $P(IB0,U,13)=7,'IBPTCCAN G STOREX  ;  ; skip claims forced to print at clearinghouse (claim check)  I $P($G(^DGCR(399,IBIFN,"TX")),U,8)=2,'IBRCBFPC G STOREX  ;  ; skip claims forced to print at clearinghouse (payer check)  I IBFT=2,PROF["PRNT",'IBRCBFPC G STOREX    ; 1500, prof payer ID  I IBFT=3,INST["PRNT",'IBRCBFPC G STOREX    ; ub, inst payer ID  ;  S IBBDA0=$G(^IBA(364.1,+IBBDA,0)) ; 0 node of batch  ;  S IBS1=$S(IBSORT=1:(99999999-IBDTX)\_U\_$P(IBBDA0,U)\_U\_$P(IBBDA0,U,14)\_U\_+$P(IBBDA0,U,5),1:$P($G(^DIC(36,+IBCURI,0)),U)\_U\_+IBCURI)  S IBS2=$S(IBSORT=1:$P(IB0,U,1),1:99999999-IBDTX)  ;  ; Meets all selection criteria - extract to sort global  S:IBS1="" IBS1=" " S:IBS2="" IBS2=" "  I '$D(^TMP("IB\_PREV\_CLAIM",$J,IBS1)) S ^TMP("IB\_PREV\_CLAIM",$J,IBS1)=$S(IBSORT=1:$$FMTE^XLFDT(IBDTX,"1"),1:IBIFN)  S ^TMP("IB\_PREV\_CLAIM",$J,IBS1,IBS2,IB364)=IBTYP  ; STOREX ;  Q  ; | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCEPTC2 | | | | | | | | |
| **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** | | | | | | | | | |
| IBCEPTC2 ;ALB/TMK - EDI PREVIOUSLY TRANSMITTED CLAIMS LIST MGR ;01/20/05  ;;2.0;INTEGRATED BILLING;\*\*296,320,348,349,547\*\*;21-MAR-94;Build 119  ;;Per VA Directive 6402, this routine should not be modified.  ; IA 3337 for file 430.3  ; IB\*2.0\*547 Variable IBLOC is pre-defined (in IBCEPTC)  ; HDR ;  K VALMHDR  I IBLOC S VALMHDR(1)="Claims Selected: "\_+$G(^TMP("IB\_PREV\_CLAIM\_SELECT",$J))\_" (marked with \*)" Q  S VALMHDR(1)="\*\* A claim may appear multiple times if transmitted more than once. \*\*"  ;  I $G(IBSORT)=1 D  . S VALMHDR(2)="Claims Selected: "\_+$G(^TMP("IB\_PREV\_CLAIM\_SELECT",$J))\_" (marked with \*)"  . Q  ;  I $G(IBSORT)=2 D  . S VALMHDR(2)="\*\* T = Test Claim \*\* R = Batch Rejected"  . S VALMHDR(3)="Claims Selected: "\_+$G(^TMP("IB\_PREV\_CLAIM\_SELECT",$J))\_" (marked with \*)"  . Q  ;  Q  ; INIT ;  S VALMCNT=0,VALMBG=1  D BLD  Q  ; BLD ; Build display lines  N IBDA,IBS1,IBS2,IBIFN,IB0,IBX,IBCNT,IBLEV1,IBBDA  K ^TMP("IB\_PREV\_CLAIM\_LIST",$J),^TMP("IB\_PREV\_CLAIM\_SELECT",$J),^TMP("IB\_PREV\_CLAIM\_BATCH",$J)  S IBCNT=0  I $O(^TMP("IB\_PREV\_CLAIM",$J,""))="" D  G BLDQ  . S IBX=" \*\* NO PREVIOUSLY "\_$S(IBLOC:"PRINTED",1:"TRANSMITTED")\_" CLAIMS EXIST FOR SEARCH CRITERIA SELECTED \*\*"  . D WRT(IBX,"",0,0,"","S","",.IBCNT,0)  ;  S IBS1="" F  S IBS1=$O(^TMP("IB\_PREV\_CLAIM",$J,IBS1)) Q:IBS1=""  D  . ; First level sort  . ; for sort by batch, display batch ID and transmit date  . I IBSORT=1 D  .. S IBLEV1=" Batch: "\_$P(IBS1,U,2)\_" Last Transmitted: "\_$G(^TMP("IB\_PREV\_CLAIM",$J,IBS1))  .. S IBBDA=+$O(^IBA(364.1,"B",$P(IBS1,U,2),0))  .. I $P(IBS1,U,3) S IBLEV1=IBLEV1\_" \*\* Test"  .. I $P(IBS1,U,4) S IBLEV1=IBLEV1\_" \*\* Rejected"  .. Q  . ;  . ; for sort by payer, display ins co name and payer address  . I IBSORT=2 D  .. S IBLEV1=" "\_$P(IBS1,U)\_" "\_$$CURRINS(+$G(^TMP("IB\_PREV\_CLAIM",$J,IBS1)),0)  .. Q  . ;  . ; output sort header line  . D WRT(IBLEV1,"",0,0,IBSORT,"S","",IBCNT,0) ; Add header line  . ;  . I IBSORT=1,IBBDA S ^TMP("IB\_PREV\_CLAIM\_BATCH",$J,IBBDA)=VALMCNT  . S IBS2="" F  S IBS2=$O(^TMP("IB\_PREV\_CLAIM",$J,IBS1,IBS2)) Q:IBS2=""  S IBDA=0 F  S IBDA=$O(^TMP("IB\_PREV\_CLAIM",$J,IBS1,IBS2,IBDA)) Q:'IBDA  D  .. N IBX,IBTEST  .. ;S IBIFN=+$G(^IBA(364,+IBDA,0)),IB0=$G(^DGCR(399,IBIFN,0))  .. S IBIFN=$S(IBLOC:+IBDA,1:+$G(^IBA(364,+IBDA,0))),IB0=$G(^DGCR(399,IBIFN,0))  .. S IBX=$P(^TMP("IB\_PREV\_CLAIM",$J,IBS1,IBS2,IBDA),U,1)  .. I IBX=1 S IBTEST=0 ; live 364 transmission  .. I IBX=2 S IBTEST=1 ; test 364 transmission  .. I IBX=3 S IBTEST=1 ; test 361.4 transmission  .. D WRT(IBS1,IBS2,IBDA,IBIFN,IBSORT,"S","",.IBCNT,0,IBTEST)  .. I IBSORT=1,IBBDA S ^TMP("IB\_PREV\_CLAIM\_BATCH",$J,IBBDA,VALMCNT)=IBIFN\_U\_IBCNT  .. Q  . Q  ; BLDQ Q  ; EXIT ; Clean up code  ;  K ^TMP("IB\_PREV\_CLAIM\_LIST",$J)  K ^TMP("IB\_PREV\_CLAIM\_SELECT",$J)  K ^TMP("IB\_PREV\_CLAIM\_LIST\_DX",$J)  K ^TMP("IB\_PREV\_CLAIM\_BATCH",$J)  D CLEAR^VALM1  Q  ; WRT(IBS1,IBS2,IBDA,IBIFN,IBSORT,IBREP,IBHDR,IBPAGE,IBSTOP,IBTEST) ; Wrt/output  ;  N IBX,IB0,Z,IBCNT,ARSTAT  S IBCNT=IBPAGE  ;  I 'IBIFN D  G WRTQ  . ;  . ; for report output  . I IBREP="R" D  Q  .. S Z="",$P(Z,"=",133)=""  .. D SET(Z,1,IBDA,IBREP,IBHDR,1,0,.IBPAGE,.IBSTOP)  .. D SET(IBS1,2,IBDA,IBREP,IBHDR,1,0,.IBPAGE,.IBSTOP)  .. Q  . ;  . ; for ListMan screen output  . D SET(IBS1,0,IBDA,IBREP,IBHDR,IBCNT+1,.VALMCNT,.IBPAGE,.IBSTOP)  . Q  ;  S IB0=$G(^DGCR(399,IBIFN,0))  S IBX=$$FO^IBCNEUT1($P(IB0,U,1),8) ; claim#  S IBX=IBX\_$S(IBSORT=2&$G(IBTEST):"T",1:" ")\_" "  S IBX=IBX\_$S($P(IB0,U,19)=2:"1500",1:"UB04")\_" "  S Z=$$INPAT^IBCEF(IBIFN) S IBX=IBX\_$S(Z:"INPT ",1:"OUTPT")  S IBX=IBX\_$J($P(IB0,U,21),3)\_" "  S Z=$$EXTERNAL^DILFD(399,.13,"",$P(IB0,U,13))  S IBX=IBX\_$$FO^IBCNEUT1(Z,11)\_" "             ; claim status  S ARSTAT=+$P($$BILL^RCJIBFN2(IBIFN),U,2) ; ien  S ARSTAT=$P($G(^PRCA(430.3,ARSTAT,0)),U,2) ; abbreviation  S IBX=IBX\_$$FO^IBCNEUT1(ARSTAT,4) ; a/r status display  ;  I IBSORT=1 D                    ; sort by batch  . N Z,IBZ,IBXDATA  . ; Print current payer, payer address, other payers, pat name  . D F^IBCEF("N-CURR INSURANCE COMPANY NAME","IBZ",,IBIFN)  . S IBX=IBX\_$$FO^IBCNEUT1(IBZ,25)\_" "                     ; ins co name  . S IBX=IBX\_$$FO^IBCNEUT1($$CURRINS(IBIFN,1),29)\_" "      ; address  . K IBZ D F^IBCEF("N-OTH INSURANCE CO. NAME","IBZ",,IBIFN)  . S IBX=IBX\_$$FO^IBCNEUT1($P($G(IBZ(1)),U,1),15)\_" "      ; other payer  . S Z=$P($G(^DPT(+$P(IB0,U,2),0)),U,1)  . S IBX=IBX\_$E(Z,1,18) ; patient name  . ;  . ; set line into list  . S IBCNT=IBCNT+1  . D SET(.IBX,1,IBDA,IBREP,IBHDR,IBCNT,.VALMCNT,.IBPAGE,.IBSTOP)  . S IBX=""  . ;  . I $G(IBZ(2))'="" D    ; other payer #2 if it exists  .. S IBX=$J("",98)\_$E($P(IBZ(2),U,1),1,15)  .. D SET(.IBX,1,IBDA,IBREP,IBHDR,IBCNT,.VALMCNT,.IBPAGE,.IBSTOP)  .. Q  . Q  ;  I IBSORT=2 D                    ; sort by payer  . N Z,IBZ  . S IBX=IBX\_" "  . ; Print other payers, patient name, date last trans, batch #, reject flag  . D F^IBCEF("N-OTH INSURANCE CO. NAME","IBZ",,IBIFN)  . S IBX=IBX\_$$FO^IBCNEUT1($P($G(IBZ(1)),U,1),18)\_" "   ; oth payer#1  . S Z=$P($G(^DPT(+$P(IB0,U,2),0)),U,1)  . S IBX=IBX\_$$FO^IBCNEUT1(Z,18)\_" "                  ; patient name  . ;  . S Z=+$P($G(^IBA(364,+IBDA,0)),U,2) ; Batch ptr  . S:IBLOC IBX=IBX\_$$FO^IBCNEUT1($$FMTE^XLFDT($P($G(^DGCR(399,IBIFN,"S")),U,14),"1"),17) ; date last printed \*547\*  . S:'IBLOC IBX=IBX\_$$FO^IBCNEUT1($$FMTE^XLFDT($P($G(^IBA(364.1,+Z,1)),U,3)\1,"1"),17) ; date last transmitted  . S:'IBLOC IBX=IBX\_$$FO^IBCNEUT1($P($G(^IBA(364.1,Z,0)),U,1),10) ; batch#  . S:IBLOC IBX=IBX\_"" ; no batch#  . S IBX=IBX\_$S($P($G(^IBA(364.1,Z,0)),U,5):" R",1:"") ; batch rejected flag  . ;  . ; set line into list  . S IBCNT=IBCNT+1  . D SET(.IBX,1,IBDA,IBREP,IBHDR,IBCNT,.VALMCNT,.IBPAGE,.IBSTOP)  . S IBX=""  . ;  . I $G(IBZ(2))'="" D       ; other payer#2 if it exists  .. S IBX=$J("",44)\_$E($P(IBZ(2),U),1,18)  .. D SET(.IBX,1,IBDA,IBREP,IBHDR,IBCNT,.VALMCNT,.IBPAGE,.IBSTOP)  .. Q  . Q  ; WRTQ I IBREP="S" S IBPAGE=IBCNT  Q  ; SET(IBX,IBLINE,IBDA,IBREP,IBHDR,IBCNT,VALMCNT,IBPAGE,IBSTOP) ;  N Q,Z,IBZ  S IBZ=IBX,IBX=""  I IBREP="R" D  Q  . D:($Y+5)>IOSL!'IBPAGE HDR^IBCEPTC1(IBHDR,IBSORT,.IBPAGE,.IBSTOP) D  . I IBLINE F Z=1:1:IBLINE W !  . W:'IBSTOP IBZ  . Q  ;  ; only display the counter if we have a line with the claim#  S VALMCNT=VALMCNT+1  I IBDA,$TR($E(IBZ,1,8)," ")'="" S IBZ=$$FO^IBCNEUT1($J(IBCNT,3),6)\_IBZ  I IBDA,$TR($E(IBZ,1,8)," ")="" S IBZ=" "\_IBZ  ;  S ^TMP("IB\_PREV\_CLAIM\_LIST",$J,VALMCNT,0)=IBZ  S ^TMP("IB\_PREV\_CLAIM\_LIST",$J,"IDX",VALMCNT,IBCNT)=""  I IBDA,$TR($E(IBZ,1,8)," ")'="" S ^TMP("IB\_PREV\_CLAIM\_LIST\_DX",$J,IBCNT)=VALMCNT\_U\_IBDA  Q  ; CURRINS(IBIFN,TRUNC) ; Returns Current insurance address for given claim  ; TRUNC = truncate flag; 1 to truncate the address and city  N IBX,IBZ,L1,CITY,ST  D F^IBCEF("N-CURR INS CO FULL ADDRESS","IBZ",,IBIFN)  S L1=$G(IBZ(1)) I +$G(TRUNC) S L1=$E(L1,1,15)  S CITY=$G(IBZ(4)) I +$G(TRUNC) S CITY=$E(CITY,1,10)  S ST=$G(IBZ(5))  I ST S ST=$P($G(^DIC(5,ST,0)),U,2)  S IBX=L1\_" "\_CITY  I CITY'="",ST'="" S IBX=IBX\_","\_ST  E  S IBX=IBX\_" "\_ST  Q IBX  ; | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCEPTC2 ;ALB/TMK - EDI PREVIOUSLY TRANSMITTED CLAIMS LIST MGR ;01/20/05  ;;2.0;INTEGRATED BILLING;\*\*296,320,348,349,547,**592**\*\*;21-MAR-94;Build 119  ;;Per VA Directive 6402, this routine should not be modified.  ; IA 3337 for file 430.3  ; IB\*2.0\*547 Variable IBLOC is pre-defined (in IBCEPTC)  ; HDR ;  K VALMHDR  I IBLOC S VALMHDR(1)="Claims Selected: "\_+$G(^TMP("IB\_PREV\_CLAIM\_SELECT",$J))\_" (marked with \*)" Q  S VALMHDR(1)="\*\* A claim may appear multiple times if transmitted more than once. \*\*"  ;  I $G(IBSORT)=1 D  . S VALMHDR(2)="Claims Selected: "\_+$G(^TMP("IB\_PREV\_CLAIM\_SELECT",$J))\_" (marked with \*)"  . Q  ;  I $G(IBSORT)=2 D  . S VALMHDR(2)="\*\* T = Test Claim \*\* R = Batch Rejected"  . S VALMHDR(3)="Claims Selected: "\_+$G(^TMP("IB\_PREV\_CLAIM\_SELECT",$J))\_" (marked with \*)"  . Q  ;  Q  ; INIT ;  S VALMCNT=0,VALMBG=1  D BLD  Q  ; BLD ; Build display lines  N IBDA,IBS1,IBS2,IBIFN,IB0,IBX,IBCNT,IBLEV1,IBBDA  K ^TMP("IB\_PREV\_CLAIM\_LIST",$J),^TMP("IB\_PREV\_CLAIM\_SELECT",$J),^TMP("IB\_PREV\_CLAIM\_BATCH",$J)  S IBCNT=0  I $O(^TMP("IB\_PREV\_CLAIM",$J,""))="" D  G BLDQ  . S IBX=" \*\* NO PREVIOUSLY "\_$S(IBLOC:"PRINTED",1:"TRANSMITTED")\_" CLAIMS EXIST FOR SEARCH CRITERIA SELECTED \*\*"  . D WRT(IBX,"",0,0,"","S","",.IBCNT,0)  ;  S IBS1="" F  S IBS1=$O(^TMP("IB\_PREV\_CLAIM",$J,IBS1)) Q:IBS1=""  D  . ; First level sort  . ; for sort by batch, display batch ID and transmit date  . I IBSORT=1 D  .. S IBLEV1=" Batch: "\_$P(IBS1,U,2)\_" Last Transmitted: "\_$G(^TMP("IB\_PREV\_CLAIM",$J,IBS1))  .. S IBBDA=+$O(^IBA(364.1,"B",$P(IBS1,U,2),0))  .. I $P(IBS1,U,3) S IBLEV1=IBLEV1\_" \*\* Test"  .. I $P(IBS1,U,4) S IBLEV1=IBLEV1\_" \*\* Rejected"  .. Q  . ;  . ; for sort by payer, display ins co name and payer address  . I IBSORT=2 D  .. S IBLEV1=" "\_$P(IBS1,U)\_" "\_$$CURRINS(+$G(^TMP("IB\_PREV\_CLAIM",$J,IBS1)),0)  .. Q  . ;  . ; output sort header line  . D WRT(IBLEV1,"",0,0,IBSORT,"S","",IBCNT,0) ; Add header line  . ;  . I IBSORT=1,IBBDA S ^TMP("IB\_PREV\_CLAIM\_BATCH",$J,IBBDA)=VALMCNT  . S IBS2="" F  S IBS2=$O(^TMP("IB\_PREV\_CLAIM",$J,IBS1,IBS2)) Q:IBS2=""  S IBDA=0 F  S IBDA=$O(^TMP("IB\_PREV\_CLAIM",$J,IBS1,IBS2,IBDA)) Q:'IBDA  D  .. N IBX,IBTEST  .. ;S IBIFN=+$G(^IBA(364,+IBDA,0)),IB0=$G(^DGCR(399,IBIFN,0))  .. S IBIFN=$S(IBLOC:+IBDA,1:+$G(^IBA(364,+IBDA,0))),IB0=$G(^DGCR(399,IBIFN,0))  .. S IBX=$P(^TMP("IB\_PREV\_CLAIM",$J,IBS1,IBS2,IBDA),U,1)  .. I IBX=1 S IBTEST=0 ; live 364 transmission  .. I IBX=2 S IBTEST=1 ; test 364 transmission  .. I IBX=3 S IBTEST=1 ; test 361.4 transmission  .. D WRT(IBS1,IBS2,IBDA,IBIFN,IBSORT,"S","",.IBCNT,0,IBTEST)  .. I IBSORT=1,IBBDA S ^TMP("IB\_PREV\_CLAIM\_BATCH",$J,IBBDA,VALMCNT)=IBIFN\_U\_IBCNT  .. Q  . Q  ; BLDQ Q  ; EXIT ; Clean up code  ;  K ^TMP("IB\_PREV\_CLAIM\_LIST",$J)  K ^TMP("IB\_PREV\_CLAIM\_SELECT",$J)  K ^TMP("IB\_PREV\_CLAIM\_LIST\_DX",$J)  K ^TMP("IB\_PREV\_CLAIM\_BATCH",$J)  D CLEAR^VALM1  Q  ; WRT(IBS1,IBS2,IBDA,IBIFN,IBSORT,IBREP,IBHDR,IBPAGE,IBSTOP,IBTEST) ; Wrt/output  ;  N IBX,IB0,Z,IBCNT,ARSTAT  S IBCNT=IBPAGE  ;  I 'IBIFN D  G WRTQ  . ;  . ; for report output  . I IBREP="R" D  Q  .. S Z="",$P(Z,"=",133)=""  .. D SET(Z,1,IBDA,IBREP,IBHDR,1,0,.IBPAGE,.IBSTOP)  .. D SET(IBS1,2,IBDA,IBREP,IBHDR,1,0,.IBPAGE,.IBSTOP)  .. Q  . ;  . ; for ListMan screen output  . D SET(IBS1,0,IBDA,IBREP,IBHDR,IBCNT+1,.VALMCNT,.IBPAGE,.IBSTOP)  . Q  ;  S IB0=$G(^DGCR(399,IBIFN,0))  S IBX=$$FO^IBCNEUT1($P(IB0,U,1),8) ; claim#  S IBX=IBX\_$S(IBSORT=2&$G(IBTEST):"T",1:" ")\_" "  **;JWS;IB\*2.0\*592 US1108 - Dental EDI 837D / form J430D**  S IBX=IBX\_$S($P(IB0,U,19)=2:"1500**",$P(IB0,U,19)=7:"J430D",**1:"UB04")\_" "  S Z=$$INPAT^IBCEF(IBIFN) S IBX=IBX\_$S(Z:"INPT ",1:"OUTPT")  S IBX=IBX\_$J($P(IB0,U,21),3)\_" "  S Z=$$EXTERNAL^DILFD(399,.13,"",$P(IB0,U,13))  S IBX=IBX\_$$FO^IBCNEUT1(Z,11)\_" "             ; claim status  S ARSTAT=+$P($$BILL^RCJIBFN2(IBIFN),U,2) ; ien  S ARSTAT=$P($G(^PRCA(430.3,ARSTAT,0)),U,2) ; abbreviation  S IBX=IBX\_$$FO^IBCNEUT1(ARSTAT,4) ; a/r status display  ;  I IBSORT=1 D                    ; sort by batch  . N Z,IBZ,IBXDATA  . ; Print current payer, payer address, other payers, pat name  . D F^IBCEF("N-CURR INSURANCE COMPANY NAME","IBZ",,IBIFN)  . S IBX=IBX\_$$FO^IBCNEUT1(IBZ,25)\_" "                     ; ins co name  . S IBX=IBX\_$$FO^IBCNEUT1($$CURRINS(IBIFN,1),29)\_" "      ; address  . K IBZ D F^IBCEF("N-OTH INSURANCE CO. NAME","IBZ",,IBIFN)  . S IBX=IBX\_$$FO^IBCNEUT1($P($G(IBZ(1)),U,1),15)\_" "      ; other payer  . S Z=$P($G(^DPT(+$P(IB0,U,2),0)),U,1)  . S IBX=IBX\_$E(Z,1,18) ; patient name  . ;  . ; set line into list  . S IBCNT=IBCNT+1  . D SET(.IBX,1,IBDA,IBREP,IBHDR,IBCNT,.VALMCNT,.IBPAGE,.IBSTOP)  . S IBX=""  . ;  . I $G(IBZ(2))'="" D    ; other payer #2 if it exists  .. S IBX=$J("",98)\_$E($P(IBZ(2),U,1),1,15)  .. D SET(.IBX,1,IBDA,IBREP,IBHDR,IBCNT,.VALMCNT,.IBPAGE,.IBSTOP)  .. Q  . Q  ;  I IBSORT=2 D                    ; sort by payer  . N Z,IBZ  . S IBX=IBX\_" "  . ; Print other payers, patient name, date last trans, batch #, reject flag  . D F^IBCEF("N-OTH INSURANCE CO. NAME","IBZ",,IBIFN)  . S IBX=IBX\_$$FO^IBCNEUT1($P($G(IBZ(1)),U,1),18)\_" "   ; oth payer#1  . S Z=$P($G(^DPT(+$P(IB0,U,2),0)),U,1)  . S IBX=IBX\_$$FO^IBCNEUT1(Z,18)\_" "                  ; patient name  . ;  . S Z=+$P($G(^IBA(364,+IBDA,0)),U,2) ; Batch ptr  . S:IBLOC IBX=IBX\_$$FO^IBCNEUT1($$FMTE^XLFDT($P($G(^DGCR(399,IBIFN,"S")),U,14),"1"),17) ; date last printed \*547\*  . S:'IBLOC IBX=IBX\_$$FO^IBCNEUT1($$FMTE^XLFDT($P($G(^IBA(364.1,+Z,1)),U,3)\1,"1"),17) ; date last transmitted  . S:'IBLOC IBX=IBX\_$$FO^IBCNEUT1($P($G(^IBA(364.1,Z,0)),U,1),10) ; batch#  . S:IBLOC IBX=IBX\_"" ; no batch#  . S IBX=IBX\_$S($P($G(^IBA(364.1,Z,0)),U,5):" R",1:"") ; batch rejected flag  . ;  . ; set line into list  . S IBCNT=IBCNT+1  . D SET(.IBX,1,IBDA,IBREP,IBHDR,IBCNT,.VALMCNT,.IBPAGE,.IBSTOP)  . S IBX=""  . ;  . I $G(IBZ(2))'="" D       ; other payer#2 if it exists  .. S IBX=$J("",44)\_$E($P(IBZ(2),U),1,18)  .. D SET(.IBX,1,IBDA,IBREP,IBHDR,IBCNT,.VALMCNT,.IBPAGE,.IBSTOP)  .. Q  . Q  ; WRTQ I IBREP="S" S IBPAGE=IBCNT  Q  ; SET(IBX,IBLINE,IBDA,IBREP,IBHDR,IBCNT,VALMCNT,IBPAGE,IBSTOP) ;  N Q,Z,IBZ  S IBZ=IBX,IBX=""  I IBREP="R" D  Q  . D:($Y+5)>IOSL!'IBPAGE HDR^IBCEPTC1(IBHDR,IBSORT,.IBPAGE,.IBSTOP) D  . I IBLINE F Z=1:1:IBLINE W !  . W:'IBSTOP IBZ  . Q  ;  ; only display the counter if we have a line with the claim#  S VALMCNT=VALMCNT+1  I IBDA,$TR($E(IBZ,1,8)," ")'="" S IBZ=$$FO^IBCNEUT1($J(IBCNT,3),6)\_IBZ  I IBDA,$TR($E(IBZ,1,8)," ")="" S IBZ=" "\_IBZ  ;  S ^TMP("IB\_PREV\_CLAIM\_LIST",$J,VALMCNT,0)=IBZ  S ^TMP("IB\_PREV\_CLAIM\_LIST",$J,"IDX",VALMCNT,IBCNT)=""  I IBDA,$TR($E(IBZ,1,8)," ")'="" S ^TMP("IB\_PREV\_CLAIM\_LIST\_DX",$J,IBCNT)=VALMCNT\_U\_IBDA  Q  ; CURRINS(IBIFN,TRUNC) ; Returns Current insurance address for given claim  ; TRUNC = truncate flag; 1 to truncate the address and city  N IBX,IBZ,L1,CITY,ST  D F^IBCEF("N-CURR INS CO FULL ADDRESS","IBZ",,IBIFN)  S L1=$G(IBZ(1)) I +$G(TRUNC) S L1=$E(L1,1,15)  S CITY=$G(IBZ(4)) I +$G(TRUNC) S CITY=$E(CITY,1,10)  S ST=$G(IBZ(5))  I ST S ST=$P($G(^DIC(5,ST,0)),U,2)  S IBX=L1\_" "\_CITY  I CITY'="",ST'="" S IBX=IBX\_","\_ST  E  S IBX=IBX\_" "\_ST  Q IBX  ; | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCEPTR | | | | | | | | |
| **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** | | | | | | | | | |
| IBCEPTR ;ALB/ESG - Test Claim Messages Report ;28-JAN-2005  ;;2.0;INTEGRATED BILLING;\*\*296,320,348,349\*\*;21-MAR-94;Build 46  ;;Per VHA Directive 2004-038, this routine should not be modified.  ;  ; eClaims Plus  ; Report on Test Claim Transmissions and Status Messages  ; EN ; Entry Point  NEW STOP,IBRMETH,IBRDATA  D SELECT I STOP G EXIT  D DEVICE EXIT ; Exit Point  Q  ; SELECT ; Determine which claim#'s or batch#'s to report on  NEW DIC,DIR,X,Y,DIRUT,DTOUT,DUOUT,DIROUT,D  S STOP=0  W @IOF  W !!?23,"Test Claim EDI Transmission Report"  W !!?7,"This report will display EDI transmission data and returned status"  W !?7,"message data for selected test claims. You may select test claims"  W !?7,"by claim number or by batch number or you may search for claims that"  W !?7,"were transmitted within a date range.",!  S DIR(0)="SO^C:Claim;B:Batch;D:Date Range (Date Transmitted)"  S DIR("A")="Selection Method",DIR("B")="D"  D ^DIR K DIR  I $D(DIRUT) S STOP=1 G SELECTX  S IBRMETH=Y  I IBRMETH'="C",IBRMETH'="B",IBRMETH'="D" S STOP=1 G SELECTX  ;  K IBRDATA  I IBRMETH="C" D  . F  D  Q:Y'>0  .. W !  .. S DIC("A")="Test Claim: "  .. I $O(IBRDATA("")) S DIC("A")="Another Test Claim: "  .. S DIC("W")="D CLMLST^IBCEPTR(Y)"  .. S DIC=361.4,DIC(0)="AEMQ",D="B" D MIX^DIC1  .. Q:Y'>0  .. S IBRDATA(+Y)=""  .. Q  . Q  ;  I IBRMETH="B" D  . F  D  Q:Y'>0  .. W !  .. S DIC("A")="Test Batch: "  .. I $O(IBRDATA("")) S DIC("A")="Another Test Batch: "  .. S DIC("S")="I $P(^(0),U,14),$O(^IBM(361.4,""C"",+Y,0))"  .. S DIC=364.1,DIC(0)="AEMQ",D="B^C" D MIX^DIC1  .. Q:Y'>0  .. S IBRDATA(+Y)=""  .. Q  . Q  ;  I IBRMETH="D" D  . W !  . S DIR(0)="DAO^:"\_DT\_":AEX",DIR("A")=" Earliest Date Claims Transmitted: "  . D ^DIR K DIR  . I $D(DIRUT)!'Y Q  . S IBRDATA(1)=Y  . W !  . S DIR(0)="DAO^"\_Y\_":"\_DT\_":AEX",DIR("A")=" Latest Date Claims Transmitted: ",DIR("B")="Today"  . D ^DIR K DIR  . I $D(DIRUT)!'Y Q  . S IBRDATA(2)=Y  . Q  ;  I '$O(IBRDATA("")) S STOP=1 G SELECTX  I IBRMETH="D",'$G(IBRDATA(1)) S STOP=1 G SELECTX  I IBRMETH="D",'$G(IBRDATA(2)) S STOP=1 G SELECTX  ; SELECTX ;  Q  ; DEVICE ; standard device selection  NEW ZTRTN,ZTDESC,ZTSAVE,POP  W !!!,"This report is 80 characters wide.",!  S ZTRTN="COMPILE^IBCEPTR"  S ZTDESC="Test Claim EDI Transmission Report"  S ZTSAVE("IBRMETH")=""  S ZTSAVE("IBRDATA")=""  D EN^XUTMDEVQ(ZTRTN,ZTDESC,.ZTSAVE,"QM") DEVX ;  Q  ; COMPILE ; compile the data into a scratch global  NEW RTN,EXTBCH,IBIFN,BCHIEN,TXDATM  S RTN="IBCEPTR"  KILL ^TMP($J,RTN) ; init scratch global  ;  I IBRMETH="C" D    ; claim search  . S EXTBCH=0  . S IBIFN=0  . F  S IBIFN=$O(IBRDATA(IBIFN)) Q:'IBIFN  D STORE(IBIFN)  . Q  ;  I IBRMETH="B" D    ; batch search  . S BCHIEN=0  . F  S BCHIEN=$O(IBRDATA(BCHIEN)) Q:'BCHIEN  D  .. S EXTBCH=$P($G(^IBA(364.1,BCHIEN,0)),U,1)  .. I EXTBCH="" S EXTBCH="~unknown"  .. S IBIFN=0  .. F  S IBIFN=$O(^IBM(361.4,"C",BCHIEN,IBIFN)) Q:'IBIFN  D STORE(IBIFN)  .. Q  . Q  ;  I IBRMETH="D" D    ; date range search  . S EXTBCH=0  . S TXDATM=$O(^IBM(361.4,"ATD",IBRDATA(1)),-1)  . F  S TXDATM=$O(^IBM(361.4,"ATD",TXDATM)) Q:'TXDATM  Q:(TXDATM\1)>IBRDATA(2) D  .. S IBIFN=0  .. F  S IBIFN=$O(^IBM(361.4,"ATD",TXDATM,IBIFN)) Q:'IBIFN  D STORE(IBIFN)  .. Q  . Q  ;  D PRINT                           ; print the report  D ^%ZISC ; close the device  KILL ^TMP($J,RTN) ; clean up scratch global  I $D(ZTQUEUED) S ZTREQ="@"        ; purge the task record COMPX ;  Q  ; STORE(IBIFN) ; Input = internal bill#; continue compilation  NEW IB0,CLAIM,IBRTXD0,TXIEN,SMIEN,DATA,TXDTM  S IB0=$G(^DGCR(399,IBIFN,0))  S CLAIM=$P(IB0,U,1) ; external claim#  I CLAIM="" S CLAIM="~unknown"  S IBRTXD0=99999999 ; initial value for earliest transmission date  ;  I IBRMETH="C" D   ; claim search for transmission data (all)  . S TXIEN=0  . F  S TXIEN=$O(^IBM(361.4,IBIFN,1,TXIEN)) Q:'TXIEN  D STORETX(IBIFN,TXIEN)  . Q  ;  I IBRMETH="B" D   ; batch search for transmission data ("C" x-ref)  . S TXIEN=0  . F  S TXIEN=$O(^IBM(361.4,"C",BCHIEN,IBIFN,TXIEN)) Q:'TXIEN  D STORETX(IBIFN,TXIEN)  . Q  ;  I IBRMETH="D" D   ; date range search for transmission data ("ATD" xref)  . S TXIEN=0  . F  S TXIEN=$O(^IBM(361.4,"ATD",TXDATM,IBIFN,TXIEN)) Q:'TXIEN  D STORETX(IBIFN,TXIEN)  . Q  ;  ; loop thru all returned messages for claim  S SMIEN=0  F  S SMIEN=$O(^IBM(361.4,IBIFN,2,SMIEN)) Q:'SMIEN  D  . S DATA=$G(^IBM(361.4,IBIFN,2,SMIEN,0)) Q:DATA=""   ; received msg data  . S TXDTM=$P(DATA,U,1) Q:'TXDTM    ; msg rec'd date/time  . ;  . ; Batch only: if this status message was received before the  . ; earliest transmission for this batch, then don't include it  . I IBRMETH="B",TXDTM'>IBRTXD0 Q  . ;  . ; Date range search only: make sure the date/time the status message  . ; was received is inside the user specified date range for this report  . I IBRMETH="D",(TXDTM\1)<IBRDATA(1) Q    ; rec'd too early  . I IBRMETH="D",(TXDTM\1)>IBRDATA(2) Q    ; rec'd too late  . ;  . ; store it  . M ^TMP($J,RTN,EXTBCH,CLAIM,TXDTM,2,SMIEN)=^IBM(361.4,IBIFN,2,SMIEN)  . Q STOREX ;  Q  ; STORETX(IBIFN,TXIEN) ; store transmission info  NEW DATA,TXDTM  S DATA=$G(^IBM(361.4,IBIFN,1,TXIEN,0))  I DATA="" G STTXXX  S TXDTM=$P(DATA,U,1) ; transmit date/time  I 'TXDTM G STTXXX  I TXDTM<IBRTXD0 S IBRTXD0=TXDTM  ;  ; store it  M ^TMP($J,RTN,EXTBCH,CLAIM,TXDTM,1,TXIEN)=^IBM(361.4,IBIFN,1,TXIEN) STTXXX ;  Q  ; PRINT ; print the report to the specified device  NEW MAXCNT,CRT,PAGECNT,STOP,DIR,X,Y,DIRUT,DUOUT,DTOUT,DIROUT  NEW BATCH,CLAIM,IBIFN,CLMD,TXD,TYPE,IEN  I IOST["C-" S MAXCNT=IOSL-3,CRT=1  E  S MAXCNT=IOSL-6,CRT=0  S PAGECNT=0,STOP=0  ;  I '$D(^TMP($J,RTN)) D HEADER W !!!?5,"No Data Found"  ;  S BATCH=""  F  S BATCH=$O(^TMP($J,RTN,BATCH)) Q:BATCH=""  D  Q:STOP  . D HEADER Q:STOP  . I BATCH'=0 W !!,"Batch#: ",BATCH  . S CLAIM=""  . F  S CLAIM=$O(^TMP($J,RTN,BATCH,CLAIM)) Q:CLAIM=""  D  Q:STOP  .. I $Y+2>MAXCNT!'PAGECNT D HEADER Q:STOP  .. I BATCH=0 W !  .. W !,"Claim#: ",CLAIM  .. S IBIFN=+$O(^DGCR(399,"B",CLAIM,""))  .. I IBIFN S CLMD=$$BT(IBIFN) W ?18,$E($P(CLMD,U,3),1,20),?40,"(",$P(CLMD,U,1),")"  .. W !,$$RJ^XLFSTR("",80,"-")  .. ;  .. S TXD=0  .. F  S TXD=$O(^TMP($J,RTN,BATCH,CLAIM,TXD)) Q:'TXD!STOP  S TYPE=0 F  S TYPE=$O(^TMP($J,RTN,BATCH,CLAIM,TXD,TYPE)) Q:'TYPE!STOP  S IEN=0 F  S IEN=$O(^TMP($J,RTN,BATCH,CLAIM,TXD,TYPE,IEN)) Q:'IEN!STOP  D  Q:STOP  ... I TYPE=1 D TXPRT  ... I TYPE=2 D SMPRT  ... Q  .. Q  . Q  ;  I STOP G PRINTX  I $Y+2>MAXCNT!'PAGECNT D HEADER I STOP G PRINTX  W !!?5,"\*\*\* End of Report \*\*\*"  I CRT,'$D(ZTQUEUED) S DIR(0)="E" D ^DIR K DIR PRINTX ;  Q  ; TXPRT ; print transmission information  NEW DATA,TXDTM,EXTBCH,TXBY,INSIEN,PAYER,PSEQ,INZ  S DATA=$G(^TMP($J,RTN,BATCH,CLAIM,TXD,TYPE,IEN,0)) I DATA="" G TXPRTX  S TXDTM=$$FMTE^XLFDT($P(DATA,U,1),"5Z")  S EXTBCH=$$EXTERNAL^DILFD(361.41,.02,,$P(DATA,U,2)) ; batch  S TXBY=$$EXTERNAL^DILFD(361.41,.03,,$P(DATA,U,3)) ; who tx  S INSIEN=+$$FINDINS^IBCEF1(IBIFN,$P(DATA,U,4)) ; insurance  S INZ=$$INSADD^IBCNSC02(INSIEN) ; ins name/addr  S PAYER=$P(INZ,U,1) ; ins name  S PSEQ=$TR($P(DATA,U,4),"123","PST") ; payer seq  ;  I $Y+2>MAXCNT!'PAGECNT D HEADER I STOP G TXPRTX  W !,"Transmission Information"  W !?1,TXDTM,?22,"Bch#",+$E(EXTBCH,4,99),?33,$E(TXBY,1,15),?50,$E(PAYER,1,20)," (",PSEQ,")"  ; display address info if not Medicare  I '$$MCRWNR^IBEFUNC(INSIEN) W !?50,$E($P(INZ,U,2),1,15),",",$E($P(INZ,U,3),1,11),",",$E($P(INZ,U,4),1,2)  W ! TXPRTX ;  Q  ; SMPRT ; print returned status message information  NEW DATA,TXDTM,SEVERITY,Z  S DATA=$G(^TMP($J,RTN,BATCH,CLAIM,TXD,TYPE,IEN,0)) I DATA="" G SMPRTX  S TXDTM=$$FMTE^XLFDT($P(DATA,U,1),"5Z")  S SEVERITY=$$EXTERNAL^DILFD(361.42,.02,,$P(DATA,U,2)) ; msg severity  ;  I $Y+2>MAXCNT!'PAGECNT D HEADER I STOP G SMPRTX  W !,"Status Message Information"  W !?1,TXDTM,?22,SEVERITY,?65,"Msg#",$P(DATA,U,3)  S Z=0  F  S Z=$O(^TMP($J,RTN,BATCH,CLAIM,TXD,TYPE,IEN,1,Z)) Q:'Z  D  Q:STOP  . I $Y+1>MAXCNT!'PAGECNT D HEADER Q:STOP  . W !?2,$G(^TMP($J,RTN,BATCH,CLAIM,TXD,TYPE,IEN,1,Z,0))  . Q  W ! SMPRTX ;  Q  ; HEADER ; page break and header  NEW LIN,HDR,TAB  S STOP=0  I CRT,PAGECNT>0,'$D(ZTQUEUED) D  I STOP G HEADX  . I MAXCNT<51 F LIN=1:1:(MAXCNT-$Y) W !  . S DIR(0)="E" D ^DIR K DIR  . I 'Y S STOP=1 Q  . Q  ;  S PAGECNT=PAGECNT+1  W @IOF,!  ;  W "Test Claim EDI Transmission Report"  S HDR="Page: "\_PAGECNT,TAB=80-$L(HDR)-1  W ?TAB,HDR  W !,"Selected ",$S(IBRMETH="B":"Batches",IBRMETH="C":"Claims",1:"Date Range")  S HDR=$$FMTE^XLFDT($$NOW^XLFDT,"1Z"),TAB=80-$L(HDR)-1  W ?TAB,HDR  W !,$$RJ^XLFSTR("",80,"=")  ;  ; check for a stop request  I $D(ZTQUEUED),$$S^%ZTLOAD() D  G HEADX  . S (ZTSTOP,STOP)=1  . W !!!?5,"\*\*\* Report Halted by TaskManager Request \*\*\*"  . Q  ; HEADX ;  Q  ; BT(IBIFN) ; bill type and info  ; [1] TYPE (form type, charge type, inp/outp)  ; [2] claim#  ; [3] patient name  NEW TYPE,IB0,F,C,S S TYPE=""  S IB0=$G(^DGCR(399,+$G(IBIFN),0)) I IB0="" Q ""  S F=$P(IB0,U,19),F=$S(F=2:"1500",1:"UB04")  S C=$P(IB0,U,27),C=$S(C=1:"Inst",1:"Prof")  S S=$$INPAT^IBCEF(IBIFN),S=$S(S=1:"Inpat",1:"Outpat")  S TYPE=F\_", "\_C\_", "\_S  Q TYPE\_U\_$P(IB0,U,1)\_U\_$P($G(^DPT(+$P(IB0,U,2),0)),U,1)  ; CLMLST(IBIFN) ; DIC lister  NEW TYPE,LTD,N1,N2  S TYPE=$P($$BT(IBIFN),U,1)  S LTD=$$FMTE^XLFDT($P($G(^IBM(361.4,IBIFN,0)),U,2),"2Z")  S N1=+$P($G(^IBM(361.4,IBIFN,1,0)),U,4) ; # transmissions  S N2=+$P($G(^IBM(361.4,IBIFN,2,0)),U,4) ; # return messages  W " ",TYPE,?34," ",LTD,?45," ",N1," Transmission",$S(N1'=1:"s",1:"")  W ?63," ",N2," Message",$S(N2'=1:"s",1:"") CLMLSTX ;  Q  ; | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCEPTR ;ALB/ESG - Test Claim Messages Report ;28-JAN-2005  ;;2.0;INTEGRATED BILLING;\*\*296,320,348,349,**592**\*\*;21-MAR-94;Build 46  ;;Per VHA Directive 2004-038, this routine should not be modified.  ;  ; eClaims Plus  ; Report on Test Claim Transmissions and Status Messages  ; EN ; Entry Point  NEW STOP,IBRMETH,IBRDATA  D SELECT I STOP G EXIT  D DEVICE EXIT ; Exit Point  Q  ; SELECT ; Determine which claim#'s or batch#'s to report on  NEW DIC,DIR,X,Y,DIRUT,DTOUT,DUOUT,DIROUT,D  S STOP=0  W @IOF  W !!?23,"Test Claim EDI Transmission Report"  W !!?7,"This report will display EDI transmission data and returned status"  W !?7,"message data for selected test claims. You may select test claims"  W !?7,"by claim number or by batch number or you may search for claims that"  W !?7,"were transmitted within a date range.",!  S DIR(0)="SO^C:Claim;B:Batch;D:Date Range (Date Transmitted)"  S DIR("A")="Selection Method",DIR("B")="D"  D ^DIR K DIR  I $D(DIRUT) S STOP=1 G SELECTX  S IBRMETH=Y  I IBRMETH'="C",IBRMETH'="B",IBRMETH'="D" S STOP=1 G SELECTX  ;  K IBRDATA  I IBRMETH="C" D  . F  D  Q:Y'>0  .. W !  .. S DIC("A")="Test Claim: "  .. I $O(IBRDATA("")) S DIC("A")="Another Test Claim: "  .. S DIC("W")="D CLMLST^IBCEPTR(Y)"  .. S DIC=361.4,DIC(0)="AEMQ",D="B" D MIX^DIC1  .. Q:Y'>0  .. S IBRDATA(+Y)=""  .. Q  . Q  ;  I IBRMETH="B" D  . F  D  Q:Y'>0  .. W !  .. S DIC("A")="Test Batch: "  .. I $O(IBRDATA("")) S DIC("A")="Another Test Batch: "  .. S DIC("S")="I $P(^(0),U,14),$O(^IBM(361.4,""C"",+Y,0))"  .. S DIC=364.1,DIC(0)="AEMQ",D="B^C" D MIX^DIC1  .. Q:Y'>0  .. S IBRDATA(+Y)=""  .. Q  . Q  ;  I IBRMETH="D" D  . W !  . S DIR(0)="DAO^:"\_DT\_":AEX",DIR("A")=" Earliest Date Claims Transmitted: "  . D ^DIR K DIR  . I $D(DIRUT)!'Y Q  . S IBRDATA(1)=Y  . W !  . S DIR(0)="DAO^"\_Y\_":"\_DT\_":AEX",DIR("A")=" Latest Date Claims Transmitted: ",DIR("B")="Today"  . D ^DIR K DIR  . I $D(DIRUT)!'Y Q  . S IBRDATA(2)=Y  . Q  ;  I '$O(IBRDATA("")) S STOP=1 G SELECTX  I IBRMETH="D",'$G(IBRDATA(1)) S STOP=1 G SELECTX  I IBRMETH="D",'$G(IBRDATA(2)) S STOP=1 G SELECTX  ; SELECTX ;  Q  ; DEVICE ; standard device selection  NEW ZTRTN,ZTDESC,ZTSAVE,POP  W !!!,"This report is 80 characters wide.",!  S ZTRTN="COMPILE^IBCEPTR"  S ZTDESC="Test Claim EDI Transmission Report"  S ZTSAVE("IBRMETH")=""  S ZTSAVE("IBRDATA")=""  D EN^XUTMDEVQ(ZTRTN,ZTDESC,.ZTSAVE,"QM") DEVX ;  Q  ; COMPILE ; compile the data into a scratch global  NEW RTN,EXTBCH,IBIFN,BCHIEN,TXDATM  S RTN="IBCEPTR"  KILL ^TMP($J,RTN) ; init scratch global  ;  I IBRMETH="C" D    ; claim search  . S EXTBCH=0  . S IBIFN=0  . F  S IBIFN=$O(IBRDATA(IBIFN)) Q:'IBIFN  D STORE(IBIFN)  . Q  ;  I IBRMETH="B" D    ; batch search  . S BCHIEN=0  . F  S BCHIEN=$O(IBRDATA(BCHIEN)) Q:'BCHIEN  D  .. S EXTBCH=$P($G(^IBA(364.1,BCHIEN,0)),U,1)  .. I EXTBCH="" S EXTBCH="~unknown"  .. S IBIFN=0  .. F  S IBIFN=$O(^IBM(361.4,"C",BCHIEN,IBIFN)) Q:'IBIFN  D STORE(IBIFN)  .. Q  . Q  ;  I IBRMETH="D" D    ; date range search  . S EXTBCH=0  . S TXDATM=$O(^IBM(361.4,"ATD",IBRDATA(1)),-1)  . F  S TXDATM=$O(^IBM(361.4,"ATD",TXDATM)) Q:'TXDATM  Q:(TXDATM\1)>IBRDATA(2) D  .. S IBIFN=0  .. F  S IBIFN=$O(^IBM(361.4,"ATD",TXDATM,IBIFN)) Q:'IBIFN  D STORE(IBIFN)  .. Q  . Q  ;  D PRINT                           ; print the report  D ^%ZISC ; close the device  KILL ^TMP($J,RTN) ; clean up scratch global  I $D(ZTQUEUED) S ZTREQ="@"        ; purge the task record COMPX ;  Q  ; STORE(IBIFN) ; Input = internal bill#; continue compilation  NEW IB0,CLAIM,IBRTXD0,TXIEN,SMIEN,DATA,TXDTM  S IB0=$G(^DGCR(399,IBIFN,0))  S CLAIM=$P(IB0,U,1) ; external claim#  I CLAIM="" S CLAIM="~unknown"  S IBRTXD0=99999999 ; initial value for earliest transmission date  ;  I IBRMETH="C" D   ; claim search for transmission data (all)  . S TXIEN=0  . F  S TXIEN=$O(^IBM(361.4,IBIFN,1,TXIEN)) Q:'TXIEN  D STORETX(IBIFN,TXIEN)  . Q  ;  I IBRMETH="B" D   ; batch search for transmission data ("C" x-ref)  . S TXIEN=0  . F  S TXIEN=$O(^IBM(361.4,"C",BCHIEN,IBIFN,TXIEN)) Q:'TXIEN  D STORETX(IBIFN,TXIEN)  . Q  ;  I IBRMETH="D" D   ; date range search for transmission data ("ATD" xref)  . S TXIEN=0  . F  S TXIEN=$O(^IBM(361.4,"ATD",TXDATM,IBIFN,TXIEN)) Q:'TXIEN  D STORETX(IBIFN,TXIEN)  . Q  ;  ; loop thru all returned messages for claim  S SMIEN=0  F  S SMIEN=$O(^IBM(361.4,IBIFN,2,SMIEN)) Q:'SMIEN  D  . S DATA=$G(^IBM(361.4,IBIFN,2,SMIEN,0)) Q:DATA=""   ; received msg data  . S TXDTM=$P(DATA,U,1) Q:'TXDTM    ; msg rec'd date/time  . ;  . ; Batch only: if this status message was received before the  . ; earliest transmission for this batch, then don't include it  . I IBRMETH="B",TXDTM'>IBRTXD0 Q  . ;  . ; Date range search only: make sure the date/time the status message  . ; was received is inside the user specified date range for this report  . I IBRMETH="D",(TXDTM\1)<IBRDATA(1) Q    ; rec'd too early  . I IBRMETH="D",(TXDTM\1)>IBRDATA(2) Q    ; rec'd too late  . ;  . ; store it  . M ^TMP($J,RTN,EXTBCH,CLAIM,TXDTM,2,SMIEN)=^IBM(361.4,IBIFN,2,SMIEN)  . Q STOREX ;  Q  ; STORETX(IBIFN,TXIEN) ; store transmission info  NEW DATA,TXDTM  S DATA=$G(^IBM(361.4,IBIFN,1,TXIEN,0))  I DATA="" G STTXXX  S TXDTM=$P(DATA,U,1) ; transmit date/time  I 'TXDTM G STTXXX  I TXDTM<IBRTXD0 S IBRTXD0=TXDTM  ;  ; store it  M ^TMP($J,RTN,EXTBCH,CLAIM,TXDTM,1,TXIEN)=^IBM(361.4,IBIFN,1,TXIEN) STTXXX ;  Q  ; PRINT ; print the report to the specified device  NEW MAXCNT,CRT,PAGECNT,STOP,DIR,X,Y,DIRUT,DUOUT,DTOUT,DIROUT  NEW BATCH,CLAIM,IBIFN,CLMD,TXD,TYPE,IEN  I IOST["C-" S MAXCNT=IOSL-3,CRT=1  E  S MAXCNT=IOSL-6,CRT=0  S PAGECNT=0,STOP=0  ;  I '$D(^TMP($J,RTN)) D HEADER W !!!?5,"No Data Found"  ;  S BATCH=""  F  S BATCH=$O(^TMP($J,RTN,BATCH)) Q:BATCH=""  D  Q:STOP  . D HEADER Q:STOP  . I BATCH'=0 W !!,"Batch#: ",BATCH  . S CLAIM=""  . F  S CLAIM=$O(^TMP($J,RTN,BATCH,CLAIM)) Q:CLAIM=""  D  Q:STOP  .. I $Y+2>MAXCNT!'PAGECNT D HEADER Q:STOP  .. I BATCH=0 W !  .. W !,"Claim#: ",CLAIM  .. S IBIFN=+$O(^DGCR(399,"B",CLAIM,""))  .. I IBIFN S CLMD=$$BT(IBIFN) W ?18,$E($P(CLMD,U,3),1,20),?40,"(",$P(CLMD,U,1),")"  .. W !,$$RJ^XLFSTR("",80,"-")  .. ;  .. S TXD=0  .. F  S TXD=$O(^TMP($J,RTN,BATCH,CLAIM,TXD)) Q:'TXD!STOP  S TYPE=0 F  S TYPE=$O(^TMP($J,RTN,BATCH,CLAIM,TXD,TYPE)) Q:'TYPE!STOP  S IEN=0 F  S IEN=$O(^TMP($J,RTN,BATCH,CLAIM,TXD,TYPE,IEN)) Q:'IEN!STOP  D  Q:STOP  ... I TYPE=1 D TXPRT  ... I TYPE=2 D SMPRT  ... Q  .. Q  . Q  ;  I STOP G PRINTX  I $Y+2>MAXCNT!'PAGECNT D HEADER I STOP G PRINTX  W !!?5,"\*\*\* End of Report \*\*\*"  I CRT,'$D(ZTQUEUED) S DIR(0)="E" D ^DIR K DIR PRINTX ;  Q  ; TXPRT ; print transmission information  NEW DATA,TXDTM,EXTBCH,TXBY,INSIEN,PAYER,PSEQ,INZ  S DATA=$G(^TMP($J,RTN,BATCH,CLAIM,TXD,TYPE,IEN,0)) I DATA="" G TXPRTX  S TXDTM=$$FMTE^XLFDT($P(DATA,U,1),"5Z")  S EXTBCH=$$EXTERNAL^DILFD(361.41,.02,,$P(DATA,U,2)) ; batch  S TXBY=$$EXTERNAL^DILFD(361.41,.03,,$P(DATA,U,3)) ; who tx  S INSIEN=+$$FINDINS^IBCEF1(IBIFN,$P(DATA,U,4)) ; insurance  S INZ=$$INSADD^IBCNSC02(INSIEN) ; ins name/addr  S PAYER=$P(INZ,U,1) ; ins name  S PSEQ=$TR($P(DATA,U,4),"123","PST") ; payer seq  ;  I $Y+2>MAXCNT!'PAGECNT D HEADER I STOP G TXPRTX  W !,"Transmission Information"  W !?1,TXDTM,?22,"Bch#",+$E(EXTBCH,4,99),?33,$E(TXBY,1,15),?50,$E(PAYER,1,20)," (",PSEQ,")"  ; display address info if not Medicare  I '$$MCRWNR^IBEFUNC(INSIEN) W !?50,$E($P(INZ,U,2),1,15),",",$E($P(INZ,U,3),1,11),",",$E($P(INZ,U,4),1,2)  W ! TXPRTX ;  Q  ; SMPRT ; print returned status message information  NEW DATA,TXDTM,SEVERITY,Z  S DATA=$G(^TMP($J,RTN,BATCH,CLAIM,TXD,TYPE,IEN,0)) I DATA="" G SMPRTX  S TXDTM=$$FMTE^XLFDT($P(DATA,U,1),"5Z")  S SEVERITY=$$EXTERNAL^DILFD(361.42,.02,,$P(DATA,U,2)) ; msg severity  ;  I $Y+2>MAXCNT!'PAGECNT D HEADER I STOP G SMPRTX  W !,"Status Message Information"  W !?1,TXDTM,?22,SEVERITY,?65,"Msg#",$P(DATA,U,3)  S Z=0  F  S Z=$O(^TMP($J,RTN,BATCH,CLAIM,TXD,TYPE,IEN,1,Z)) Q:'Z  D  Q:STOP  . I $Y+1>MAXCNT!'PAGECNT D HEADER Q:STOP  . W !?2,$G(^TMP($J,RTN,BATCH,CLAIM,TXD,TYPE,IEN,1,Z,0))  . Q  W ! SMPRTX ;  Q  ; HEADER ; page break and header  NEW LIN,HDR,TAB  S STOP=0  I CRT,PAGECNT>0,'$D(ZTQUEUED) D  I STOP G HEADX  . I MAXCNT<51 F LIN=1:1:(MAXCNT-$Y) W !  . S DIR(0)="E" D ^DIR K DIR  . I 'Y S STOP=1 Q  . Q  ;  S PAGECNT=PAGECNT+1  W @IOF,!  ;  W "Test Claim EDI Transmission Report"  S HDR="Page: "\_PAGECNT,TAB=80-$L(HDR)-1  W ?TAB,HDR  W !,"Selected ",$S(IBRMETH="B":"Batches",IBRMETH="C":"Claims",1:"Date Range")  S HDR=$$FMTE^XLFDT($$NOW^XLFDT,"1Z"),TAB=80-$L(HDR)-1  W ?TAB,HDR  W !,$$RJ^XLFSTR("",80,"=")  ;  ; check for a stop request  I $D(ZTQUEUED),$$S^%ZTLOAD() D  G HEADX  . S (ZTSTOP,STOP)=1  . W !!!?5,"\*\*\* Report Halted by TaskManager Request \*\*\*"  . Q  ; HEADX ;  Q  ; BT(IBIFN) ; bill type and info  ; [1] TYPE (form type, charge type, inp/outp)  ; [2] claim#  ; [3] patient name  NEW TYPE,IB0,F,C,S S TYPE=""  S IB0=$G(^DGCR(399,+$G(IBIFN),0)) I IB0="" Q ""  **;**S F=$P(IB0,U,19),F=$S(F=2:"1500",1:"UB04")) **;JRA IB\*2.0\*592 ';'**  S F=$P(IB0,U,19),F=$S(F=2:"1500",**F=7:"J430D",**1:"UB04") **;JRA IB\*2.0\*592 Add Dental Form 'J430D'**  S C=$P(IB0,U,27),C=$S(C=1:"Inst",1:"Prof")  S S=$$INPAT^IBCEF(IBIFN),S=$S(S=1:"Inpat",1:"Outpat")  S TYPE=F\_", "\_C\_", "\_S  Q TYPE\_U\_$P(IB0,U,1)\_U\_$P($G(^DPT(+$P(IB0,U,2),0)),U,1)  ; CLMLST(IBIFN) ; DIC lister  NEW TYPE,LTD,N1,N2  S TYPE=$P($$BT(IBIFN),U,1)  S LTD=$$FMTE^XLFDT($P($G(^IBM(361.4,IBIFN,0)),U,2),"2Z")  S N1=+$P($G(^IBM(361.4,IBIFN,1,0)),U,4) ; # transmissions  S N2=+$P($G(^IBM(361.4,IBIFN,2,0)),U,4) ; # return messages  W " ",TYPE,?34," ",LTD,?45," ",N1," Transmission",$S(N1'=1:"s",1:"")  W ?63," ",N2," Message",$S(N2'=1:"s",1:"") CLMLSTX ;  Q  ; | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCEQ1 | | | | | | | | |
| **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** | | | | | | | | | |
| IBCEQ1 ;BSL,ALB/TMK - PROVIDER ID QUERY ;25-AUG-03  ;;2.0;INTEGRATED BILLING;\*\*232,356,349\*\*;21-MAR-94;Build 46  ;;Per VHA Directive 2004-038, this routine should not be modified.  ;  ;QUERY TOOL HELPS IDENTIFY PLANS THAT ARE LACKING PROVIDER ID  ;INFO OR HAVE BAD PROVIDER ID DATA FOR E-BILLING  ;  ;CONDITIONS TO IDENTIFY:  ;1-BLUE CROSS LINKED TO 1500 ONLY (1) HARD ERROR  ;2-BLUE SHIELD LINKED TO UB-04 ONLY (2) WARNING  ;3-BLUE CROSS ID APPLIED TO BOTH FORMS (0) WARNING  ;4-BLUE CROSS OR BLUE SHIELD IDs EXIST FOR AN INS CO, BUT ONE OR  ; MORE OF THE INSURANCE COMPANY'S PLANS DOES NOT HAVE AN  ; ELECTRONIC PLAN TYPE OF 'BL'  ;5-NON BLUE CROSS/SHIELD ID FOR AN INS COMPANY WITH BLUE PLAN(S)  ;6-VAD000 as an ID but not flagged as a UPIN  ; EN ;  N POP,%ZIS,ZTSK,ZTRTN,ZTDESC,IBREBLD,IBSENDM,IBTO,DIR,X,Y,DUOUT,DTOUT,Z  S IBREBLD=$S('$D(^XTMP("IB\_PLAN232")):1,1:0)  I $D(^XTMP("IB\_PLAN232")) D  . S DIR("?")="IF YOU ANSWER NO, REPORT WILL BE RUN FROM THE EXISTING QUERY DATA",DIR("?",1)="IF YOU ANSWER YES, A NEW QUERY WILL BE RUN"  . S DIR(0)="YA",DIR("A",1)="THE EXTRACT GLOBAL FOR THIS QUERY ALREADY EXISTS",DIR("A")="DO YOU WANT TO DELETE IT AND RERUN THE QUERY?: ",DIR("B")="NO" W ! D ^DIR K DIR  . Q:$D(DUOUT)!$D(DTOUT)!'Y  . S IBREBLD=1  ;  N XMINSTR,Z,ZTSAVE  K ^TMP("XMY",$J),^TMP("XMY0",$J)  S XMINSTR("ADDR FLAGS")="R"  D TOWHOM^XMXAPIU(DUZ,"","S",.XMINSTR)  S Z="" F  S Z=$O(^TMP("XMY",$J,Z)) Q:Z=""  S IBTO(Z)=""  K ^TMP("XMY",$J),^TMP("XMY0",$J)  ;  S %ZIS="QM" D ^%ZIS G:POP EN1Q  I $D(IO("Q")) D  G EN1Q  . S ZTRTN="ENT^IBCEQ1("\_IBREBLD\_",.IBTO)",ZTDESC="IB - HIPAA ENHANCEMENTS PROV ID QUERY",ZTSAVE("IBTO(")=""  . D ^%ZTLOAD  . W !!,$S($D(ZTSK):"Task # "\_ZTSK\_" has been queued.",1:"Unable to queue this job.")  . K ZTSK,IO("Q") D HOME^%ZIS  U IO  D ENT(IBREBLD,.IBTO) EN1Q Q  ; ENT(IBREBLD,IBTO) ; Queued job enter here  ;  N LOOP,Z  K ^TMP($J,"SENDMSG")  S ^TMP($J,"SENDMSG")=$S(IBREBLD:1,1:0)  S Z="" F  S Z=$O(IBTO(Z)) Q:Z=""  S ^TMP($J,"SENDMSG",0,Z)=""  I $G(IBREBLD) D  . ; Rebld query  . K ^XTMP("IB\_PLAN232")  . S ^XTMP("IB\_PLAN232")="",^XTMP("IB\_PLAN232",0)=$$FMADD^XLFDT(DT,45)\_U\_DT\_"^IB PATCH 232 PROV ID QUERY"  . ;  . ; loop thru 355.91 (IB INSURANCE CO LEVEL BILLING PROV ID)  . ; then 355.9 (IB BILLING PRACTITIONER ID)  . F LOOP=355.91,355.9 D LP  . ;  ;  D RPTOUT^IBCEQ1A  K ^TMP($J,"SENDMSG")  Q  ; LP ; Loop through ids  N IB,PTYP,PAYER,PLANIEN,FTA,IEPLAN,IPROV,PPROV,EDII,EDIP,PAYERP,TYPCOV,IBPMBPID,PTYPNM,IBI3,IBI0,SEQ,BLUE,TOT,NBLUE,DIR,DTOUT,DUOUT,X,Z,Z0,Z1,BL,UPIN,BCR,BSH  S (SEQ,X,TOT,NBLUE,BLUE)=0,(BCR,BSH,UPIN)=""  S Z="" F  S Z=$O(^IBE(355.97,Z)) Q:'Z  S Z0=$G(^(Z,0)) D  . I $P(Z,U)["BLUE CROSS" S BCR=Z Q  . I $P(Z,U)["BLUE SHIELD" S BSH=Z Q  . I $P(Z,U)["UPIN" S UPIN=Z Q  S:UPIN="" UPIN=22 S:BCR="" BCR=1 S:BSH="" BSH=2  F  S X=$O(^IBA(LOOP,X)) Q:+X=0 D  . S (PAYER,FTA,PLANIEN,IEPLAN,IPROV,PPROV,EDII,EDIP,PAYERP,TYPCOV,IBPMBPID,PTYPNM)=""  . S SEQ=SEQ+1  . S IB=$G(^IBA(LOOP,X,0))  . S PTYP=$P(IB,U,6) ; prov id type ien  . Q:PTYP=""  ; no prov type  . S PTYPNM=$P($G(^IBE(355.97,PTYP,0)),U) ; prov id type desc  . S PAYERP=$S(LOOP[".91":+IB,1:+$P(IB,U,2)) ;ins co ien  . S IBI0=$G(^DIC(36,PAYERP,0)),IBI3=$G(^(3)),PAYER=$P(IBI0,U)  . Q:$P(IBI0,U,5)!(IBI0="") ; ins co inactive/deleted  . S EDIP=$P(IBI3,U,2) ; edi id# prof  . S EDII=$P(IBI3,U,4) ; edi id# inst  . S IEPLAN=$P(IBI3,U,9) ; elec ins type ?1N  . S PPROV=$P(IBI0,U,17) ; prof. prov#  . S IPROV=$P(IBI0,U,11) ; hosp. prov#  . S TYPCOV=$P(IBI0,U,13) ; type of cov ien;file 355.2  . S FTA=$P(IB,U,4) ; form type applied; 0:both, 1:ub, 2:1500  . S IBPMBPID=X\_";"\_LOOP  . I $P(IB,U,7)="VAD000",PTYP'=UPIN D SET(6)  . ;  . I PTYP'=BCR&(PTYP'=BSH) D  Q    ; not BC/BS  .. ; Only do following check once per insurance co  .. Q:$D(^XTMP("IB\_PLAN232",3,PAYERP))  .. S ^XTMP("IB\_PLAN232",3,PAYERP)=""  .. ; Check if BC/BS ids exist at all for ins co  .. Q:$O(^IBA(355.9,"AC",1,PAYERP,0))!$O(^IBA(355.9,"AC",2,PAYERP,0))!$O(^IBA(355.91,"AC",PAYERP,1,0))!$O(^IBA(355.91,"AC",PAYERP,2,0))  .. S BL=0  .. S Z1=0 F  S Z1=$O(^IBA(355.3,"B",PAYERP,Z1)) Q:'Z1  D  ... I '$P($G(^IBA(355.3,Z1,0)),U,11),$P($G(^(0)),U,15)="BL" S PLANIEN=Z1,BL=1 D SET(5)  .. S:BL NBLUE=NBLUE+1  . ;  . S BLUE=BLUE+1  . ; ERROR - FORM TYPE=2:1500 AND PTYP=1:BC  . I PTYP=1&(FTA=2) D SET(1) Q  . ;  . I PTYP=2&(FTA=1) D SET(2) Q  ; BS applied to just UB  . I FTA=0&(PTYP=1) D SET(3) Q  ; BC applied to both forms  . ;  . ; Only do following check once per insurance co  . I '$D(^XTMP("IB\_PLAN232",2,PAYERP)) D  ; Checks plans not BL  .. S Z1=0,^XTMP("IB\_PLAN232",2,PAYERP)=""  .. F  S Z1=$O(^IBA(355.3,"B",PAYERP,Z1)) Q:'Z1  D  ... I $P($G(^IBA(355.3,Z1,0)),U,15)'="BL",'$P(^(0),U,11) S PLANIEN=Z1 D SET(4) Q  ;  ; 3RD PC XTMP(IB\_PLAN232)=TOTAL BLUES WITH NO BLUE IDS  S $P(^XTMP("IB\_PLAN232"),U,3)=$P($G(^XTMP("IB\_PLAN232")),U,3)+NBLUE  ;  ; 4TH PC XTMP(IB\_PLAN232)=TOT NUMBER SCANNED  S $P(^XTMP("IB\_PLAN232"),U,4)=$P($G(^XTMP("IB\_PLAN232")),U,4)+SEQ  ;  ; 5TH PC XTMP(IB\_PLAN232)=TOT BLUES IDS FOUND  S $P(^XTMP("IB\_PLAN232"),U,5)=$P($G(^XTMP("IB\_PLAN232")),U,5)+BLUE  ;  ; 6TH PC XTMP(IB\_PLAN232)=TOTAL ERRORS FOUND  S $P(^XTMP("IB\_PLAN232"),U,6)=$P($G(^XTMP("IB\_PLAN232")),U,6)+TOT  Q  ; SET(Z) ;SET VALUES INTO SAVE GLOBAL  ; Z=REASON WHY WE'RE SETTING IT  ; 1. PAYER-ins co name (36)  ; 2. PLAN-grp name (355.3)  ; 3. GROUP-grp # (355.3)  ; 4. FTA-form typ (355.9)  ; 5. EPLAN-"BL" (355.3)  ; 6. IEPLAN-elec ins typ (36)  ; 7. IPROV-hosp prov# (36)  ; 8. PPROV-prof prov# (36)  ; 9. EDII-inst edi id# (36)  ;10. EDIP-prof edi id# (36)  ;11. PAYERP-ins co ien (36)  ;12. TYPCOV-type of cov ien (36)  ;13. PLANIEN-ien of file (355.3)  ;14. IBPMBPID-355.9 or 355.91;ien of file  ;15. PTYPNM-prov id type desc (355.9)  ;16. Z-reason  ;  N A,DUP  ;  S A=$O(^XTMP("IB\_PLAN232",1," "),-1)+1,TOT=TOT+1  S ^XTMP("IB\_PLAN232",1,A,0)=PAYER\_U\_""\_U\_""\_U\_FTA\_U\_""\_U\_IEPLAN\_U\_""\_U\_""\_U\_""\_U\_""\_U\_PAYERP\_U\_TYPCOV\_U\_PLANIEN\_U\_IBPMBPID\_U\_PTYPNM\_U\_Z  Q  ; | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCEQ1 ;BSL,ALB/TMK - PROVIDER ID QUERY ;25-AUG-03  ;;2.0;INTEGRATED BILLING;\*\*232,356,349,592\*\*;21-MAR-94;Build 46  ;;Per VHA Directive 2004-038, this routine should not be modified.  ;  ;QUERY TOOL HELPS IDENTIFY PLANS THAT ARE LACKING PROVIDER ID  ;INFO OR HAVE BAD PROVIDER ID DATA FOR E-BILLING  ;  ;CONDITIONS TO IDENTIFY:  ;1-BLUE CROSS LINKED TO 1500 ONLY (1) HARD ERROR  ;2-BLUE SHIELD LINKED TO UB-04 ONLY (2) WARNING  ;3-BLUE CROSS ID APPLIED TO BOTH FORMS (0) WARNING  ;4-BLUE CROSS OR BLUE SHIELD IDs EXIST FOR AN INS CO, BUT ONE OR  ; MORE OF THE INSURANCE COMPANY'S PLANS DOES NOT HAVE AN  ; ELECTRONIC PLAN TYPE OF 'BL'  ;5-NON BLUE CROSS/SHIELD ID FOR AN INS COMPANY WITH BLUE PLAN(S)  ;6-VAD000 as an ID but not flagged as a UPIN  ; EN ;  N POP,%ZIS,ZTSK,ZTRTN,ZTDESC,IBREBLD,IBSENDM,IBTO,DIR,X,Y,DUOUT,DTOUT,Z  S IBREBLD=$S('$D(^XTMP("IB\_PLAN232")):1,1:0)  I $D(^XTMP("IB\_PLAN232")) D  . S DIR("?")="IF YOU ANSWER NO, REPORT WILL BE RUN FROM THE EXISTING QUERY DATA",DIR("?",1)="IF YOU ANSWER YES, A NEW QUERY WILL BE RUN"  . S DIR(0)="YA",DIR("A",1)="THE EXTRACT GLOBAL FOR THIS QUERY ALREADY EXISTS",DIR("A")="DO YOU WANT TO DELETE IT AND RERUN THE QUERY?: ",DIR("B")="NO" W ! D ^DIR K DIR  . Q:$D(DUOUT)!$D(DTOUT)!'Y  . S IBREBLD=1  ;  N XMINSTR,Z,ZTSAVE  K ^TMP("XMY",$J),^TMP("XMY0",$J)  S XMINSTR("ADDR FLAGS")="R"  D TOWHOM^XMXAPIU(DUZ,"","S",.XMINSTR)  S Z="" F  S Z=$O(^TMP("XMY",$J,Z)) Q:Z=""  S IBTO(Z)=""  K ^TMP("XMY",$J),^TMP("XMY0",$J)  ;  S %ZIS="QM" D ^%ZIS G:POP EN1Q  I $D(IO("Q")) D  G EN1Q  . S ZTRTN="ENT^IBCEQ1("\_IBREBLD\_",.IBTO)",ZTDESC="IB - HIPAA ENHANCEMENTS PROV ID QUERY",ZTSAVE("IBTO(")=""  . D ^%ZTLOAD  . W !!,$S($D(ZTSK):"Task # "\_ZTSK\_" has been queued.",1:"Unable to queue this job.")  . K ZTSK,IO("Q") D HOME^%ZIS  U IO  D ENT(IBREBLD,.IBTO) EN1Q Q  ; ENT(IBREBLD,IBTO) ; Queued job enter here  ;  N LOOP,Z  K ^TMP($J,"SENDMSG")  S ^TMP($J,"SENDMSG")=$S(IBREBLD:1,1:0)  S Z="" F  S Z=$O(IBTO(Z)) Q:Z=""  S ^TMP($J,"SENDMSG",0,Z)=""  I $G(IBREBLD) D  . ; Rebld query  . K ^XTMP("IB\_PLAN232")  . S ^XTMP("IB\_PLAN232")="",^XTMP("IB\_PLAN232",0)=$$FMADD^XLFDT(DT,45)\_U\_DT\_"^IB PATCH 232 PROV ID QUERY"  . ;  . ; loop thru 355.91 (IB INSURANCE CO LEVEL BILLING PROV ID)  . ; then 355.9 (IB BILLING PRACTITIONER ID)  . F LOOP=355.91,355.9 D LP  . ;  ;  D RPTOUT^IBCEQ1A  K ^TMP($J,"SENDMSG")  Q  ; LP ; Loop through ids  N IB,PTYP,PAYER,PLANIEN,FTA,IEPLAN,IPROV,PPROV,EDII,EDIP,PAYERP,TYPCOV,IBPMBPID,PTYPNM,IBI3,IBI0,SEQ,BLUE,TOT,NBLUE,DIR,DTOUT,DUOUT,X,Z,Z0,Z1,BL,UPIN,BCR,BSH  S (SEQ,X,TOT,NBLUE,BLUE)=0,(BCR,BSH,UPIN)=""  S Z="" F  S Z=$O(^IBE(355.97,Z)) Q:'Z  S Z0=$G(^(Z,0)) D  . I $P(Z,U)["BLUE CROSS" S BCR=Z Q  . I $P(Z,U)["BLUE SHIELD" S BSH=Z Q  . I $P(Z,U)["UPIN" S UPIN=Z Q  S:UPIN="" UPIN=22 S:BCR="" BCR=1 S:BSH="" BSH=2  F  S X=$O(^IBA(LOOP,X)) Q:+X=0 D  . S (PAYER,FTA,PLANIEN,IEPLAN,IPROV,PPROV,EDII,EDIP,PAYERP,TYPCOV,IBPMBPID,PTYPNM)=""  . S SEQ=SEQ+1  . S IB=$G(^IBA(LOOP,X,0))  . S PTYP=$P(IB,U,6) ; prov id type ien  . Q:PTYP=""  ; no prov type  . S PTYPNM=$P($G(^IBE(355.97,PTYP,0)),U) ; prov id type desc  . S PAYERP=$S(LOOP[".91":+IB,1:+$P(IB,U,2)) ;ins co ien  . S IBI0=$G(^DIC(36,PAYERP,0)),IBI3=$G(^(3)),PAYER=$P(IBI0,U)  . Q:$P(IBI0,U,5)!(IBI0="") ; ins co inactive/deleted  . S EDIP=$P(IBI3,U,2) ; edi id# prof  . S EDII=$P(IBI3,U,4) ; edi id# inst  . S IEPLAN=$P(IBI3,U,9) ; elec ins type ?1N  . S PPROV=$P(IBI0,U,17) ; prof. prov#  . S IPROV=$P(IBI0,U,11) ; hosp. prov#  . S TYPCOV=$P(IBI0,U,13) ; type of cov ien;file 355.2  **. ; JWS;IB\*2.0\*592: Dental form type J430D**  . S FTA=$P(IB,U,4) ; form type applied; 0:both, 1:ub, 2:1500**&J430D**  . S IBPMBPID=X\_";"\_LOOP  . I $P(IB,U,7)="VAD000",PTYP'=UPIN D SET(6)  . I PTYP'=BCR&(PTYP'=BSH) D  Q    ; not BC/BS  .. ; Only do following check once per insurance co  .. Q:$D(^XTMP("IB\_PLAN232",3,PAYERP))  .. S ^XTMP("IB\_PLAN232",3,PAYERP)=""  .. ; Check if BC/BS ids exist at all for ins co  .. Q:$O(^IBA(355.9,"AC",1,PAYERP,0))!$O(^IBA(355.9,"AC",2,PAYERP,0))!$O(^IBA(355.91,"AC",PAYERP,1,0))!$O(^IBA(355.91,"AC",PAYERP,2,0))  .. S BL=0  .. S Z1=0 F  S Z1=$O(^IBA(355.3,"B",PAYERP,Z1)) Q:'Z1  D  ... I '$P($G(^IBA(355.3,Z1,0)),U,11),$P($G(^(0)),U,15)="BL" S PLANIEN=Z1,BL=1 D SET(5)  .. S:BL NBLUE=NBLUE+1  . ;  . S BLUE=BLUE+1  **. ; JWS;IB\*2.0\*592: Dental form type J430D**  . ; ERROR - FORM TYPE=2:1500**&J430D** AND PTYP=1:BC  . I PTYP=1&(FTA=2) D SET(1) Q  . I PTYP=2&(FTA=1) D SET(2) Q  ; BS applied to just UB  . I FTA=0&(PTYP=1) D SET(3) Q  ; BC applied to both forms  . ;  . ; Only do following check once per insurance co  . I '$D(^XTMP("IB\_PLAN232",2,PAYERP)) D  ; Checks plans not BL  .. S Z1=0,^XTMP("IB\_PLAN232",2,PAYERP)=""  .. F  S Z1=$O(^IBA(355.3,"B",PAYERP,Z1)) Q:'Z1  D  ... I $P($G(^IBA(355.3,Z1,0)),U,15)'="BL",'$P(^(0),U,11) S PLANIEN=Z1 D SET(4) Q  ;  ; 3RD PC XTMP(IB\_PLAN232)=TOTAL BLUES WITH NO BLUE IDS  S $P(^XTMP("IB\_PLAN232"),U,3)=$P($G(^XTMP("IB\_PLAN232")),U,3)+NBLUE  ;  ; 4TH PC XTMP(IB\_PLAN232)=TOT NUMBER SCANNED  S $P(^XTMP("IB\_PLAN232"),U,4)=$P($G(^XTMP("IB\_PLAN232")),U,4)+SEQ  ;  ; 5TH PC XTMP(IB\_PLAN232)=TOT BLUES IDS FOUND  S $P(^XTMP("IB\_PLAN232"),U,5)=$P($G(^XTMP("IB\_PLAN232")),U,5)+BLUE  ;  ; 6TH PC XTMP(IB\_PLAN232)=TOTAL ERRORS FOUND  S $P(^XTMP("IB\_PLAN232"),U,6)=$P($G(^XTMP("IB\_PLAN232")),U,6)+TOT  Q  ; SET(Z) ;SET VALUES INTO SAVE GLOBAL  ; Z=REASON WHY WE'RE SETTING IT  ; 1. PAYER-ins co name (36)  ; 2. PLAN-grp name (355.3)  ; 3. GROUP-grp # (355.3)  ; 4. FTA-form typ (355.9)  ; 5. EPLAN-"BL" (355.3)  ; 6. IEPLAN-elec ins typ (36)  ; 7. IPROV-hosp prov# (36)  ; 8. PPROV-prof prov# (36)  ; 9. EDII-inst edi id# (36)  ;10. EDIP-prof edi id# (36)  ;11. PAYERP-ins co ien (36)  ;12. TYPCOV-type of cov ien (36)  ;13. PLANIEN-ien of file (355.3)  ;14. IBPMBPID-355.9 or 355.91;ien of file  ;15. PTYPNM-prov id type desc (355.9)  ;16. Z-reason  ;  N A,DUP  ;  S A=$O(^XTMP("IB\_PLAN232",1," "),-1)+1,TOT=TOT+1  S ^XTMP("IB\_PLAN232",1,A,0)=PAYER\_U\_""\_U\_""\_U\_FTA\_U\_""\_U\_IEPLAN\_U\_""\_U\_""\_U\_""\_U\_""\_U\_PAYERP\_U\_TYPCOV\_U\_PLANIEN\_U\_IBPMBPID\_U\_PTYPNM\_U\_Z  Q  ; | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCEQ1A | | | | | | | | |
| **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** | | | | | | | | | |
| IBCEQ1A ;ALB/BSL,TMK - PROVIDER ID QUERY REPORT ;25-AUG-03  ;;2.0;INTEGRATED BILLING;\*\*232,348,349,516\*\*;21-MAR-94;Build 123  ;;Per VA Directive 6402, this routine should not be modified.  ; RPTOUT ; Print from data in ^XTMP  N IBP,IBA,IBI,IBIN,IBPNM,IBPNUM,IBSTOP,IBX,IBZ,IBPG,IBICONT,Z  K ^TMP($J,"IBZ232")  F Z=1:1:6 S ^TMP($J,"IBZ232",Z)=""  S (IBPG,IBSTOP)=0  S IBA=0 F  S IBA=$O(^XTMP("IB\_PLAN232",1,IBA)) Q:'IBA  D  . S IBX=$G(^XTMP("IB\_PLAN232",1,IBA,0))  . ; Sort by err type, ins co ien  . S ^TMP($J,"IBZ232",+$P(IBX,U,16),+$P(IBX,U,11),IBA)=IBX  ;  S IBZ=0 F  S IBZ=$O(^TMP($J,"IBZ232",IBZ)) Q:'IBZ!IBSTOP!(IBZ>6) D HDR1(.IBPG,.IBSTOP,IBZ,0) S IBI=0 F  S IBI=$O(^TMP($J,"IBZ232",IBZ,IBI)) Q:'IBI!IBSTOP  D  . S IBIN=$P($G(^DIC(36,+IBI,0)),U)\_" ("\_$S(+$G(^(3))=1:"",1:"NOT ")\_"SET TO TRANSMIT LIVE)"  . D INSHDR(.IBPG,.IBSTOP,IBIN,IBZ,0) S IBICONT=0  . S IBA=0 F  S IBA=$O(^TMP($J,"IBZ232",IBZ,IBI,IBA)) Q:'IBA!IBSTOP  S IBX=$G(^(IBA)) D  .. I ($Y+5)>IOSL D INSHDR(.IBPG,.IBSTOP,IBIN,IBZ,IBICONT) Q:IBSTOP  .. ;  .. I IBZ'=4,IBZ'=5 D  ... S IBP=+$P(IBX,U,14)  ... I $P(IBX,U,14)[".91" S IBPNM="ALL PROVIDERS"  ... I $P(IBX,U,14)'[".91" D  .... N Z  .... S Z=$P($G(^IBA(355.9,IBP,0)),U)  .... S IBPNM=$S(Z["VA(200":"",1:"#")\_$$EXTERNAL^DILFD(355.9,.01,"",Z)  ... S IBPNUM=$P($G(^IBA(+$P($P(IBX,U,14),";",2),IBP,0)),U,7)  ... D WRT(1," "\_$E($P("BOTH^UB-04^CMS-1500",U,$P(IBX,U,4)+1)\_$J("",9),1,9)\_" "\_$E($P(IBX,U,15)\_$J("",23),1,23)\_" "\_$E(IBPNM\_$J("",28),1,28)\_" "\_$E(IBPNUM,1,11))  .. ;  .. I IBZ=4!(IBZ=5) D  ... ;IB\*516/TAZ - Change Group Name from piece 3 to field 2.01, and group Number from piece 4 to field 2.02  ... ;N Z  ... N GNUM,GNAM,EPTYP  ... ;S Z=$G(^IBA(355.3,+$P(IBX,U,13),0))  ... ;D WRT(1," "\_$E($P(Z,U,3)\_$J("",20),1,20)\_" "\_$E($P(Z,U,4)\_$J("",17),1,17)\_" "\_$$EXTERNAL^DILFD(355.3,.15,"",$P(Z,U,15)))  ... S GNUM=$$GET1^DIQ(355.3,+$P(IBX,U,13)\_",",2.02) ;Group Number  ... S GNAM=$$GET1^DIQ(355.3,+$P(IBX,U,13)\_",",2.01) ;Group Name  ... S EPTYP=$$GET1^DIQ(355.3,+$P(IBX,U,13)\_",",.15) ;Electronic Plan Type  ... D WRT(1," "\_$E(GNUM\_$J("",20),1,20)\_" "\_$E(GNAM\_$J("",17),1,17)\_" "\_EPTYP)  .. S:'IBICONT IBICONT=1  ;  I 'IBSTOP D  ;Totals  . N Z  . S Z=$G(^XTMP("IB\_PLAN232"))  . I ($Y+10)>IOSL!'IBPG D HDR(.IBPG,.IBSTOP,"") Q:IBSTOP  . D WRT(2,$J("",25)\_"TOTAL # OF IDs CHECKED: "\_+$P(Z,U,4))  . D WRT(1,$J("",14)\_"TOT # BLUE CROSS/SHIELD IDS FOUND: "\_+$P(Z,U,5))  . D WRT(1,"TOTAL # OF INS CO. W/BLUE PLANS AND NO BLUE IDS: "\_+$P(Z,U,3))  . D WRT(1,$J("",21)\_"TOTAL # OF ERRORS/WARNINGS: "\_+$P(Z,U,6))  ;  I '$D(ZTQUEUED) D ^%ZISC I 'IBSTOP,IBPG D ASK()  I $D(ZTQUEUED),'IBSTOP S ZTREQ="@"  I $G(^TMP($J,"SENDMSG")),'IBSTOP D  . N XMDUZ,XMSUBJ,XMBODY,XMTO,XMZ  . S XMDUZ=DUZ,XMSUBJ=$E("PROVIDER ID QUERY FROM "\_$P($G(^DIC(4,+$P($G(^IBE(350.9,1,0)),U,2),0)),U),1,65),XMBODY="^TMP($J,""SENDMSG"",1)"  . M XMTO=^TMP($J,"SENDMSG",0)  . S Z="" F  S Z=$O(^TMP($J,"SENDMSG",0,Z)) Q:Z=""  S XMZ(Z)=""  . D SENDMSG^XMXAPI(XMDUZ,XMSUBJ,XMBODY,.XMTO,"",.XMZ)  K ^TMP($J,"IBZ232"),^TMP($J,"SENDMSG")  Q  ; HDR(IBPG,IBSTOP,IBZ,FF) ; Main hdr  ; FF = 0 if continuation pg so it writes it to report, but not mail msg  N Z,IBT  Q:$G(IBSTOP)  I $D(ZTQUEUED),$$S^%ZTLOAD S (IBSTOP,ZTSTOP)=1 K ZTREQ I +$G(IBPG) D WRT(2,"\*\*\*TASK STOPPED BY USER\*\*\*") Q  I IBPG&($E(IOST,1,2)="C-") D ASK(.IBSTOP) Q:IBSTOP  S IBT=$S(IBPG:1,1:0)  S IBPG=IBPG+1  S Z="PROVIDER ID VERIFICATION QUERY REPORT"  S Z=$$SETSTR^VALM1($J("",80-$L(Z)\2)\_Z,"",1,79)  S Z=$$SETSTR^VALM1("Page: "\_IBPG,Z,70,10)  D WRT(0,"@IOF",$G(FF))  D WRT(1,Z,$G(FF))  S Z="RUN DATE: "\_$$FMTE^XLFDT(DT,2),Z=$J("",80-$L(Z)\2)\_Z  D WRT(1,Z,$G(FF))  I IBZ'="",IBZ'=4,IBZ'=5 D  . D WRT(2," FORM TYPE PROV ID TYPE"\_$J("",12)\_"PROVIDER NAME (#=Non-VA)"\_$J("",6)\_"PROV ID",$G(FF))  I IBZ=4!(IBZ=5) D  . D WRT(2," GROUP NAME"\_$J("",12)\_"GROUP NUMBER"\_$J("",7)\_"ELECTRONIC PLAN TYPE",$G(FF))  D WRT(1,$TR($J("",IOM-1)," ","-"),$G(FF))  Q  ; HDR1(IBPG,IBSTOP,IBZ,IBCONT) ; Hdr err typ  N Z,Z0,Z1  D HDR(.IBPG,.IBSTOP,IBZ,IBCONT) Q:IBSTOP  S Z="",$P(Z,"\*",80)="" D WRT(1,Z,IBCONT)  S Z0="\* "\_$S(IBZ>1:"WARNING: ",1:"ERROR: ")  ;  I IBZ'=4,IBZ'=5 D  . N X  . S X="BLUE CROSS ID FOUND FOR A 1500 FORM TYPE ONLY^BLUE SHIELD ID FOUND FOR A UB-04 FORM TYPE ONLY^BLUE CROSS ID FOUND FOR BOTH FORM TYPES^BL CROSS/BL SHIELD IDs FOUND FOR PLANS NOT HAVING 'BL' ELECTRONIC PLAN TYPE"  . S Z0=Z0\_$S(IBZ<6:$P(X,U,IBZ),IBZ=6:"""VAD000"" PROVIDER ID FOUND NOT SET UP AS A UPIN PROVIDER ID TYPE",1:"")  I IBZ=4 D  . S Z0=Z0\_"BL CROSS/BL SHIELD IDs FOUND FOR PLANS NOT HAVING 'BL' ELECTRONIC" D WRT(1,Z0\_$J("",78-$L(Z0))\_"\*",IBCONT)  . S Z0="\*"\_$J("",10)\_"PLAN TYPE"  ;  I IBZ=5 D  . S Z0=Z0\_"INSURANCE CO HAS BL CROSS/SHIELD PLANS, BUT NO BL CROSS/SHIELD IDs"  ;  S Z0=Z0\_$S(IBCONT:" (CONT)",1:"")  D WRT(1,Z0\_$J("",78-$L(Z0))\_"\*",IBCONT)  ;  I 'IBCONT D  . I IBZ=1 D  .. D WRT(1,"\*"\_$J("",77)\_"\*",IBCONT)  .. S Z0="\* SOLUTION: THIS ID WILL NEVER BE USED ELECTRONICALLY." D WRT(1,Z0\_$J("",78-$L(Z0))\_"\*",IBCONT)  .. S Z0="\*"\_$J("",11)\_"CHANGE PROVIDER ID TYPE TO BLUE SHIELD IF THIS ID SHOULD BE" D WRT(1,Z0\_$J("",78-$L(Z0))\_"\*",IBCONT)  .. S Z0="\*"\_$J("",11)\_"TRANSMITTED ON A 1500." D WRT(1,Z0\_$J("",78-$L(Z0))\_"\*",IBCONT)  . ;  . I IBZ=2 D  .. D WRT(1,"\*"\_$J("",77)\_"\*",IBCONT)  .. S Z0="\* SUGGESTION: VISTA WILL TRANSMIT THIS ID ELECTRONICALLY, BUT IT IS OPTIMAL"  D WRT(1,Z0\_$J("",78-$L(Z0))\_"\*",IBCONT)  .. S Z0="\*"\_$J("",13)\_"TO HAVE THIS ID SET UP AS BLUE CROSS." D WRT(1,Z0\_$J("",78-$L(Z0))\_"\*",IBCONT)  . ;  . I IBZ=3 D  .. D WRT(1,"\*"\_$J("",77)\_"\*",IBCONT)  .. S Z0="\* SUGGESTION: A BLUE CROSS ID CAN ONLY BE APPLIED TO A UB-04 FORM TYPE." D WRT(1,Z0\_$J("",78-$L(Z0))\_"\*",IBCONT)  .. S Z0="\*"\_$J("",13)\_"EDIT THE 'APPLIED TO FORM TYPE' FOR THE ID TO BE UB-04 ONLY." D WRT(1,Z0\_$J("",78-$L(Z0))\_"\*",IBCONT)  .. S Z0="\*"\_$J("",13)\_"IF YOU NEED THIS ID ON A 1500, SET IT UP AS A BLUE SHIELD ID" D WRT(1,Z0\_$J("",78-$L(Z0))\_"\*",IBCONT)  .. S Z0="\*"\_$J("",13)\_"APPLIED TO A CMS-1500 FORM TYPE." D WRT(1,Z0\_$J("",78-$L(Z0))\_"\*",IBCONT)  . ;  . I IBZ=4 D  .. D WRT(1,"\*"\_$J("",77)\_"\*",IBCONT)  .. S Z0="\* SUGGESTION: A BLUE CROSS OR BLUE SHIELD ID IS DEFINED FOR THE INSURANCE" D WRT(1,Z0\_$J("",78-$L(Z0))\_"\*",IBCONT)  .. S Z0="\*"\_$J("",13)\_"COMPANY, BUT THE ELECTRONIC PLAN TYPE FOR ONE OR MORE OF THE" D WRT(1,Z0\_$J("",78-$L(Z0))\_"\*",IBCONT)  .. S Z0="\*"\_$J("",13)\_"COMPANY'S PLANS IS NOT SET TO 'BL' (BLUE CROSS/BLUE SHIELD)." D WRT(1,Z0\_$J("",78-$L(Z0))\_"\*",IBCONT)  .. S Z0="\*"\_$J("",13)\_"IF BLUE CROSS/BLUE SHIELD IDs ARE NEEDED TO PRINT FOR ANY" D WRT(1,Z0\_$J("",78-$L(Z0))\_"\*",IBCONT)  .. S Z0="\*"\_$J("",13)\_"OF THESE PLANS, ITS ELECTRONIC PLAN TYPE MUST BE CHANGED TO BL." D WRT(1,Z0\_$J("",78-$L(Z0))\_"\*",IBCONT)  . ;  . I IBZ=5 D  .. D WRT(1,"\*"\_$J("",77)\_"\*",IBCONT)  .. S Z0="\* SUGGESTION: A BLUE CROSS OR BLUE SHIELD PLAN IS DEFINED FOR THE INSURANCE" D WRT(1,Z0\_$J("",78-$L(Z0))\_"\*",IBCONT)  .. S Z0="\*"\_$J("",13)\_"COMPANY, BUT YOU HAVE ONLY NON-BLUE CROSS/SHIELD IDS SET UP." D WRT(1,Z0\_$J("",78-$L(Z0))\_"\*",IBCONT)  .. S Z0="\*"\_$J("",13)\_"YOU MUST SET UP THE APPROPRIATE BLUE CROSS/BLUE SHIELD IDs" D WRT(1,Z0\_$J("",78-$L(Z0))\_"\*",IBCONT)  .. S Z0="\*"\_$J("",13)\_"FOR THE INSURANCE COMPANY." D WRT(1,Z0\_$J("",78-$L(Z0))\_"\*",IBCONT)  . ;  . I IBZ=6 D  .. D WRT(1,"\*"\_$J("",77)\_"\*",IBCONT)  .. S Z0="\* SUGGESTION: CHANGE PROVIDER ID TYPE TO UPIN."  .. D WRT(1,Z0\_$J("",78-$L(Z0))\_"\*",IBCONT)  .. S Z0="\*"\_$J("",13)\_"ONCE ALL PAYERS FULLY IMPLEMENT HIPAA EDITS, YOU"  .. D WRT(1,Z0\_$J("",78-$L(Z0))\_"\*",IBCONT)  .. S Z0="\*"\_$J("",13)\_"MUST USE THE CORRECT ID TYPE FOR THE ID ENTERED."  .. D WRT(1,Z0\_$J("",78-$L(Z0))\_"\*",IBCONT)  . ;  . D WRT(1,"\*"\_$J("",77)\_"\*",IBCONT)  . S Z1="\*"\_$J("",$S(IBZ'=1:13,1:11))\_"VISTA OPTION TO USE: "  . I IBZ'=4 D  .. S Z0=Z1\_"PROVIDER ID MAINTENANCE"  . I IBZ=4 D  .. S Z0=Z1\_"INSURANCE COMPANY ENTRY/EDIT"  . D WRT(1,Z0\_$J("",78-$L(Z0))\_"\*",IBCONT)  ;  D WRT(1,Z,IBCONT)  ;  I '$O(^TMP($J,"IBZ232",IBZ,0)) D WRT(2,"\*\*\*\*\* NOTHING FOUND FOR THIS ERROR/WARNING \*\*\*\*\*",IBCONT)  Q  ; INSHDR(IBPG,IBSTOP,IBINM,IBZ,IBICONT) ; Ins Co info  I ($Y+7)>IOSL D HDR1(.IBPG,.IBSTOP,IBZ,1)  Q:IBSTOP  D WRT(2,"INSURANCE CO NAME: "\_IBINM\_$S($G(IBICONT):" (Continued)",1:""),IBICONT)  Q  ; ASK(IBSTOP) ; Ask continue  ; If passed by ref, IBSTOP returned = 1 if print aborted  I $E(IOST,1,2)'["C-" Q  N DIR,DIROUT,DIRUT,DTOUT,DUOUT  S DIR(0)="E" W ! D ^DIR  I ($D(DIRUT))!($D(DUOUT)) S IBSTOP=1 Q  Q  ; WRT(FF,TEXT,NOT) ; Wrt/store line  N Z,A  S A=+$O(^TMP($J,"SENDMSG",1,""),-1),NOT=$G(NOT)  I FF F Z=1:1:FF W ! I $G(^TMP($J,"SENDMSG")),'NOT,Z>1 S A=A+1,^TMP($J,"SENDMSG",1,A)=" "  ;  I TEXT="@IOF" D  Q  . W @IOF  . I $G(^TMP($J,"SENDMSG")),'NOT,IBPG>1 D  .. S A=A+1,^TMP($J,"SENDMSG",1,A)=" "  .. F Z=1:1:2 S A=A+1,^TMP($J,"SENDMSG",1,A)="\*\*\* TOP OF NEW PAGE \*\*\*"  .. S A=A+1,^TMP($J,"SENDMSG",1,A)=" "  ;  W TEXT  I $G(^TMP($J,"SENDMSG")),'NOT S A=A+1,^TMP($J,"SENDMSG",1,A)=TEXT  Q  ; | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCEQ1A ;ALB/BSL,TMK - PROVIDER ID QUERY REPORT ;25-AUG-03  ;;2.0;INTEGRATED BILLING;\*\*232,348,349,516,**592**\*\*;21-MAR-94;Build 123  ;;Per VA Directive 6402, this routine should not be modified.  ; RPTOUT ; Print from data in ^XTMP  N IBP,IBA,IBI,IBIN,IBPNM,IBPNUM,IBSTOP,IBX,IBZ,IBPG,IBICONT,Z  K ^TMP($J,"IBZ232")  F Z=1:1:6 S ^TMP($J,"IBZ232",Z)=""  S (IBPG,IBSTOP)=0  S IBA=0 F  S IBA=$O(^XTMP("IB\_PLAN232",1,IBA)) Q:'IBA  D  . S IBX=$G(^XTMP("IB\_PLAN232",1,IBA,0))  . ; Sort by err type, ins co ien  . S ^TMP($J,"IBZ232",+$P(IBX,U,16),+$P(IBX,U,11),IBA)=IBX  ;  S IBZ=0 F  S IBZ=$O(^TMP($J,"IBZ232",IBZ)) Q:'IBZ!IBSTOP!(IBZ>6) D HDR1(.IBPG,.IBSTOP,IBZ,0) S IBI=0 F  S IBI=$O(^TMP($J,"IBZ232",IBZ,IBI)) Q:'IBI!IBSTOP  D  . S IBIN=$P($G(^DIC(36,+IBI,0)),U)\_" ("\_$S(+$G(^(3))=1:"",1:"NOT ")\_"SET TO TRANSMIT LIVE)"  . D INSHDR(.IBPG,.IBSTOP,IBIN,IBZ,0) S IBICONT=0  . S IBA=0 F  S IBA=$O(^TMP($J,"IBZ232",IBZ,IBI,IBA)) Q:'IBA!IBSTOP  S IBX=$G(^(IBA)) D  .. I ($Y+5)>IOSL D INSHDR(.IBPG,.IBSTOP,IBIN,IBZ,IBICONT) Q:IBSTOP  .. ;  .. I IBZ'=4,IBZ'=5 D  ... S IBP=+$P(IBX,U,14)  ... I $P(IBX,U,14)[".91" S IBPNM="ALL PROVIDERS"  ... I $P(IBX,U,14)'[".91" D  .... N Z  .... S Z=$P($G(^IBA(355.9,IBP,0)),U)  .... S IBPNM=$S(Z["VA(200":"",1:"#")\_$$EXTERNAL^DILFD(355.9,.01,"",Z)  ... S IBPNUM=$P($G(^IBA(+$P($P(IBX,U,14),";",2),IBP,0)),U,7) **... ; JWS;IB\*2.0\*592: Dental form type J430D, handle same as CMS-1500**  ... D WRT(1," "\_$E($P("ALL^UB-04^CMS-1500**&J430D**",U,$P(IBX,U,4)+1)\_$J("",9),1,9)\_" "\_$E($P(IBX,U,15)\_$J("",23),1,23)\_" "\_$E(IBPNM\_$J("",28),1,28)\_" "\_$E(IBPNUM,1,11))  .. ;  .. I IBZ=4!(IBZ=5) D  ... ;IB\*516/TAZ - Change Group Name from piece 3 to field 2.01, and group Number from piece 4 to field 2.02  ... ;N Z  ... N GNUM,GNAM,EPTYP  ... ;S Z=$G(^IBA(355.3,+$P(IBX,U,13),0))  ... ;D WRT(1," "\_$E($P(Z,U,3)\_$J("",20),1,20)\_" "\_$E($P(Z,U,4)\_$J("",17),1,17)\_" "\_$$EXTERNAL^DILFD(355.3,.15,"",$P(Z,U,15)))  ... S GNUM=$$GET1^DIQ(355.3,+$P(IBX,U,13)\_",",2.02) ;Group Number  ... S GNAM=$$GET1^DIQ(355.3,+$P(IBX,U,13)\_",",2.01) ;Group Name  ... S EPTYP=$$GET1^DIQ(355.3,+$P(IBX,U,13)\_",",.15) ;Electronic Plan Type  ... D WRT(1," "\_$E(GNUM\_$J("",20),1,20)\_" "\_$E(GNAM\_$J("",17),1,17)\_" "\_EPTYP)  .. S:'IBICONT IBICONT=1  ;  I 'IBSTOP D  ;Totals  . N Z  . S Z=$G(^XTMP("IB\_PLAN232"))  . I ($Y+10)>IOSL!'IBPG D HDR(.IBPG,.IBSTOP,"") Q:IBSTOP  . D WRT(2,$J("",25)\_"TOTAL # OF IDs CHECKED: "\_+$P(Z,U,4))  . D WRT(1,$J("",14)\_"TOT # BLUE CROSS/SHIELD IDS FOUND: "\_+$P(Z,U,5))  . D WRT(1,"TOTAL # OF INS CO. W/BLUE PLANS AND NO BLUE IDS: "\_+$P(Z,U,3))  . D WRT(1,$J("",21)\_"TOTAL # OF ERRORS/WARNINGS: "\_+$P(Z,U,6))  ;  I '$D(ZTQUEUED) D ^%ZISC I 'IBSTOP,IBPG D ASK()  I $D(ZTQUEUED),'IBSTOP S ZTREQ="@"  I $G(^TMP($J,"SENDMSG")),'IBSTOP D  . N XMDUZ,XMSUBJ,XMBODY,XMTO,XMZ  . S XMDUZ=DUZ,XMSUBJ=$E("PROVIDER ID QUERY FROM "\_$P($G(^DIC(4,+$P($G(^IBE(350.9,1,0)),U,2),0)),U),1,65),XMBODY="^TMP($J,""SENDMSG"",1)"  . M XMTO=^TMP($J,"SENDMSG",0)  . S Z="" F  S Z=$O(^TMP($J,"SENDMSG",0,Z)) Q:Z=""  S XMZ(Z)=""  . D SENDMSG^XMXAPI(XMDUZ,XMSUBJ,XMBODY,.XMTO,"",.XMZ)  K ^TMP($J,"IBZ232"),^TMP($J,"SENDMSG")  Q  ; HDR(IBPG,IBSTOP,IBZ,FF) ; Main hdr  ; FF = 0 if continuation pg so it writes it to report, but not mail msg  N Z,IBT  Q:$G(IBSTOP)  I $D(ZTQUEUED),$$S^%ZTLOAD S (IBSTOP,ZTSTOP)=1 K ZTREQ I +$G(IBPG) D WRT(2,"\*\*\*TASK STOPPED BY USER\*\*\*") Q  I IBPG&($E(IOST,1,2)="C-") D ASK(.IBSTOP) Q:IBSTOP  S IBT=$S(IBPG:1,1:0)  S IBPG=IBPG+1  S Z="PROVIDER ID VERIFICATION QUERY REPORT"  S Z=$$SETSTR^VALM1($J("",80-$L(Z)\2)\_Z,"",1,79)  S Z=$$SETSTR^VALM1("Page: "\_IBPG,Z,70,10)  D WRT(0,"@IOF",$G(FF))  D WRT(1,Z,$G(FF))  S Z="RUN DATE: "\_$$FMTE^XLFDT(DT,2),Z=$J("",80-$L(Z)\2)\_Z  D WRT(1,Z,$G(FF))  I IBZ'="",IBZ'=4,IBZ'=5 D  . D WRT(2," FORM TYPE PROV ID TYPE"\_$J("",12)\_"PROVIDER NAME (#=Non-VA)"\_$J("",6)\_"PROV ID",$G(FF))  I IBZ=4!(IBZ=5) D  . D WRT(2," GROUP NAME"\_$J("",12)\_"GROUP NUMBER"\_$J("",7)\_"ELECTRONIC PLAN TYPE",$G(FF))  D WRT(1,$TR($J("",IOM-1)," ","-"),$G(FF))  Q  ; HDR1(IBPG,IBSTOP,IBZ,IBCONT) ; Hdr err typ  N Z,Z0,Z1  D HDR(.IBPG,.IBSTOP,IBZ,IBCONT) Q:IBSTOP  S Z="",$P(Z,"\*",80)="" D WRT(1,Z,IBCONT)  S Z0="\* "\_$S(IBZ>1:"WARNING: ",1:"ERROR: ")  ;  I IBZ'=4,IBZ'=5 D  . N X  **. ;JWS;IB\*2.0\*592: Dental form type J430D**  . S X="BLUE CROSS ID FOUND FOR A 1500**&J430D** FORM TYPES ONLY^BLUE SHIELD ID FOUND FOR A UB-04 FORM TYPE ONLY^BLUE CROSS ID FOUND FOR ALL FORM TYPES^BL CROSS/BL SHIELD IDs FOUND FOR PLANS NOT HAVING 'BL' ELECTRONIC PLAN TYPE"  . S Z0=Z0\_$S(IBZ<6:$P(X,U,IBZ),IBZ=6:"""VAD000"" PROVIDER ID FOUND NOT SET UP AS A UPIN PROVIDER ID TYPE",1:"")  I IBZ=4 D  . S Z0=Z0\_"BL CROSS/BL SHIELD IDs FOUND FOR PLANS NOT HAVING 'BL' ELECTRONIC" D WRT(1,Z0\_$J("",78-$L(Z0))\_"\*",IBCONT)  . S Z0="\*"\_$J("",10)\_"PLAN TYPE"  ;  I IBZ=5 D  . S Z0=Z0\_"INSURANCE CO HAS BL CROSS/SHIELD PLANS, BUT NO BL CROSS/SHIELD IDs"  ;  S Z0=Z0\_$S(IBCONT:" (CONT)",1:"")  D WRT(1,Z0\_$J("",78-$L(Z0))\_"\*",IBCONT)  ;  I 'IBCONT D  . I IBZ=1 D  .. D WRT(1,"\*"\_$J("",77)\_"\*",IBCONT)  .. S Z0="\* SOLUTION: THIS ID WILL NEVER BE USED ELECTRONICALLY." D WRT(1,Z0\_$J("",78-$L(Z0))\_"\*",IBCONT)  .. S Z0="\*"\_$J("",11)\_"CHANGE PROVIDER ID TYPE TO BLUE SHIELD IF THIS ID SHOULD BE" D WRT(1,Z0\_$J("",78-$L(Z0))\_"\*",IBCONT) **.. ;JWS;IB\*2.0\*592: Dental form type J430D**  .. S Z0="\*"\_$J("",11)\_"TRANSMITTED ON A 1500 **or J430D**." D WRT(1,Z0\_$J("",78-$L(Z0))\_"\*",IBCONT)  . ;  . I IBZ=2 D  .. D WRT(1,"\*"\_$J("",77)\_"\*",IBCONT)  .. S Z0="\* SUGGESTION: VISTA WILL TRANSMIT THIS ID ELECTRONICALLY, BUT IT IS OPTIMAL"  D WRT(1,Z0\_$J("",78-$L(Z0))\_"\*",IBCONT)  .. S Z0="\*"\_$J("",13)\_"TO HAVE THIS ID SET UP AS BLUE CROSS." D WRT(1,Z0\_$J("",78-$L(Z0))\_"\*",IBCONT)  . ;  . I IBZ=3 D  .. D WRT(1,"\*"\_$J("",77)\_"\*",IBCONT)  .. S Z0="\* SUGGESTION: A BLUE CROSS ID CAN ONLY BE APPLIED TO A UB-04 FORM TYPE." D WRT(1,Z0\_$J("",78-$L(Z0))\_"\*",IBCONT)  .. S Z0="\*"\_$J("",13)\_"EDIT THE 'APPLIED TO FORM TYPE' FOR THE ID TO BE UB-04 ONLY." D WRT(1,Z0\_$J("",78-$L(Z0))\_"\*",IBCONT)  **.. ;JWS;IB\*2.0\*592: Dental form type J430D**  .. S Z0="\*"\_$J("",13)\_"IF YOU NEED THIS ID ON A 1500 **or J430D**, SET IT UP AS A BLUE SHIELD ID" D WRT(1,Z0\_$J("",78-$L(Z0))\_"\*",IBCONT)  .. S Z0="\*"\_$J("",13)\_"APPLIED TO A CMS-1500 and J430D FORM TYPE." D WRT(1,Z0\_$J("",78-$L(Z0))\_"\*",IBCONT)  . ;  . I IBZ=4 D  .. D WRT(1,"\*"\_$J("",77)\_"\*",IBCONT)  .. S Z0="\* SUGGESTION: A BLUE CROSS OR BLUE SHIELD ID IS DEFINED FOR THE INSURANCE" D WRT(1,Z0\_$J("",78-$L(Z0))\_"\*",IBCONT)  .. S Z0="\*"\_$J("",13)\_"COMPANY, BUT THE ELECTRONIC PLAN TYPE FOR ONE OR MORE OF THE" D WRT(1,Z0\_$J("",78-$L(Z0))\_"\*",IBCONT)  .. S Z0="\*"\_$J("",13)\_"COMPANY'S PLANS IS NOT SET TO 'BL' (BLUE CROSS/BLUE SHIELD)." D WRT(1,Z0\_$J("",78-$L(Z0))\_"\*",IBCONT)  .. S Z0="\*"\_$J("",13)\_"IF BLUE CROSS/BLUE SHIELD IDs ARE NEEDED TO PRINT FOR ANY" D WRT(1,Z0\_$J("",78-$L(Z0))\_"\*",IBCONT)  .. S Z0="\*"\_$J("",13)\_"OF THESE PLANS, ITS ELECTRONIC PLAN TYPE MUST BE CHANGED TO BL." D WRT(1,Z0\_$J("",78-$L(Z0))\_"\*",IBCONT)  . ;  . I IBZ=5 D  .. D WRT(1,"\*"\_$J("",77)\_"\*",IBCONT)  .. S Z0="\* SUGGESTION: A BLUE CROSS OR BLUE SHIELD PLAN IS DEFINED FOR THE INSURANCE" D WRT(1,Z0\_$J("",78-$L(Z0))\_"\*",IBCONT)  .. S Z0="\*"\_$J("",13)\_"COMPANY, BUT YOU HAVE ONLY NON-BLUE CROSS/SHIELD IDS SET UP." D WRT(1,Z0\_$J("",78-$L(Z0))\_"\*",IBCONT)  .. S Z0="\*"\_$J("",13)\_"YOU MUST SET UP THE APPROPRIATE BLUE CROSS/BLUE SHIELD IDs" D WRT(1,Z0\_$J("",78-$L(Z0))\_"\*",IBCONT)  .. S Z0="\*"\_$J("",13)\_"FOR THE INSURANCE COMPANY." D WRT(1,Z0\_$J("",78-$L(Z0))\_"\*",IBCONT)  . ;  . I IBZ=6 D  .. D WRT(1,"\*"\_$J("",77)\_"\*",IBCONT)  .. S Z0="\* SUGGESTION: CHANGE PROVIDER ID TYPE TO UPIN."  .. D WRT(1,Z0\_$J("",78-$L(Z0))\_"\*",IBCONT)  .. S Z0="\*"\_$J("",13)\_"ONCE ALL PAYERS FULLY IMPLEMENT HIPAA EDITS, YOU"  .. D WRT(1,Z0\_$J("",78-$L(Z0))\_"\*",IBCONT)  .. S Z0="\*"\_$J("",13)\_"MUST USE THE CORRECT ID TYPE FOR THE ID ENTERED."  .. D WRT(1,Z0\_$J("",78-$L(Z0))\_"\*",IBCONT)  . ;  . D WRT(1,"\*"\_$J("",77)\_"\*",IBCONT)  . S Z1="\*"\_$J("",$S(IBZ'=1:13,1:11))\_"VISTA OPTION TO USE: "  . I IBZ'=4 D  .. S Z0=Z1\_"PROVIDER ID MAINTENANCE"  . I IBZ=4 D  .. S Z0=Z1\_"INSURANCE COMPANY ENTRY/EDIT"  . D WRT(1,Z0\_$J("",78-$L(Z0))\_"\*",IBCONT)  ;  D WRT(1,Z,IBCONT)  ;  I '$O(^TMP($J,"IBZ232",IBZ,0)) D WRT(2,"\*\*\*\*\* NOTHING FOUND FOR THIS ERROR/WARNING \*\*\*\*\*",IBCONT)  Q  ; INSHDR(IBPG,IBSTOP,IBINM,IBZ,IBICONT) ; Ins Co info  I ($Y+7)>IOSL D HDR1(.IBPG,.IBSTOP,IBZ,1)  Q:IBSTOP  D WRT(2,"INSURANCE CO NAME: "\_IBINM\_$S($G(IBICONT):" (Continued)",1:""),IBICONT)  Q  ; ASK(IBSTOP) ; Ask continue  ; If passed by ref, IBSTOP returned = 1 if print aborted  I $E(IOST,1,2)'["C-" Q  N DIR,DIROUT,DIRUT,DTOUT,DUOUT  S DIR(0)="E" W ! D ^DIR  I ($D(DIRUT))!($D(DUOUT)) S IBSTOP=1 Q  Q  ; WRT(FF,TEXT,NOT) ; Wrt/store line  N Z,A  S A=+$O(^TMP($J,"SENDMSG",1,""),-1),NOT=$G(NOT)  I FF F Z=1:1:FF W ! I $G(^TMP($J,"SENDMSG")),'NOT,Z>1 S A=A+1,^TMP($J,"SENDMSG",1,A)=" "  ;  I TEXT="@IOF" D  Q  . W @IOF  . I $G(^TMP($J,"SENDMSG")),'NOT,IBPG>1 D  .. S A=A+1,^TMP($J,"SENDMSG",1,A)=" "  .. F Z=1:1:2 S A=A+1,^TMP($J,"SENDMSG",1,A)="\*\*\* TOP OF NEW PAGE \*\*\*"  .. S A=A+1,^TMP($J,"SENDMSG",1,A)=" "  ;  W TEXT  I $G(^TMP($J,"SENDMSG")),'NOT S A=A+1,^TMP($J,"SENDMSG",1,A)=TEXT  Q  ; | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCERP6 | | | | | | | | |
| **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** | | | | | | | | | |
| IBCERP6 ;ALB/JEH - MRA/EDI CLAIMS READY FOR EXTRACT ;12/10/99  ;;2.0;INTEGRATED BILLING;\*\*137,211,155,348,349\*\*;21-MAR-94;Build 46  ;;Per VHA Directive 2004-038, this routine should not be modified.  ; EN ;Entry point from option  W !!,"This report provides a list of claims held in a"  W !,"Ready for Extract status. Users can select all bills"  W !,"in a Ready for extract status or only those trapped due to"  W !,"the EDI/MRA Parameters being turned off."  ;  S IBQUIT=0 D SELECT I IBQUIT G ENQ1  S IBQUIT=0 D PARAM I IBQUIT G ENQ1  ;  W !!,"This report requires a 132 column printer.",!!  ; - Ask device  N %ZIS,ZTRTN,ZTSAVE,ZTDESC  S %ZIS="QM" D ^%ZIS G:POP ENQ1  I $D(IO("Q")) D  G ENQ1  .S ZTRTN="BLD^IBCERP6",ZTDESC="IB - EDI/MRA Claims in Waiting Transmission Status"  .S ZTSAVE("IB\*")=""  .D ^%ZTLOAD  .W !!,$S($D(ZTSK):"Your task number"\_ZTSK\_" has been queued.",1:"Unable to queue this job.")  .K ZTSK,IO("Q") D HOME^%ZIS  U IO  ; BLD ; - Tasked entry point  ;  N IBSTAT,IBILL,IBREC,IBIFN,IBSTAT,IBVSIT,IBCAT,IBI,IBINS,IBPREC,IBEVDT,IBTYP,IBPG,IBCHK  K ^TMP("IBCERP6",$J)  S (IBI,IBIFN)=0 F  S IBI=$O(^IBA(364,"ASTAT","X",IBI)) Q:'IBI  S IBIFN=+$G(^IBA(364,IBI,0)) D  .S IBQUIT=0  .S IBSTAT=$$WNRBILL^IBEFUNC(IBIFN)  .I IBSEL=2 D  I IBQUIT Q  ..I 'IBSTAT,13[IBPARAM S IBQUIT=1 Q  ..I IBSTAT,23[IBPARAM S IBQUIT=1 Q  .S IBSTAT=$S(IBSTAT:"MRA",1:"EDI")  .S IBREC=$G(^DGCR(399,+IBIFN,0))  .S IBVSIT=$S($$INPAT^IBCEF(IBIFN,1)=1:"INP",1:"OPT")  .S IBCAT=$S($$FT^IBCEF(IBIFN)=3:"UB04",1:"1500")  .S IBILL=$$BN1^PRCAFN(IBIFN)  .S IBINS=$P($G(^DIC(36,+$$CURR^IBCEF2(IBIFN),0)),U)  .S IBPREC=$$PT^IBEFUNC(+$P(IBREC,U,2))  .S IBEVDT=$P($G(^DGCR(399,IBIFN,"U")),U) ;Statement from date  .;S IBTYP=$P(IBREC,U,24)\_U\_$P($G(^DGCR(399.1,+$P(IBREC,U,25),0)),U)\_U\_$P(IBREC,U,26)  .S IBTYP=$$GET1^DIQ(399,IBIFN,.24)\_U\_$$GET1^DIQ(399,IBIFN,.25)\_U\_$$GET1^DIQ(399,IBIFN,.26)  .S ^TMP("IBCERP6",$J,IBSTAT,IBILL)=IBILL\_U\_IBVSIT\_U\_IBCAT\_U\_$P(IBPREC,U)\_U\_$E($P(IBPREC,U,2),8,11)\_U\_IBEVDT\_U\_IBTYP\_U\_IBINS  ; PRINT ;Prints report  S (IBQUIT,IBPG,IBEDI,IBMRA,IBTOT)=0 D HDR  I '$D(^TMP("IBCERP6",$J)) W !!,"There are no "\_$S(IBPARAM=1:"EDI",IBPARAM=2:"MRA",1:"EDI/MRA")\_" records"\_$S(IBSEL=2:" trapped",1:"")\_" in a ready for extract status" G ENQ1  S IBSTAT="" F  S IBSTAT=$O(^TMP("IBCERP6",$J,IBSTAT)) Q:IBSTAT=""!(IBQUIT=1) D  .S IBILL="" F  S IBILL=$O(^TMP("IBCERP6",$J,IBSTAT,IBILL)) Q:IBILL=""!(IBQUIT=1) S IBREC=^(IBILL) D  ..I ($Y+5)>IOSL D  I IBQUIT Q  ...D ASK I IBQUIT Q  ...D HDR  ..;  ..W !,?2,$P(IBREC,U),?15,$P(IBREC,U,2),?22,$P(IBREC,U,3)  ..W ?28,$E($P(IBREC,U,4),1,20),?50,$P(IBREC,U,5)  ..W ?57,$$FMTE^XLFDT($P(IBREC,U,6)),?73,$E($P(IBREC,U,7),1,8)\_", "\_$E($P(IBREC,U,8),1,3)\_", "\_$E($P(IBREC,U,9),1,16),?110,$E($P(IBREC,U,10),1,20)  ..I IBSTAT="EDI" S IBEDI=IBEDI+1  ..E  S IBMRA=IBMRA+1  ..S IBTOT=IBTOT+1  W !!  I IBEDI>0 W !,?3,"Total EDI Bills ",IBEDI  I IBMRA>0 W !,?3,"Total MRA Bills ",IBMRA  W !!,?3,"Total bills ",IBTOT  K ^TMP("IBCERP6",$J)  I $D(ZTQUEUED) S ZTREQ="@"  I '$D(ZTQUEUED) D ^%ZISC ENQ1 K IBPARAM,IBQUIT,IBSEL,Y,IBEDI,IBMRA,IBTOT Q  ; PARAM ;  S IBPARAM=$P($G(^IBE(350.9,1,8)),U,10) ;Get MRA/EDI site parameter setting  I IBPARAM="" D  .W !!,"Your EDI/MRA site parameter setting is incomplete."  .W !,"Please contact your coordinator.",!  .S IBQUIT=1  ;  I IBSEL=2,IBPARAM=3 D  .W !!,"Your site parameters are set to allow both EDI and MRA"  .W !,"transmissions. There is no need to run this report.",!  .S IBQUIT=1  Q  ; HDR ;Prints report heading  ; IB\*2.0\*211  ;I $E(IOST,1,2)="C-" W @IOF,\*13  I $S(IBPG:1,1:$E(IOST,1,2)="C-") W @IOF,\*13  S IBPG=IBPG+1  W !!,?45,$S(IBSEL=2:"Trapped ",1:"")\_" Claims Ready for Extract",?90,$$FMTE^XLFDT(DT),?110,"Page: ",IBPG  W !!,?15,"Inpt/",?23,"Inst/",!,?4,"Bill #",?15,"Opt",?23,"Prof",?32,"Name"  W ?51,"SSN",?57,"Statement Date",?89,"Type",?110,"Insurance Co."  W !,$TR($J("",IOM)," ","=")  Q  ; ASK ;  I $E(IOST,1,2)'["C-" Q  N DIR,DIROUT,DIRUT,DTOUT,DUOUT  S DIR(0)="E" D ^DIR  I ($D(DIRUT))!($D(DUOUT)) S IBQUIT=1 Q  Q  ; SELECT ;Report selection  N DIR,DIROUT,DTOUT,DUOUT,DTOUT  S IBSEL=0  W !! S DIR("A",1)="Do you want to print a list of:"  S DIR("A",2)=""  S DIR("A",3)=" 1 - All bills in Ready for Extract status"  S DIR("A",4)=" 2 - Bills trapped due to EDI/MRA parameter being turned off"  S DIR("A",5)=""  S DIR(0)="SAXB^1:All bills;2:Trapped bills"  W !  S DIR("A")="Select Number: ",DIR("B")=1  D ^DIR  I +Y'>0 S IBQUIT=1 Q  S IBSEL=+Y  Q | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCERP6 ;ALB/JEH - MRA/EDI CLAIMS READY FOR EXTRACT ;12/10/99  ;;2.0;INTEGRATED BILLING;\*\*137,211,155,348,349**,592**\*\*;21-MAR-94;Build 46  ;;Per VHA Directive 2004-038, this routine should not be modified.  ; EN ;Entry point from option  W !!,"This report provides a list of claims held in a"  W !,"Ready for Extract status. Users can select all bills"  W !,"in a Ready for extract status or only those trapped due to"  W !,"the EDI/MRA Parameters being turned off."  ;  S IBQUIT=0 D SELECT I IBQUIT G ENQ1  S IBQUIT=0 D PARAM I IBQUIT G ENQ1  ;  W !!,"This report requires a 132 column printer.",!!  ; - Ask device  N %ZIS,ZTRTN,ZTSAVE,ZTDESC  S %ZIS="QM" D ^%ZIS G:POP ENQ1  I $D(IO("Q")) D  G ENQ1  .S ZTRTN="BLD^IBCERP6",ZTDESC="IB - EDI/MRA Claims in Waiting Transmission Status"  .S ZTSAVE("IB\*")=""  .D ^%ZTLOAD  .W !!,$S($D(ZTSK):"Your task number"\_ZTSK\_" has been queued.",1:"Unable to queue this job.")  .K ZTSK,IO("Q") D HOME^%ZIS  U IO  ; BLD ; - Tasked entry point  ;  N IBSTAT,IBILL,IBREC,IBIFN,IBSTAT,IBVSIT,IBCAT,IBI,IBINS,IBPREC,IBEVDT,IBTYP,IBPG,IBCHK  K ^TMP("IBCERP6",$J)  S (IBI,IBIFN)=0 F  S IBI=$O(^IBA(364,"ASTAT","X",IBI)) Q:'IBI  S IBIFN=+$G(^IBA(364,IBI,0)) D  .S IBQUIT=0  .S IBSTAT=$$WNRBILL^IBEFUNC(IBIFN)  .I IBSEL=2 D  I IBQUIT Q  ..I 'IBSTAT,13[IBPARAM S IBQUIT=1 Q  ..I IBSTAT,23[IBPARAM S IBQUIT=1 Q  .S IBSTAT=$S(IBSTAT:"MRA",1:"EDI")  .S IBREC=$G(^DGCR(399,+IBIFN,0))  .S IBVSIT=$S($$INPAT^IBCEF(IBIFN,1)=1:"INP",1:"OPT") **.;JWS;IB\*2.0\*592:Dental form 7**  .S IBCAT=$S($$FT^IBCEF(IBIFN)=3:"UB04**",$$FT^IBCEF(IBIFN)=7:"J430D",**1:"1500")  .S IBILL=$$BN1^PRCAFN(IBIFN)  .S IBINS=$P($G(^DIC(36,+$$CURR^IBCEF2(IBIFN),0)),U)  .S IBPREC=$$PT^IBEFUNC(+$P(IBREC,U,2))  .S IBEVDT=$P($G(^DGCR(399,IBIFN,"U")),U) ;Statement from date  .;S IBTYP=$P(IBREC,U,24)\_U\_$P($G(^DGCR(399.1,+$P(IBREC,U,25),0)),U)\_U\_$P(IBREC,U,26)  .S IBTYP=$$GET1^DIQ(399,IBIFN,.24)\_U\_$$GET1^DIQ(399,IBIFN,.25)\_U\_$$GET1^DIQ(399,IBIFN,.26)  .S ^TMP("IBCERP6",$J,IBSTAT,IBILL)=IBILL\_U\_IBVSIT\_U\_IBCAT\_U\_$P(IBPREC,U)\_U\_$E($P(IBPREC,U,2),8,11)\_U\_IBEVDT\_U\_IBTYP\_U\_IBINS  ; PRINT ;Prints report  S (IBQUIT,IBPG,IBEDI,IBMRA,IBTOT)=0 D HDR  I '$D(^TMP("IBCERP6",$J)) W !!,"There are no "\_$S(IBPARAM=1:"EDI",IBPARAM=2:"MRA",1:"EDI/MRA")\_" records"\_$S(IBSEL=2:" trapped",1:"")\_" in a ready for extract status" G ENQ1  S IBSTAT="" F  S IBSTAT=$O(^TMP("IBCERP6",$J,IBSTAT)) Q:IBSTAT=""!(IBQUIT=1) D  .S IBILL="" F  S IBILL=$O(^TMP("IBCERP6",$J,IBSTAT,IBILL)) Q:IBILL=""!(IBQUIT=1) S IBREC=^(IBILL) D  ..I ($Y+5)>IOSL D  I IBQUIT Q  ...D ASK I IBQUIT Q  ...D HDR  ..;  ..W !,?2,$P(IBREC,U),?15,$P(IBREC,U,2),?22,$P(IBREC,U,3)  ..W ?28,$E($P(IBREC,U,4),1,20),?50,$P(IBREC,U,5)  ..W ?57,$$FMTE^XLFDT($P(IBREC,U,6)),?73,$E($P(IBREC,U,7),1,8)\_", "\_$E($P(IBREC,U,8),1,3)\_", "\_$E($P(IBREC,U,9),1,16),?110,$E($P(IBREC,U,10),1,20)  ..I IBSTAT="EDI" S IBEDI=IBEDI+1  ..E  S IBMRA=IBMRA+1  ..S IBTOT=IBTOT+1  W !!  I IBEDI>0 W !,?3,"Total EDI Bills ",IBEDI  I IBMRA>0 W !,?3,"Total MRA Bills ",IBMRA  W !!,?3,"Total bills ",IBTOT  K ^TMP("IBCERP6",$J)  I $D(ZTQUEUED) S ZTREQ="@"  I '$D(ZTQUEUED) D ^%ZISC ENQ1 K IBPARAM,IBQUIT,IBSEL,Y,IBEDI,IBMRA,IBTOT Q  ; PARAM ;  S IBPARAM=$P($G(^IBE(350.9,1,8)),U,10) ;Get MRA/EDI site parameter setting  I IBPARAM="" D  .W !!,"Your EDI/MRA site parameter setting is incomplete."  .W !,"Please contact your coordinator.",!  .S IBQUIT=1  ;  I IBSEL=2,IBPARAM=3 D  .W !!,"Your site parameters are set to allow both EDI and MRA"  .W !,"transmissions. There is no need to run this report.",!  .S IBQUIT=1  Q  ; HDR ;Prints report heading  ; IB\*2.0\*211  ;I $E(IOST,1,2)="C-" W @IOF,\*13  I $S(IBPG:1,1:$E(IOST,1,2)="C-") W @IOF,\*13  S IBPG=IBPG+1  W !!,?45,$S(IBSEL=2:"Trapped ",1:"")\_" Claims Ready for Extract",?90,$$FMTE^XLFDT(DT),?110,"Page: ",IBPG  W !!,?15,"Inpt/",?23,"Inst/",!,?4,"Bill #",?15,"Opt",?23,"Prof",?32,"Name"  W ?51,"SSN",?57,"Statement Date",?89,"Type",?110,"Insurance Co."  W !,$TR($J("",IOM)," ","=")  Q  ; ASK ;  I $E(IOST,1,2)'["C-" Q  N DIR,DIROUT,DIRUT,DTOUT,DUOUT  S DIR(0)="E" D ^DIR  I ($D(DIRUT))!($D(DUOUT)) S IBQUIT=1 Q  Q  ; SELECT ;Report selection  N DIR,DIROUT,DTOUT,DUOUT,DTOUT  S IBSEL=0  W !! S DIR("A",1)="Do you want to print a list of:"  S DIR("A",2)=""  S DIR("A",3)=" 1 - All bills in Ready for Extract status"  S DIR("A",4)=" 2 - Bills trapped due to EDI/MRA parameter being turned off"  S DIR("A",5)=""  S DIR(0)="SAXB^1:All bills;2:Trapped bills"  W !  S DIR("A")="Select Number: ",DIR("B")=1  D ^DIR  I +Y'>0 S IBQUIT=1 Q  S IBSEL=+Y  Q | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCEST | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** |  | | | | | | | | |
| **Related Options** | None | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  |  | | | | |  | | | |
| **Data Dictionary (DD) References** |  | | | | | | | | |
| **Related Protocols** | None | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Current Logic** | | | | | | | | | |
| IBCEST ;ALB/TMP - 837 EDI STATUS MESSAGE PROCESSING ;17-APR-96  ;;2.0;INTEGRATED BILLING;\*\*137,189,197,135,283,320,368,397,407\*\*;21-MAR-94;Build 29  ;;Per VHA Directive 2004-038, this routine should not be modified.  ; IA 4043 for call to AUDITX^PRCAUDT  Q  ; UPD361(IBTDA) ; Update IB BILL STATUS MESSAGES file  ; IBTDA = ien of return message in file 364.2  ;  N IB,IB0,IBSEQ,IB00,IBBILL,IBBTCH,IBMNUM  ;  I '$$LOCK^IBCEM(IBTDA) G UPDQ ;Lock message in file 364.2  ;  S IB0=$G(^IBA(364.2,IBTDA,0))  S IBMNUM=$P(IB0,U) ; Message number  S IB00=$G(^IBA(364,+$P(IB0,U,5),0)) ; Transmit bill entry  S IBBILL=+IB00 ; Actual bill ien in file 399  S IBBTCH=$P(IB0,U,4) ; Batch #  ;  ; Auto-audit bills based on status code on '10' record of status msg  ; flat file  I IBBILL,$P($T(PRCAUDT+1^PRCAUDT),"\*\*",2)[",173" D  . N Z,Z0,Z1,OK  . Q:+$$STA^PRCAFN(IBBILL)'=104  . S (Z,OK)=0  . F  S Z=$O(^IBA(364.2,IBTDA,2,Z)) Q:'Z  S Z0=$P($G(^(Z,0)),"##RAW DATA: ",2) I +Z0=10 S Z0=$P(Z0,U,5) D  Q:OK  .. ; Strip leading spaces  .. S Z0=$$TRIM^XLFSTR(Z0)  .. Q:Z0=""  .. I $$SCODE^IBCEST1(Z0),$P($G(^DGCR(399.3,+$P($G(^DGCR(399,IBBILL,0)),U,7),0)),U,11) D AUDITX^PRCAUDT(IBBILL) S OK=1 ; IA 4043  ;  I $S(IBMNUM="":1,1:'IBBILL&(IBBTCH="")) D DELMSG^IBCESRV2(IBTDA) G UPDQ  ;  ; Individual bill  I IBBILL D  G UPDQ  . N IBA1,IBMSG0,IBPID  . S IBPID="",IBA1=0  . F  S IBA1=$O(^IBA(364.2,IBTDA,2,IBA1)) Q:'IBA1  S IBMSG0=$P($G(^(IBA1,0)),"##RAW DATA: ",2) I +IBMSG0=277,$P(IBMSG0,U,5)="N" S IBPID=$P(IBMSG0,U,11) Q  . S IBSEQ=$P(IB00,U,8) S:IBSEQ="" IBSEQ="P"  . D STORE(IB0,IBBTCH,IBMNUM,IBTDA,IBBILL,IBSEQ,IBPID,1)  ;  ; Batch - update each bill separately  S IBBILL=""  F  S IBBILL=$O(^IBA(364,"ABABI",+IBBTCH,IBBILL)) Q:'IBBILL  D  . Q:$D(^TMP("IBCONF",$J,IBBILL)) ;Bill was rejected  . S IB=$O(^IBA(364,"ABABI",+IBBTCH,IBBILL,0)) Q:'IB  . S IBSEQ=$P($G(^IBA(364,IB,0)),U,8) S:IBSEQ="" IBSEQ="P"  . D STORE(IB0,IBBTCH,IBMNUM,IBTDA,IBBILL,IBSEQ,"",0)  ;  Q  ; STORE(IB0,IBBTCH,IBMNUM,IBTDA,IBBILL,IBSEQ,IBPID,IB1) ;  ;  ; IB0 = 0-node of message in file 364.2  ; IBBTCH = ien of batch in file 364.1  ; IBMNUM = actual message number  ; IBTDA = ien of message in file 364.2  ; IBBILL = ien of bill in 399  ; IBSEQ = P/S/T/ for COB sequence related to message  ; IBPID = the payer id returned from clearinghouse for the claim  ; IB1 = flag that says if the message was for a single bill or a batch.  ; Batch statuses have an additional standard text entry.  ; 1 = single bill 0 = batch  ;   N DA,DIK,DIE,DIC,X,Y,DR,DO,DD,DLAYGO,Z,Z0,Z1,Z2,Z3,IBT,IBDUP,IBFLDS,IBY,IBAUTO,IBLN  ;  S X=IBBILL,IBDUP=0  ;  S IBFLDS=".02////"\_$P(IB0,U,3)  S IBFLDS=IBFLDS\_";.03////"\_$S($$EXTERNAL^DILFD(364.2,.02,"U",$P(IB0,U,2))["REJ":"R",1:"I")\_";.05////"\_IBBTCH\_";.06////"\_IBMNUM\_";.04////"\_+$P(IB0,U,8)\_";.07////"\_IBSEQ\_$S($P(IB0,U,5):";.11////"\_$P(IB0,U,5),1:"")  S IBFLDS=IBFLDS\_";.12////"\_$P(IB0,U,10)\_";.09////0"  S IBFLDS=IBFLDS\_";.15////"\_$$CHKSUM^IBCEST1("^IBA(364.2,"\_IBTDA\_",2)")  I IBPID'="" D  . S IBPID("TYPE")=$S($$FT^IBCEF(IBBILL)=2:"P",1:"I")  . D UPDINS(.IBPID,$$POLICY^IBCEF(IBBILL,1,$TR(IBSEQ,"PST","123")),IBBILL)  ;  I IBDUP D  I $D(Y) G UPDQ  . ; Stuff fields into existing entry  . ; (may be needed for reprocessing of aborted updates)  . S DIE="^IBM(361,",DA=IBDUP,DR=IBFLDS\_";1///@"  . D ^DIE  . I $D(Y) S IBY=-1 Q  ;Update not successful  . S IBY=IBDUP  ;  K IBT  I 'IBDUP D  ; Create new entry and stuff fields  . S DIC(0)="L",DIC="^IBM(361,",DLAYGO=361  . S DIC("DR")=IBFLDS  . D FILE^DICN  . K DO,DD,DLAYGO,DIC  . S IBY=+Y  . Q:IBY'>0  . ;  . ; IB\*2\*320 - Check for duplicate status message  . NEW IBNEW,IBOLD,PCE,Z,DIK,DA  . S IBNEW=""  . F PCE=3,4,5,7,8,11,15 S IBNEW=IBNEW\_$P($G(^IBM(361,IBY,0)),U,PCE)\_U  . S Z=0  . F  S Z=$O(^IBM(361,"B",IBBILL,Z)) Q:'Z  I Z'=IBY D  Q:IBY'>0  .. S IBOLD=""  .. F PCE=3,4,5,7,8,11,15 S IBOLD=IBOLD\_$P($G(^IBM(361,Z,0)),U,PCE)\_U  .. I IBNEW'=IBOLD Q   ; no duplicate so get the next one  .. S DIK="^IBM(361,",DA=IBY,IBY=-1 D ^DIK D DELMSG^IBCESRV2(IBTDA)  .. Q  . Q  ;  I IBY>0 D  ;Move text over  . K IBT  . ;  . D BLDMSG(IB1,IBTDA,.IBT,.IBAUTO)  . ;  . ; IB\*2\*368 - ymg - 2Q,RE,RP messages will be filed as informational  . ; Z0 is the flag for 2Q code  . ; Z1 is the flag for RE code  . ; Z2 is the flag for RP code  . ; Z3 is the flag for autofiling the message  . I $P($G(^IBM(361,+IBY,0)),U,3)="R" D  .. S Z="",(Z0,Z1,Z2,Z3)=0 F  S Z=$O(IBT(Z)) Q:Z=""!(Z3=1) D  ... S IBLN=$$UP^XLFSTR($G(IBT(Z)))  ... I (Z0!Z1!Z2)=0 D  .... S:IBLN?.E1"CODE:".P1"2Q".E Z0=1  .... S:IBLN?.E1"CODE:".P1"RE".E Z1=1  .... S:IBLN?.E1"CODE:".P1"RP".E Z2=1  ... I Z0=1 S:IBLN?.P1"CLAIM".P1"REJECTED".P1"BY".P1"CLEARINGHOUSE".E Z3=1  ... I Z1=1 S:IBLN?.P1"ELECTRONIC".P1"CLAIM".P1"REJECTED".P1"BY".P1"EMDEON".E Z3=1  ... I Z2=1 S:IBLN?.P1"PAPER".P1"CLAIM".P1"REJECTED".P1"BY".P1"EMDEON".E Z3=1  .. I Z3=1 S IBAUTO=1,DIE=361,DA=+IBY,DR=".03////I" D ^DIE  .. Q  . ;  . ; if info msg, ck for no review needed based on first line of text  . I $G(IBAUTO),$P($G(^IBM(361,+IBY,0)),U,3)="I" D  .. S DIE="^IBM(361,",DR=".09////2;.14////1;.1////F",DA=+IBY D ^DIE  .. I IB1,$P($G(^IBM(361,+IBY,0)),U,11) S Z="",Z0=0 F  S Z=$O(IBT(Z)) Q:Z=""!(Z0=1) D  ... S Z0=$$PRINTUPD^IBCEU0($$UP^XLFSTR($G(IBT(Z))),$P($G(^IBM(361,+IBY,0)),U,11))  . ;  . D MSGLNSZ(.IBT) ; Convert Message Lines in IBT to be no longer than 70 chars  . D WP^DIE(361,+IBY\_",",1,"A","IBT") ; file message text  . ;  . ; Delete message after it successfully updates the database.  . D DELMSG^IBCESRV2(IBTDA)  . Q  ; UPDQ L -^IBA(364.2,IBTDA,0)  Q  ; BLDMSG(IB1,IBTDA,IBT,IBAUTO) ; Builds message text  ; IB1 = flag for batch message  ; IBTDA = ien of entry in file 364.2  ; IBT = array returned with message text  ; IBAUTO = if passed by reference, returns 1 if text indicates review  ; not needed  N IBDATA,IBCK,IBZ,IBZ0,IBZ1,Z  S (IBZ,IBZ0,IBDATA,IBAUTO,IBCK)=0  I 'IB1 S IBT(1)="Status message received for batch "\_$P($G(^IBA(364.1,IBBTCH,0)),U)\_" dated "\_$$FMTE^XLFDT($P($G(^IBA(364.2,IBTDA,0)),U,10),2),IBZ0=1  ; Don't move the raw data over, just move the text of the message  F  S IBZ=$O(^IBA(364.2,IBTDA,2,IBZ)) Q:'IBZ  S IBZ1=$G(^(IBZ,0)) S IBDATA=($E(IBZ1,1,2)="##") Q:IBDATA  S IBZ0=IBZ0+1,IBT(IBZ0)=IBZ1 I 'IBCK S Z=$$CKREVU^IBCEM4(IBZ1,,,.IBCK),IBAUTO=$S(IBCK:0,Z:1,1:IBAUTO)  Q  ; UPDINS(IBPID,IBINS,IBIFN) ; Update the insurance id or the bill printed at  ; the EDI contractor's print shop and mailed to the ins co.  ; IBPID = the id returned from the EDI contractor for the ins co  ; ("TYPE") = P if professional id or I if institutional id  ; IBINS = the ien of the insurance co it was sent to (file 36)  ; IBIFN = the ien of the claim (file 399)  ;  N IBID,IBIDFLD,IBPRT,IBLOOK,DA,DR,DIE,X,Y,Z  ;  Q:'$G(IBINS)!($G(IBPID)="")  ;  ; Strip spaces off the end of data  S IBLOOK=""  I $L(IBPID) F Z=$L(IBPID):-1:1 I $E(IBPID,Z)'=" " S IBLOOK=$E(IBPID,1,Z) Q  ;  S IBPRT=($E(IBLOOK,2,5)="PRNT")  I IBPRT D  ; Set printed via EDI field on bill  . S DA=IBIFN,DIE="^DGCR(399,",DR="26////1" D ^DIE  ;  S IBLOOK=$E($S('IBPRT:$P(IBLOOK,"PAYID=",2),1:""),1,5)  Q:IBLOOK=""!($E(IBLOOK,2,5)="PRNT")  S IBIDFLD="3.0"\_$S($G(IBPID("TYPE"))="I":4,1:2)  S IBID=$P($G(^DIC(36,+IBINS,3)),U,IBIDFLD\*100#100)  Q:IBID=IBLOOK  I IBID="" D  G UPDINSQ ; Update insurance co electronic id # if blank  . S DIE="^DIC(36,",DR=IBIDFLD\_"////"\_IBLOOK,DA=IBINS D ^DIE  I IBID'="",IBLOOK'="" D  ; Bulletin that the id on file and id returned  . ; are different  . N XMTO,XMDUZ,XMBODY,IBXM,XMSUBJ,XMZ  . S XMTO("I:G.IB EDI")=""  . S XMDUZ="",XMBODY="IBXM",XMSUBJ="PAYER ID RETURNED IS DIFFERENT THAN PAYER ID ON FILE"  . S IBXM(1)="BILL # : "\_$P($G(^DGCR(399,IBIFN,0)),U)  . S IBXM(2)="PAYER : "\_$P($G(^DIC(36,+IBINS,0)),U)  . S IBXM(3)="BILL TYPE : "\_$S($G(IBPID("TYPE"))="I":"INSTITUT",1:"PROFESS")\_"IONAL"  . S IBXM(4)="ID ON FILE : "\_IBID  . S IBXM(5)="ID RETURNED: "\_IBLOOK  . S IBXM(6)=" ",IBXM(7)=" Please determine which id number is correct and correct the id in the",IBXM(8)="insurance file for this payer, if needed"  . D SENDMSG^XMXAPI(XMDUZ,XMSUBJ,XMBODY,.XMTO,,.XMZ)  ; UPDINSQ Q  ; MSGLNSZ(MSG) ; Change Input Message Lines to be no more than 70 characters long each  ;  ; Input/Output: MSG - array of Input Message Lines; this is also the Output Message  ; which is an array of Converted Message Lines (with lines no more than 70 chars each)  ;  N LN,XARY,XARYLN,CNT,OUTMSG,TMPMSG,LDNGSP,LDNGSPN  S LN="",CNT=0 F  S LN=$O(MSG(LN)) Q:LN=""  D  ;  . ; Find any leading spaces in original message line,   . ; to be used if line got split below  . S TMPMSG=$$TRIM^XLFSTR(MSG(LN),"L"," ") ;Trim Leading Spaces  . S LDNGSP=$P(MSG(LN),TMPMSG,1) ;get leading spaces if any  . S LDNGSPN=$L(LDNGSP) S:LDNGSPN>30 LDNGSP=$E(LDNGSP,1,30) ;make sure there are no more than 30 leading spaces   . ; Converts a single line to multiple lines with a maximum width of 70 each  . ; If line is 70 chars or less, this call returns the exact line  . K XARY D FSTRNG^IBJU1(TMPMSG,70-LDNGSPN,.XARY)  . ; Scan lines and merge them into the final output array (OUTMSG)  . ; On lines 2 and higher, add Leading Spaces found above, if any.  . S XARYLN="" F  S XARYLN=$O(XARY(XARYLN)) Q:XARYLN=""  S CNT=CNT+1,OUTMSG(CNT)=LDNGSP\_XARY(XARYLN)  ;  ; Move the final Message Lines (OUTMSG) into MSG array to be returned  K MSG M MSG=OUTMSG  Q  ; | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCEST ;ALB/TMP - 837 EDI STATUS MESSAGE PROCESSING ;17-APR-96  ;;2.0;INTEGRATED BILLING;\*\*137,189,197,135,283,320,368,397,407,577,**592**\*\*;21-MAR-94;Build 1  ;;Per VA Directive 6402, this routine should not be modified.  ; IA 4043 for call to AUDITX^PRCAUDT  Q  ; UPD361(IBTDA) ; Update IB BILL STATUS MESSAGES file  ; IBTDA = ien of return message in file 364.2  ;  N IB,IB0,IBSEQ,IB00,IBBILL,IBBTCH,IBMNUM,IBDATE,IBTYP  ;  I '$$LOCK^IBCEM(IBTDA) G UPDQ ;Lock message in file 364.2  ;  S IB0=$G(^IBA(364.2,IBTDA,0))  S IBMNUM=$P(IB0,U) ; Message number  S IB00=$G(^IBA(364,+$P(IB0,U,5),0)) ; Transmit bill entry  S IBBILL=+IB00 ; Actual bill ien in file 399  S IBBTCH=$P(IB0,U,4) ; Batch #  ;  ; Auto-audit bills based on status code on '10' record of status msg  ; flat file  I IBBILL,$P($T(PRCAUDT+1^PRCAUDT),"\*\*",2)[",173" D  . N Z,Z0,Z1,OK  . Q:+$$STA^PRCAFN(IBBILL)'=104  . S (Z,OK)=0  . F  S Z=$O(^IBA(364.2,IBTDA,2,Z)) Q:'Z  S Z0=$P($G(^(Z,0)),"##RAW DATA: ",2) I +Z0=10 S Z0=$P(Z0,U,5) D  Q:OK  .. ; Strip leading spaces  .. S Z0=$$TRIM^XLFSTR(Z0)  .. Q:Z0=""  .. I $$SCODE^IBCEST1(Z0),$P($G(^DGCR(399.3,+$P($G(^DGCR(399,IBBILL,0)),U,7),0)),U,11) D AUDITX^PRCAUDT(IBBILL) S OK=1 ; IA 4043  ;  I $S(IBMNUM="":1,1:'IBBILL&(IBBTCH="")) D DELMSG^IBCESRV2(IBTDA) G UPDQ  ;  ; Individual bill ; KDM US129 IB\*2\*577 rework Individual vs. Batch to Correct Storage of Payer ID  I IBBILL D UPDTBILL() G UPDQ  ;  ; Batch - update each bill separately  S IBBILL=""  F  S IBBILL=$O(^IBA(364,"ABABI",+IBBTCH,IBBILL)) Q:'IBBILL  D  . Q:$D(^TMP("IBCONF",$J,IBBILL)) ;Bill was rejected  . S IB=$O(^IBA(364,"ABABI",+IBBTCH,IBBILL,0))  . Q:'IB  . D UPDTBILL() ;KDM US129 IB\*2\*577 Correct Storage of PAYER ID  ;  Q  ; UPDTBILL() ;KDM US129 IB\*2\*577 New section to Correct Storage of PAYER ID  N IBA1,IBMSG0,IBPID  S IBPID="",IBA1=0  ;  F  S IBA1=$O(^IBA(364.2,IBTDA,2,IBA1)) Q:'IBA1  D  Q:IBPID]""  . S IBMSG0=$P($G(^(IBA1,0)),"##RAW DATA: ",2)  . I +IBMSG0=277,$P(IBMSG0,U,5)="N" S IBPID=$P(IBMSG0,U,11)  ;  S IBSEQ=$P(IB00,U,8) S:IBSEQ="" IBSEQ="P"  D STORE(IB0,IBBTCH,IBMNUM,IBTDA,IBBILL,IBSEQ,IBPID,1)  Q  ; STORE(IB0,IBBTCH,IBMNUM,IBTDA,IBBILL,IBSEQ,IBPID,IB1) ;  ;  ; IB0 = 0-node of message in file 364.2  ; IBBTCH = ien of batch in file 364.1  ; IBMNUM = actual message number  ; IBTDA = ien of message in file 364.2  ; IBBILL = ien of bill in 399  ; IBSEQ = P/S/T/ for COB sequence related to message  ; IBPID = the payer id returned from clearinghouse for the claim  ; IB1 = flag that says if the message was for a single bill or a batch.  ; Batch statuses have an additional standard text entry.  ; 1 = single bill 0 = batch  ;   N DA,DIK,DIE,DIC,X,Y,DR,DO,DD,DLAYGO,Z,Z0,Z1,Z2,Z3,IBT,IBDUP,IBFLDS,IBY,IBAUTO,IBLN  ;  S X=IBBILL,IBDUP=0  ;  S IBFLDS=".02////"\_$P(IB0,U,3)  S IBFLDS=IBFLDS\_";.03////"\_$S($$EXTERNAL^DILFD(364.2,.02,"U",$P(IB0,U,2))["REJ":"R",1:"I")\_";.05////"\_IBBTCH\_";.06////"\_IBMNUM\_";.04////"\_+$P(IB0,U,8)\_";.07////"\_IBSEQ\_$S($P(IB0,U,5):";.11////"\_$P(IB0,U,5),1:"")  S IBFLDS=IBFLDS\_";.12////"\_$P(IB0,U,10)\_";.09////0"  S IBFLDS=IBFLDS\_";.15////"\_$$CHKSUM^IBCEST1("^IBA(364.2,"\_IBTDA\_",2)")  I IBPID'="" D  **. ;JWS;IB\*2.0\*592;Dental Form 7**  . S IBPID("TYPE")=$S($$FT^IBCEF(IBBILL)=2:"P**",$$FT^IBCEF(IBBILL)=7:"P",**1:"I")  . D UPDINS(.IBPID,$$POLICY^IBCEF(IBBILL,1,$TR(IBSEQ,"PST","123")),IBBILL,IBTDA) ;KDM US129 IB\*2\*577  ;  I IBDUP D  I $D(Y) G UPDQ  . ; Stuff fields into existing entry  . ; (may be needed for reprocessing of aborted updates)  . S DIE="^IBM(361,",DA=IBDUP,DR=IBFLDS\_";1///@"  . D ^DIE  . I $D(Y) S IBY=-1 Q  ;Update not successful  . S IBY=IBDUP  ;  K IBT  I 'IBDUP D  ; Create new entry and stuff fields  . S DIC(0)="L",DIC="^IBM(361,",DLAYGO=361  . S DIC("DR")=IBFLDS  . D FILE^DICN  . K DO,DD,DLAYGO,DIC  . S IBY=+Y  . Q:IBY'>0  . ;  . ; IB\*2\*320 - Check for duplicate status message  . NEW IBNEW,IBOLD,PCE,Z,DIK,DA  . S IBNEW=""  . F PCE=3,4,5,7,8,11,15 S IBNEW=IBNEW\_$P($G(^IBM(361,IBY,0)),U,PCE)\_U  . S Z=0  . F  S Z=$O(^IBM(361,"B",IBBILL,Z)) Q:'Z  I Z'=IBY D  Q:IBY'>0  .. S IBOLD=""  .. F PCE=3,4,5,7,8,11,15 S IBOLD=IBOLD\_$P($G(^IBM(361,Z,0)),U,PCE)\_U  .. I IBNEW'=IBOLD Q   ; no duplicate so get the next one  .. S DIK="^IBM(361,",DA=IBY,IBY=-1 D ^DIK D DELMSG^IBCESRV2(IBTDA)  .. Q  . Q  ;  I IBY>0 D  ;Move text over  . K IBT  . ;  . D BLDMSG(IB1,IBTDA,.IBT,.IBAUTO)  . ;  . ; IB\*2\*368 - ymg - 2Q,RE,RP messages will be filed as informational  . ; Z0 is the flag for 2Q code  . ; Z1 is the flag for RE code  . ; Z2 is the flag for RP code  . ; Z3 is the flag for autofiling the message  . I $P($G(^IBM(361,+IBY,0)),U,3)="R" D  .. S Z="",(Z0,Z1,Z2,Z3)=0 F  S Z=$O(IBT(Z)) Q:Z=""!(Z3=1) D  ... S IBLN=$$UP^XLFSTR($G(IBT(Z)))  ... I (Z0!Z1!Z2)=0 D  .... S:IBLN?.E1"CODE:".P1"2Q".E Z0=1  .... S:IBLN?.E1"CODE:".P1"RE".E Z1=1  .... S:IBLN?.E1"CODE:".P1"RP".E Z2=1  ... I Z0=1 S:IBLN?.P1"CLAIM".P1"REJECTED".P1"BY".P1"CLEARINGHOUSE".E Z3=1  ... I Z1=1 S:IBLN?.P1"ELECTRONIC".P1"CLAIM".P1"REJECTED".P1"BY".P1"EMDEON".E Z3=1  ... I Z2=1 S:IBLN?.P1"PAPER".P1"CLAIM".P1"REJECTED".P1"BY".P1"EMDEON".E Z3=1  .. I Z3=1 S IBAUTO=1,DIE=361,DA=+IBY,DR=".03////I" D ^DIE  .. Q  . ;  . ; if info msg, ck for no review needed based on first line of text  . I $G(IBAUTO),$P($G(^IBM(361,+IBY,0)),U,3)="I" D  .. S DIE="^IBM(361,",DR=".09////2;.14////1;.1////F",DA=+IBY D ^DIE  .. I IB1,$P($G(^IBM(361,+IBY,0)),U,11) S Z="",Z0=0 F  S Z=$O(IBT(Z)) Q:Z=""!(Z0=1) D  ... S Z0=$$PRINTUPD^IBCEU0($$UP^XLFSTR($G(IBT(Z))),$P($G(^IBM(361,+IBY,0)),U,11))  . ;  . D MSGLNSZ(.IBT) ; Convert Message Lines in IBT to be no longer than 70 chars  . D WP^DIE(361,+IBY\_",",1,"A","IBT") ; file message text  . ;  . ; Delete message after it successfully updates the database.  . D DELMSG^IBCESRV2(IBTDA)  . Q  ; UPDQ L -^IBA(364.2,IBTDA,0)  Q  ; BLDMSG(IB1,IBTDA,IBT,IBAUTO) ; Builds message text  ; IB1 = flag for batch message  ; IBTDA = ien of entry in file 364.2  ; IBT = array returned with message text  ; IBAUTO = if passed by reference, returns 1 if text indicates review  ; not needed  N IBDATA,IBCK,IBZ,IBZ0,IBZ1,Z  S (IBZ,IBZ0,IBDATA,IBAUTO,IBCK)=0  I 'IB1 S IBT(1)="Status message received for batch "\_$P($G(^IBA(364.1,IBBTCH,0)),U)\_" dated "\_$$FMTE^XLFDT($P($G(^IBA(364.2,IBTDA,0)),U,10),2),IBZ0=1  ; Don't move the raw data over, just move the text of the message  F  S IBZ=$O(^IBA(364.2,IBTDA,2,IBZ)) Q:'IBZ  S IBZ1=$G(^(IBZ,0)) S IBDATA=($E(IBZ1,1,2)="##") Q:IBDATA  S IBZ0=IBZ0+1,IBT(IBZ0)=IBZ1 I 'IBCK S Z=$$CKREVU^IBCEM4(IBZ1,,,.IBCK),IBAUTO=$S(IBCK:0,Z:1,1:IBAUTO)  Q  ; UPDINS(IBPID,IBINS,IBIFN,IBTDA) ;KDM US129 IB\*2\*577  ; Update the insurance id or the bill printed at  ; the EDI contractor's print shop and mailed to the ins co.  ; IBPID = the id returned from the EDI contractor for the ins co  ; ("TYPE") = P if professional id or I if institutional id  ; IBINS = the ien of the insurance co it was sent to (file 36)  ; IBIFN = the ien of the claim (file 399)  ; IBTDA = ien of entry in file 364.2 ;KDM US129 IB\*2\*577  ;  N IBID,IBIDFLD,IBPRT,IBLOOK,DA,DR,DIE,X,Y,Z,UPD     ;KDM US129 IB\*2\*577  ;  Q:'$G(IBINS)!($G(IBPID)="")  ;  ; Strip spaces off the end of data  S IBLOOK=""  I $L(IBPID) F Z=$L(IBPID):-1:1 I $E(IBPID,Z)'=" " S IBLOOK=$E(IBPID,1,Z) Q  ;  S IBPRT=($E(IBLOOK,2,5)="PRNT")  I IBPRT D  ; Set printed via EDI field on bill  . S DA=IBIFN,DIE="^DGCR(399,",DR="26////1" D ^DIE  ;  ; KDM US129 IB\*2\*577 correct payer ID storage  ;S IBLOOK=$E($S('IBPRT:$P(IBLOOK,"PAYID=",2),1:""),1,5)  ;Q:IBLOOK=""!($E(IBLOOK,2,5)="PRNT")  I IBPRT Q  I IBLOOK'["PAYID=",IBLOOK'["COBID=" Q     ;KDM US129 IB\*2\*577  S IBLOOK=$E($P(IBLOOK,"ID=",2),1,5)  Q:IBLOOK=""  ;  S IBIDFLD="3.0"\_$S($G(IBPID("TYPE"))="I":4,1:2)  S IBID=$P($G(^DIC(36,+IBINS,3)),U,IBIDFLD\*100#100)  Q:IBID=IBLOOK  S IBDATE=DT,IBTYP=$G(IBPID("TYPE")) ;KDM US129 IB\*2\*577  I IBID="" D  G UPDINSQ ; Update insurance co electronic id # if blank  . S DIE="^DIC(36,",DR=IBIDFLD\_"////"\_IBLOOK,DA=IBINS D ^DIE  . D UPDLOG(1,IBDATE,IBINS,IBLOOK,IBTYP,IBID) ;KDM US129 IB\*2\*577  I IBID'="",IBLOOK'="" D  ; Bulletin that the id on file and id returned  . ; are different  . N XMTO,XMDUZ,XMBODY,IBXM,XMSUBJ,XMZ  . S XMTO("I:G.IB EDI")=""  . S XMDUZ="",XMBODY="IBXM",XMSUBJ="PAYER ID RETURNED IS DIFFERENT THAN PAYER ID ON FILE"  . S IBXM(1)="BILL # : "\_$P($G(^DGCR(399,IBIFN,0)),U)  . S IBXM(2)="PAYER : "\_$P($G(^DIC(36,+IBINS,0)),U)  . S IBXM(3)="BILL TYPE : "\_$S($G(IBPID("TYPE"))="I":"INSTITUT",1:"PROFESS")\_"IONAL"  . S IBXM(4)="ID ON FILE : "\_IBID  . S IBXM(5)="ID RETURNED: "\_IBLOOK  . S IBXM(6)=" ",IBXM(7)=" Please determine which id number is correct and correct the id in the",IBXM(8)="insurance file for this payer, if needed"  . D SENDMSG^XMXAPI(XMDUZ,XMSUBJ,XMBODY,.XMTO,,.XMZ)  . D UPDLOG(0,IBDATE,IBINS,IBLOOK,IBTYP,IBID) ;KDM US129, US976 IB\*2\*577  ; UPDINSQ Q  ; UPDLOG(UPD,IBDATE,IBINS,IBLOOK,IBTYP,IBID) ;KDM US129, US976 IB\*2\*577 New section for New Payer Report   ; store flds for reporting purposes when updating or attempting to update Payer information (US129)  ; ^DIC(36 -17.0 277EDI ID Number  ; 17.01 277EDI ID Number  ; 17.02 277Date EDI ID Number  ; 17.03 277EDI Type (P)ROF or (I)nst  ; 17.04 277EDI ID NUMBER ON FILE ;if blank it was an update otherwise it was an attempted update.   ;  Q:(($D(^DIC(36,"AEDIX",IBDATE,IBINS,IBLOOK,IBTYP)))&(UPD=0)) ;store only one attempt a day  N ERROR,IBFDA,LEV  S LEV="+2,"\_IBINS\_","  S IBFDA(36.017,LEV,.01)=IBLOOK     ;New Value from 277STAT  S IBFDA(36.017,LEV,.02)=IBDATE     ;Date transaction is processed  S IBFDA(36.017,LEV,.03)=IBTYP      ;"P" or "I"  S IBFDA(36.017,LEV,.04)=$G(IBID) ;Value already on file- if blank it was an update, otherwise attempted update  D UPDATE^DIE("","IBFDA","","ERROR")  Q  ; MSGLNSZ(MSG) ; Change Input Message Lines to be no more than 70 characters long each  ;  ; Input/Output: MSG - array of Input Message Lines; this is also the Output Message  ; which is an array of Converted Message Lines (with lines no more than 70 chars each)  ;  N LN,XARY,XARYLN,CNT,OUTMSG,TMPMSG,LDNGSP,LDNGSPN  S LN="",CNT=0 F  S LN=$O(MSG(LN)) Q:LN=""  D  ;  . ; Find any leading spaces in original message line,   . ; to be used if line got split below  . S TMPMSG=$$TRIM^XLFSTR(MSG(LN),"L"," ") ;Trim Leading Spaces  . S LDNGSP=$P(MSG(LN),TMPMSG,1) ;get leading spaces if any  . S LDNGSPN=$L(LDNGSP) S:LDNGSPN>30 LDNGSP=$E(LDNGSP,1,30) ;make sure there are no more than 30 leading spaces   . ; Converts a single line to multiple lines with a maximum width of 70 each  . ; If line is 70 chars or less, this call returns the exact line  . K XARY D FSTRNG^IBJU1(TMPMSG,70-LDNGSPN,.XARY)  . ; Scan lines and merge them into the final output array (OUTMSG)  . ; On lines 2 and higher, add Leading Spaces found above, if any.  . S XARYLN="" F  S XARYLN=$O(XARY(XARYLN)) Q:XARYLN=""  S CNT=CNT+1,OUTMSG(CNT)=LDNGSP\_XARY(XARYLN)  ;  ; Move the final Message Lines (OUTMSG) into MSG array to be returned  K MSG M MSG=OUTMSG  Q  ; | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCEU | | | | | | | | |
| **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** | | | | | | | | | |
| IBCEU ;ALB/TMP - EDI UTILITIES ;02-OCT-96  ;;2.0;INTEGRATED BILLING;\*\*51,137,207,232,349,432\*\*;21-MAR-94;Build 192  ;;Per VHA Directive 2004-038, this routine should not be modified.  ; DBIA SUPPORTED REF: GET^XUA4A72 = 1625  ; DBIA SUPPORTED REF: $$ESBLOCK^XUSESIG1 = 1557  Q  ; TESTPT(DFN) ; Determine if pt is test pt  ; Returns 1 if a test pt, 0 if not  Q $E($P($G(^DPT(+DFN,0)),U,9),1,5)="00000"  ; MAINPRV(IBIFN) ; Returns name^id^ien^type code of 'main' prov on bill IBIFN  N IBPRV,IBCOB,IBQ,Z  D GETPRV(IBIFN,"3,4",.IBPRV)  S IBQ="",IBCOB=$$COBN^IBCEF(IBIFN)  F Z=3,4 I $G(IBPRV(Z,1))'="" D  Q  . S IBQ=IBPRV(Z,1),$P(IBQ,U,4)=Z  . I $G(IBPRV(Z,1,IBCOB))'="" S $P(IBQ,U,2)=IBPRV(Z,1,IBCOB)  Q IBQ  ; PRVOK(VAL,IBIFN) ; Check bill form & prov function agree  ; VAL = internal value of prov function  ;  N OK,IBBT  S OK=0  Q:VAL="" OK  Q:'IBIFN OK  S IBBT=$$FT^IBCEF(IBIFN) ; 2 If CMS-1500, 3 If UB-04  I IBBT=2 D  . I VAL=1 S OK=1 Q   ; CMS-1500, REFERRING  . I VAL=3 S OK=1 Q   ; CMS-1500, RENDERING  . I VAL=5 S OK=1 Q   ; CMS-1500, SUPERVISING  I 'OK,IBBT=3 D  . I VAL=1 S OK=1 Q   ; UB-04, REFERRING  . I VAL=2 S OK=1 Q   ; UB-04, OPERATING  . I VAL=3 S OK=1 Q   ; UB-04, RENDERING  . I VAL=4 S OK=1 Q   ; UB-04, ATTENDING  . I VAL=9 S OK=1 Q   ; UB-04, OTHER  ;  Q OK  ; PRVOK1(VAL,IBIFN) ; Check for both attending and rendering on bill  N OK  S OK=1  Q:$$FT^IBCEF(IBIFN)=3 1 ; both are allowed on UB  I $S("34"'[VAL:0,1:$D(^DGCR(399,IBIFN,"PRV","B",$S(VAL=3:4,1:3)))) D EN^DDIOL($S(VAL=3:"ATTENDING",1:"RENDERING")\_" ALREADY EXISTS - CAN'T HAVE BOTH ON ONE BILL") S OK=0  Q OK  ; SPEC(IBPRV,IBDT) ; Returns spec code for vp ien IBPRV from file 355.9  ; (for new person entries, as of date in IBDT)  ; DBIA 1625  N IBSPEC  S:'$G(IBDT) IBDT=DT  I IBPRV'["IBA(355.93" S IBSPEC=$S(IBPRV:$P($$GET^XUA4A72(+IBPRV,IBDT),U,8),1:"") ; VA  I IBPRV["IBA(355.93" S IBSPEC=$P($G(^IBA(355.93,+IBPRV,0)),U,4) ; Non-VA  Q IBSPEC  ; CRED(IBPRV,IBIFN,IBPIEN,IBTYP) ; Returns prov credentials  ; IBPRV = vp of provider for file 200 or 355.93  ; IBIFN = bill ien in file 399 (optional)  ; IBPIEN = prov ien - file 399.0222 (optional)  ; DEM;432 - prov ien can be from file 399.0404  ; as well (optional).  ; IBTYP = the prov type  ;  N IBCRED  S IBCRED=""  ;  ; DEM;432 - Provider can come from either file 399.0222, or  ; file 399.0404. Variable IBLNPRV is the flag  ; that indicates we want prov ien from file 399.0404.  ;  I '$G(IBLNPRV),$G(IBIFN),'$D(^DGCR(399,IBIFN,"PRV",0)) G CREDQ  ;  ; DEM;432 - Next line if for line level provider. Variable IBPROCP,  ; if it exist, is the procedure ien. File 399.0404 is a  ; multiple of the Procedure File 399.0304.  ;  I $G(IBLNPRV),$G(IBIFN),$G(IBPROCP),'$D(^DGCR(399,IBIFN,"CP",IBPROCP,"LNPRV",0)) G CREDQ  I '$G(IBLNPRV),$G(IBIFN),($G(IBPIEN)!$G(IBTYP)) D  . I '$G(IBPIEN) S IBPIEN=+$O(^DGCR(399,IBIFN,"PRV","B",IBTYP,0))  . S IBCRED=$P($G(^DGCR(399,IBIFN,"PRV",IBPIEN,0)),U,3)  ;  I $G(IBLNPRV),$G(IBIFN),$G(IBPROCP),($G(IBPIEN)!$G(IBTYP)) D  ; DEM;432 - Line Provider File 399.0404.  . I '$G(IBPIEN) S IBPIEN=+$O(^DGCR(399,IBIFN,"CP",IBPROCP,"B",IBTYP,0))  . S IBCRED=$P($G(^DGCR(399,IBIFN,"CP",IBPROCP,"LNPRV",IBPIEN,0)),U,3)  ; CREDQ ;  I $G(IBPRV),IBCRED="" D  . I IBPRV'["IBA(355.93" S IBCRED=$P($$ESBLOCK^XUSESIG1(+IBPRV),U,2)  . I IBPRV["IBA(355.93" S IBCRED=$P($G(^IBA(355.93,+IBPRV,0)),U,3)  Q IBCRED  ; GETPRV(IBIFN,IBTYP,IBPRV) ; Returns prov(s) of type(s) IBTYP for  ; bill ien IBIFN.  ; IBTYP = prov types needed, separated by ',' or ALL  ;   ; OUTPUT:  ; IBPRV array: IBPRV(type)= 1 if prov is from old prov flds  ; IBPRV(type,ct)=name^current COB id^vp provider ien^cred  ; IBPRV(type,ct,seq)=COB seq specific id  ; IBPRV(type)=default nm^def id  ; IBPRV(type,"NOTOPT")= defined if a required prov type  ;  N IB,IBCT,IBD,IBY,IBZ,IBMRAND,IBID,IBWNR,IBPNM,Z ;,IBZFID  ;S IBZFID=""  D F^IBCEF("N-CURRENT INS POLICY TYPE","IBZ",,IBIFN)  ;I IBZ="CI" D F^IBCEF("N-FEDERAL TAX ID","IBZFID",,IBIFN) S IBZFID=$TR(IBZFID,"-")  S IBPRV=U\_$G(IBZ),IBY=0  S IBMRAND=$$MCRONBIL^IBEFUNC(IBIFN)  ;WCJ;IB\*2.0\*432;Remove Default  I IBMRAND D  .; F Z=1:1:3,5,6,7,8,9 S:Z=3&($$FT^IBCEF(IBIFN)=3) Z=4 S IBPRV(Z)=$S(Z=3!(Z=4):"DEPT VETERANS AFFAIRS",1:"")\_"^VAD000"  . F Z=1:1:9 S IBPRV(Z)="^VAD000"  . I '$$INPAT^IBCEF(IBIFN,1),$$FT^IBCEF(IBIFN)=3 S IBPRV(4,1)="^SLF000"  ;WCJ;IB\*2.0\*432;End changes  ;  ; For backwards compatability (before the claim level provider mulitple)  I '$D(^DGCR(399,+IBIFN,"PRV",0)) D  G GETQ  . N IBALL  . S IBALL=(IBTYP="ALL")  . I IBTYP[4!IBALL S:$P($G(^DGCR(399,+IBIFN,"U1")),U,13)'="" IBPRV(4,1)=$P(^("U1"),U,13),IBPRV(4)=1 Q:IBTYP=4  . I IBTYP[3!IBALL S:$P($G(^DGCR(399,+IBIFN,"UF2")),U)'="" IBPRV(3,1)=$P(^("UF2"),U),IBPRV(3)=1 Q:IBTYP=3  . I IBTYP[9!IBALL S:$P($G(^DGCR(399,+IBIFN,"U1")),U,14)'="" IBPRV(9,1)=$P(^("U1"),U,14),IBPRV(9)=1  ;  S IBID=4+$$COBN^IBCEF(IBIFN),IBWNR=$$WNRBILL^IBEFUNC(IBIFN)  F IBZ=1:1:$S(IBTYP="ALL":99,1:$L(IBTYP,",")) S (IBCT,IB)=0,IBY=$S(IBTYP'="ALL":$P(IBTYP,",",IBZ),1:$O(^DGCR(399,+IBIFN,"PRV","B",IBY))) Q:IBY=""  F  S IB=$O(^DGCR(399,+IBIFN,"PRV","B",IBY,IB)) Q:'IB  D  . S IBCT=IBCT+1  . S IBD=$G(^DGCR(399,+IBIFN,"PRV",IB,0))  . Q:'$P(IBD,U,2)  . S IBPNM=$$EXPAND^IBTRE(399.0222,.02,$P(IBD,U,2))  . I IBWNR Q:'$D(IBPRV(IBY)) S $P(IBD,U,IBID)=$P(IBPRV(IBY),U,2)  . S IBPRV(IBY,IBCT)=IBPNM\_U\_$S($P(IBD,U,IBID)'="":$P(IBD,U,IBID),$P($G(IBPRV(IBY)),U,2)'="":$P(IBPRV(IBY),U,2),1:$P($$DEFID^IBCEF74(IBIFN,IB),U,IBID-4))\_U\_$P(IBD,U,2)  . S $P(IBPRV(IBY,IBCT),U,4)=$$CRED($P(IBPRV(IBY,IBCT),U,3),IBIFN,$S($P(IBD,U,3)'=""!'$P(IBPRV(IBY,IBCT),U,3):IB,1:""))  . F Z=1:1:3 D  .. ;I IBZFID'="",'$$INPAT^IBCEF(IBIFN,1),$P(IBPRV(IBY,IBCT),U,2)="SLF000" S IBZFID=""  .. ;I $S(Z=1:1,1:$D(^DGCR(399,IBIFN,"I"\_Z))) S IBPRV(IBY,IBCT,Z)=$S($G(IBZFID)'="":IBZFID,$P(IBD,U,Z+4)'="":$P(IBD,U,Z+4),1:"")  .. I $S(Z=1:1,1:$D(^DGCR(399,IBIFN,"I"\_Z))) S IBPRV(IBY,IBCT,Z)=$S($P(IBD,U,Z+4)'="":$P(IBD,U,Z+4),1:$P($$DEFID^IBCEF74(IBIFN,IB),U,Z)) GETQ D NEEDPRV(IBIFN,IBTYP,.IBPRV)  Q  ; NEEDPRV(IBIFN,IBTYP,IBPRV) ; Check for needed prov  ; If needed, not entered, insert defaults for MCR only  N IB0,IBINP,IBFT,IBMRAND,IBTOB  S IB0=$G(^DGCR(399,+IBIFN,0))  S IBFT=($$FT^IBCEF(IBIFN)=3),IBINP=$$INPAT^IBCEF(IBIFN,1),IBTOB=$$TOB^IBCBB(IB0)  ; Only allow defaults for MCR  S IBMRAND=$$WNRBILL^IBEFUNC(IBIFN) ;$$MCRONBIL^IBEFUNC(IBIFN)  ;  I IBTYP="ALL"!((IBTYP\_",")["1,") D  . ; DEM;432 - UB-04 or CMS-1500 SITUATIONAL  . S IBPRV(1,"SITUATIONAL")=1  . Q  ;  I IBTYP="ALL"!((IBTYP\_",")["2,") D:IBFT  . ; only for bill type inpt - 11X, outpt - 83X  . S IBPRV(2,"SITUATIONAL")=1 ; DEM;432 - Default to "SITUATIONAL". If conditions below are met, then IBPRV(2,"SITUATIONAL") is KILLED and IBRPV is SET according to conditions.  . Q:$S(IBINP:$E(IBTOB,1,2)'="11",1:$E(IBTOB,1,2)'="83")  . ; UB-04 bill includes HCPCS procs - operating phys situational  . N Z  . S Z=0 F  S Z=$O(^DGCR(399,IBIFN,"CP",Z)) Q:'Z  I $P($G(^(Z,0)),U)["ICP" D  Q  .. K IBPRV(2,"SITUATIONAL") ; DEM;432 - We have met one of the condtions, so KILL IBPRV(2,"SITUATIONAL").   .. I IBINP S IBPRV(2,"SITUATIONAL")=1 Q  ; DEM;432 - If UB-04 (inpatient), then operating provider situational.  .. I 'IBINP S IBPRV(2,"NOTOPT")=1 ; DEM;432 - If UB-04 (outpatient), then operating provider required.  .. Q:'IBMRAND  .. I '$O(IBPRV(2,0)) S IBPRV(2,"REQ")=1,IBPRV(2,1)=$G(IBPRV(2))  ;  I IBTYP="ALL"!((IBTYP\_",")["3,") D  . ; if a CMS-1500 bill, rendering is required  . I 'IBFT S IBPRV(3,"NOTOPT")=1  . ; DEM;432 - if UB-04, rendering is situational.  . I IBFT S IBPRV(3,"SITUATIONAL")=1 Q  . Q:'IBMRAND  . I '$O(IBPRV(3,0)) S IBPRV(3,1)=$G(IBPRV(3)),IBPRV(3,"REQ")=1  ;  I IBTYP="ALL"!((IBTYP\_",")["4,") D:IBFT  . ; if a UB-04, attending required  . S IBPRV(4,"NOTOPT")=1  . Q:'IBMRAND  . I '$O(IBPRV(4,0)) S IBPRV(4,1)=$G(IBPRV(4)),IBPRV(4,"REQ")=1  Q  ; CKPROV(IBIFN,IBTYP,IBVAL) ; Checks if prov of type IBTYP in 'PRV' node  ; of bill IBIFN  ; If IBVAL = 1, skips the check for an existing provider, just looks  ; for existence of the function itself  N OK,IBFT,Z,R  S OK=0,IBFT=$$FT^IBCEF(IBIFN)  S Z=+$O(^DGCR(399,IBIFN,"PRV","B",+IBTYP,0))  I $G(^DGCR(399,IBIFN,"PRV",Z,0))'="" D  . ; Only outpt UB-04 can have SLF000 as prov ID with no name  . I IBFT=3,'$$INPAT^IBCEF(IBIFN,1),$P(^DGCR(399,IBIFN,"PRV",Z,0),U,2)="",$P(^(0),U,5)="SLF000" S OK=1 Q  . I '$G(IBVAL) Q:$P(^DGCR(399,IBIFN,"PRV",Z,0),U,2)=""  . S OK=1  Q OK  ; XFER(IBQ) ; Transfer DILIST  ; IBQ = # of entries already found  N Z,IBZ  S (Z,IBZ)=0  F  S Z=$O(^TMP("DILIST",$J,1,Z)) Q:'Z  S IBZ=IBZ+1,^TMP("IBLIST",$J,1,IBZ+IBQ)=^TMP("DILIST",$J,1,Z),^TMP("IBLIST",$J,2,IBZ+IBQ)=^TMP("DILIST",$J,2,Z) M ^TMP("IBLIST",$J,"ID",IBZ+IBQ)=^TMP("DILIST",$J,"ID",Z)  ;  I $D(^TMP("DILIST",$J,0)) S ^TMP("IBLIST",$J,0)=^TMP("DILIST",$J,0)  S $P(^TMP("IBLIST",$J,0),U)=IBQ+IBZ  Q  ; DATE(X) ; Convert date X in YYYYMMDD or YYMMDD to FM format  ; FP = flag to indicate if past or future dates are expected  N %DT,Y  I $L(X)=8,$E(X,1,4)<2100,$E(X,5,6)<13,$E(X,7,8)<32 S X=$E(X,1,4)-1700\_$E(X,5,8) G DTQ  I $L(X)=6,$E(X,3,4)<13,$E(X,5,6)<32 S X=$E(X,3,4)\_"/"\_$E(X,5,6)\_"/"\_$E(X,1,2),%DT="N" D ^%DT I Y>0 S X=Y DTQ Q X  ; BCLASS(IBIFN) ; Returns actual bill classif. code from ptr fld  ; .25 in file 399 for bill ien IBIFN  Q $P($G(^DGCR(399.1,+$P($G(^DGCR(399,IBIFN,0)),U,25),0)),U,2)  ; ADMHR(IBIFN,IBDTTM) ; Extract admit hr from admit dt/tm  ; Default 00 if no time and bill is 11X or 18X  N TM  S TM=$P(IBDTTM,".",2)  I TM="","18"[$$BCLASS(IBIFN),$P($G(^DGCR(399,IBIFN,0)),U,24)=1 S TM="00"  I TM'="",TM'="00" S TM=$E(TM\_"0000",1,4)  Q TM  ; OLAB(IBIFN) ; Returns 1 if bill IBIFN is outside lab  N IBL,IBLAB  S IBL=0  S IBLAB=$P($G(^DGCR(399,IBIFN,"U2")),U,11)  I IBLAB,"24"[IBLAB S IBL=1  Q IBL  ; PSRV(IBIFN) ; Returns 1 if bill IBIFN has any purch services  N IBZ,IBXDATA,IBXSAVE,Z  S IBZ=0  D F^IBCEF("N-HCFA 1500 PROCEDURES",,,IBIFN)  S Z=0 F  S Z=$O(IBXSAVE("BOX24",Z)) Q:'Z  I $P(IBXSAVE("BOX24",Z),U,11) S IBZ=1 Q  Q IBZ  ; SEQBILL(IBIFN) ; Returns the ien's of all bills in COB sequence for bill IBIFN  ; Return value is "^" delimited: primary ien^secondary ien^tertiary ien  N IBSEQ,Z  S IBSEQ=$P($G(^DGCR(399,IBIFN,"M1")),U,5,7)  S Z=$$COBN^IBCEF(IBIFN)  I $P(IBSEQ,U,Z)="" S $P(IBSEQ,U,Z)=IBIFN  Q IBSEQ  ;  ;IB\*2.0\*432/TAZ Added to take into account the line level providers. GETPRV1(IBIFN,IBTYP,IBPRV) ; Returns prov(s) of type(s) IBTYP for  ; bill ien IBIFN for TPJI display  ; IBTYP = prov types needed, separated by ',' or ALL  ;   ; OUTPUT:  ; IBPRV array: IBPRV(level,type,ct)=name^current COB id^vp provider ien^cred  ;  N IB,IBCT,IBD,IBY,IBZ,IBMRAND,IBID,IBWNR,IBPNM,Z,IBPRTYP  D F^IBCEF("N-CURRENT INS POLICY TYPE","IBZ",,IBIFN)  S IBPRV=U\_$G(IBZ),IBY=0  D ALLIDS^IBCEFP(IBIFN,.IBXSAVE)  S IBCT=0  F  S IBCT=$O(IBXSAVE("PROVINF",IBIFN,"C",IBCT)) Q:'IBCT  D  . S IBPRTYP=""  . F  S IBPRTYP=$O(IBXSAVE("PROVINF",IBIFN,"C",IBCT,IBPRTYP)) Q:'IBPRTYP  D  .. I IBTYP'="ALL",IBTYP'[IBPRTYP Q  ;Screen out unwanted providers  .. N IBPRIEN,OBPRNM,IBCOBID  .. S IBPRIEN=$P(IBXSAVE("PROVINF",IBIFN,"C",IBCT,IBPRTYP),U)  .. S $P(IBPRV(1,IBCT,IBPRTYP),U,1)=$$EXPAND^IBTRE(399.0222,.02,IBPRIEN)  .. S $P(IBPRV(1,IBCT,IBPRTYP),U,2)=IBXSAVE("PROVINF",IBIFN,"C",IBCT,IBPRTYP,"COBID")  .. S $P(IBPRV(1,IBCT,IBPRTYP),U,3)=IBPRIEN  .. S $P(IBPRV(1,IBCT,IBPRTYP),U,4)=$P(IBXSAVE("PROVINF",IBIFN,"C",IBCT,IBPRTYP,"NAME"),U,4)  S IBCT=0  F  S IBCT=$O(IBXSAVE("L-PROV",IBIFN,IBCT)) Q:'IBCT  D  . S IBPRTYP=""  . F  S IBPRTYP=$O(IBXSAVE("L-PROV",IBIFN,IBCT,"C",1,IBPRTYP)) Q:'IBPRTYP  D  .. I IBTYP'="ALL",IBTYP'[IBPRTYP Q  ;Screen out unwanted providers  .. N IBPRIEN  .. S IBPRIEN=$P(IBXSAVE("L-PROV",IBIFN,IBCT,"C",1,IBPRTYP),U)  .. S IBPRV(2,IBCT,IBPRTYP)=$$EXPAND^IBTRE(399.0222,.02,IBPRIEN)  .. S $P(IBPRV(2,IBCT,IBPRTYP),U,2)=IBXSAVE("L-PROV",IBIFN,IBCT,"C",1,IBPRTYP,"COBID")  .. S $P(IBPRV(2,IBCT,IBPRTYP),U,3)=IBPRIEN  .. S $P(IBPRV(2,IBCT,IBPRTYP),U,4)=$P(IBXSAVE("L-PROV",IBIFN,IBCT,"C",1,IBPRTYP,"NAME"),U,4)  Q | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCEU ;ALB/TMP - EDI UTILITIES ;02-OCT-96  ;;2.0;INTEGRATED BILLING;\*\*51,137,207,232,349,432,**592**\*\*;21-MAR-94;Build 192  ;;Per VHA Directive 2004-038, this routine should not be modified.  ; DBIA SUPPORTED REF: GET^XUA4A72 = 1625  ; DBIA SUPPORTED REF: $$ESBLOCK^XUSESIG1 = 1557  Q  ; TESTPT(DFN) ; Determine if pt is test pt  ; Returns 1 if a test pt, 0 if not  Q $E($P($G(^DPT(+DFN,0)),U,9),1,5)="00000"  ; MAINPRV(IBIFN) ; Returns name^id^ien^type code of 'main' prov on bill IBIFN  N IBPRV,IBCOB,IBQ,Z  D GETPRV(IBIFN,"3,4",.IBPRV)  S IBQ="",IBCOB=$$COBN^IBCEF(IBIFN)  F Z=3,4 I $G(IBPRV(Z,1))'="" D  Q  . S IBQ=IBPRV(Z,1),$P(IBQ,U,4)=Z  . I $G(IBPRV(Z,1,IBCOB))'="" S $P(IBQ,U,2)=IBPRV(Z,1,IBCOB)  Q IBQ  ; PRVOK(VAL,IBIFN) ; Check bill form & prov function agree  ; VAL = internal value of prov function  ;  N OK,IBBT  S OK=0  Q:VAL="" OK  Q:'IBIFN OK **; JWS;IB\*2.0\*592 US1108 - add Dental form (7) check**  S IBBT=$$FT^IBCEF(IBIFN) ; 2 If CMS-1500, 3 If UB-04**, 7 if J430D Dental**  I IBBT=2!(**IBBT=7**) D  . I VAL=1 S OK=1 Q   ; CMS-1500, REFERRING  . I VAL=3 S OK=1 Q   ; CMS-1500, RENDERING  . I VAL=5 S OK=1 Q   ; CMS-1500, SUPERVISING  **. I IBBT=7,VAL=6 S OK=1 Q  ;J430D, ASSISTANT SURGEON**  **; JWS;IB\*2.0\*592 US1108 - end**  I 'OK,IBBT=3 D  . I VAL=1 S OK=1 Q   ; UB-04, REFERRING  . I VAL=2 S OK=1 Q   ; UB-04, OPERATING  . I VAL=3 S OK=1 Q   ; UB-04, RENDERING  . I VAL=4 S OK=1 Q   ; UB-04, ATTENDING  . I VAL=9 S OK=1 Q   ; UB-04, OTHER  ;  Q OK  ; PRVOK1(VAL,IBIFN) ; Check for both attending and rendering on bill  N OK  S OK=1  Q:$$FT^IBCEF(IBIFN)=3 1 ; both are allowed on UB  I $S("34"'[VAL:0,1:$D(^DGCR(399,IBIFN,"PRV","B",$S(VAL=3:4,1:3)))) D EN^DDIOL($S(VAL=3:"ATTENDING",1:"RENDERING")\_" ALREADY EXISTS - CAN'T HAVE BOTH ON ONE BILL") S OK=0  Q OK  ; SPEC(IBPRV,IBDT) ; Returns spec code for vp ien IBPRV from file 355.9  ; (for new person entries, as of date in IBDT)  ; DBIA 1625  N IBSPEC  S:'$G(IBDT) IBDT=DT  I IBPRV'["IBA(355.93" S IBSPEC=$S(IBPRV:$P($$GET^XUA4A72(+IBPRV,IBDT),U,8),1:"") ; VA  I IBPRV["IBA(355.93" S IBSPEC=$P($G(^IBA(355.93,+IBPRV,0)),U,4) ; Non-VA  Q IBSPEC  ; CRED(IBPRV,IBIFN,IBPIEN,IBTYP) ; Returns prov credentials  ; IBPRV = vp of provider for file 200 or 355.93  ; IBIFN = bill ien in file 399 (optional)  ; IBPIEN = prov ien - file 399.0222 (optional)  ; DEM;432 - prov ien can be from file 399.0404  ; as well (optional).  ; IBTYP = the prov type  ;  N IBCRED  S IBCRED=""  ;  ; DEM;432 - Provider can come from either file 399.0222, or  ; file 399.0404. Variable IBLNPRV is the flag  ; that indicates we want prov ien from file 399.0404.  ;  I '$G(IBLNPRV),$G(IBIFN),'$D(^DGCR(399,IBIFN,"PRV",0)) G CREDQ  ;  ; DEM;432 - Next line if for line level provider. Variable IBPROCP,  ; if it exist, is the procedure ien. File 399.0404 is a  ; multiple of the Procedure File 399.0304.  ;  I $G(IBLNPRV),$G(IBIFN),$G(IBPROCP),'$D(^DGCR(399,IBIFN,"CP",IBPROCP,"LNPRV",0)) G CREDQ  I '$G(IBLNPRV),$G(IBIFN),($G(IBPIEN)!$G(IBTYP)) D  . I '$G(IBPIEN) S IBPIEN=+$O(^DGCR(399,IBIFN,"PRV","B",IBTYP,0))  . S IBCRED=$P($G(^DGCR(399,IBIFN,"PRV",IBPIEN,0)),U,3)  ;  I $G(IBLNPRV),$G(IBIFN),$G(IBPROCP),($G(IBPIEN)!$G(IBTYP)) D  ; DEM;432 - Line Provider File 399.0404.  . I '$G(IBPIEN) S IBPIEN=+$O(^DGCR(399,IBIFN,"CP",IBPROCP,"B",IBTYP,0))  . S IBCRED=$P($G(^DGCR(399,IBIFN,"CP",IBPROCP,"LNPRV",IBPIEN,0)),U,3)  ; CREDQ ;  I $G(IBPRV),IBCRED="" D  . I IBPRV'["IBA(355.93" S IBCRED=$P($$ESBLOCK^XUSESIG1(+IBPRV),U,2)  . I IBPRV["IBA(355.93" S IBCRED=$P($G(^IBA(355.93,+IBPRV,0)),U,3)  Q IBCRED  ; GETPRV(IBIFN,IBTYP,IBPRV) ; Returns prov(s) of type(s) IBTYP for  ; bill ien IBIFN.  ; IBTYP = prov types needed, separated by ',' or ALL  ;   ; OUTPUT:  ; IBPRV array: IBPRV(type)= 1 if prov is from old prov flds  ; IBPRV(type,ct)=name^current COB id^vp provider ien^cred  ; IBPRV(type,ct,seq)=COB seq specific id  ; IBPRV(type)=default nm^def id  ; IBPRV(type,"NOTOPT")= defined if a required prov type  ;  N IB,IBCT,IBD,IBY,IBZ,IBMRAND,IBID,IBWNR,IBPNM,Z ;,IBZFID  ;S IBZFID=""  D F^IBCEF("N-CURRENT INS POLICY TYPE","IBZ",,IBIFN)  ;I IBZ="CI" D F^IBCEF("N-FEDERAL TAX ID","IBZFID",,IBIFN) S IBZFID=$TR(IBZFID,"-")  S IBPRV=U\_$G(IBZ),IBY=0  S IBMRAND=$$MCRONBIL^IBEFUNC(IBIFN)  ;WCJ;IB\*2.0\*432;Remove Default  I IBMRAND D  . ; F Z=1:1:3,5,6,7,8,9 S:Z=3&($$FT^IBCEF(IBIFN)=3) Z=4 S IBPRV(Z)=$S(Z=3!(Z=4):"DEPT VETERANS AFFAIRS",1:"")\_"^VAD000"  . F Z=1:1:9 S IBPRV(Z)="^VAD000"  . I '$$INPAT^IBCEF(IBIFN,1),$$FT^IBCEF(IBIFN)=3 S IBPRV(4,1)="^SLF000"  ;WCJ;IB\*2.0\*432;End changes  ;  ; For backwards compatability (before the claim level provider mulitple)  I '$D(^DGCR(399,+IBIFN,"PRV",0)) D  G GETQ  . N IBALL  . S IBALL=(IBTYP="ALL")  . I IBTYP[4!IBALL S:$P($G(^DGCR(399,+IBIFN,"U1")),U,13)'="" IBPRV(4,1)=$P(^("U1"),U,13),IBPRV(4)=1 Q:IBTYP=4  . I IBTYP[3!IBALL S:$P($G(^DGCR(399,+IBIFN,"UF2")),U)'="" IBPRV(3,1)=$P(^("UF2"),U),IBPRV(3)=1 Q:IBTYP=3  . I IBTYP[9!IBALL S:$P($G(^DGCR(399,+IBIFN,"U1")),U,14)'="" IBPRV(9,1)=$P(^("U1"),U,14),IBPRV(9)=1  ;  S IBID=4+$$COBN^IBCEF(IBIFN),IBWNR=$$WNRBILL^IBEFUNC(IBIFN)  F IBZ=1:1:$S(IBTYP="ALL":99,1:$L(IBTYP,",")) S (IBCT,IB)=0,IBY=$S(IBTYP'="ALL":$P(IBTYP,",",IBZ),1:$O(^DGCR(399,+IBIFN,"PRV","B",IBY))) Q:IBY=""  F  S IB=$O(^DGCR(399,+IBIFN,"PRV","B",IBY,IB)) Q:'IB  D  . S IBCT=IBCT+1  . S IBD=$G(^DGCR(399,+IBIFN,"PRV",IB,0))  . Q:'$P(IBD,U,2)  . S IBPNM=$$EXPAND^IBTRE(399.0222,.02,$P(IBD,U,2))  . I IBWNR Q:'$D(IBPRV(IBY)) S $P(IBD,U,IBID)=$P(IBPRV(IBY),U,2)  . S IBPRV(IBY,IBCT)=IBPNM\_U\_$S($P(IBD,U,IBID)'="":$P(IBD,U,IBID),$P($G(IBPRV(IBY)),U,2)'="":$P(IBPRV(IBY),U,2),1:$P($$DEFID^IBCEF74(IBIFN,IB),U,IBID-4))\_U\_$P(IBD,U,2)  . S $P(IBPRV(IBY,IBCT),U,4)=$$CRED($P(IBPRV(IBY,IBCT),U,3),IBIFN,$S($P(IBD,U,3)'=""!'$P(IBPRV(IBY,IBCT),U,3):IB,1:""))  . F Z=1:1:3 D  .. ;I IBZFID'="",'$$INPAT^IBCEF(IBIFN,1),$P(IBPRV(IBY,IBCT),U,2)="SLF000" S IBZFID=""  .. ;I $S(Z=1:1,1:$D(^DGCR(399,IBIFN,"I"\_Z))) S IBPRV(IBY,IBCT,Z)=$S($G(IBZFID)'="":IBZFID,$P(IBD,U,Z+4)'="":$P(IBD,U,Z+4),1:"")  .. I $S(Z=1:1,1:$D(^DGCR(399,IBIFN,"I"\_Z))) S IBPRV(IBY,IBCT,Z)=$S($P(IBD,U,Z+4)'="":$P(IBD,U,Z+4),1:$P($$DEFID^IBCEF74(IBIFN,IB),U,Z)) GETQ D NEEDPRV(IBIFN,IBTYP,.IBPRV)  Q  ; NEEDPRV(IBIFN,IBTYP,IBPRV) ; Check for needed prov  ; If needed, not entered, insert defaults for MCR only  N IB0,IBINP,IBFT,IBMRAND,IBTOB  S IB0=$G(^DGCR(399,+IBIFN,0))  S IBFT=($$FT^IBCEF(IBIFN)=3),IBINP=$$INPAT^IBCEF(IBIFN,1),IBTOB=$$TOB^IBCBB(IB0)  ; Only allow defaults for MCR  S IBMRAND=$$WNRBILL^IBEFUNC(IBIFN) ;$$MCRONBIL^IBEFUNC(IBIFN)  ;  I IBTYP="ALL"!((IBTYP\_",")["1,") D  . ; DEM;432 - UB-04 or CMS-1500 SITUATIONAL  . S IBPRV(1,"SITUATIONAL")=1  . Q  ;  I IBTYP="ALL"!((IBTYP\_",")["2,") D:IBFT  . ; only for bill type inpt - 11X, outpt - 83X  . S IBPRV(2,"SITUATIONAL")=1 ; DEM;432 - Default to "SITUATIONAL". If conditions below are met, then IBPRV(2,"SITUATIONAL") is KILLED and IBRPV is SET according to conditions.  . Q:$S(IBINP:$E(IBTOB,1,2)'="11",1:$E(IBTOB,1,2)'="83")  . ; UB-04 bill includes HCPCS procs - operating phys situational  . N Z  . S Z=0 F  S Z=$O(^DGCR(399,IBIFN,"CP",Z)) Q:'Z  I $P($G(^(Z,0)),U)["ICP" D  Q  .. K IBPRV(2,"SITUATIONAL") ; DEM;432 - We have met one of the condtions, so KILL IBPRV(2,"SITUATIONAL").   .. I IBINP S IBPRV(2,"SITUATIONAL")=1 Q  ; DEM;432 - If UB-04 (inpatient), then operating provider situational.  .. I 'IBINP S IBPRV(2,"NOTOPT")=1 ; DEM;432 - If UB-04 (outpatient), then operating provider required.  .. Q:'IBMRAND  .. I '$O(IBPRV(2,0)) S IBPRV(2,"REQ")=1,IBPRV(2,1)=$G(IBPRV(2))  ;  I IBTYP="ALL"!((IBTYP\_",")["3,") D  . ; if a CMS-1500 bill, rendering is required **. ; JWS;IB\*2.0\*592 US1108 - exclude dental form**  . I 'IBFT**,$$FT^IBCEF(IBIFN)'=7** S IBPRV(3,"NOTOPT")=1  . ; DEM;432 - if UB-04, rendering is situational.  **. ; JWS;IB\*2.0\*592 US1108 - dental form check**  . I IBFT**!($$FT^IBCEF(IBIFN)=7**) S IBPRV(3,"SITUATIONAL")=1 Q  . Q:'IBMRAND  . I '$O(IBPRV(3,0)) S IBPRV(3,1)=$G(IBPRV(3)),IBPRV(3,"REQ")=1  ;  I IBTYP="ALL"!((IBTYP\_",")["4,") D:IBFT  . ; if a UB-04, attending required  . S IBPRV(4,"NOTOPT")=1  . Q:'IBMRAND  . I '$O(IBPRV(4,0)) S IBPRV(4,1)=$G(IBPRV(4)),IBPRV(4,"REQ")=1  Q  ; CKPROV(IBIFN,IBTYP,IBVAL) ; Checks if prov of type IBTYP in 'PRV' node  ; of bill IBIFN  ; If IBVAL = 1, skips the check for an existing provider, just looks  ; for existence of the function itself  N OK,IBFT,Z,R  S OK=0,IBFT=$$FT^IBCEF(IBIFN)  S Z=+$O(^DGCR(399,IBIFN,"PRV","B",+IBTYP,0))  I $G(^DGCR(399,IBIFN,"PRV",Z,0))'="" D  . ; Only outpt UB-04 can have SLF000 as prov ID with no name  . I IBFT=3,'$$INPAT^IBCEF(IBIFN,1),$P(^DGCR(399,IBIFN,"PRV",Z,0),U,2)="",$P(^(0),U,5)="SLF000" S OK=1 Q  . I '$G(IBVAL) Q:$P(^DGCR(399,IBIFN,"PRV",Z,0),U,2)=""  . S OK=1  Q OK  ; XFER(IBQ) ; Transfer DILIST  ; IBQ = # of entries already found  N Z,IBZ  S (Z,IBZ)=0  F  S Z=$O(^TMP("DILIST",$J,1,Z)) Q:'Z  S IBZ=IBZ+1,^TMP("IBLIST",$J,1,IBZ+IBQ)=^TMP("DILIST",$J,1,Z),^TMP("IBLIST",$J,2,IBZ+IBQ)=^TMP("DILIST",$J,2,Z) M ^TMP("IBLIST",$J,"ID",IBZ+IBQ)=^TMP("DILIST",$J,"ID",Z)  ;  I $D(^TMP("DILIST",$J,0)) S ^TMP("IBLIST",$J,0)=^TMP("DILIST",$J,0)  S $P(^TMP("IBLIST",$J,0),U)=IBQ+IBZ  Q  ; DATE(X) ; Convert date X in YYYYMMDD or YYMMDD to FM format  ; FP = flag to indicate if past or future dates are expected  N %DT,Y  I $L(X)=8,$E(X,1,4)<2100,$E(X,5,6)<13,$E(X,7,8)<32 S X=$E(X,1,4)-1700\_$E(X,5,8) G DTQ  I $L(X)=6,$E(X,3,4)<13,$E(X,5,6)<32 S X=$E(X,3,4)\_"/"\_$E(X,5,6)\_"/"\_$E(X,1,2),%DT="N" D ^%DT I Y>0 S X=Y DTQ Q X  ; BCLASS(IBIFN) ; Returns actual bill classif. code from ptr fld  ; .25 in file 399 for bill ien IBIFN  Q $P($G(^DGCR(399.1,+$P($G(^DGCR(399,IBIFN,0)),U,25),0)),U,2)  ; ADMHR(IBIFN,IBDTTM) ; Extract admit hr from admit dt/tm  ; Default 00 if no time and bill is 11X or 18X  N TM  S TM=$P(IBDTTM,".",2)  I TM="","18"[$$BCLASS(IBIFN),$P($G(^DGCR(399,IBIFN,0)),U,24)=1 S TM="00"  I TM'="",TM'="00" S TM=$E(TM\_"0000",1,4)  Q TM  ; OLAB(IBIFN) ; Returns 1 if bill IBIFN is outside lab  N IBL,IBLAB  S IBL=0  S IBLAB=$P($G(^DGCR(399,IBIFN,"U2")),U,11)  I IBLAB,"24"[IBLAB S IBL=1  Q IBL  ; PSRV(IBIFN) ; Returns 1 if bill IBIFN has any purch services  N IBZ,IBXDATA,IBXSAVE,Z  S IBZ=0  D F^IBCEF("N-HCFA 1500 PROCEDURES",,,IBIFN)  S Z=0 F  S Z=$O(IBXSAVE("BOX24",Z)) Q:'Z  I $P(IBXSAVE("BOX24",Z),U,11) S IBZ=1 Q  Q IBZ  ; SEQBILL(IBIFN) ; Returns the ien's of all bills in COB sequence for bill IBIFN  ; Return value is "^" delimited: primary ien^secondary ien^tertiary ien  N IBSEQ,Z  S IBSEQ=$P($G(^DGCR(399,IBIFN,"M1")),U,5,7)  S Z=$$COBN^IBCEF(IBIFN)  I $P(IBSEQ,U,Z)="" S $P(IBSEQ,U,Z)=IBIFN  Q IBSEQ  ;  ;IB\*2.0\*432/TAZ Added to take into account the line level providers. GETPRV1(IBIFN,IBTYP,IBPRV) ; Returns prov(s) of type(s) IBTYP for  ; bill ien IBIFN for TPJI display  ; IBTYP = prov types needed, separated by ',' or ALL  ;   ; OUTPUT:  ; IBPRV array: IBPRV(level,type,ct)=name^current COB id^vp provider ien^cred  ;  N IB,IBCT,IBD,IBY,IBZ,IBMRAND,IBID,IBWNR,IBPNM,Z,IBPRTYP  D F^IBCEF("N-CURRENT INS POLICY TYPE","IBZ",,IBIFN)  S IBPRV=U\_$G(IBZ),IBY=0  D ALLIDS^IBCEFP(IBIFN,.IBXSAVE)  S IBCT=0  F  S IBCT=$O(IBXSAVE("PROVINF",IBIFN,"C",IBCT)) Q:'IBCT  D  . S IBPRTYP=""  . F  S IBPRTYP=$O(IBXSAVE("PROVINF",IBIFN,"C",IBCT,IBPRTYP)) Q:'IBPRTYP  D  .. I IBTYP'="ALL",IBTYP'[IBPRTYP Q  ;Screen out unwanted providers  .. N IBPRIEN,OBPRNM,IBCOBID  .. S IBPRIEN=$P(IBXSAVE("PROVINF",IBIFN,"C",IBCT,IBPRTYP),U)  .. S $P(IBPRV(1,IBCT,IBPRTYP),U,1)=$$EXPAND^IBTRE(399.0222,.02,IBPRIEN)  .. S $P(IBPRV(1,IBCT,IBPRTYP),U,2)=IBXSAVE("PROVINF",IBIFN,"C",IBCT,IBPRTYP,"COBID")  .. S $P(IBPRV(1,IBCT,IBPRTYP),U,3)=IBPRIEN  .. S $P(IBPRV(1,IBCT,IBPRTYP),U,4)=$P(IBXSAVE("PROVINF",IBIFN,"C",IBCT,IBPRTYP,"NAME"),U,4)  S IBCT=0  F  S IBCT=$O(IBXSAVE("L-PROV",IBIFN,IBCT)) Q:'IBCT  D  . S IBPRTYP=""  . F  S IBPRTYP=$O(IBXSAVE("L-PROV",IBIFN,IBCT,"C",1,IBPRTYP)) Q:'IBPRTYP  D  .. I IBTYP'="ALL",IBTYP'[IBPRTYP Q  ;Screen out unwanted providers  .. N IBPRIEN  .. S IBPRIEN=$P(IBXSAVE("L-PROV",IBIFN,IBCT,"C",1,IBPRTYP),U)  .. S IBPRV(2,IBCT,IBPRTYP)=$$EXPAND^IBTRE(399.0222,.02,IBPRIEN)  .. S $P(IBPRV(2,IBCT,IBPRTYP),U,2)=IBXSAVE("L-PROV",IBIFN,IBCT,"C",1,IBPRTYP,"COBID")  .. S $P(IBPRV(2,IBCT,IBPRTYP),U,3)=IBPRIEN  .. S $P(IBPRV(2,IBCT,IBPRTYP),U,4)=$P(IBXSAVE("L-PROV",IBIFN,IBCT,"C",1,IBPRTYP,"NAME"),U,4)  Q **;/IB\*2.0\*592** **RTYPOK(VAL,IBIFN) ;sceen for field 399,285 Attachment Report Type - Check for a valid Report Type depending on Claim Type  ; VAL = internal value of report type file#353.3  ; IBIFN = file 399 ien  ;  N OK,IBBT  S OK=0  Q:VAL="" OK  Q:'IBIFN OK  S IBBT=$$FT^IBCEF(IBIFN) ;2 if CMS-1500, 3 if UB-04, 7 if J430D Dental  I IBBT'=7 S:VAL'="P6" OK=1 Q OK  ;not a Dental Claim, periodontal charts not applicable  ; following for Dental claims  I "^B4^DA^DG^EB^OZ^P6^RB^RR^"[(U\_VAL\_U) S OK=1  Q OK  ; IB\*2.0\*592 end**  ; | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCEU0 | | | | | | | | |
| **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** | | | | | | | | | |
| IBCEU0 ;ALB/TMP - EDI UTILITIES ;02-OCT-96  ;;2.0;INTEGRATED BILLING;\*\*137,197,155,296,349,417,432\*\*;21-MAR-94;Build 192  ;;Per VHA Directive 2004-038, this routine should not be modified.  ; NOTECHG(IBDA,IBNTEXT) ; Enter who/when review stat change was entered  ; IBDA = ien of entry in file 361.1  ; IBNTEXT = array containing the lines of text to store if not using the  ; default text IBNTEXT = # of lines IBNTEXT(#)=line text  N IBIEN,IBTEXT,DA,X,Y,DIC,DO,DLAYGO,DD  S DA(1)=IBDA,DIC="^IBM(361.1,"\_DA(1)\_",2,",DIC(0)="L",DLAYGO=361.121  S X=$$NOW^XLFDT  D FILE^DICN K DIC,DD,DO,DLAYGO  Q:Y'>0  S DA(2)=DA(1),DA(1)=+Y,IBIEN=DA(1)\_","\_DA(2)\_","  I $G(IBNTEXT) D  . M IBTEXT=IBNTEXT  E  D  . S IBTEXT(1)="REVIEW STATUS CHANGED TO '"\_$$EXTERNAL^DILFD(361.1,.2,,$P(^IBM(361.1,DA(2),0),U,20))\_"' BY: "\_$$EXTERNAL^DILFD(361.121,.02,,+$G(DUZ))  D WP^DIE(361.121,IBIEN,.03,,"IBTEXT") K ^TMP("DIERR",$J)  Q  ; LOCK(IBFILE,IBREC) ; Lock record # IBREC in file #IBFILE (361 or 361.1)  N OK  S OK=0  L +^IBM(IBFILE,IBREC):3 I $T S OK=1  I 'OK D  . W !,"Another user has locked this record - try again later"  . D PAUSE^VALM1  Q OK  ; UNLOCK(IBFILE,IBREC) ; Unlock record # IBREC in file #IBFILE  I $G(IBREC) L -^IBM(IBFILE,IBREC)  Q  ; MSTAT ; Enter reviewed by selected range  N IBDAX,IBA,IBCLOSE,IBLOOK,IBOK,IBSTOP,IBREBLD,IBCLOK,DA,DIR,X,Y,DIE,DR  D FULL^VALM1  D SEL^IBCECSA4(.IBDAX)  S IBREBLD=0  I $O(IBDAX(""))="" G MSTATQ  S DIR("?,1")="ONLY SELECT TO CLOSE THE TRANSMIT RECORDS IF YOU KNOW THESE ARE THE FINAL",DIR("?",2)=" ELECTRONIC MESSAGES YOU WILL RECEIVE FOR ALL THE BILLS REFERENCED BY",DIR("?")=" THESE MESSAGES"  S DIR(0)="YA",DIR("A",1)="DO YOU WANT TO AUTOMATICALLY CLOSE THE TRANSMIT RECORDS FOR ANY MESSAGES",DIR("A")=" THAT AREN'T REJECTS?: ",DIR("B")="NO" W ! D ^DIR K DIR W !  G:$D(DIRUT) MSTATQ  S IBCLOSE=(Y=1)  S DIR(0)="YA",DIR("A")="DO YOU WANT TO SEE EACH MESSAGE BEFORE MARKING IT REVIEWED?: ",DIR("B")="NO"  S DIR("?",1)="IF YOU OPT TO SEE EACH MESSAGE, YOU CAN CONTROL WHETHER OR NOT THE MESSAGE",DIR("?",2)=" IS MARKED AS REVIEWED"  I 'IBCLOSE S DIR("?")=DIR("?",2) K DIR("?",2)  I IBCLOSE S DIR("?",2)=DIR("?",2)\_" AND, FOR NON-REJECTS, WHETHER OR NOT TO CLOSE THE",DIR("?")=" TRANSMIT RECORD FOR THE BILL"  W ! D ^DIR K DIR W !  G:$D(DIRUT) MSTATQ  S IBLOOK=(Y=1)  S IBDAX=0,IBSTOP=0  F  S IBDAX=+$O(IBDAX(IBDAX)) Q:'IBDAX  D  Q:IBSTOP  . S IBA=$G(IBDAX(IBDAX))  . S DIE="^IBM(361,",DA=$P(IBA,U,2),DR=""  . I DA D  .. S IBOK=1  .. S IBCLOK=$S(IBCLOSE:1,1:0)  .. I IBLOOK D  Q:'IBOK  ... S DIC="^IBM(361," D EN^DIQ  ... I '$$LOCK(361,DA) W ! S IBOK=0 Q  ... S DIR(0)="YA",DIR("A")="OK TO MARK REVIEWED?: ",DIR("B")="YES",DIR("?",1)="IF YOU ENTER YES, THIS MESSAGE WILL BE MARKED REVIEWED"  ... S DIR("?",2)="IF YOU ENTER NO, THIS MESSAGE WILL NOT BE ALTERED",DIR("?",3)="IF YOU ENTER AN ^, THIS MESSAGE WILL NOT BE ALTERED & NONE OF THE",DIR("?")=" REMAINING MESSAGES WILL BE PROCESSED" D ^DIR K DIR  ... I Y'>0 S IBOK=0 S:$D(DIRUT) IBSTOP=1 Q  ... I 'IBCLOSE D  .... S DIR(0)="YA",DIR("A")="OK TO CLOSE THIS BILL'S TRANSMIT RECORD?: ",DIR("B")="NO"  .... S DIR("?",1)="If you respond YES to this prompt, the transmit status of this bill will",DIR("?",2)=" be set to CLOSED. No further electronic processing of this bill will be"  .... S DIR("?",3)=" allowed. If you respond NO to this prompt, this electronic message will",DIR("?",4)=" be filed as reviewed, but the bill's transmit status will not be changed."  .... S DIR("?",5)=" You may wish to periodically print a list of bills with a non-final",DIR("?",6)=" (closed/cancelled/etc) status to ensure the electronic processing of all"  .... S DIR("?",7)=" bills has been completed. Closing the transmit bill record here will",DIR("?")=" eliminate the bill from this list."  .... W ! D ^DIR K DIR W !  .... I Y'=1 S IBCLOK=0  .. I 'IBLOOK,$P($G(^IBM(361,DA,0)),U,3)="R" D  Q:'IBOK  ... S DR="1",DIC="^IBM(361," D EN^DIQ W !,"Bill Number: ",$$EXPAND^IBTRE(361,.01,+^IBM(361,DA,0))  ... S DIR(0)="YA",DIR("A")="THIS IS A REJECTION ... ARE YOU SURE YOU WANT TO MARK IT REVIEWED?: ",DIR("B")="NO"  ... S DIR("?",1)="IF YOU ENTER YES, THIS MESSAGE WILL BE MARKED REVIEWED"  ... S DIR("?",2)="IF YOU ENTER NO, THIS MESSAGE WILL NOT BE ALTERED",DIR("?",3)="IF YOU ENTER AN ^, THIS MESSAGE WILL NOT BE ALTERED & NONE OF THE",DIR("?")=" MESSAGES FOLLOWING THIS ONE WILL BE PROCESSED" D ^DIR K DIR  ... I Y'=1 S IBOK=0 S:$D(DIRUT) IBSTOP=1  .. S:'IBREBLD IBREBLD=1  .. S DR=".09////2;.1////F" D ^DIE  .. N IBUPD  .. S IBUPD=0  .. I $$PRINTUPD($G(^IBM(361,DA,1,1,0)),+$P(^IBM(361,DA,0),U,11)) S IBUPD=1  .. I $G(^IBM(361,DA,1,1,0))["CLAIM SENT TO PAYER" D UPDTX^IBCECSA2(+$P(^IBM(361,DA,0),U,11),$S(IBCLOK:"Z",1:"A2")) S IBUPD=1  .. I $G(^IBM(361,DA,1,1,0))["CLAIM REJECTED" D UPDTX^IBCECSA2(+$P(^IBM(361,DA,0),U,11),"E") S IBUPD=1  .. I IBCLOK,'IBUPD D UPDTX^IBCECSA2(+$P(^IBM(361,DA,0),U,11),"Z")  .. I 'IBLOOK D  ... W !,"Seq #: ",IBDAX," Bill number: ",$$EXPAND^IBTRE(361,.01,+^IBM(361,DA,0)),?45,"REVIEWED"  .. D NOTECHG^IBCECSA2(DA,1)  .. D UNLOCK(361,DA)  W !!,"LAST SELECTION PROCESSED",!  D PAUSE^VALM1 MSTATQ S VALMBCK="R"  I IBREBLD D BLD^IBCECSA1  Q  ; PRPAY(IBIFN,IBMCR) ; Returns total amount of prior payments applied to  ; bill ien IBIFN  ; IBMCR = flag passed in as 1 if MRA total should be included  ;  N IBTOT,IBZ,IBSEQ  S IBSEQ=$$COBN^IBCEF(IBIFN)  I IBSEQ'>1 S IBTOT=0 G PRPAYQ  D F^IBCEF("N-PRIOR PAYMENTS","IBZ",,IBIFN)  S IBTOT=IBZ  I $G(IBMCR),$$MCRONBIL^IBEFUNC(IBIFN)=1 D  ; MCR on bill before curr ins  . N Z,Z0,Z2,Q  . F Z=1:1:IBSEQ-1 I $$WNRBILL^IBEFUNC(IBIFN,Z) D  .. S IBTOT=+$$MCRPAY(IBIFN) PRPAYQ Q IBTOT  ; PRINTUPD(IBTEXT,IBDA) ; If the status message indicates claim was printed  ; or the claim record in file 399 says it was, update the transmit  ; message status to closed  ; IBTEXT = the first line text of the status message (optional)  ; IBDA = the ien of the transmission record in file 364  ;  ; FUNCTION returns 1 if message status changed  ;  N IBP,IBP1  S IBP=0,IBP1=$P($G(^DGCR(399,+$G(^IBA(364,+$G(IBDA),0)),"TX")),U,7)  I $G(IBTEXT)["CLAIM RECEIVED, PRINTED AND MAILED BY PRINT CENTER"!IBP1 D  . N Z  . S Z=$E($P($G(^IBA(364,IBDA,0)),U,3),1)  . I "AP"'[Z Q  ; Only change if status is pending or received/accepted  . D UPDTX^IBCECSA2(IBDA,"Z") S IBP=1  Q IBP  ; MCRPAY(IBIFN) ; Calculate MRA total for the bill IBIFN  N IBPAY,Q,Z0  S IBPAY=0  ;include eligible bill for process  ; 432 - added MRA flag to IBCEU1 to not always screen out non-MRA's  S Q=0 F  S Q=$O(^IBM(361.1,"B",IBIFN,Q)) Q:'Q  I $$EOBELIG^IBCEU1(Q,1) S IBPAY=IBPAY+$P($G(^IBM(361.1,Q,1)),U,1)  Q IBPAY  ; PREOBTOT(IBIFN,IBMRANOT) ; Function - Calculates Patient Responsibility Amount  ; Input: IBIFN - ien of Bill Number (ien of file 399)  ; IBMRANOT - flag to indicate that this is NOT and MRA  ; Output Function returns: Patient Responsibility Amount for all EOB's for bill  ;  N FRMTYP,IBPTRES  S IBPTRES=0  ; Form Type 2=CMS-1500; 3=UB-04  S FRMTYP=$$FT^IBCEF(IBIFN)  ;  ; For bills w/CMS-1500 Form Type, total up Pt Resp amount from top  ; level of EOB (field 1.02) for All MRA type EOB's on file for that  ; bill (IBIFN)  ;   I FRMTYP=2 D  Q IBPTRES  . N IBEOB,EOBREC,EOBREC1,IBPRTOT  . S (IBEOB,IBPRTOT,IBPTRES)=0  . F  S IBEOB=$O(^IBM(361.1,"B",IBIFN,IBEOB)) Q:'IBEOB  D  ;  . . S EOBREC=$G(^IBM(361.1,IBEOB,0)),EOBREC1=$G(^(1))  .. ; IB\*2.0\*432 allow for non-MRA's  . . I $G(IBMRANOT)'=1,$P(EOBREC,U,4)'=1 Q  ;make sure it's an MRA  . . Q:$D(^IBM(361.1,IBEOB,"ERR")) ;no filing error  . . ; Total up Pt Resp Amounts on all valid MRA's  . . S IBPTRES=IBPTRES+$P(EOBREC1,U,2)  ;  ; For bills w/UB-04 Form Type, loop through all EOB's and sum up amounts  ; on both Line level and on Claim level  N EOBADJ,IBEOB,LNLVL  S IBEOB=0  F  S IBEOB=$O(^IBM(361.1,"B",IBIFN,IBEOB)) Q:'IBEOB  D  ;  . ; IB\*2.0\*432 allow for non-MRA's  . I $G(IBMRANOT)'=1,$P($G(^IBM(361.1,IBEOB,0)),U,4)'=1 Q    ; must be an MRA  . Q:$D(^IBM(361.1,IBEOB,"ERR")) ; no filing error  . ; get claim level adjustments  . K EOBADJ M EOBADJ=^IBM(361.1,IBEOB,10)  . S IBPTRES=IBPTRES+$$CALCPR(.EOBADJ)  . ;  . ; get line level adjustments  . S LNLVL=0  . F  S LNLVL=$O(^IBM(361.1,IBEOB,15,LNLVL)) Q:'LNLVL  D  ;  . . K EOBADJ M EOBADJ=^IBM(361.1,IBEOB,15,LNLVL,1)  . . S IBPTRES=IBPTRES+$$CALCPR(.EOBADJ)  Q IBPTRES  ; CALCPR(EOBADJ) ; Function - Calculate Patient Responsibilty Amount  ; For Group Code PR; Ignore the PR-AAA kludge  ; Input - EOBADJ = Array of Group Codes & Reason Codes from either the Claim  ; Level (10) or Service Line Level (15) of EOB file (#361.1)  ; Output - Function returns Patient Responsibility Amount  ;  N GRPLVL,RSNCD,RSNAMT,PTRESP  S (GRPLVL,PTRESP)=0  F  S GRPLVL=$O(EOBADJ(GRPLVL)) Q:'GRPLVL  D  . I $P($G(EOBADJ(GRPLVL,0)),U)'="PR" Q  ;grp code must be PR  . S RSNCD=0  . F  S RSNCD=$O(EOBADJ(GRPLVL,1,RSNCD)) Q:'RSNCD  D  . . I $P($G(EOBADJ(GRPLVL,1,RSNCD,0)),U,1)="AAA" Q   ; ignore PR-AAA  . . S RSNAMT=$P($G(EOBADJ(GRPLVL,1,RSNCD,0)),U,2)  . . S PTRESP=PTRESP+RSNAMT  Q PTRESP  ; COBMOD(IBXSAVE,IBXDATA,SEQ) ; output the modifiers from the COB  ; SEQ is which modifier we're extracting (1-4)  ; Build IBXDATA(line#)=Modifier# SEQ  NEW LN,N,Z,MOD,LNSEQ  KILL IBXDATA  I '$G(SEQ) Q  S (LN,LNSEQ)=0  F  S LN=$O(IBXSAVE("LCOB",LN)) Q:'LN  D  . S LNSEQ=LNSEQ+1  . S (N,Z)=0  . F  S Z=$O(IBXSAVE("LCOB",LN,"COBMOD",Z)) Q:'Z  D  .. S N=N+1  .. S MOD(LNSEQ,N)=$P($G(IBXSAVE("LCOB",LN,"COBMOD",Z,0)),U,1)  .. Q  . S MOD=$G(MOD(LNSEQ,SEQ))  . I MOD'="" S IBXDATA(LNSEQ)=MOD  . Q  Q  ; | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCEU0 ;ALB/TMP - EDI UTILITIES ;02-OCT-96  ;;2.0;INTEGRATED BILLING;\*\*137,197,155,296,349,417,432,**592**\*\*;21-MAR-94;Build 192  ;;Per VHA Directive 2004-038, this routine should not be modified.  ; NOTECHG(IBDA,IBNTEXT) ; Enter who/when review stat change was entered  ; IBDA = ien of entry in file 361.1  ; IBNTEXT = array containing the lines of text to store if not using the  ; default text IBNTEXT = # of lines IBNTEXT(#)=line text  N IBIEN,IBTEXT,DA,X,Y,DIC,DO,DLAYGO,DD  S DA(1)=IBDA,DIC="^IBM(361.1,"\_DA(1)\_",2,",DIC(0)="L",DLAYGO=361.121  S X=$$NOW^XLFDT  D FILE^DICN K DIC,DD,DO,DLAYGO  Q:Y'>0  S DA(2)=DA(1),DA(1)=+Y,IBIEN=DA(1)\_","\_DA(2)\_","  I $G(IBNTEXT) D  . M IBTEXT=IBNTEXT  E  D  . S IBTEXT(1)="REVIEW STATUS CHANGED TO '"\_$$EXTERNAL^DILFD(361.1,.2,,$P(^IBM(361.1,DA(2),0),U,20))\_"' BY: "\_$$EXTERNAL^DILFD(361.121,.02,,+$G(DUZ))  D WP^DIE(361.121,IBIEN,.03,,"IBTEXT") K ^TMP("DIERR",$J)  Q  ; LOCK(IBFILE,IBREC) ; Lock record # IBREC in file #IBFILE (361 or 361.1)  N OK  S OK=0  L +^IBM(IBFILE,IBREC):3 I $T S OK=1  I 'OK D  . W !,"Another user has locked this record - try again later"  . D PAUSE^VALM1  Q OK  ; UNLOCK(IBFILE,IBREC) ; Unlock record # IBREC in file #IBFILE  I $G(IBREC) L -^IBM(IBFILE,IBREC)  Q  ; MSTAT ; Enter reviewed by selected range  N IBDAX,IBA,IBCLOSE,IBLOOK,IBOK,IBSTOP,IBREBLD,IBCLOK,DA,DIR,X,Y,DIE,DR  D FULL^VALM1  D SEL^IBCECSA4(.IBDAX)  S IBREBLD=0  I $O(IBDAX(""))="" G MSTATQ  S DIR("?,1")="ONLY SELECT TO CLOSE THE TRANSMIT RECORDS IF YOU KNOW THESE ARE THE FINAL",DIR("?",2)=" ELECTRONIC MESSAGES YOU WILL RECEIVE FOR ALL THE BILLS REFERENCED BY",DIR("?")=" THESE MESSAGES"  S DIR(0)="YA",DIR("A",1)="DO YOU WANT TO AUTOMATICALLY CLOSE THE TRANSMIT RECORDS FOR ANY MESSAGES",DIR("A")=" THAT AREN'T REJECTS?: ",DIR("B")="NO" W ! D ^DIR K DIR W !  G:$D(DIRUT) MSTATQ  S IBCLOSE=(Y=1)  S DIR(0)="YA",DIR("A")="DO YOU WANT TO SEE EACH MESSAGE BEFORE MARKING IT REVIEWED?: ",DIR("B")="NO"  S DIR("?",1)="IF YOU OPT TO SEE EACH MESSAGE, YOU CAN CONTROL WHETHER OR NOT THE MESSAGE",DIR("?",2)=" IS MARKED AS REVIEWED"  I 'IBCLOSE S DIR("?")=DIR("?",2) K DIR("?",2)  I IBCLOSE S DIR("?",2)=DIR("?",2)\_" AND, FOR NON-REJECTS, WHETHER OR NOT TO CLOSE THE",DIR("?")=" TRANSMIT RECORD FOR THE BILL"  W ! D ^DIR K DIR W !  G:$D(DIRUT) MSTATQ  S IBLOOK=(Y=1)  S IBDAX=0,IBSTOP=0  F  S IBDAX=+$O(IBDAX(IBDAX)) Q:'IBDAX  D  Q:IBSTOP  . S IBA=$G(IBDAX(IBDAX))  . S DIE="^IBM(361,",DA=$P(IBA,U,2),DR=""  . I DA D  .. S IBOK=1  .. S IBCLOK=$S(IBCLOSE:1,1:0)  .. I IBLOOK D  Q:'IBOK  ... S DIC="^IBM(361," D EN^DIQ  ... I '$$LOCK(361,DA) W ! S IBOK=0 Q  ... S DIR(0)="YA",DIR("A")="OK TO MARK REVIEWED?: ",DIR("B")="YES",DIR("?",1)="IF YOU ENTER YES, THIS MESSAGE WILL BE MARKED REVIEWED"  ... S DIR("?",2)="IF YOU ENTER NO, THIS MESSAGE WILL NOT BE ALTERED",DIR("?",3)="IF YOU ENTER AN ^, THIS MESSAGE WILL NOT BE ALTERED & NONE OF THE",DIR("?")=" REMAINING MESSAGES WILL BE PROCESSED" D ^DIR K DIR  ... I Y'>0 S IBOK=0 S:$D(DIRUT) IBSTOP=1 Q  ... I 'IBCLOSE D  .... S DIR(0)="YA",DIR("A")="OK TO CLOSE THIS BILL'S TRANSMIT RECORD?: ",DIR("B")="NO"  .... S DIR("?",1)="If you respond YES to this prompt, the transmit status of this bill will",DIR("?",2)=" be set to CLOSED. No further electronic processing of this bill will be"  .... S DIR("?",3)=" allowed. If you respond NO to this prompt, this electronic message will",DIR("?",4)=" be filed as reviewed, but the bill's transmit status will not be changed."  .... S DIR("?",5)=" You may wish to periodically print a list of bills with a non-final",DIR("?",6)=" (closed/cancelled/etc) status to ensure the electronic processing of all"  .... S DIR("?",7)=" bills has been completed. Closing the transmit bill record here will",DIR("?")=" eliminate the bill from this list."  .... W ! D ^DIR K DIR W !  .... I Y'=1 S IBCLOK=0  .. I 'IBLOOK,$P($G(^IBM(361,DA,0)),U,3)="R" D  Q:'IBOK  ... S DR="1",DIC="^IBM(361," D EN^DIQ W !,"Bill Number: ",$$EXPAND^IBTRE(361,.01,+^IBM(361,DA,0))  ... S DIR(0)="YA",DIR("A")="THIS IS A REJECTION ... ARE YOU SURE YOU WANT TO MARK IT REVIEWED?: ",DIR("B")="NO"  ... S DIR("?",1)="IF YOU ENTER YES, THIS MESSAGE WILL BE MARKED REVIEWED"  ... S DIR("?",2)="IF YOU ENTER NO, THIS MESSAGE WILL NOT BE ALTERED",DIR("?",3)="IF YOU ENTER AN ^, THIS MESSAGE WILL NOT BE ALTERED & NONE OF THE",DIR("?")=" MESSAGES FOLLOWING THIS ONE WILL BE PROCESSED" D ^DIR K DIR  ... I Y'=1 S IBOK=0 S:$D(DIRUT) IBSTOP=1  .. S:'IBREBLD IBREBLD=1  .. S DR=".09////2;.1////F" D ^DIE  .. N IBUPD  .. S IBUPD=0  .. I $$PRINTUPD($G(^IBM(361,DA,1,1,0)),+$P(^IBM(361,DA,0),U,11)) S IBUPD=1  .. I $G(^IBM(361,DA,1,1,0))["CLAIM SENT TO PAYER" D UPDTX^IBCECSA2(+$P(^IBM(361,DA,0),U,11),$S(IBCLOK:"Z",1:"A2")) S IBUPD=1  .. I $G(^IBM(361,DA,1,1,0))["CLAIM REJECTED" D UPDTX^IBCECSA2(+$P(^IBM(361,DA,0),U,11),"E") S IBUPD=1  .. I IBCLOK,'IBUPD D UPDTX^IBCECSA2(+$P(^IBM(361,DA,0),U,11),"Z")  .. I 'IBLOOK D  ... W !,"Seq #: ",IBDAX," Bill number: ",$$EXPAND^IBTRE(361,.01,+^IBM(361,DA,0)),?45,"REVIEWED"  .. D NOTECHG^IBCECSA2(DA,1)  .. D UNLOCK(361,DA)  W !!,"LAST SELECTION PROCESSED",!  D PAUSE^VALM1 MSTATQ S VALMBCK="R"  I IBREBLD D BLD^IBCECSA1  Q  ; PRPAY(IBIFN,IBMCR) ; Returns total amount of prior payments applied to  ; bill ien IBIFN  ; IBMCR = flag passed in as 1 if MRA total should be included  ;  N IBTOT,IBZ,IBSEQ  S IBSEQ=$$COBN^IBCEF(IBIFN)  I IBSEQ'>1 S IBTOT=0 G PRPAYQ  D F^IBCEF("N-PRIOR PAYMENTS","IBZ",,IBIFN)  S IBTOT=IBZ  I $G(IBMCR),$$MCRONBIL^IBEFUNC(IBIFN)=1 D  ; MCR on bill before curr ins  . N Z,Z0,Z2,Q  . F Z=1:1:IBSEQ-1 I $$WNRBILL^IBEFUNC(IBIFN,Z) D  .. S IBTOT=+$$MCRPAY(IBIFN) PRPAYQ Q IBTOT  ; PRINTUPD(IBTEXT,IBDA) ; If the status message indicates claim was printed  ; or the claim record in file 399 says it was, update the transmit  ; message status to closed  ; IBTEXT = the first line text of the status message (optional)  ; IBDA = the ien of the transmission record in file 364  ;  ; FUNCTION returns 1 if message status changed  ;  N IBP,IBP1  S IBP=0,IBP1=$P($G(^DGCR(399,+$G(^IBA(364,+$G(IBDA),0)),"TX")),U,7)  I $G(IBTEXT)["CLAIM RECEIVED, PRINTED AND MAILED BY PRINT CENTER"!IBP1 D  . N Z  . S Z=$E($P($G(^IBA(364,IBDA,0)),U,3),1)  . I "AP"'[Z Q  ; Only change if status is pending or received/accepted  . D UPDTX^IBCECSA2(IBDA,"Z") S IBP=1  Q IBP  ; MCRPAY(IBIFN) ; Calculate MRA total for the bill IBIFN  N IBPAY,Q,Z0  S IBPAY=0  ;include eligible bill for process  ; 432 - added MRA flag to IBCEU1 to not always screen out non-MRA's  S Q=0 F  S Q=$O(^IBM(361.1,"B",IBIFN,Q)) Q:'Q  I $$EOBELIG^IBCEU1(Q,1) S IBPAY=IBPAY+$P($G(^IBM(361.1,Q,1)),U,1)  Q IBPAY  ; PREOBTOT(IBIFN,IBMRANOT) ; Function - Calculates Patient Responsibility Amount  ; Input: IBIFN - ien of Bill Number (ien of file 399)  ; IBMRANOT - flag to indicate that this is NOT and MRA  ; Output Function returns: Patient Responsibility Amount for all EOB's for bill  ;  N FRMTYP,IBPTRES  S IBPTRES=0 **;JWS;IB\*2.0\*592: Dental form 7**  ; Form Type 2=CMS-1500; 3=UB-04; **7=J430D Dental**  S FRMTYP=$$FT^IBCEF(IBIFN)  ;  ; For bills w/CMS-1500 Form Type, total up Pt Resp amount from top  ; level of EOB (field 1.02) for All MRA type EOB's on file for that  ; bill (IBIFN)  ;  **;JWS;IB\*2.0\*592: Dental form 7**  I FRMTYP=2!(**FRMTYP=7**) D  Q IBPTRES  . N IBEOB,EOBREC,EOBREC1,IBPRTOT  . S (IBEOB,IBPRTOT,IBPTRES)=0  . F  S IBEOB=$O(^IBM(361.1,"B",IBIFN,IBEOB)) Q:'IBEOB  D  ;  . . S EOBREC=$G(^IBM(361.1,IBEOB,0)),EOBREC1=$G(^(1))  .. ; IB\*2.0\*432 allow for non-MRA's  . . I $G(IBMRANOT)'=1,$P(EOBREC,U,4)'=1 Q  ;make sure it's an MRA  . . Q:$D(^IBM(361.1,IBEOB,"ERR")) ;no filing error  . . ; Total up Pt Resp Amounts on all valid MRA's  . . S IBPTRES=IBPTRES+$P(EOBREC1,U,2)  ;  ; For bills w/UB-04 Form Type, loop through all EOB's and sum up amounts  ; on both Line level and on Claim level  N EOBADJ,IBEOB,LNLVL  S IBEOB=0  F  S IBEOB=$O(^IBM(361.1,"B",IBIFN,IBEOB)) Q:'IBEOB  D  ;  . ; IB\*2.0\*432 allow for non-MRA's  . I $G(IBMRANOT)'=1,$P($G(^IBM(361.1,IBEOB,0)),U,4)'=1 Q    ; must be an MRA  . Q:$D(^IBM(361.1,IBEOB,"ERR")) ; no filing error  . ; get claim level adjustments  . K EOBADJ M EOBADJ=^IBM(361.1,IBEOB,10)  . S IBPTRES=IBPTRES+$$CALCPR(.EOBADJ)  . ;  . ; get line level adjustments  . S LNLVL=0  . F  S LNLVL=$O(^IBM(361.1,IBEOB,15,LNLVL)) Q:'LNLVL  D  ;  . . K EOBADJ M EOBADJ=^IBM(361.1,IBEOB,15,LNLVL,1)  . . S IBPTRES=IBPTRES+$$CALCPR(.EOBADJ)  Q IBPTRES  ; CALCPR(EOBADJ) ; Function - Calculate Patient Responsibilty Amount  ; For Group Code PR; Ignore the PR-AAA kludge  ; Input - EOBADJ = Array of Group Codes & Reason Codes from either the Claim  ; Level (10) or Service Line Level (15) of EOB file (#361.1)  ; Output - Function returns Patient Responsibility Amount  ;  N GRPLVL,RSNCD,RSNAMT,PTRESP  S (GRPLVL,PTRESP)=0  F  S GRPLVL=$O(EOBADJ(GRPLVL)) Q:'GRPLVL  D  . I $P($G(EOBADJ(GRPLVL,0)),U)'="PR" Q  ;grp code must be PR  . S RSNCD=0  . F  S RSNCD=$O(EOBADJ(GRPLVL,1,RSNCD)) Q:'RSNCD  D  . . I $P($G(EOBADJ(GRPLVL,1,RSNCD,0)),U,1)="AAA" Q   ; ignore PR-AAA  . . S RSNAMT=$P($G(EOBADJ(GRPLVL,1,RSNCD,0)),U,2)  . . S PTRESP=PTRESP+RSNAMT  Q PTRESP  ; COBMOD(IBXSAVE,IBXDATA,SEQ) ; output the modifiers from the COB  ; SEQ is which modifier we're extracting (1-4)  ; Build IBXDATA(line#)=Modifier# SEQ  NEW LN,N,Z,MOD,LNSEQ  KILL IBXDATA  I '$G(SEQ) Q  S (LN,LNSEQ)=0  F  S LN=$O(IBXSAVE("LCOB",LN)) Q:'LN  D  . S LNSEQ=LNSEQ+1  . S (N,Z)=0  . F  S Z=$O(IBXSAVE("LCOB",LN,"COBMOD",Z)) Q:'Z  D  .. S N=N+1  .. S MOD(LNSEQ,N)=$P($G(IBXSAVE("LCOB",LN,"COBMOD",Z,0)),U,1)  .. Q  . S MOD=$G(MOD(LNSEQ,SEQ))  . I MOD'="" S IBXDATA(LNSEQ)=MOD  . Q  Q  ; | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCEU3 | | | | | | | | |
| **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** | | | | | | | | | |
| IBCEU3 ;ALB/TMP - EDI UTILITIES FOR 1500 CLAIM FORM ;12/29/05 9:58am  ;;2.0;INTEGRATED BILLING;\*\*51,137,155,323,348,371,400,432,488,519\*\*;21-MAR-94;Build 56  ;;Per VA Directive 6402, this routine should not be modified.  ; BOX19(IBIFN) ; New Box 19 added for patch 488. This is for workman's comp?  ; This returns the Paperwork Attachment   ; Information in the following format:  ; PWKNNFX12348907CHEY<3 Spaces>Next set if more than one on claim  ; PWK is the qualifier for data, followed by the appropriate Report Type   ;Code, the appropriate Transmission Type Code, then the Attachment Control   ;Number. Do not enter spaces between qualifiers and data.  ;  ; This information can be at either the Line Level or the Claim Level.  ; Check all Lines first and print as many as possible - 71 characters   ; maximum. Then check the Claim Level  N IBRTP,LN,U8,IBBX19,IB19,DATA,I,DEL  S IB19="",DEL=" ",LN=0  ; Get rate type  S IBRTP=$P($G(^DGCR(399,IBIFN,0)),U,7)  ; Get data entered for box 19  S IBBX19=$P($G(^DGCR(399,IBIFN,"UF31")),U,3)  ; check the line Level first  I IBRTP=11 D  .F  S LN=$O(^DGCR(399,IBIFN,"CP",LN)) Q:LN=""  Q:LN'?.N  D  ..S DATA=$G(^DGCR(399,IBIFN,"CP",LN,1))  ..I $P(DATA,U,2)'="" S IB19=IB19\_$S(IB19="":"",1:DEL)\_$$FORMAT(DATA)  .; check the Claim Level next  .S DATA=""  .S DATA=$G(^DGCR(399,IBIFN,"U8"))  .I DATA'="" S IB19=IB19\_$S(IB19="":"",1:DEL)\_$$FORMAT(DATA)  ; If any room left add user entered box 19 info  I IBBX19'="",IB19'="",$L(IB19)<84 D  .F I=1:1:$L(IBBX19,DEL) S DATA=$P(IBBX19,DEL,I) I DATA'="" D  ..I $L(IB19\_DEL\_DATA)<84 S IB19=IB19\_$S(IB19="":"",1:DEL)\_DATA  I IB19="",IBBX19'="" S IB19=IBBX19  ;  Q IB19  ; FORMAT(DATA) ; format data for ouput  N ART,OUT  S ART=$P(DATA,U,2)  S ART=$P(^IBE(353.3,ART,0),U,1)  S OUT="PWK"\_ART\_$P(DATA,U,3)\_$P(DATA,U,1)  Q OUT  ;  ; BELOW NO LONGER USED -> BAA \*488\* OBOX19(IBIFN) ; THIS IS NOLONGER USED. IT WAS REPLACE WITH ABOVE.  ; Returns the text that should print in box 19 of the CMS-1500  ; for bill ien IBIFN  ; Data is derived from a combo of data throughout  ; the system and is limited to 80 characters. The hierarchy for  ; including data is as follows (until 80 characters have been used):  ; DATE LAST SEEN and REFERRING PHYSICIAN ID# (physical therapy)  ; specialty codes = 025,065,073,067,048  ; LAST X-RAY DATE (chiropractic) specialty code = 35  ; HOMEBOUND INDICATOR (independent lab renders an EKG or obtains  ; a specimen from a homebound patient)  ; NO ASSIGNMENT OF BENEFITS (if no assignment of benefits indicated)  ; Hearing aid testing (if applicable)  ; ATTENDING PHYSICIAN NOT HOSPICE EMPLOYEE (if applicable)  ; SPECIAL PROGRAM indicator if Medicare demonstration project for  ; lung volume reduction surgery study is set  ; COMMENTS FOUND IN BOX 19 DATA FIELD FOR THE CLAIM  ; REMARKS FOUND IN BILL COMMENT FOR THE CLAIM, INCLUDING PROSTHETICS  ; DETAIL  ;  N IBGO,IBHOSP,IBID,IBLSDT,IBXDATA,IB19,IBHAID,IBXRAY,IBSPEC,Z,Z0,IBSUB,IBPRT,IBREM,IBSPI  S IB19="",IBGO=1  S IBSUB=$S('$G(^TMP("IBTX",$J,IBIFN)):"BOX24",1:"OUTPT")  I $D(IBXSAVE(IBSUB)) N IBXSAVE  S IBPRT=(IBSUB["24")  ;  S IBSPEC=$$BILLSPEC(IBIFN)  G:'IBPRT NPRT  ; Check for chiropractic services  I $P($G(^DGCR(399,IBIFN,"U3")),U,5)'="" S:$P($G(^DGCR(399,IBIFN,"U3")),U,4)'="" IBGO=$$LENOK("Last X-ray: "\_$TR($$DATE^IBCF2($P(^DGCR(399,IBIFN,"U3"),U,4))," ","/"),.IB19)  G:'IBGO BOX19Q  ;  I "^25^65^73^67^48^"[(U\_IBSPEC\_U) D  . K IBXDATA D F^IBCEF("N-DATE LAST SEEN",,,IBIFN)  . I IBXDATA'="" S IBID="",IBLSDT=$$DATE^IBCF2(IBXDATA,0,1) D  I IBLSDT'="" S IBGO=$$LENOK("Date Last Seen:"\_IBLSDT\_IBID,.IB19)  .. ; Only print if specialty is OT or PT or proc for routine foot care  .. D F^IBCEF("N-REFERRING PROVIDER ID",,,IBIFN) I IBXDATA'="" S IBID=" By:"\_IBXDATA  ;  G:'IBGO BOX19Q  K IBXDATA D F^IBCEF("N-HOMEBOUND",,,IBIFN)  I IBXDATA G:'$$LENOK("Homebound",.IB19) BOX19Q  ;  K IBXDATA D F^IBCEF("N-ASSIGN OF BENEFITS INDICATOR",,,IBIFN)  I "Nn0"[IBXDATA&(IBXDATA'="") G:'$$LENOK("Patient refuses to assign benefits",.IB19) BOX19Q  ;  I '$D(IBXSAVE(IBSUB)) D B24^IBCEF3(.IBXSAVE,IBIFN,$S($G(IBNOSHOW)=0:0,1:1))  ;  S (IBHAID,IBHOSP,IBXRAY)=0  ;  S Z=0 F  S Z=$O(IBXSAVE(IBSUB,Z)) Q:'Z  D  G:'IBGO BOX19Q  . I $D(IBXSAVE(IBSUB,Z,"RX")),$P(IBXSAVE(IBSUB,Z,"RX"),U,3)="" S IBGO=$$LENOK("NOC Drug:"\_$P(IBXSAVE(IBSUB,Z,"RX"),U,2)\_" Units:"\_+$P(IBXSAVE(IBSUB,Z,"RX"),U,6),.IB19)  . ;  . Q:'IBGO  . I 'IBHAID,$P(IBXSAVE(IBSUB,Z),U,5)="V5010",$$COBCT^IBCEF(IBIFN)>1 D  Q  .. S IBHAID=1,IBGO=$$LENOK("Testing for hearing aid",.IB19) Q  . ;  . Q:'IBGO  . I 'IBHOSP,$P($G(IBXSAVE(IBSUB,Z,"AUX")),U,3) S IBHOSP=1,IBGO=$$LENOK("Attending physician,not hospice employee",.IB19) Q  G:'IBGO BOX19Q  K IBXDATA D F^IBCEF("N-SPECIAL PROGRAM",,,IBIFN)  I IBXDATA=30 G:'$$LENOK("Medicare demonstration project for lung volume reduction surgery study",.IB19) BOX19Q  ;  ; SPECIAL PROGRAM INDICATOR field code.  S IBSPI=$$GET1^DIQ(399,IBIFN\_",",238,"E")  I IBSPI'="" S IBGO=$$LENOK(IBSPI,.IB19)  ;  G:'IBGO BOX19Q NPRT K IBXDATA D F^IBCEF("N-HCFA 1500 BOX 19 RAW DATA",,,IBIFN)  S IBREM=0  I IBXDATA'="" G:'$$LENOK("Remarks:"\_IBXDATA,.IB19) BOX19Q S IBREM=1  K IBXDATA D F^IBCEF("N-BILL REMARKS",,,IBIFN)  I IBXDATA'="" G:'$$LENOK($S('IBREM:"Remarks:",1:"")\_IBXDATA,.IB19) BOX19Q  ; BOX19Q Q IB19  ; ALL OF THE ABOVE TO OBOX19 IS NO LONGER USED \*488\*  ; LENOK(IBDATA,IB19) ; Add text IBDATA to box 19 string (IB19 passed by ref)  ; Check length of box 19 data - truncate at 71 (max length)  ; Returns 0 if max length reached or exceeded, otherwise, 1  ; Changed 96 to 71 for new 1500 form  N OK  S OK=1  S IB19=IB19\_$S(IB19'="":" ",1:"")\_$G(IBDATA)  I $L(IB19)'<83 S OK=0,IB19=$E(IB19,1,71) G LENOKQ LENOKQ Q OK  ; ASK19(IBIFN) ; Ask to display CMS-1500 box 19 data for current IBIFN  ; changed to 71 length.  N DIR,DIC,X,Y,DIE,DR,Z  S DIR(0)="YA",DIR("B")="NO",DIR("A")="DISPLAY THE FULL CMS-1500 BOX 19?: "  D ^DIR  K DIR("B")  I Y=1 D  .S Z=$$BOX19(IBIFN) W !!,?4,"19",?45,$E(Z,1,23) W:$L(Z)>23 !,?4,$E(Z,24,71),!  .S DIR(0)="E",DIR("A")="Enter <RET> to Continue " W ! D ^DIR K DIR  Q  ; ONLAB(IBIFN) ; Functions returns 1 if the bill IBIFN is outside non-lab  N IBP,IBPUR  S IBP=0  S IBPUR=$P($G(^DGCR(399,IBIFN,"U2")),U,11)  I IBPUR,"13"[IBPUR S IBP=1  Q IBP  ; TEXT24(FLD,IBXSAVE,IBXDATA,IBSUB) ; Format the text line of box 24 by fld  ; INPUT:  ; FLD = the letter of the field in box 24 (A-J)  ; IBXSAVE = passed by reference = extracted data for the box 24 lines  ; IBSUB = the subscript of the IBXSAVE array to use.  ; If null, use "BOX24"  ; OUTPUT:  ; IBXDATA = passed by reference, set to the correct part of the  ; text that will print in the field's positions  ;  ; esg - 8/14/06 - modified for the new cms-1500 form - IB\*2\*348  ;  N Z,IBLINE,IBVAL,IBS,IBE,IBTEXT,IBAUX,IBDAT,IBZ,IBREN,IBRENQ,IBRENNPI,IBRENSID  K IBXDATA  S (IBLINE,Z)=0 S:$G(IBSUB)="" IBSUB="BOX24"  ;  I FLD="I"!(FLD="J") D   ; extract the Rendering provider data  . I '$G(IBXIEN) Q       ; assume that the claim# exists  . S IBREN=$$CFIDS^IBCEF77(IBXIEN)  . S IBRENQ=$P(IBREN,U,1) ; qual  . S IBRENSID=$P(IBREN,U,2) ; id  . S IBRENNPI=$P(IBREN,U,3) ; npi  . Q  ;  F  S Z=$O(IBXSAVE(IBSUB,Z)) Q:'Z  D  . S IBDAT=$G(IBXSAVE(IBSUB,Z))  . S IBAUX=$G(IBXSAVE(IBSUB,Z,"AUX"))  . S IBTEXT=$G(IBXSAVE(IBSUB,Z,"TEXT"))  . S IBZ=$P(IBAUX,U,9)  . I IBZ="" S IBZ=" "  . S IBTEXT=IBZ\_IBTEXT  . ;  . I $S($G(IBAC)=4:$S($D(IBXSAVE(IBSUB,Z,"ARX")):1,1:$D(IBXSAVE(IBSUB,Z,"A"))),$D(IBXSAVE(IBSUB,Z,"RX")):0,1:$G(IBNOSHOW)) S IBTEXT=""  . ;  . I FLD="AF" S IBVAL=$P(IBDAT,U),IBS=1,IBE=9 D   ; From date of service  .. S IBVAL=$E(IBVAL,1,2)\_" "\_$E(IBVAL,3,4)\_" "\_$E(IBVAL,7,8)  .. Q  . ;  . I FLD="AT" S IBVAL=$S($P(IBDAT,U,2):$P(IBDAT,U,2),1:$P(IBDAT,U)),IBS=10,IBE=18 D    ; To date of service  .. S IBVAL=$E(IBVAL,1,2)\_" "\_$E(IBVAL,3,4)\_" "\_$E(IBVAL,7,8)  .. Q  . ;  . I FLD="B" S IBVAL=$P(IBDAT,U,3),IBS=19,IBE=21 ; place of service  . I FLD="C" S IBVAL=$S($P(IBDAT,U,13)=1:"Y",1:""),IBS=22,IBE=24 ; emergency indicator  . I FLD="D" S IBVAL=$P(IBDAT,U,5),IBS=25,IBE=44 D   ; procedures and modifiers  .. N M S M=$$MODLST^IBEFUNC($P(IBDAT,U,10)) ; modifier list  .. S IBVAL=$$FO^IBCNEUT1(IBVAL,6)\_" "            ; procedure code  .. S IBVAL=IBVAL\_$$FO^IBCNEUT1($P(M,",",1),3) ; mod#1  .. S IBVAL=IBVAL\_$$FO^IBCNEUT1($P(M,",",2),3) ; mod#2  .. S IBVAL=IBVAL\_$$FO^IBCNEUT1($P(M,",",3),3) ; mod#3  .. S IBVAL=IBVAL\_$$FO^IBCNEUT1($P(M,",",4),3) ; mod#4  .. Q  . ;  . I FLD="E" D  .. N NUM,IN,OUT,LET  .. S IN="1,2,3,4,5,6,7,8,9"  .. S OUT="A,B,C,D,E,F,G,H,I"  .. S IBVAL=$P(IBDAT,U,7)  .. F I=1:1:4 S NUM=$P(IBVAL,",",I) D  ... I NUM<10 S $P(LET,",",I)=$TR(NUM,IN,OUT)  ... I NUM=10 S $P(LET,",",I)="J"  ... I NUM=11 S $P(LET,",",I)="K"  ... I NUM=12 S $P(LET,",",I)="L"  .. S IBVAL=$TR(LET,","),IBS=45,IBE=48 ; diagnosis pointer  . I FLD="F" S IBVAL=$P(IBDAT,U,8)\*$P(IBDAT,U,9),IBS=49,IBE=57 D  .. ; total charges \*\*519 returned field length back to 8, 9 is too long for BOX24F  .. S IBVAL=$$DOL^IBCEF77(IBVAL,8)  .. I $L(IBVAL)>8 S IBVAL=$E(IBVAL,$L(IBVAL)-7,$L(IBVAL))  .. Q  . ;  . I FLD="G" S IBVAL=$S($P(IBDAT,U,12):$P(IBDAT,U,12),1:$P(IBDAT,U,9)),IBS=58,IBE=61 D  .. ; days or units or anesthesia minutes  .. S IBVAL=$J(+IBVAL,4)  .. Q  . ;  . ; columns H,I,J don't have any free text supplemental information  . ;  . I FLD="H" D     ; epsdt family plan  .. S IBVAL=$P(IBAUX,U,7),IBS=0,IBE=0,IBTEXT=""   ; line 1 blank  .. I IBVAL S IBVAL="Y"  .. Q  . I FLD="I" D     ; ID qualifier for rendering provider  .. S IBVAL="",IBS=1,IBE=2 ; line 2 blank  .. S IBTEXT=$G(IBRENQ) ; qualifier on line 1  .. Q  . I FLD="J" D     ; rendering provider ID and NPI  .. S IBTEXT=$G(IBRENSID),IBS=1,IBE=11 ; secondary ID line 1  .. S IBVAL=$G(IBRENNPI) ; NPI# line 2  .. Q  . ;  . S IBLINE=IBLINE+1 ; top line  . S IBXDATA(IBLINE)=$E(IBTEXT,IBS,IBE) ; text in shaded area (top)  . S IBLINE=IBLINE+1 ; bottom line  . S IBXDATA(IBLINE)=IBVAL       ; field value in unshaded area (bottom)  . Q  ;  Q  ; LINSPEC(IBIFN) ; Checks the specialities of line and claim level providers  ; called from IBCBB2 to check for Chiro codes & IBCBB9 to check for 99's on Medicare  ; Default = 99 if no valid SPEC code found for line and claim level provider  ; Get rendering for professional, attending for institutional  ; If multiple lines w/ rendering or attending, returns a string of spec codes  N Z,IBSPEC,IBINS,IBDT,IBCP,IBSPC  S IBSPC=""  S IBDT=$P($G(^DGCR(399,+IBIFN,"U")),U,1) ; use statement from date  S IBINS=($$FT^IBCEF(IBIFN)=3)  D GETPRV^IBCEU(IBIFN,"ALL",.IBPRV)  S Z=$S('IBINS:3,1:4)  ; check claim level  I $G(IBPRV(Z,1))'="" D  . I $P(IBPRV(Z,1),U,3) S IBSPEC=$$SPEC^IBCEU($P($G(IBPRV(Z,1)),U,3),IBDT) I IBSPEC'="" S IBSPC=IBSPC\_U\_IBSPEC Q  . S Z0=+$O(^DGCR(399,IBIFN,"PRV","B",Z,0))  . I Z0 S IBSPEC=$P($G(^DGCR(399,IBIFN,"PRV",Z0,0)),U,8) S:IBSPEC="" IBSPEC=99 S IBSPC=IBSPC\_U\_IBSPEC  ; Check line level  S IBCP=0 F  S IBCP=$O(^DGCR(399,IBIFN,"CP",IBCP)) Q:'IBCP  D  .S Z0=+$O(^DGCR(399,IBIFN,"CP",IBCP,"LNPRV","B",Z,0))  .I Z0 S IBSPEC=$P($G(^DGCR(399,IBIFN,"CP",IBCP,"LNPRV",Z0,0)),U,8) S:IBSPEC="" IBSPEC="99" S IBSPC=IBSPC\_U\_IBSPEC  S:IBSPC="" IBSPC=99  Q IBSPC  ; BILLSPEC(IBIFN,IBPRV) ; Returns the specialty of the provider on bill IBIFN  ; If IBPRV is supplied, returns the data for that provider, otherwise,  ; returns the specialty of the 'main/required' provider on the bill.  ; Default = 99 if no valid code found  ; IBPRV = vp of provider (file 200 or 355.93)  N Z,IBSPEC,IBINS,IBDT  S IBSPEC="",IBPRV=$G(IBPRV)  S IBDT=$P($G(^DGCR(399,+IBIFN,"U")),U,1) ; use statement from date  ;  I $G(IBPRV) D  G SPECQ  . S IBSPEC=$$SPEC^IBCEU(IBPRV,IBDT)  ;  ;Get rendering for professional, attending for institutional,  S IBINS=($$FT^IBCEF(IBIFN)=3)  D GETPRV^IBCEU(IBIFN,"ALL",.IBPRV)  S Z=$S('IBINS:3,1:4)  I $G(IBPRV(Z,1))'="" D  . I $P(IBPRV(Z,1),U,3) S IBSPEC=$$SPEC^IBCEU($P($G(IBPRV(Z,1)),U,3),IBDT) Q:IBSPEC'=""  . S Z0=+$O(^DGCR(399,IBIFN,"PRV","B",Z,0))  . I Z0,$P($G(^DGCR(399,IBIFN,"PRV",Z0,0)),U,8)'="" S IBSPEC=$P(^(0),U,8)  ; SPECQ I IBSPEC="" S IBSPEC="99"  Q IBSPEC  ; CHAMPVA(IBIFN) ; Returns 1 if the bill IBIFN has a CHAMPVA rate type  Q $E($P($G(^DGCR(399.3,+$P($G(^DGCR(399,IBIFN,0)),U,7),0)),U),1,7)="CHAMPVA"  ; FAC(IBIFN) ; Obsolete function. Used by old output formatter field and data element N-RENDERING INSTITUTION  Q ""  ; MCR24K(IBIFN,IBPRV) ;Function returns MEDICARE id# for professional (CMS-1500) box 24k for bill IBIFN if appropriate  ;\*432/TAZ - Added IBPRV to allow circumvent the call to F^IBCEF("N-SPECIALTY CODE","IBZ",,IBIFN) in MCRSPEC^IBCEU4  Q $S($$FT^IBCEF(IBIFN)=2&$$MCRONBIL^IBEFUNC(IBIFN):"V"\_$$MCRSPEC^IBCEU4(IBIFN,1,$G(IBPRV))\_$P($$SITE^VASITE,U,3),1:"") | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCEU3 ;ALB/TMP - EDI UTILITIES FOR 1500 CLAIM FORM ;12/29/05 9:58am  ;;2.0;INTEGRATED BILLING;\*\*51,137,155,323,348,371,400,432,488,519,**592**\*\*;21-MAR-94;Build 56  ;;Per VA Directive 6402, this routine should not be modified.  ; BOX19(IBIFN) ; New Box 19 added for patch 488. This is for workman's comp?  ; This returns the Paperwork Attachment   ; Information in the following format:  ; PWKNNFX12348907CHEY<3 Spaces>Next set if more than one on claim  ; PWK is the qualifier for data, followed by the appropriate Report Type   ;Code, the appropriate Transmission Type Code, then the Attachment Control   ;Number. Do not enter spaces between qualifiers and data.  ;  ; This information can be at either the Line Level or the Claim Level.  ; Check all Lines first and print as many as possible - 71 characters   ; maximum. Then check the Claim Level  N IBRTP,LN,U8,IBBX19,IB19,DATA,I,DEL  S IB19="",DEL=" ",LN=0  ; Get rate type  S IBRTP=$P($G(^DGCR(399,IBIFN,0)),U,7)  ; Get data entered for box 19  S IBBX19=$P($G(^DGCR(399,IBIFN,"UF31")),U,3)  ; check the line Level first  I IBRTP=11 D  .F  S LN=$O(^DGCR(399,IBIFN,"CP",LN)) Q:LN=""  Q:LN'?.N  D  ..S DATA=$G(^DGCR(399,IBIFN,"CP",LN,1))  ..I $P(DATA,U,2)'="" S IB19=IB19\_$S(IB19="":"",1:DEL)\_$$FORMAT(DATA)  .; check the Claim Level next  .S DATA=""  .S DATA=$G(^DGCR(399,IBIFN,"U8"))  .I DATA'="" S IB19=IB19\_$S(IB19="":"",1:DEL)\_$$FORMAT(DATA)  ; If any room left add user entered box 19 info  I IBBX19'="",IB19'="",$L(IB19)<84 D  .F I=1:1:$L(IBBX19,DEL) S DATA=$P(IBBX19,DEL,I) I DATA'="" D  ..I $L(IB19\_DEL\_DATA)<84 S IB19=IB19\_$S(IB19="":"",1:DEL)\_DATA  I IB19="",IBBX19'="" S IB19=IBBX19  ;  Q IB19  ; FORMAT(DATA) ; format data for ouput  N ART,OUT  S ART=$P(DATA,U,2)  S ART=$P(^IBE(353.3,ART,0),U,1)  S OUT="PWK"\_ART\_$P(DATA,U,3)\_$P(DATA,U,1)  Q OUT  ;  ; BELOW NO LONGER USED -> BAA \*488\* OBOX19(IBIFN) ; THIS IS NOLONGER USED. IT WAS REPLACE WITH ABOVE.  ; Returns the text that should print in box 19 of the CMS-1500  ; for bill ien IBIFN  ; Data is derived from a combo of data throughout  ; the system and is limited to 80 characters. The hierarchy for  ; including data is as follows (until 80 characters have been used):  ; DATE LAST SEEN and REFERRING PHYSICIAN ID# (physical therapy)  ; specialty codes = 025,065,073,067,048  ; LAST X-RAY DATE (chiropractic) specialty code = 35  ; HOMEBOUND INDICATOR (independent lab renders an EKG or obtains  ; a specimen from a homebound patient)  ; NO ASSIGNMENT OF BENEFITS (if no assignment of benefits indicated)  ; Hearing aid testing (if applicable)  ; ATTENDING PHYSICIAN NOT HOSPICE EMPLOYEE (if applicable)  ; SPECIAL PROGRAM indicator if Medicare demonstration project for  ; lung volume reduction surgery study is set  ; COMMENTS FOUND IN BOX 19 DATA FIELD FOR THE CLAIM  ; REMARKS FOUND IN BILL COMMENT FOR THE CLAIM, INCLUDING PROSTHETICS  ; DETAIL  ;  N IBGO,IBHOSP,IBID,IBLSDT,IBXDATA,IB19,IBHAID,IBXRAY,IBSPEC,Z,Z0,IBSUB,IBPRT,IBREM,IBSPI  S IB19="",IBGO=1  S IBSUB=$S('$G(^TMP("IBTX",$J,IBIFN)):"BOX24",1:"OUTPT")  I $D(IBXSAVE(IBSUB)) N IBXSAVE  S IBPRT=(IBSUB["24")  ;  S IBSPEC=$$BILLSPEC(IBIFN)  G:'IBPRT NPRT  ; Check for chiropractic services  I $P($G(^DGCR(399,IBIFN,"U3")),U,5)'="" S:$P($G(^DGCR(399,IBIFN,"U3")),U,4)'="" IBGO=$$LENOK("Last X-ray: "\_$TR($$DATE^IBCF2($P(^DGCR(399,IBIFN,"U3"),U,4))," ","/"),.IB19)  G:'IBGO BOX19Q  ;  I "^25^65^73^67^48^"[(U\_IBSPEC\_U) D  . K IBXDATA D F^IBCEF("N-DATE LAST SEEN",,,IBIFN)  . I IBXDATA'="" S IBID="",IBLSDT=$$DATE^IBCF2(IBXDATA,0,1) D  I IBLSDT'="" S IBGO=$$LENOK("Date Last Seen:"\_IBLSDT\_IBID,.IB19)  .. ; Only print if specialty is OT or PT or proc for routine foot care  .. D F^IBCEF("N-REFERRING PROVIDER ID",,,IBIFN) I IBXDATA'="" S IBID=" By:"\_IBXDATA  ;  G:'IBGO BOX19Q  K IBXDATA D F^IBCEF("N-HOMEBOUND",,,IBIFN)  I IBXDATA G:'$$LENOK("Homebound",.IB19) BOX19Q  ;  K IBXDATA D F^IBCEF("N-ASSIGN OF BENEFITS INDICATOR",,,IBIFN)  I "Nn0"[IBXDATA&(IBXDATA'="") G:'$$LENOK("Patient refuses to assign benefits",.IB19) BOX19Q  ;  I '$D(IBXSAVE(IBSUB)) D B24^IBCEF3(.IBXSAVE,IBIFN,$S($G(IBNOSHOW)=0:0,1:1))  ;  S (IBHAID,IBHOSP,IBXRAY)=0  ;  S Z=0 F  S Z=$O(IBXSAVE(IBSUB,Z)) Q:'Z  D  G:'IBGO BOX19Q  . I $D(IBXSAVE(IBSUB,Z,"RX")),$P(IBXSAVE(IBSUB,Z,"RX"),U,3)="" S IBGO=$$LENOK("NOC Drug:"\_$P(IBXSAVE(IBSUB,Z,"RX"),U,2)\_" Units:"\_+$P(IBXSAVE(IBSUB,Z,"RX"),U,6),.IB19)  . ;  . Q:'IBGO  . I 'IBHAID,$P(IBXSAVE(IBSUB,Z),U,5)="V5010",$$COBCT^IBCEF(IBIFN)>1 D  Q  .. S IBHAID=1,IBGO=$$LENOK("Testing for hearing aid",.IB19) Q  . ;  . Q:'IBGO  . I 'IBHOSP,$P($G(IBXSAVE(IBSUB,Z,"AUX")),U,3) S IBHOSP=1,IBGO=$$LENOK("Attending physician,not hospice employee",.IB19) Q  G:'IBGO BOX19Q  K IBXDATA D F^IBCEF("N-SPECIAL PROGRAM",,,IBIFN)  I IBXDATA=30 G:'$$LENOK("Medicare demonstration project for lung volume reduction surgery study",.IB19) BOX19Q  ;  ; SPECIAL PROGRAM INDICATOR field code.  S IBSPI=$$GET1^DIQ(399,IBIFN\_",",238,"E")  I IBSPI'="" S IBGO=$$LENOK(IBSPI,.IB19)  ;  G:'IBGO BOX19Q NPRT K IBXDATA D F^IBCEF("N-HCFA 1500 BOX 19 RAW DATA",,,IBIFN)  S IBREM=0  I IBXDATA'="" G:'$$LENOK("Remarks:"\_IBXDATA,.IB19) BOX19Q S IBREM=1  K IBXDATA D F^IBCEF("N-BILL REMARKS",,,IBIFN)  I IBXDATA'="" G:'$$LENOK($S('IBREM:"Remarks:",1:"")\_IBXDATA,.IB19) BOX19Q  ; BOX19Q Q IB19  ; ALL OF THE ABOVE TO OBOX19 IS NO LONGER USED \*488\*  ; LENOK(IBDATA,IB19) ; Add text IBDATA to box 19 string (IB19 passed by ref)  ; Check length of box 19 data - truncate at 71 (max length)  ; Returns 0 if max length reached or exceeded, otherwise, 1  ; Changed 96 to 71 for new 1500 form  N OK  S OK=1  S IB19=IB19\_$S(IB19'="":" ",1:"")\_$G(IBDATA)  I $L(IB19)'<83 S OK=0,IB19=$E(IB19,1,71) G LENOKQ LENOKQ Q OK  ; ASK19(IBIFN) ; Ask to display CMS-1500 box 19 data for current IBIFN  ; changed to 71 length.  N DIR,DIC,X,Y,DIE,DR,Z  S DIR(0)="YA",DIR("B")="NO",DIR("A")="DISPLAY THE FULL CMS-1500 BOX 19?: "  D ^DIR  K DIR("B")  I Y=1 D  .S Z=$$BOX19(IBIFN) W !!,?4,"19",?45,$E(Z,1,23) W:$L(Z)>23 !,?4,$E(Z,24,71),!  .S DIR(0)="E",DIR("A")="Enter <RET> to Continue " W ! D ^DIR K DIR  Q  ; ONLAB(IBIFN) ; Functions returns 1 if the bill IBIFN is outside non-lab  N IBP,IBPUR  S IBP=0  S IBPUR=$P($G(^DGCR(399,IBIFN,"U2")),U,11)  I IBPUR,"13"[IBPUR S IBP=1  Q IBP  ; TEXT24(FLD,IBXSAVE,IBXDATA,IBSUB) ; Format the text line of box 24 by fld  ; INPUT:  ; FLD = the letter of the field in box 24 (A-J)  ; IBXSAVE = passed by reference = extracted data for the box 24 lines  ; IBSUB = the subscript of the IBXSAVE array to use.  ; If null, use "BOX24"  ; OUTPUT:  ; IBXDATA = passed by reference, set to the correct part of the  ; text that will print in the field's positions  ;  ; esg - 8/14/06 - modified for the new cms-1500 form - IB\*2\*348  ;  N Z,IBLINE,IBVAL,IBS,IBE,IBTEXT,IBAUX,IBDAT,IBZ,IBREN,IBRENQ,IBRENNPI,IBRENSID  K IBXDATA  S (IBLINE,Z)=0 S:$G(IBSUB)="" IBSUB="BOX24"  ;  I FLD="I"!(FLD="J") D   ; extract the Rendering provider data  . I '$G(IBXIEN) Q       ; assume that the claim# exists  . S IBREN=$$CFIDS^IBCEF77(IBXIEN)  . S IBRENQ=$P(IBREN,U,1) ; qual  . S IBRENSID=$P(IBREN,U,2) ; id  . S IBRENNPI=$P(IBREN,U,3) ; npi  . Q  ;  F  S Z=$O(IBXSAVE(IBSUB,Z)) Q:'Z  D  . S IBDAT=$G(IBXSAVE(IBSUB,Z))  . S IBAUX=$G(IBXSAVE(IBSUB,Z,"AUX"))  . S IBTEXT=$G(IBXSAVE(IBSUB,Z,"TEXT"))  . S IBZ=$P(IBAUX,U,9)  . I IBZ="" S IBZ=" "  . S IBTEXT=IBZ\_IBTEXT  . ;  . I $S($G(IBAC)=4:$S($D(IBXSAVE(IBSUB,Z,"ARX")):1,1:$D(IBXSAVE(IBSUB,Z,"A"))),$D(IBXSAVE(IBSUB,Z,"RX")):0,1:$G(IBNOSHOW)) S IBTEXT=""  . ;  . I FLD="AF" S IBVAL=$P(IBDAT,U),IBS=1,IBE=9 D   ; From date of service  .. S IBVAL=$E(IBVAL,1,2)\_" "\_$E(IBVAL,3,4)\_" "\_$E(IBVAL,7,8)  .. Q  . ;  . I FLD="AT" S IBVAL=$S($P(IBDAT,U,2):$P(IBDAT,U,2),1:$P(IBDAT,U)),IBS=10,IBE=18 D    ; To date of service  .. S IBVAL=$E(IBVAL,1,2)\_" "\_$E(IBVAL,3,4)\_" "\_$E(IBVAL,7,8)  .. Q  . ;  . I FLD="B" S IBVAL=$P(IBDAT,U,3),IBS=19,IBE=21 ; place of service  . I FLD="C" S IBVAL=$S($P(IBDAT,U,13)=1:"Y",1:""),IBS=22,IBE=24 ; emergency indicator  . I FLD="D" S IBVAL=$P(IBDAT,U,5),IBS=25,IBE=44 D   ; procedures and modifiers  .. N M S M=$$MODLST^IBEFUNC($P(IBDAT,U,10)) ; modifier list  .. S IBVAL=$$FO^IBCNEUT1(IBVAL,6)\_" "            ; procedure code  .. S IBVAL=IBVAL\_$$FO^IBCNEUT1($P(M,",",1),3) ; mod#1  .. S IBVAL=IBVAL\_$$FO^IBCNEUT1($P(M,",",2),3) ; mod#2  .. S IBVAL=IBVAL\_$$FO^IBCNEUT1($P(M,",",3),3) ; mod#3  .. S IBVAL=IBVAL\_$$FO^IBCNEUT1($P(M,",",4),3) ; mod#4  .. Q  . ;  . I FLD="E" D  .. N NUM,IN,OUT,LET  .. S IN="1,2,3,4,5,6,7,8,9"  .. S OUT="A,B,C,D,E,F,G,H,I"  .. S IBVAL=$P(IBDAT,U,7)  .. F I=1:1:4 S NUM=$P(IBVAL,",",I) D  ... I NUM<10 S $P(LET,",",I)=$TR(NUM,IN,OUT)  ... I NUM=10 S $P(LET,",",I)="J"  ... I NUM=11 S $P(LET,",",I)="K"  ... I NUM=12 S $P(LET,",",I)="L"  .. S IBVAL=$TR(LET,","),IBS=45,IBE=48 ; diagnosis pointer  . I FLD="F" S IBVAL=$P(IBDAT,U,8)\*$P(IBDAT,U,9),IBS=49,IBE=57 D  .. ; total charges \*\*519 returned field length back to 8, 9 is too long for BOX24F  .. S IBVAL=$$DOL^IBCEF77(IBVAL,8)  .. I $L(IBVAL)>8 S IBVAL=$E(IBVAL,$L(IBVAL)-7,$L(IBVAL))  .. Q  . ;  . I FLD="G" S IBVAL=$S($P(IBDAT,U,12):$P(IBDAT,U,12),1:$P(IBDAT,U,9)),IBS=58,IBE=61 D  .. ; days or units or anesthesia minutes  .. S IBVAL=$J(+IBVAL,4)  .. Q  . ;  . ; columns H,I,J don't have any free text supplemental information  . ;  . I FLD="H" D     ; epsdt family plan  .. S IBVAL=$P(IBAUX,U,7),IBS=0,IBE=0,IBTEXT=""   ; line 1 blank  .. I IBVAL S IBVAL="Y"  .. Q  . I FLD="I" D     ; ID qualifier for rendering provider  .. S IBVAL="",IBS=1,IBE=2 ; line 2 blank  .. S IBTEXT=$G(IBRENQ) ; qualifier on line 1  .. Q  . I FLD="J" D     ; rendering provider ID and NPI  .. S IBTEXT=$G(IBRENSID),IBS=1,IBE=11 ; secondary ID line 1  .. S IBVAL=$G(IBRENNPI) ; NPI# line 2  .. Q  . ;  . S IBLINE=IBLINE+1 ; top line  . S IBXDATA(IBLINE)=$E(IBTEXT,IBS,IBE) ; text in shaded area (top)  . S IBLINE=IBLINE+1 ; bottom line  . S IBXDATA(IBLINE)=IBVAL       ; field value in unshaded area (bottom)  . Q  ;  Q  ; LINSPEC(IBIFN) ; Checks the specialities of line and claim level providers  ; called from IBCBB2 to check for Chiro codes & IBCBB9 to check for 99's on Medicare  ; Default = 99 if no valid SPEC code found for line and claim level provider  ; Get rendering for professional, attending for institutional  ; If multiple lines w/ rendering or attending, returns a string of spec codes  N Z,IBSPEC,IBINS,IBDT,IBCP,IBSPC  S IBSPC=""  S IBDT=$P($G(^DGCR(399,+IBIFN,"U")),U,1) ; use statement from date  S IBINS=($$FT^IBCEF(IBIFN)=3)  D GETPRV^IBCEU(IBIFN,"ALL",.IBPRV)  S Z=$S('IBINS:3,1:4)  ; check claim level  I $G(IBPRV(Z,1))'="" D  . I $P(IBPRV(Z,1),U,3) S IBSPEC=$$SPEC^IBCEU($P($G(IBPRV(Z,1)),U,3),IBDT) I IBSPEC'="" S IBSPC=IBSPC\_U\_IBSPEC Q  . S Z0=+$O(^DGCR(399,IBIFN,"PRV","B",Z,0))  . I Z0 S IBSPEC=$P($G(^DGCR(399,IBIFN,"PRV",Z0,0)),U,8) S:IBSPEC="" IBSPEC=99 S IBSPC=IBSPC\_U\_IBSPEC  ; Check line level  S IBCP=0 F  S IBCP=$O(^DGCR(399,IBIFN,"CP",IBCP)) Q:'IBCP  D  .S Z0=+$O(^DGCR(399,IBIFN,"CP",IBCP,"LNPRV","B",Z,0))  .I Z0 S IBSPEC=$P($G(^DGCR(399,IBIFN,"CP",IBCP,"LNPRV",Z0,0)),U,8) S:IBSPEC="" IBSPEC="99" S IBSPC=IBSPC\_U\_IBSPEC  S:IBSPC="" IBSPC=99  Q IBSPC  ; BILLSPEC(IBIFN,IBPRV) ; Returns the specialty of the provider on bill IBIFN  ; If IBPRV is supplied, returns the data for that provider, otherwise,  ; returns the specialty of the 'main/required' provider on the bill.  ; Default = 99 if no valid code found  ; IBPRV = vp of provider (file 200 or 355.93)  N Z,IBSPEC,IBINS,IBDT  S IBSPEC="",IBPRV=$G(IBPRV)  S IBDT=$P($G(^DGCR(399,+IBIFN,"U")),U,1) ; use statement from date  ;  I $G(IBPRV) D  G SPECQ  . S IBSPEC=$$SPEC^IBCEU(IBPRV,IBDT)  ;  ;Get rendering for professional, attending for institutional,  S IBINS=($$FT^IBCEF(IBIFN)=3)  D GETPRV^IBCEU(IBIFN,"ALL",.IBPRV)  S Z=$S('IBINS:3,1:4)  I $G(IBPRV(Z,1))'="" D  . I $P(IBPRV(Z,1),U,3) S IBSPEC=$$SPEC^IBCEU($P($G(IBPRV(Z,1)),U,3),IBDT) Q:IBSPEC'=""  . S Z0=+$O(^DGCR(399,IBIFN,"PRV","B",Z,0))  . I Z0,$P($G(^DGCR(399,IBIFN,"PRV",Z0,0)),U,8)'="" S IBSPEC=$P(^(0),U,8)  ; SPECQ I IBSPEC="" S IBSPEC="99"  Q IBSPEC  ; CHAMPVA(IBIFN) ; Returns 1 if the bill IBIFN has a CHAMPVA rate type  Q $E($P($G(^DGCR(399.3,+$P($G(^DGCR(399,IBIFN,0)),U,7),0)),U),1,7)="CHAMPVA"  ; FAC(IBIFN) ; Obsolete function. Used by old output formatter field and data element N-RENDERING INSTITUTION  Q ""  ; MCR24K(IBIFN,IBPRV) ;Function returns MEDICARE id# for professional (CMS-1500) box 24k for bill IBIFN if appropriate  ;\*432/TAZ - Added IBPRV to allow circumvent the call to F^IBCEF("N-SPECIALTY CODE","IBZ",,IBIFN) in MCRSPEC^IBCEU4 **;JWS;IB\*2.0\*592:Added dental form to check for compatibility**  Q $S(($$FT^IBCEF(IBIFN)=2**!$$FT^IBCEF(IBIFN)=7)**&$$MCRONBIL^IBEFUNC(IBIFN):"V"\_$$MCRSPEC^IBCEU4(IBIFN,1,$G(IBPRV))\_$P($$SITE^VASITE,U,3),1:"") | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCEU5 | | | | | | | | |
| **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** | | | | | | | | | |
| IBCEU5 ;ALB/TMP - EDI UTILITIES (continued) FOR CMS-1500 ;13-DEC-99  ;;2.0;INTEGRATED BILLING;\*\*51,137,232,348,349,432\*\*;21-MAR-94;Build 192  ;;Per VHA Directive 2004-038, this routine should not be modified.  Q  ; EXTCR(IBPRV) ; Called by trigger on field .02 of file 399.0222  ; Also called by trigger on field .02 of file 399.0404 (DEM;432).  ; Function returns the first 3 digits of the provider's degree if  ; a VA provider or the credentials in file 355.9 if non-VA provider  ; IBPRV = vp to file 200 or 355.93  Q $E($$CRED^IBCEU(IBPRV),1,3)  ;  FTPRV(IBIFN,NOASK) ; If form type changes from UB-04 to CMS-1500 or vice  ; versa, ask to change provider function to appropriate function for  ; form type (ATTENDING = UB-04, RENDERING = CMS-1500)  ; IBIFN = ien of bill in file 399  ; NOASK (flag) = 1 if change should happen without asking first  N ATT,REN,FT  S FT=$$FT^IBCEF(IBIFN)  S REN=$$CKPROV^IBCEU(IBIFN,3,1)  S ATT=$$CKPROV^IBCEU(IBIFN,4,1)  I $S(FT=2:'REN&ATT,FT=3:'ATT&REN,1:0) D  . I '$G(NOASK) D TXFERPRV(IBIFN,FT) Q  . D PRVCHG(IBIFN,FT)  D CLEANUP(IBIFN,FT)  Q  ; TXFERPRV(IBIFN,FT) ; Ask to change the function of the main provider on  ; bill IBIFN to the function appropriate to the form type FT  ;   N DIR,X,Y,Z,DIE,DA,DR,HAVE,NEED,IBZ  ; DEM;432 - Changed the prompt from uppercase to mixed case.  W ! S DIR("A")=" Change the Claim Level "\_$S(FT=3:"Rendering",1:"Attending")\_" provider's function to "\_$S(FT=3:"Attending",1:"Rendering")\_"?: "  S DIR(0)="YA",DIR("B")="NO",DIR("?",1)="If you answer YES here, you will make the claim level provider functions",DIR("?")=" consistent with the form type of the bill"  D ^DIR K DIR  I Y'=1 Q  D PRVCHG(IBIFN,FT)  Q  ; PRVCHG(IBIFN,IBFT) ; Change provider type to type consistent with current  ; data on bill  N Z,IBZ,HAVE,NEED,DIE,DA,X,Y  S HAVE=$S(IBFT=3:3,1:4)  S NEED=$S(IBFT=3:4,1:3)  S Z=$O(^DGCR(399,IBIFN,"PRV","B",HAVE,0))  I Z D  . S DA(1)=IBIFN,DA=+Z  . D FDA^DILF(399.0222,.DA,.01,,NEED,"IBZ")  . D FILE^DIE(,"IBZ")  ;I Z S DA(1)=IBIFN,DIE="^DGCR(399,"\_DA(1)\_",""PRV"",",DA=+Z,DR=".01////"\_NEED D FILE^DIE(,DIE  Q  ; CLEANUP(IBIFN,FT) ; If form type changes remove any extra provider FUNCTIONS.  N X,PRV,CLEAN,DA,DIE  ;  ; (3) If form type changes from CMS-1500 to UB-04, remove any extra provider FUNCTIONS.   I FT=3 F X=5 D  ; 5-SUPERVISING  .I $D(^DGCR(399,IBIFN,"PRV","B",X)) D  .. S PRV=0 F  S PRV=$O(^DGCR(399,IBIFN,"PRV","B",X,PRV)) Q:+PRV=0 D  ... S DA(1)=IBIFN,DA=PRV D FDA^DILF(399.0222,.DA,.01,,"@","CLEAN")  ;  ; (2) If form type changes from UB-04 to CMS-1500, remove any extra provider FUNCTIONS.   I FT=2 F X=2,4,9 D  ; 2-OPERATING, 4-ATTENDING, 9-OTHER  .I $D(^DGCR(399,IBIFN,"PRV","B",X)) D  .. S PRV=0 F  S PRV=$O(^DGCR(399,IBIFN,"PRV","B",X,PRV)) Q:+PRV=0 D  ... S DA(1)=IBIFN,DA=PRV D FDA^DILF(399.0222,.DA,.01,,"@","CLEAN")  ;  I $D(CLEAN) D FILE^DIE(,"CLEAN")  Q  ; PRVHELP ; Text for the provider function help  Q:$G(X)'="??"  N IBZ,IBQUIT,IB,IB1,DIR,Z  S IBQUIT=0  S Z=""  I '$D(IOSL)!'$D(IOST) D HOME^%ZIS  Q:IOST'["C-"  D:$G(D0) SPECIFIC(D0)  N DIR,X,Y S DIR(0)="E" D ^DIR K DIR W @IOF  S:$G(D0) Z=$$FT^IBCEF(D0)  S IB=IOSL,IB1=1  F IBZ=1:1 S:$P($T(HLPTXT+IBZ),";;",2)="" IBQUIT=1 Q:IBQUIT  S IB1=1 D  . I $Y>(IB-3) N DIR,X,Y S IB1=0,DIR(0)="E" D ^DIR K DIR S IB=IB+IOSL I Y'=1 S IBQUIT=1 Q  . W !,$P($T(HLPTXT+IBZ),";;",2)  I IB1 D  . N DIR,X,Y S DIR(0)="E" D ^DIR K DIR  W @IOF  Q  ; SPECIFIC(IBIFN) ; Display specific provider requirements for the bill IBIFN  N IBFT,IBPRV,IBR,ONBILL,Z,IBZ  S IBFT=$$FT^IBCEF(IBIFN)  D GETPRV^IBCEU(IBIFN,"ALL",.IBPRV) ;Returns needed providers  W !,"This bill is ",$S(IBFT=3:"UB-04",1:"CMS-1500"),"/",$S($$INPAT^IBCEF(IBIFN):"Inpatient",1:"Outpatient")  W !!,"The valid provider functions for this bill are:"  F IBZ=1:1:5,9 I $$PRVOK^IBCEU(IBZ,IBIFN) D  . S ONBILL=$$CKPROV^IBCEU(IBIFN,IBZ)  . S IBR=$S($G(IBPRV(IBZ,"NOTOPT")):1,$G(IBPRV(IBZ,"SITUATIONAL")):2,1:0) ; DEM;432 added "SITUATIONAL" check.  . ; ib2.0\*432  . ; W !,IBZ," ",$$EXPAND^IBTRE(399.0222,.01,IBZ),?18,$S(IBR&'ONBILL:"\*\*",1:""),?20,$S(IBR:"REQUIRED",1:"OPTIONAL"),$S(ONBILL:" - ALREADY ON BILL",1:" - NOT ON BILL")  . W !,IBZ," ",$$EXPAND^IBTRE(399.0222,.01,IBZ),?18,$S(IBR&'ONBILL:"\*\*",1:""),?20,$S(IBR=1:"REQUIRED",IBR=2:"SITUATIONAL",1:"OPTIONAL")  W !  Q  ; HLPTXT ; Helptext for provider function  ;;   ;;PROVIDER FUNCTION requirements:  ;;   ;;RENDERING: UB-04 Situational or CMS-1500 REQUIRED (CMS-1500)  ;; This is the provider who performed a service.  ;;   ;;ATTENDING: UB-04 REQUIRED  ;; The physician who has primary responsibility  ;; for the patient's medical care and treatment.   ;;   ;;OPERATING: UB-04 SITUATIONAL   ;; The provider who performed the principal procedure(s)  ;; being billed.  ;; UB-04 (inpatient): Situational IF type of bill has first 2  ;; digits of 11, and there is a principal  ;; procedure that will print in Form  ;; Locator 74 of the claim, there must be  ;; an Operating or Rendering Provider.  ;; UB-04 (outpatient):REQUIRED IF type of bill has first 2  ;; digits of 83, and there is a principal  ;; procedure that will print in Form  ;; Locator 74 of the claim.  ;;   ;;REFERRING: UB-04 or CMS-1500 SITUATIONAL  ;; The provider who referred the patient for the services being billed.   ;;   ;;SUPERVISING: CMS-1500 OPTIONAL  ;; Required when the rendering provider is supervised  ;; by another provider. Data will not be printed.  ;;   ;;OTHER OPERATING: UB-04 SITUATIONAL  ;; Used to report another Operating Physician. There must  ;; also be an Operating Physician on the claim.  ;;   ;; 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.  ;;  ; LINKRX(IBIFN,IBREV) ; Ask for revenue code's RX if not already there  N DIR,X,Y,IBZ,IBRX,Z,Z0,DA  Q:$P($G(^DGCR(399,IBIFN,"RC",IBREV,0)),U,11)!($P($G(^(0)),U,10)'=3)  S Z=0 F  S Z=$O(^DGCR(399,IBIFN,"RC",Z)) Q:'Z  I Z'=IBREV S Z0=$G(^(Z,0)) I $P(Z0,U,10)=3,$P(Z0,U,11) S IBRX(+$P(Z0,U,11))=""  S DIR(0)="PAO^IBA(362.4,:AEMQ",DIR("S")="I $P(^(0),U,2)=IBIFN,'$D(IBRX(+Y))"  S DIR("A")="Select Rx for this charge: "  S DIR("?",1)="Enter an Rx# for this revenue code"  S DIR("?")=" The Rx must not already have an associated revenue code"  D ^DIR K DIR  I Y>0 D  . S DA(1)=IBIFN,DA=IBREV,IBZ=""  . D FDA^DILF(399.042,.DA,.11,"R",+Y,"IBZ")  . D FILE^DIE(,"IBZ")  Q  ; LINKCPT(IBIFN,IBREV) ; Ask for revenue code's CPT  N DIR,X,Y,IBZ,IBCP,Z,Z0,Z1,DA,IBRC,IBP  S IBRC=$G(^DGCR(399,IBIFN,"RC",IBREV,0))  Q:$P(IBRC,U,8)!($P(IBRC,U,10)'=4)  S IBP=+$P(IBRC,U,6)  I $P(IBRC,U,11) W !,"PROCEDURE #"\_$P(IBRC,U,11)\_" HAS BEEN ASSOCIATED WITH THIS MANUAL CHARGE"  I '$P(IBRC,U,11) D  Q:IBRC=""  . S DIR("?",1)="Respond YES if this revenue code charge specifically references the data for"  . S DIR("?",2)=" a particular procedure that was manually entered on the previous screen."  . S DIR("?",3)=" For outpatient UB-04 bills, associating a manual revenue code charge with",DIR("?")=" a procedure is the only way to print a modifier in box 44"  . S DIR(0)="YA",DIR("A")="SHOULD A PROCEDURE ENTRY BE ASSOCIATED WITH THIS CHARGE?: ",DIR("B")=$S(IBP:"YES",1:"NO") W ! D ^DIR K DIR W !  . I Y'=1 S IBRC="" Q  I $P(IBRC,U,11) D  . S DIR("?",1)="Respond YES if you no longer want this revenue code charge to reference a",DIR("?")=" specific manually entered procedure"  . S DIR(0)="YA",DIR("A")="DELETE THE EXISTING PROCEDURE ASSOCIATION?: ",DIR("B")="NO" W ! D ^DIR K DIR  . I Y=1 D UPDPTR(IBIFN,IBREV,"") S $P(IBRC,U,11)=""  S Z=0 F  S Z=$O(^DGCR(399,IBIFN,"RC",Z)) Q:'Z  S Z0=$G(^(Z,0)) I IBREV'=Z,$P(Z0,U,11) D  . ; Don't allow to link to 'used' proc  . I $P(Z0,U,10)=4 S IBCP($P(Z0,U,11))="" Q  . I $P(Z0,U,10)=3,$P(Z0,U,15) S IBCP($P(Z0,U,15))=""  S DIR(0)="PAO^DGCR(399,"\_IBIFN\_",""CP"",:AEMQ",DIR("S")="I '$D(IBCP(+Y)),$P(^(0),U)[""CPT"",+^(0)="\_+$P($G(^DGCR(399,IBIFN,"RC",IBREV,0)),U,6)  S DIR("A")="SELECT A PROCEDURE ENTRY: "\_$S($P(IBRC,U,11):"#"\_$P(IBRC,U,11)\_" - "\_$$EXPAND^IBTRE(399.0304,.01,$P($G(^DGCR(399,IBIFN,"CP",$P(IBRC,U,11),0)),U))\_"// ",1:"")  S DIR("?")="Enter a manually-added CPT procedure to associate with this charge"  S DA(1)=IBIFN  D ^DIR K DIR W !  I Y>0 D UPDPTR(IBIFN,IBREV,+Y)  Q  ; UPDPTR(IBIFN,IBREV,Y) ;  N IBZ,DA  S DA(1)=IBIFN,DA=IBREV,IBZ=""  D FDA^DILF(399.042,.DA,.11,"R",$S(Y:+Y,1:""),"IBZ")  D FILE^DIE(,"IBZ")  Q  ; INSFT(IBIFN) ; Returns 1 if form type is UB-04, 0 if CMS-1500  Q ($$FT^IBCEF(IBIFN)=3) | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCEU5 ;ALB/TMP - EDI UTILITIES (continued) FOR CMS-1500 ;13-DEC-99  ;;2.0;INTEGRATED BILLING;\*\*51,137,232,348,349,432,592\*\*;21-MAR-94;Build 192  ;;Per VHA Directive 2004-038, this routine should not be modified.  Q  ; EXTCR(IBPRV) ; Called by trigger on field .02 of file 399.0222  ; Also called by trigger on field .02 of file 399.0404 (DEM;432).  ; Function returns the first 3 digits of the provider's degree if  ; a VA provider or the credentials in file 355.9 if non-VA provider  ; IBPRV = vp to file 200 or 355.93  Q $E($$CRED^IBCEU(IBPRV),1,3)  ;  FTPRV(IBIFN,NOASK) ; If form type changes from UB-04 to CMS-1500 or vice  ; versa, ask to change provider function to appropriate function for  ; form type (ATTENDING = UB-04, RENDERING = CMS-1500)  ; IBIFN = ien of bill in file 399  ; NOASK (flag) = 1 if change should happen without asking first  N ATT,REN,FT  S FT=$$FT^IBCEF(IBIFN)  S REN=$$CKPROV^IBCEU(IBIFN,3,1)  S ATT=$$CKPROV^IBCEU(IBIFN,4,1)  ;JWS;IB\*2.0\*592;add Dental form check   I $S(FT=2:'REN&ATT,FT=3:'ATT&REN,FT=7:'REN&ATT,1:0) D  . I '$G(NOASK) D TXFERPRV(IBIFN,FT) Q  . D PRVCHG(IBIFN,FT)  D CLEANUP(IBIFN,FT)  Q  ; TXFERPRV(IBIFN,FT) ; Ask to change the function of the main provider on  ; bill IBIFN to the function appropriate to the form type FT  ;   N DIR,X,Y,Z,DIE,DA,DR,HAVE,NEED,IBZ  ; DEM;432 - Changed the prompt from uppercase to mixed case.  W ! S DIR("A")=" Change the Claim Level "\_$S(FT=3:"Rendering",1:"Attending")\_" provider's function to "\_$S(FT=3:"Attending",1:"Rendering")\_"?: "  S DIR(0)="YA",DIR("B")="NO",DIR("?",1)="If you answer YES here, you will make the claim level provider functions",DIR("?")=" consistent with the form type of the bill"  D ^DIR K DIR  I Y'=1 Q  D PRVCHG(IBIFN,FT)  Q  ; PRVCHG(IBIFN,IBFT) ; Change provider type to type consistent with current  ; data on bill  N Z,IBZ,HAVE,NEED,DIE,DA,X,Y  S HAVE=$S(IBFT=3:3,1:4)  S NEED=$S(IBFT=3:4,1:3)  S Z=$O(^DGCR(399,IBIFN,"PRV","B",HAVE,0))  I Z D  . S DA(1)=IBIFN,DA=+Z  . D FDA^DILF(399.0222,.DA,.01,,NEED,"IBZ")  . D FILE^DIE(,"IBZ")  ;I Z S DA(1)=IBIFN,DIE="^DGCR(399,"\_DA(1)\_",""PRV"",",DA=+Z,DR=".01////"\_NEED D FILE^DIE(,DIE  Q  ; CLEANUP(IBIFN,FT) ; If form type changes remove any extra provider FUNCTIONS.  N X,PRV,CLEAN,DA,DIE  ;  ;JWS;IB\*2.0\*592 US1108 - If form type changes to (7) J430D - Dental, default Bill Charge Type  I FT=7 S CLEAN(399,IBIFN\_",",.27)=2  ; (3) If form type changes from CMS-1500 to UB-04, remove any extra provider FUNCTIONS.   ;JWS;IB\*2.0\*592 US1108 - added 6-ASSISTANT SURGEON  I FT=3 F X=5,6 D  ; 5-SUPERVISING, 6-ASSISTANT SURGEON  . I $D(^DGCR(399,IBIFN,"PRV","B",X)) D  .. S PRV=0 F  S PRV=$O(^DGCR(399,IBIFN,"PRV","B",X,PRV)) Q:+PRV=0 D  ... S DA(1)=IBIFN,DA=PRV D FDA^DILF(399.0222,.DA,.01,,"@","CLEAN")  ;  ; (2) If form type changes from UB-04 to CMS-1500, remove any extra provider FUNCTIONS.   ;JWS;IB\*2.0\*592 US1108 - added 6-ASSISTANT SURGEON  I FT=2 F X=2,4,6,9 D  ; 2-OPERATING, 4-ATTENDING, 6-ASSISTANT SURGEON, 9-OTHER  . I $D(^DGCR(399,IBIFN,"PRV","B",X)) D  .. S PRV=0 F  S PRV=$O(^DGCR(399,IBIFN,"PRV","B",X,PRV)) Q:+PRV=0 D  ... S DA(1)=IBIFN,DA=PRV D FDA^DILF(399.0222,.DA,.01,,"@","CLEAN")  ;  I $D(CLEAN) D FILE^DIE(,"CLEAN")  Q  ; PRVHELP ; Text for the provider function help  Q:$G(X)'="??"  N IBZ,IBQUIT,IB,IB1,DIR,Z  S IBQUIT=0  S Z=""  I '$D(IOSL)!'$D(IOST) D HOME^%ZIS  Q:IOST'["C-"  D:$G(D0) SPECIFIC(D0)  N DIR,X,Y S DIR(0)="E" D ^DIR K DIR W @IOF  S:$G(D0) Z=$$FT^IBCEF(D0)  S IB=IOSL,IB1=1  F IBZ=1:1 S:$P($T(HLPTXT+IBZ),";;",2)="" IBQUIT=1 Q:IBQUIT  S IB1=1 D  . I $Y>(IB-3) N DIR,X,Y S IB1=0,DIR(0)="E" D ^DIR K DIR S IB=IB+IOSL I Y'=1 S IBQUIT=1 Q  . W !,$P($T(HLPTXT+IBZ),";;",2)  I IB1 D  . N DIR,X,Y S DIR(0)="E" D ^DIR K DIR  W @IOF  Q  ; SPECIFIC(IBIFN) ; Display specific provider requirements for the bill IBIFN  N IBFT,IBPRV,IBR,ONBILL,Z,IBZ  S IBFT=$$FT^IBCEF(IBIFN)  D GETPRV^IBCEU(IBIFN,"ALL",.IBPRV) ;Returns needed providers  ;JWS;IB\*2.0\*592 US1108 - added Dental form #7  W !,"This bill is ",$S(IBFT=7:"J430D",IBFT=3:"UB-04",1:"CMS-1500"),"/",$S($$INPAT^IBCEF(IBIFN):"Inpatient",1:"Outpatient")  W !!,"The valid provider functions for this bill are:"  ;JWS;IB\*2.0\*592 US1108 - changed loop from :5 to :6 for Assistant Surgeon  F IBZ=1:1:6,9 I $$PRVOK^IBCEU(IBZ,IBIFN) D  . S ONBILL=$$CKPROV^IBCEU(IBIFN,IBZ)  . S IBR=$S($G(IBPRV(IBZ,"NOTOPT")):1,$G(IBPRV(IBZ,"SITUATIONAL")):2,1:0) ; DEM;432 added "SITUATIONAL" check.  . ;JWS;IB\*2.0\*592 US1108 - dental form#7  . I IBFT=7 S IBR=2  . ; ib2.0\*432  . ; W !,IBZ," ",$$EXPAND^IBTRE(399.0222,.01,IBZ),?18,$S(IBR&'ONBILL:"\*\*",1:""),?20,$S(IBR:"REQUIRED",1:"OPTIONAL"),$S(ONBILL:" - ALREADY ON BILL",1:" - NOT ON BILL")  . W !,IBZ," ",$$EXPAND^IBTRE(399.0222,.01,IBZ),?18,$S(IBR&'ONBILL:"\*\*",1:""),?23,$S(IBR=1:"REQUIRED",IBR=2:"SITUATIONAL",1:"OPTIONAL")  W !  Q  ; HLPTXT ; Helptext for provider function  ;;   ;;PROVIDER FUNCTION requirements:  ;;   ;;RENDERING: UB-04 Situational, CMS-1500 REQUIRED (CMS-1500), or J430D Situational  ;; This is the provider who performed a service.  ;;   ;;ATTENDING: UB-04 REQUIRED  ;; The physician who has primary responsibility  ;; for the patient's medical care and treatment.   ;;   ;;OPERATING: UB-04 SITUATIONAL   ;; The provider who performed the principal procedure(s)  ;; being billed.  ;; UB-04 (inpatient): Situational IF type of bill has first 2  ;; digits of 11, and there is a principal  ;; procedure that will print in Form  ;; Locator 74 of the claim, there must be  ;; an Operating or Rendering Provider.  ;; UB-04 (outpatient):REQUIRED IF type of bill has first 2  ;; digits of 83, and there is a principal  ;; procedure that will print in Form  ;; Locator 74 of the claim.  ;;   ;;REFERRING: UB-04, CMS-1500, or J430D SITUATIONAL  ;; The provider who referred the patient for the services being billed.   ;;   ;;SUPERVISING: CMS-1500 OPTIONAL or J430D SITUATIONAL  ;; Required when the rendering provider is supervised  ;; by another provider. Data will not be printed.  ;;   ;;OTHER OPERATING: UB-04 SITUATIONAL  ;; Used to report another Operating Physician. There must  ;; also be an Operating Physician on the claim.  ;;  ;;ASSISTANT SURGEON: J430D SITUATIONAL  ;; User when the Rendering Provider provided these services in the role  ;; of the Assisting Surgeon.  ;;   ;; 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.  ;;  ; LINKRX(IBIFN,IBREV) ; Ask for revenue code's RX if not already there  N DIR,X,Y,IBZ,IBRX,Z,Z0,DA  Q:$P($G(^DGCR(399,IBIFN,"RC",IBREV,0)),U,11)!($P($G(^(0)),U,10)'=3)  S Z=0 F  S Z=$O(^DGCR(399,IBIFN,"RC",Z)) Q:'Z  I Z'=IBREV S Z0=$G(^(Z,0)) I $P(Z0,U,10)=3,$P(Z0,U,11) S IBRX(+$P(Z0,U,11))=""  S DIR(0)="PAO^IBA(362.4,:AEMQ",DIR("S")="I $P(^(0),U,2)=IBIFN,'$D(IBRX(+Y))"  S DIR("A")="Select Rx for this charge: "  S DIR("?",1)="Enter an Rx# for this revenue code"  S DIR("?")=" The Rx must not already have an associated revenue code"  D ^DIR K DIR  I Y>0 D  . S DA(1)=IBIFN,DA=IBREV,IBZ=""  . D FDA^DILF(399.042,.DA,.11,"R",+Y,"IBZ")  . D FILE^DIE(,"IBZ")  Q  ; LINKCPT(IBIFN,IBREV) ; Ask for revenue code's CPT  N DIR,X,Y,IBZ,IBCP,Z,Z0,Z1,DA,IBRC,IBP  S IBRC=$G(^DGCR(399,IBIFN,"RC",IBREV,0))  Q:$P(IBRC,U,8)!($P(IBRC,U,10)'=4)  S IBP=+$P(IBRC,U,6)  I $P(IBRC,U,11) W !,"PROCEDURE #"\_$P(IBRC,U,11)\_" HAS BEEN ASSOCIATED WITH THIS MANUAL CHARGE"  I '$P(IBRC,U,11) D  Q:IBRC=""  . S DIR("?",1)="Respond YES if this revenue code charge specifically references the data for"  . S DIR("?",2)=" a particular procedure that was manually entered on the previous screen."  . S DIR("?",3)=" For outpatient UB-04 bills, associating a manual revenue code charge with",DIR("?")=" a procedure is the only way to print a modifier in box 44"  . S DIR(0)="YA",DIR("A")="SHOULD A PROCEDURE ENTRY BE ASSOCIATED WITH THIS CHARGE?: ",DIR("B")=$S(IBP:"YES",1:"NO") W ! D ^DIR K DIR W !  . I Y'=1 S IBRC="" Q  I $P(IBRC,U,11) D  . S DIR("?",1)="Respond YES if you no longer want this revenue code charge to reference a",DIR("?")=" specific manually entered procedure"  . S DIR(0)="YA",DIR("A")="DELETE THE EXISTING PROCEDURE ASSOCIATION?: ",DIR("B")="NO" W ! D ^DIR K DIR  . I Y=1 D UPDPTR(IBIFN,IBREV,"") S $P(IBRC,U,11)=""  S Z=0 F  S Z=$O(^DGCR(399,IBIFN,"RC",Z)) Q:'Z  S Z0=$G(^(Z,0)) I IBREV'=Z,$P(Z0,U,11) D  . ; Don't allow to link to 'used' proc  . I $P(Z0,U,10)=4 S IBCP($P(Z0,U,11))="" Q  . I $P(Z0,U,10)=3,$P(Z0,U,15) S IBCP($P(Z0,U,15))=""  S DIR(0)="PAO^DGCR(399,"\_IBIFN\_",""CP"",:AEMQ",DIR("S")="I '$D(IBCP(+Y)),$P(^(0),U)[""CPT"",+^(0)="\_+$P($G(^DGCR(399,IBIFN,"RC",IBREV,0)),U,6)  S DIR("A")="SELECT A PROCEDURE ENTRY: "\_$S($P(IBRC,U,11):"#"\_$P(IBRC,U,11)\_" - "\_$$EXPAND^IBTRE(399.0304,.01,$P($G(^DGCR(399,IBIFN,"CP",$P(IBRC,U,11),0)),U))\_"// ",1:"")  S DIR("?")="Enter a manually-added CPT procedure to associate with this charge"  S DA(1)=IBIFN  D ^DIR K DIR W !  I Y>0 D UPDPTR(IBIFN,IBREV,+Y)  Q  ; UPDPTR(IBIFN,IBREV,Y) ;  N IBZ,DA  S DA(1)=IBIFN,DA=IBREV,IBZ=""  D FDA^DILF(399.042,.DA,.11,"R",$S(Y:+Y,1:""),"IBZ")  D FILE^DIE(,"IBZ")  Q  ; INSFT(IBIFN) ; Returns 1 if form type is UB-04, 0 if CMS-1500 or J430D  Q ($$FT^IBCEF(IBIFN)=3) | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCEU7 | | | | | | | | |
| **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** | | | | | | | | | |
| IBCEU7 ;ALB/DEM - EDI UTILITIES ;26-SEP-2010  ;;2.0;INTEGRATED BILLING;\*\*432\*\*;21-MAR-94;Build 192  ;;Per VHA Directive 2004-038, this routine should not be modified.  Q  ; LNPRVOK(VAL,IBIFN) ; Check bill form & line prov function agree  ; DEM;432 - New routine for Claim Line Provider.  ; VAL = internal value of prov function  ;  ; Allowable line provider functions for UB04 (FORM TYPE = 3)  ; Inpatient and UB04 Outpatient:  ; - Rendering Provider(3).  ; - Referring Provider(1).  ; - Operating Physician(2).  ; - Other Operating Physician(9).  ;  ; Allowable line provider functions for CMS 1500 (FORM TYPE = 2)  ; Inpatient and CMS 1500 Outpatient:  ; - Rendering Provider(3).  ; - Referring Provider(1).  ; - Supervising Provider(5).  ;  N OK,IBUB  S VAL=$$UP^XLFSTR(VAL)  S OK=$S(VAL'="":1,1:0)  G:'OK!'$G(IBIFN) PRVQ  ;  S IBUB=($$FT^IBCEF(IBIFN)=3) ; 1 if UB-04 ; 0 if CMS-1500  ;  ;  S OK=0  S:(IBUB)&("1239"[VAL) OK=1 ; UB-04  S:('IBUB)&("135"[VAL) OK=1 ; CMS-1500  ; PRVQ Q OK  ; LNPRVHLP ;Helptext for line provider function.  ;  N IBZ,IBQUIT,VALUE,FORMAT  F IBZ=1:1 S:$P($T(HLPTXT+IBZ),";;",2)="END" IBQUIT=1 Q:$G(IBQUIT) D  . S VALUE=$P($T(HLPTXT+IBZ),";;",2)  . S FORMAT=$S(VALUE="":"!",1:"")  . D EN^DDIOL(VALUE,"",FORMAT)  . Q  Q  ; HLPTXT ; Helptext for line provider function.  ;;  ;;Enter the name of the line level provider who provided this service.  ;;Line level providers are optional and should only be entered if  ;;different from the claim level provider.  ;;  ;;  ;;END  ; HLPTXT2 ; \*\*\*Currently, not activated\*\*\* - Helptext for line provider function.  ;;  ;;LINE PROVIDER FUNCTION requirements:  ;;  ;;Allowable line provider functions for UB04 Inpatient and Outpatient:  ;;  ;; - Rendering Provider(3).  ;; - Referring Provider(1).  ;; - Operating Physician(2).  ;; - Other Operating Physician(9).  ;;  ;;Allowable line provider functions for CMS 1500 Inpatient and Outpatient:  ;;  ;; - Rendering Provider(3).  ;; - Referring Provider(1).  ;; - Supervising Provider(5).  ;;  ;;END  Q  ; LNPRVFT(IBFT,IBLNPRV) ; DEM;432 - Field Index "AK" (#301) on FORM TYPE field (399,.19).  ;  ; Description:  ;  ; This function is called by the FORM TYPE (399,.19) "AK" field index.  ; In the case when the FORM TYPE field is changed, then the line  ; provider types are checked to see if any, or all, line providers  ; need to be deleted from the claim.  ;  ; Input:  ;  ; IBFT = FORM TYPE = 2 = (CMS-1500), or FORM TYPE = 3 = (UB-04).  ; Must be either FORM TYPE 2, or FORM TYPE 3 to continue.  ; See allowable line provider functions by FORM TYPE below.  ; IBLNPRV = Array passed by reference.  ;  ; Output:  ;  ; OK = 1 = line providers to delete, OK = 0 = no line providers to delete.  ; IBLNPRV Array = If line providers to delete, then array contains  ; these line providers - IBLNPRV(399.0404,"IENS",.01)="@"  ;  ; Allowable line provider functions for UB04 (FORM TYPE = 3)  ; Inpatient and UB04 Outpatient:  ; - Rendering Provider(VAL=3).  ; - Referring Provider(VAL=1).  ; - Operating Physician(VAL=2).  ; - Other Operating Physician(VAL=9).  ;  ; Allowable line provider functions for CMS 1500 (FORM TYPE = 2)  ; Inpatient and CMS 1500 Outpatient:  ; - Rendering Provider(VAL=3).  ; - Referring Provider(VAL=1).  ; - Supervising Provider(VAL=5).  ;  Q:'$G(IBIFN) 0 ; QUIT 0 if no claim number.  Q:'$G(IBFT) 0 ; QUIT 0 if no FORM TYPE.  Q:(IBFT'=2)&(IBFT'=3) 0 ; QUIT 0 - Must be CMS-1500 (2) or UB-04 (3) FORM TYPE.  ;  N IBPRVFUN,OK  S:IBFT=3 IBPRVFUN("VAL",IBFT)="1239"  ; Allowable LINE PROVIDER FUNCTIONs for UB-04.  S:IBFT=2 IBPRVFUN("VAL",IBFT)="135"  ; Allowable LINE PROVIDER FUNCTIONs for CMS-1500.  ;  S OK=0 ; Initialize OK=0.  ;  N IBPROCP,IBLPIEN,IBLNPROV,DA  S IBPROCP=0 F  S IBPROCP=$O(^DGCR(399,IBIFN,"CP",IBPROCP)) Q:'IBPROCP  D  ; Loop on PROCEDURES multiple.  . Q:'($D(^DGCR(399,IBIFN,"CP",IBPROCP,0))#10) ; No zero node for procedure.  . S IBPRVFUN=0 F  S IBPRVFUN=$O(^DGCR(399,IBIFN,"CP",IBPROCP,"LNPRV","B",IBPRVFUN)) Q:'IBPRVFUN  D:IBPRVFUN("VAL",IBFT)'[IBPRVFUN  . . S IBLPIEN=0 F  S IBLPIEN=$O(^DGCR(399,IBIFN,"CP",IBPROCP,"LNPRV","B",IBPRVFUN,IBLPIEN)) Q:'IBLPIEN  D  . . . Q:'($D(^DGCR(399,IBIFN,"CP",IBPROCP,"LNPRV",IBLPIEN,0))#10) ; No zero node for line level provider.  . . . S IBLNPROV=$P(^DGCR(399,IBIFN,"CP",IBPROCP,"LNPRV",IBLPIEN,0),U,2)  . . . Q:'IBLNPROV  ; No line provider for this line provider function.  . . . S OK=1,IBLNPRV(399.0404,IBLPIEN\_","\_IBPROCP\_","\_IBIFN\_",",.01)="@"  ; We have at least one line provider to delete from claim.  . . . Q  . . Q  . Q  ;  Q OK  ; REMOVE(IBIFN,IBFT) ; This will be used to remove all line level providers and all line level attachments from inpatient UB claims  ;  ; Input IBIFN - Claim Number  ;  Q:IBFT'=3 ; Only worried about UBs  N IBINPAT  S IBINPAT=$$INPAT^IBCEF(IBIFN) Q:'IBINPAT   ; Quit if it's not an inpatient  ;  ; If we got here, we have an inpatient UB  ; In which case, we should not have any line level providers or line level attachment control numbers  ; If we do, then let's remove them  ;  N CPIEN,LNPRVIEN,FDA,ERR  S CPIEN=0 F  S CPIEN=$O(^DGCR(399,IBIFN,"CP",CPIEN)) Q:'+CPIEN  D  . ;  . ; Remove the Line level attachments  . S FDA(399.0304,CPIEN\_","\_IBIFN\_",",70)="@"  . S FDA(399.0304,CPIEN\_","\_IBIFN\_",",71)="@"  . S FDA(399.0304,CPIEN\_","\_IBIFN\_",",72)="@"  . D FILE^DIE("E","FDA")  . ;  . K FDA  . S LNPRVIEN=0 F  S LNPRVIEN=$O(^DGCR(399,IBIFN,"CP",CPIEN,"LNPRV",LNPRVIEN)) Q:'+LNPRVIEN  D  .. ;  .. ;Remove the line level providers  .. S FDA(399.0404,LNPRVIEN\_","\_CPIEN\_","\_IBIFN\_",",.01)="@"  . I $D(FDA) D FILE^DIE("E","FDA")  Q | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCEU7 ;ALB/DEM - EDI UTILITIES ;26-SEP-2010  ;;2.0;INTEGRATED BILLING;\*\*432,592\*\*;21-MAR-94;Build 192  ;;Per VHA Directive 2004-038, this routine should not be modified.  Q  ; LNPRVOK(VAL,IBIFN) ; Check bill form & line prov function agree  ; DEM;432 - New routine for Claim Line Provider.  ; VAL = internal value of prov function  ;  ; Allowable line provider functions for UB04 (FORM TYPE = 3)  ; Inpatient and UB04 Outpatient:  ; - Rendering Provider(3).  ; - Referring Provider(1).  ; - Operating Physician(2).  ; - Other Operating Physician(9).  ;  ; Allowable line provider functions for CMS 1500 (FORM TYPE = 2)  ; Inpatient and CMS 1500 Outpatient:  ; - Rendering Provider(3).  ; - Referring Provider(1).  ; - Supervising Provider(5).  ;  ;JWS;IB\*2.0\*592 US1108  ; Allowable line provider functions for J430D Dental (FORM TYPE = 7)  ; Inpatient and CMS 1500 Outpatient:  ; - Rendering Provider(3).  ; - Referring Provider(1).  ; - Supervising Provider(5).  ; - Assistant Surgeon (6).  ;  N OK,IBUB  S VAL=$$UP^XLFSTR(VAL)  S OK=$S(VAL'="":1,1:0)  G:'OK!'$G(IBIFN) PRVQ  ;  ;JWS;IB\*2.0\*592 US1108 - 2 for form#7 Dental  S IBUB=$S($$FT^IBCEF(IBIFN)=7:2,1:($$FT^IBCEF(IBIFN)=3)) ; 1 if UB-04 ; 0 if CMS-1500 ; 2 if J430D Dental form  ;  S OK=0  I IBUB=1,"1239"[VAL S OK=1 ; UB-04  I 'IBUB,"135"[VAL S OK=1 ; CMS-1500  ;JWS;IB\*2.0\*592 US1108 J430D Dental  I IBUB=2,"1356"[VAL S OK=1  ; PRVQ Q OK  ; LNPRVHLP ;Helptext for line provider function.  ;  N IBZ,IBQUIT,VALUE,FORMAT  F IBZ=1:1 S:$P($T(HLPTXT+IBZ),";;",2)="END" IBQUIT=1 Q:$G(IBQUIT) D  . S VALUE=$P($T(HLPTXT+IBZ),";;",2)  . S FORMAT=$S(VALUE="":"!",1:"")  . D EN^DDIOL(VALUE,"",FORMAT)  . Q  Q  ; HLPTXT ; Helptext for line provider function.  ;;  ;;Enter the name of the line level provider who provided this service.  ;;Line level providers are optional and should only be entered if  ;;different from the claim level provider.  ;;  ;;  ;;END  ; HLPTXT2 ; \*\*\*Currently, not activated\*\*\* - Helptext for line provider function.  ;;  ;;LINE PROVIDER FUNCTION requirements:  ;;  ;;Allowable line provider functions for UB04 Inpatient and Outpatient:  ;;  ;; - Rendering Provider(3).  ;; - Referring Provider(1).  ;; - Operating Physician(2).  ;; - Other Operating Physician(9).  ;;  ;;Allowable line provider functions for CMS 1500 Inpatient and Outpatient:  ;;  ;; - Rendering Provider(3).  ;; - Referring Provider(1).  ;; - Supervising Provider(5).  ;;  ;; Allowable line provider functions for J430D Dental (FORM TYPE = 7)  ;; Inpatient and CMS 1500 Outpatient:  ;; - Rendering Provider(3).  ;; - Referring Provider(1).  ;; - Supervising Provider(5).  ;; - Assistant Surgeon (6).  ;;  ;;END  Q  ; LNPRVFT(IBFT,IBLNPRV) ; DEM;432 - Field Index "AK" (#301) on FORM TYPE field (399,.19).  ;  ; Description:  ;  ; This function is called by the FORM TYPE (399,.19) "AK" field index.  ; In the case when the FORM TYPE field is changed, then the line  ; provider types are checked to see if any, or all, line providers  ; need to be deleted from the claim.  ;  ; Input:  ;  ; IBFT = FORM TYPE = 2 = (CMS-1500), or FORM TYPE = 3 = (UB-04).  ; Must be either FORM TYPE 2, or FORM TYPE 3 to continue.  ; See allowable line provider functions by FORM TYPE below.  ; IBLNPRV = Array passed by reference.  ;  ; Output:  ;  ; OK = 1 = line providers to delete, OK = 0 = no line providers to delete.  ; IBLNPRV Array = If line providers to delete, then array contains  ; these line providers - IBLNPRV(399.0404,"IENS",.01)="@"  ;  ; Allowable line provider functions for UB04 (FORM TYPE = 3)  ; Inpatient and UB04 Outpatient:  ; - Rendering Provider(VAL=3).  ; - Referring Provider(VAL=1).  ; - Operating Physician(VAL=2).  ; - Other Operating Physician(VAL=9).  ;  ; Allowable line provider functions for CMS 1500 (FORM TYPE = 2)  ; Inpatient and CMS 1500 Outpatient:  ; - Rendering Provider(VAL=3).  ; - Referring Provider(VAL=1).  ; - Supervising Provider(VAL=5).  ;  ; Allowable line provider functions for J430D (FORM TYPE = 7)  ; Dental:  ; - Rendering Provider(VAL=3).  ; - Referring Provider(VAL=1).  ; - Supervising Provider(VAL=5).  ; - Assistant Surgeon(VAL=6).  ;  Q:'$G(IBIFN) 0 ; QUIT 0 if no claim number.  Q:'$G(IBFT) 0 ; QUIT 0 if no FORM TYPE.  ;JWS;IB\*2.0\*592 US1108  Q:(IBFT'=2)&(IBFT'=3)&(IBFT'=7) 0 ; QUIT 0 - Must be CMS-1500 (2) or UB-04 (3) or J430D (7) FORM TYPE.  ;  N IBPRVFUN,OK  S:IBFT=3 IBPRVFUN("VAL",IBFT)="1239"  ; Allowable LINE PROVIDER FUNCTIONs for UB-04.  S:IBFT=2 IBPRVFUN("VAL",IBFT)="135"  ; Allowable LINE PROVIDER FUNCTIONs for CMS-1500.  ;JWS;IB\*2.0\*592 US1108  S:IBFT=7 IBPRVFUN("VAL",IBFT)="1356"  ;Allowable LINE PROVIDER FUNCTIONs for J430D.  ;  S OK=0 ; Initialize OK=0.  ;  N IBPROCP,IBLPIEN,IBLNPROV,DA  S IBPROCP=0 F  S IBPROCP=$O(^DGCR(399,IBIFN,"CP",IBPROCP)) Q:'IBPROCP  D  ; Loop on PROCEDURES multiple.  . Q:'($D(^DGCR(399,IBIFN,"CP",IBPROCP,0))#10) ; No zero node for procedure.  . S IBPRVFUN=0 F  S IBPRVFUN=$O(^DGCR(399,IBIFN,"CP",IBPROCP,"LNPRV","B",IBPRVFUN)) Q:'IBPRVFUN  D:IBPRVFUN("VAL",IBFT)'[IBPRVFUN  . . S IBLPIEN=0 F  S IBLPIEN=$O(^DGCR(399,IBIFN,"CP",IBPROCP,"LNPRV","B",IBPRVFUN,IBLPIEN)) Q:'IBLPIEN  D  . . . Q:'($D(^DGCR(399,IBIFN,"CP",IBPROCP,"LNPRV",IBLPIEN,0))#10) ; No zero node for line level provider.  . . . S IBLNPROV=$P(^DGCR(399,IBIFN,"CP",IBPROCP,"LNPRV",IBLPIEN,0),U,2)  . . . Q:'IBLNPROV  ; No line provider for this line provider function.  . . . S OK=1,IBLNPRV(399.0404,IBLPIEN\_","\_IBPROCP\_","\_IBIFN\_",",.01)="@"  ; We have at leaset one line provider to delete from claim.  . . . Q  . . Q  . Q  ;  Q OK  ; REMOVE(IBIFN,IBFT) ; This will be used to remove all line level providers and all line level attachments from inpatient UB claims  ;  ; Input IBIFN - Claim Number  ;  Q:IBFT'=3 ; Only worried about UBs  N IBINPAT  S IBINPAT=$$INPAT^IBCEF(IBIFN) Q:'IBINPAT   ; Quit if it's not an inpatient  ;  ; If we got here, we have an inpatient UB  ; In which case, we should not have any line level providers or line level attachment control numbers  ; If we do, then let's remove them  ;  N CPIEN,LNPRVIEN,FDA,ERR  S CPIEN=0 F  S CPIEN=$O(^DGCR(399,IBIFN,"CP",CPIEN)) Q:'+CPIEN  D  . ;  . ; Remove the Line level attachments  . S FDA(399.0304,CPIEN\_","\_IBIFN\_",",70)="@"  . S FDA(399.0304,CPIEN\_","\_IBIFN\_",",71)="@"  . S FDA(399.0304,CPIEN\_","\_IBIFN\_",",72)="@"  . D FILE^DIE("E","FDA")  . ;  . K FDA  . S LNPRVIEN=0 F  S LNPRVIEN=$O(^DGCR(399,IBIFN,"CP",CPIEN,"LNPRV",LNPRVIEN)) Q:'+LNPRVIEN  D  .. ;  .. ;Remove the line level providers  .. S FDA(399.0404,LNPRVIEN\_","\_CPIEN\_","\_IBIFN\_",",.01)="@"  . I $D(FDA) D FILE^DIE("E","FDA")  Q | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCEXTRP | | | | | | | | |
| **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** | | | | | | | | | |
| IBCEXTRP ;ALB/JEH - VIEW/PRINT EDI EXTRACT DATA ;4/22/03 9:59am  ;;2.0;INTEGRATED BILLING;\*\*137,197,211,348,349,377\*\*;21-MAR-94;Build 23  ;;Per VHA Directive 2004-038, this routine should not be modified.  ; EN ; INIT ;  W !!,"This option will display the EDI extract data for a bill.",!  N IBREC1,IBIEN,IBINC,DIC,X,Y,DIR,IB364IEN,IBVNUM,IBSEG,STOP,POP,DTOUT,DUOUT  ;  N DPTNOFZY S DPTNOFZY=1 ; Suppress PATIENT file fuzzy lookups  S DIC="^DGCR(399,",DIC(0)="AEMQ",DIC("S")="I 234[$P(^(0),U,13)" D ^DIC  I Y<1 G EXITQ  S IBIEN=+Y,IBREC1=$G(^DGCR(399,IBIEN,0))  S IB364IEN=$$LAST364^IBCEF4(IBIEN) I +$G(IB364IEN)=0 D  G EXITQ  . W !,"There is no entry in the EDI Transmit Bill file for this bill number."  S IBVNUM=$P($G(^IBA(364,IB364IEN,0)),U,2) I +$G(IBVNUM)=0 D  G EXITQ  . W !!,"There is no batch # for this bill. It has not been transmitted."  S IBVNUM=$P($G(^IBA(364.1,IBVNUM,0)),U)  S DIR("A")="Include Fields With No Data?: ",DIR("B")="NO",DIR(0)="YA"  W ! D ^DIR K DIR  I $D(DTOUT)!$D(DUOUT) G EXITQ  S IBINC=+Y  ;  ; IB\*2\*377 - esg - Ask for specific EDI segments to view  ;  W !  S DIR(0)="SA^A:All EDI Segments;S:Selected EDI Segments"  S DIR("A")="Include (A)ll or (S)elected EDI Segments?: "  S DIR("B")="All EDI Segments"  D ^DIR K DIR  I $D(DTOUT)!$D(DUOUT) G EXITQ  I Y="A" G DEV                    ; all segments, skip to device prompt  ;  W !  K IBSEG  S STOP=0  F  D  Q:STOP  . S DIR(0)="FO^3:4"  . S DIR("A")=" Select EDI Segment"  . I $D(IBSEG) S DIR("A")="Another EDI Segment"  . S DIR("?")="Enter the name of the EDI segment to include."  . D ^DIR K DIR  . I $D(DTOUT)!$D(DUOUT) S STOP=1 Q  . S Y=$$UP^XLFSTR(Y),Y=$$TRIM^XLFSTR(Y) ; uppercase/trim spaces  . I Y="" S STOP=1 Q  . S IBSEG(Y)=""  . Q  I $D(DTOUT)!$D(DUOUT) G EXITQ  ; DEV ; - Select device  N %ZIS,ZTRTN,ZTSAVE,ZTDESC  W !  S %ZIS="QM" D ^%ZIS G:POP EXITQ  I $D(IO("Q")) D  G EXITQ  . S ZTRTN="LIST^IBCEXTRP",ZTDESC="Transmitted Bill Extract Data"  . S ZTSAVE("IB\*")=""  . D ^%ZTLOAD  . W !!,$S($D(ZTSK):"Your task number "\_ZTSK\_" has been queued.",1:"Unable to queue this job.")  .K ZTSK,IO("Q") D HOME^%ZIS  U IO  ; LIST ; - set up array and print data  N IBPG,IBSEQ,IBPC,IBDA,IBREC,IBQUIT,IBILL,IBLINE,IBXDATA,IBERR,IBXERR,Z,Z0,Z1  D EXTRACT(IBIEN,IBVNUM,8,1)  S (IBPG,IBQUIT,IBSEQ,IBPC,IBDA,IBLINE)=0  K ^TMP($J,"IBLINES")  ;IB\*2.0\*211 - rely on form type instead of bill charge type  N IBFMTYP S IBFMTYP=$$FT^IBCEF(IBIEN)  S IBFMTYP=$S(IBFMTYP=2:"CMS-1500",IBFMTYP=3:"UB-04",1:"OTHER"\_"("\_IBFMTYP\_")")  S IBILL=$S($$INPAT^IBCEF(IBIEN,1):"Inpt",1:"Oupt")\_"/"\_IBFMTYP  ;  I $D(^TMP("IBXERR",$J)) D  G EXITQ  . S IBERR=0 F  S IBERR=$O(^TMP("IBXERR",$J,IBERR)) Q:'IBERR  W !,$G(^TMP("IBXERR",$J,IBERR))  . Q  ;  F  S IBSEQ=$O(^IBA(364.6,"ASEQ",8,IBSEQ)) Q:'IBSEQ  I $$INCLUDE(IBSEQ) F  S IBPC=$O(^IBA(364.6,"ASEQ",8,IBSEQ,1,IBPC)) Q:'IBPC  F  S IBDA=$O(^IBA(364.6,"ASEQ",8,IBSEQ,1,IBPC,IBDA)) Q:'IBDA  D  . N IBOK,Z,IBMULT,DSP,IBDATA,PCD,SN  . S IBREC=$G(^IBA(364.6,IBDA,0))  . I $P(IBREC,U,11)=1 Q     ; calculate only field  . ;  . ; processing for piece 1 of this EDI segment to see if there is any  . ; other data that exists in this segment  . I IBPC=1 S IBOK=0 D  .. S Z=1 F  S Z=$O(^TMP("IBXDATA",$J,1,IBSEQ,1,Z)) Q:'Z  I $G(^(Z))'="" S IBOK=1 Q  .. I IBOK Q   ; data exists so include segment normally  .. S SN=$P($G(^TMP("IBXDATA",$J,1,IBSEQ,1,1)),U,1) ; segment name  .. I SN="" S SN=$P($P(IBREC,U,10),"'",2)  .. S SN=SN\_" (No Data - Record Not Sent)"  .. S $P(^TMP("IBXDATA",$J,1,IBSEQ,1,1),U,1)=SN  .. Q  . ;  . ; loop thru all multiple occurrences of this segment  . S IBMULT=0 F  S IBMULT=$O(^TMP("IBXDATA",$J,1,IBSEQ,IBMULT)) Q:'IBMULT   D  .. ;  .. ; field with no data; check user preference  .. I '$G(IBINC),$P($G(^TMP("IBXDATA",$J,1,IBSEQ,IBMULT,IBPC)),U,1)="" Q  .. ;  .. ; build display data  .. S PCD="["\_IBPC\_"] "      ; piece#  .. S DSP=$P(IBREC,U,10) ; short description field  .. S IBDATA=$P($G(^TMP("IBXDATA",$J,1,IBSEQ,IBMULT,IBPC)),U,1) ; data  .. S DSP=$J(PCD,5)\_$$FO^IBCNEUT1(DSP,40)\_": "\_IBDATA  .. S ^TMP($J,"IBLINES",IBSEQ,IBMULT,IBPC)=DSP  .. Q  . Q  ;  S IBQUIT=0  W:$E(IOST,1,2)["C-" @IOF ; initial form feed for screen print  N IBFMTYP S IBFMTYP=$$FT^IBCEF(IBIEN)  S IBFMTYP=$S(IBFMTYP=2:"CMS-1500",IBFMTYP=3:"UB-04",1:"OTHER"\_"("\_IBFMTYP\_")")  S IBILL=$S($$INPAT^IBCEF(IBIEN,1):"Inpt",1:"Oupt")\_"/"\_IBFMTYP  D HDR  S Z=0 F  S Z=$O(^TMP($J,"IBLINES",Z)) Q:'Z!IBQUIT  S Z0=0 F  S Z0=$O(^TMP($J,"IBLINES",Z,Z0)) Q:'Z0!IBQUIT  S Z1=0 F  S Z1=$O(^TMP($J,"IBLINES",Z,Z0,Z1)) Q:'Z1!IBQUIT  D  Q:IBQUIT  . I IBLINE>(IOSL-3) D HDR Q:IBQUIT  . W !,^TMP($J,"IBLINES",Z,Z0,Z1)  . S IBLINE=IBLINE+1  . I IBLINE>(IOSL-3) D HDR Q:IBQUIT  . ;  . ; end of segment add an extra line feed  . I '$O(^TMP($J,"IBLINES",Z,Z0,Z1)) W ! S IBLINE=IBLINE+1  . Q  ;  K ^TMP($J,"IBLINES")  G EXITQ  ;  ; HDR ; - Report header  N DIR,Y  I IBPG D  Q:IBQUIT  . I $E(IOST,1,2)["C-" K DIR S DIR(0)="E" D ^DIR K DIR S IBQUIT=('Y) Q:IBQUIT  . W @IOF  ;  S IBPG=IBPG+1  W !,?25,"EDI Transmitted Bill Extract Data",!,"Bill #",?11,"Type",?27,"Patient Name",?52,"SSN",?57,$$FMTE^XLFDT(DT),?71,"Page: "\_IBPG  W !,$TR($J("",IOM)," ","=")  W !,$P(IBREC1,U)\_" "\_"("\_IBILL\_")",?27,$P($G(^DPT(+$P(IBREC1,U,2),0)),U),?52,$P($G(^DPT($P(IBREC1,U,2),0)),U,9),!  S IBLINE=6  Q  ; EXITQ ; - clean up and exit  I $E(IOST,1,2)["C-",'$G(IBQUIT) K DIR S DIR(0)="E" W ! D ^DIR K DIR  K ^TMP("IBXERR",$J),^TMP("IBXDATA",$J),IBXERR  D CLEAN^DILF  Q  ; EXTRACT(IBIFN,IBBATCH,IBFORM,IBLOCAL) ; Extracts transmitted form data into global  ; ^TMP("IBXDATA",$J). Errors are in ^TMP("IBXERR",$J,err\_num)=text.  ; IBBATCH = Batch # of bill (if known), otherwise, set to 1. This  ; variable must be > 0 to prevent a new batch from being added  ; IBFORM = the ien of the form in file 353  ; IBLOCAL = 1 if OK to use local form, 0 if not  N IBVNUM,IBL,IBINC,IBSEG  D FORMPRE^IBCFP1  S IBVNUM=$G(IBBATCH)  S IBL=$S('$G(IBLOCAL):IBFORM,1:"") ; No local form ... set = main form  ; Get local form associated with parent, if any  I IBL="" S IBL=$S($P($G(^IBE(353,+IBFORM,2)),U,8):$P(^(2),U,8),1:IBFORM)  D SETUP^IBCE837(1)  D ROUT^IBCFP1(IBFORM,1,IBIFN,0,IBL)  Q  ; INCLUDE(IBSEQ) ; Function to determine if segment should be included or not  N OK,LZ,SEGNAME  S OK=1 ; default is to include it  I '$D(IBSEG) G INCLX     ; if nothing in array, then include all  I '$D(^TMP("IBXDATA",$J,1,IBSEQ)) S OK=0 G INCLX        ; no data there  S LZ=+$O(^TMP("IBXDATA",$J,1,IBSEQ,"")) ; first line# found in data  S SEGNAME=$P($G(^TMP("IBXDATA",$J,1,IBSEQ,LZ,1)),U,1) ; piece 1  S SEGNAME=$$TRIM^XLFSTR(SEGNAME)  I SEGNAME'="",'$D(IBSEG(SEGNAME)) S OK=0 ; don't include INCLX ;  Q OK  ; | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCEXTRP ;ALB/JEH - VIEW/PRINT EDI EXTRACT DATA ;4/22/03 9:59am  ;;2.0;INTEGRATED BILLING;\*\*137,197,211,348,349,377,**592\***\*;21-MAR-94;Build 23  ;;Per VHA Directive 2004-038, this routine should not be modified.  ; EN ; INIT ;  W !!,"This option will display the EDI extract data for a bill.",!  N IBREC1,IBIEN,IBINC,DIC,X,Y,DIR,IB364IEN,IBVNUM,IBSEG,STOP,POP,DTOUT,DUOUT  ;  N DPTNOFZY S DPTNOFZY=1 ; Suppress PATIENT file fuzzy lookups  S DIC="^DGCR(399,",DIC(0)="AEMQ",DIC("S")="I 234[$P(^(0),U,13)" D ^DIC  I Y<1 G EXITQ  S IBIEN=+Y,IBREC1=$G(^DGCR(399,IBIEN,0))  S IB364IEN=$$LAST364^IBCEF4(IBIEN) I +$G(IB364IEN)=0 D  G EXITQ  . W !,"There is no entry in the EDI Transmit Bill file for this bill number."  S IBVNUM=$P($G(^IBA(364,IB364IEN,0)),U,2) I +$G(IBVNUM)=0 D  G EXITQ  . W !!,"There is no batch # for this bill. It has not been transmitted."  S IBVNUM=$P($G(^IBA(364.1,IBVNUM,0)),U)  S DIR("A")="Include Fields With No Data?: ",DIR("B")="NO",DIR(0)="YA"  W ! D ^DIR K DIR  I $D(DTOUT)!$D(DUOUT) G EXITQ  S IBINC=+Y  ;  ; IB\*2\*377 - esg - Ask for specific EDI segments to view  ;  W !  S DIR(0)="SA^A:All EDI Segments;S:Selected EDI Segments"  S DIR("A")="Include (A)ll or (S)elected EDI Segments?: "  S DIR("B")="All EDI Segments"  D ^DIR K DIR  I $D(DTOUT)!$D(DUOUT) G EXITQ  I Y="A" G DEV                    ; all segments, skip to device prompt  ;  W !  K IBSEG  S STOP=0  F  D  Q:STOP  . S DIR(0)="FO^3:4"  . S DIR("A")=" Select EDI Segment"  . I $D(IBSEG) S DIR("A")="Another EDI Segment"  . S DIR("?")="Enter the name of the EDI segment to include."  . D ^DIR K DIR  . I $D(DTOUT)!$D(DUOUT) S STOP=1 Q  . S Y=$$UP^XLFSTR(Y),Y=$$TRIM^XLFSTR(Y) ; uppercase/trim spaces  . I Y="" S STOP=1 Q  . S IBSEG(Y)=""  . Q  I $D(DTOUT)!$D(DUOUT) G EXITQ  ; DEV ; - Select device  N %ZIS,ZTRTN,ZTSAVE,ZTDESC  W !  S %ZIS="QM" D ^%ZIS G:POP EXITQ  I $D(IO("Q")) D  G EXITQ  . S ZTRTN="LIST^IBCEXTRP",ZTDESC="Transmitted Bill Extract Data"  . S ZTSAVE("IB\*")=""  . D ^%ZTLOAD  . W !!,$S($D(ZTSK):"Your task number "\_ZTSK\_" has been queued.",1:"Unable to queue this job.")  .K ZTSK,IO("Q") D HOME^%ZIS  U IO  ; LIST ; - set up array and print data  N IBPG,IBSEQ,IBPC,IBDA,IBREC,IBQUIT,IBILL,IBLINE,IBXDATA,IBERR,IBXERR,Z,Z0,Z1  D EXTRACT(IBIEN,IBVNUM,8,1)  S (IBPG,IBQUIT,IBSEQ,IBPC,IBDA,IBLINE)=0  K ^TMP($J,"IBLINES")  ;IB\*2.0\*211 - rely on form type instead of bill charge type  N IBFMTYP S IBFMTYP=$$FT^IBCEF(IBIEN)  **;JWS;IB\*2.0\*592 - Dental form 7 (J430D)**  S IBFMTYP=$S(IBFMTYP=2:"CMS-1500",IBFMTYP=3:"UB-04",**IBFMTYP=7:"J430D",**1:"OTHER"\_"("\_IBFMTYP\_")")  S IBILL=$S($$INPAT^IBCEF(IBIEN,1):"Inpt",1:"Oupt")\_"/"\_IBFMTYP  ;  I $D(^TMP("IBXERR",$J)) D  G EXITQ  . S IBERR=0 F  S IBERR=$O(^TMP("IBXERR",$J,IBERR)) Q:'IBERR  W !,$G(^TMP("IBXERR",$J,IBERR))  . Q  ;  F  S IBSEQ=$O(^IBA(364.6,"ASEQ",8,IBSEQ)) Q:'IBSEQ  I $$INCLUDE(IBSEQ) F  S IBPC=$O(^IBA(364.6,"ASEQ",8,IBSEQ,1,IBPC)) Q:'IBPC  F  S IBDA=$O(^IBA(364.6,"ASEQ",8,IBSEQ,1,IBPC,IBDA)) Q:'IBDA  D  . N IBOK,Z,IBMULT,DSP,IBDATA,PCD,SN  . S IBREC=$G(^IBA(364.6,IBDA,0))  . I $P(IBREC,U,11)=1 Q     ; calculate only field  . ;  . ; processing for piece 1 of this EDI segment to see if there is any  . ; other data that exists in this segment  . I IBPC=1 S IBOK=0 D  .. S Z=1 F  S Z=$O(^TMP("IBXDATA",$J,1,IBSEQ,1,Z)) Q:'Z  I $G(^(Z))'="" S IBOK=1 Q  .. I IBOK Q   ; data exists so include segment normally  .. S SN=$P($G(^TMP("IBXDATA",$J,1,IBSEQ,1,1)),U,1) ; segment name  .. I SN="" S SN=$P($P(IBREC,U,10),"'",2)  .. S SN=SN\_" (No Data - Record Not Sent)"  .. S $P(^TMP("IBXDATA",$J,1,IBSEQ,1,1),U,1)=SN  .. Q  . ;  . ; loop thru all multiple occurrences of this segment  . S IBMULT=0 F  S IBMULT=$O(^TMP("IBXDATA",$J,1,IBSEQ,IBMULT)) Q:'IBMULT   D  .. ;  .. ; field with no data; check user preference  .. I '$G(IBINC),$P($G(^TMP("IBXDATA",$J,1,IBSEQ,IBMULT,IBPC)),U,1)="" Q  .. ;  .. ; build display data  .. S PCD="["\_IBPC\_"] "      ; piece#  .. S DSP=$P(IBREC,U,10) ; short description field  .. S IBDATA=$P($G(^TMP("IBXDATA",$J,1,IBSEQ,IBMULT,IBPC)),U,1) ; data  .. S DSP=$J(PCD,5)\_$$FO^IBCNEUT1(DSP,40)\_": "\_IBDATA  .. S ^TMP($J,"IBLINES",IBSEQ,IBMULT,IBPC)=DSP  .. Q  . Q  ;  S IBQUIT=0  W:$E(IOST,1,2)["C-" @IOF ; initial form feed for screen print  N IBFMTYP S IBFMTYP=$$FT^IBCEF(IBIEN) **;JWS;IB\*2.0\*592 - Dental form 7 (J430D)**  S IBFMTYP=$S(IBFMTYP=2:"CMS-1500",IBFMTYP=3:"UB-04",**IBFMTYP=7:"J430D",**1:"OTHER"\_"("\_IBFMTYP\_")")  S IBILL=$S($$INPAT^IBCEF(IBIEN,1):"Inpt",1:"Oupt")\_"/"\_IBFMTYP  D HDR  S Z=0 F  S Z=$O(^TMP($J,"IBLINES",Z)) Q:'Z!IBQUIT  S Z0=0 F  S Z0=$O(^TMP($J,"IBLINES",Z,Z0)) Q:'Z0!IBQUIT  S Z1=0 F  S Z1=$O(^TMP($J,"IBLINES",Z,Z0,Z1)) Q:'Z1!IBQUIT  D  Q:IBQUIT  . I IBLINE>(IOSL-3) D HDR Q:IBQUIT  . W !,^TMP($J,"IBLINES",Z,Z0,Z1)  . S IBLINE=IBLINE+1  . I IBLINE>(IOSL-3) D HDR Q:IBQUIT  . ;  . ; end of segment add an extra line feed  . I '$O(^TMP($J,"IBLINES",Z,Z0,Z1)) W ! S IBLINE=IBLINE+1  . Q  ;  K ^TMP($J,"IBLINES")  G EXITQ  ;  ; HDR ; - Report header  N DIR,Y  I IBPG D  Q:IBQUIT  . I $E(IOST,1,2)["C-" K DIR S DIR(0)="E" D ^DIR K DIR S IBQUIT=('Y) Q:IBQUIT  . W @IOF  ;  S IBPG=IBPG+1  W !,?25,"EDI Transmitted Bill Extract Data",!,"Bill #",?11,"Type",?27,"Patient Name",?52,"SSN",?57,$$FMTE^XLFDT(DT),?71,"Page: "\_IBPG  W !,$TR($J("",IOM)," ","=")  W !,$P(IBREC1,U)\_" "\_"("\_IBILL\_")",?27,$P($G(^DPT(+$P(IBREC1,U,2),0)),U),?52,$P($G(^DPT($P(IBREC1,U,2),0)),U,9),!  S IBLINE=6  Q  ; EXITQ ; - clean up and exit  I $E(IOST,1,2)["C-",'$G(IBQUIT) K DIR S DIR(0)="E" W ! D ^DIR K DIR  K ^TMP("IBXERR",$J),^TMP("IBXDATA",$J),IBXERR  D CLEAN^DILF  Q  ; EXTRACT(IBIFN,IBBATCH,IBFORM,IBLOCAL) ; Extracts transmitted form data into global  ; ^TMP("IBXDATA",$J). Errors are in ^TMP("IBXERR",$J,err\_num)=text.  ; IBBATCH = Batch # of bill (if known), otherwise, set to 1. This  ; variable must be > 0 to prevent a new batch from being added  ; IBFORM = the ien of the form in file 353  ; IBLOCAL = 1 if OK to use local form, 0 if not  N IBVNUM,IBL,IBINC,IBSEG  D FORMPRE^IBCFP1  S IBVNUM=$G(IBBATCH)  S IBL=$S('$G(IBLOCAL):IBFORM,1:"") ; No local form ... set = main form  ; Get local form associated with parent, if any  I IBL="" S IBL=$S($P($G(^IBE(353,+IBFORM,2)),U,8):$P(^(2),U,8),1:IBFORM)  D SETUP^IBCE837(1)  D ROUT^IBCFP1(IBFORM,1,IBIFN,0,IBL)  Q  ; INCLUDE(IBSEQ) ; Function to determine if segment should be included or not  N OK,LZ,SEGNAME  S OK=1 ; default is to include it  I '$D(IBSEG) G INCLX     ; if nothing in array, then include all  I '$D(^TMP("IBXDATA",$J,1,IBSEQ)) S OK=0 G INCLX        ; no data there  S LZ=+$O(^TMP("IBXDATA",$J,1,IBSEQ,"")) ; first line# found in data  S SEGNAME=$P($G(^TMP("IBXDATA",$J,1,IBSEQ,LZ,1)),U,1) ; piece 1  S SEGNAME=$$TRIM^XLFSTR(SEGNAME)  I SEGNAME'="",'$D(IBSEG(SEGNAME)) S OK=0 ; don't include INCLX ;  Q OK  ; | | | | | | | | | |

| **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** | IBCSC10 | | | | | | | | |
| **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** | | | | | | | | | |
| IBCSC10 ;ALB/MJB - MCCR SCREEN 10 (UB-82 BILL SPECIFIC INFO) ;27 MAY 88 10:20  ;;2.0;INTEGRATED BILLING;\*\*432,547,574\*\*;21-MAR-94;Build 12  ;;Per VA Directive 6402, this routine should not be modified.  ;  ;MAP TO DGCRSC8  ;  ; DEM;432 - Moved IBCSC8\* billing screen routines to IBCSC10\* billing screen  ; routines and created a new billing screen 8 routine IBCSC8.  ; EN S IBCUBFT=$$FT^IBCU3(IBIFN) I IBCUBFT=2!(IBCUBFT=7) K IBCUBFT G ^IBCSC10H ; hcfa 1500  I IBCUBFT=3 K IBCUBFT G ^IBCSC102 ; ub-92  ;I $P(^DGCR(399,IBIFN,0),"^",19)=2 G ^IBCSC10H ;hcfa 1500  D ^IBCSCU S IBSR=10,IBSR1="",IBV1="000000000" S:IBV IBV1="111111111" F I="U","U1",0 S IB(I)=$S($D(^DGCR(399,IBIFN,I)):^(I),1:"")  D H^IBCSCU  S Z=1,IBW=1 X IBWW W " Bill Remark : ",$S($P(IB("U1"),U,8)]"":$P(IB("U1"),U,8),1:IBUN)  S IBX="^^^2^9^27^45" F I=4:1:7 S Z=(I-2),IBW=1 X IBWW W " Form Locator ",$P(IBX,U,I),$S($E($P(IBX,U,I),2)="":" : ",1:": "),$S($P(IB("U1"),U,I)]"":$P(IB("U1"),U,I),1:IBUN)  S IBX=91 F I=13,14 S Z=(I-7),IBW=1,IBX=IBX+1 X IBWW W " Form Locator ",IBX,": ",$S($P(IB("U1"),U,I)]"":$P(IB("U1"),U,I),1:IBUN)  S Z=8,IBW=1 X IBWW W " Tx Auth. Code : ",$S($P(IB("U"),U,13)]"":$P(IB("U"),U,13),1:IBUN)  G ^IBCSCP Q Q  ;  ;WCJ;IB\*2.0\*547 ACINTEL(IBINSDAT,IBNEXT) ; build some intelligence in this Alternate ID branching logic called from both screen 10 templates.  ;  ; Input:  ; IBINSDAT - INS DATA node  ; IBNEXT -where to branch if not correct plan  ;  ; Returns - where to branch to  ;  N IBPLAN,IBEPT,IBINSPRF  S IBPLAN=$P(IBINSDAT,U,18)  I IBPLAN=""  Q IBNEXT  S IBPLAN=$G(^IBA(355.3,+IBPLAN,0))  I IBPLAN="" Q IBNEXT  S IBEPT=$P(IBPLAN,U,15)  I IBEPT="" Q IBNEXT  I IBEPT="MX" Q:'$D(^IBE(350.9,1,81,"B")) IBNEXT  ; no Medicare set up in site parameters  I IBEPT'="MX" Q:'$D(^IBE(350.9,1,82,"B")) IBNEXT   ; no commercial set up in site parameters  ; Use form type not charge type 09/07/2016  ;S IBINSPRF=$$INSPRF^IBCEF(IBIFN)  S IBINSPRF=$$FT^IBCEF(+IBIFN)=3 ; set IBINST flag=1 if it is institutional,0 for professional.  ;  ; Institutional  I IBINSPRF=1 Q:'$D(^DIC(36,+IBINSDAT,15,"B")) IBNEXT   ; this insurance company has no institutional set up  ;  ; Professional  I IBINSPRF=0 Q:'$D(^DIC(36,+IBINSDAT,16,"B")) IBNEXT  ; this insurance company has no professional set up  ;  ; now it gets complicated :)  ; there needs to be one set up for this form type in the ins comp file  ; and also set up for Medicare/commercial in the site parameter file  N IBTMPINS,IBTMPSP,IBLOOP,IBFOUND  M IBTMPINS=^DIC(36,+IBINSDAT,$S(IBINSPRF=1:15,1:16),"B")  M IBTMPSP=^IBE(350.9,1,$S(IBEPT="MX":81,1:82),"B")  S IBLOOP="",IBFOUND=0  F  S IBLOOP=$O(IBTMPINS(IBLOOP)) Q:IBLOOP=""  D  Q:IBFOUND  . Q:'$D(IBTMPSP(IBLOOP))  . S IBFOUND=1  I IBFOUND Q ""  Q IBNEXT  ;IBCSC10 | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCSC10 ;ALB/MJB - MCCR SCREEN 10 (UB-82 BILL SPECIFIC INFO) ;27 MAY 88 10:20  ;;2.0;INTEGRATED BILLING;\*\*432,547,574,**592**\*\*;21-MAR-94;Build 12  ;;Per VA Directive 6402, this routine should not be modified.  ;  ;MAP TO DGCRSC8  ;  ; DEM;432 - Moved IBCSC8\* billing screen routines to IBCSC10\* billing screen  ; routines and created a new billing screen 8 routine IBCSC8.  ;  **;JWS;IB\*2.0\*592 US1108 - Dental form 7** EN S IBCUBFT=$$FT^IBCU3(IBIFN) I IBCUBFT=2**!(IBCUBFT=7)** K IBCUBFT G ^IBCSC10H ; hcfa 1500 ;JWS 3/6/17 Dental Form  I IBCUBFT=3 K IBCUBFT G ^IBCSC102 ; ub-92  ;I $P(^DGCR(399,IBIFN,0),"^",19)=2 G ^IBCSC10H ;hcfa 1500  D ^IBCSCU S IBSR=10,IBSR1="",IBV1="000000000" S:IBV IBV1="111111111" F I="U","U1",0 S IB(I)=$S($D(^DGCR(399,IBIFN,I)):^(I),1:"")  D H^IBCSCU  S Z=1,IBW=1 X IBWW W " Bill Remark : ",$S($P(IB("U1"),U,8)]"":$P(IB("U1"),U,8),1:IBUN)  S IBX="^^^2^9^27^45" F I=4:1:7 S Z=(I-2),IBW=1 X IBWW W " Form Locator ",$P(IBX,U,I),$S($E($P(IBX,U,I),2)="":" : ",1:": "),$S($P(IB("U1"),U,I)]"":$P(IB("U1"),U,I),1:IBUN)  S IBX=91 F I=13,14 S Z=(I-7),IBW=1,IBX=IBX+1 X IBWW W " Form Locator ",IBX,": ",$S($P(IB("U1"),U,I)]"":$P(IB("U1"),U,I),1:IBUN)  S Z=8,IBW=1 X IBWW W " Tx Auth. Code : ",$S($P(IB("U"),U,13)]"":$P(IB("U"),U,13),1:IBUN)  G ^IBCSCP Q Q  ;  ;WCJ;IB\*2.0\*547 ACINTEL(IBINSDAT,IBNEXT) ; build some intelligence in this Alternate ID branching logic called from both screen 10 templates.  ;  ; Input:  ; IBINSDAT - INS DATA node  ; IBNEXT -where to branch if not correct plan  ;  ; Returns - where to branch to  ;  N IBPLAN,IBEPT,IBINSPRF  S IBPLAN=$P(IBINSDAT,U,18)  I IBPLAN=""  Q IBNEXT  S IBPLAN=$G(^IBA(355.3,+IBPLAN,0))  I IBPLAN="" Q IBNEXT  S IBEPT=$P(IBPLAN,U,15)  I IBEPT="" Q IBNEXT  I IBEPT="MX" Q:'$D(^IBE(350.9,1,81,"B")) IBNEXT  ; no medicare set up in site parameters  I IBEPT'="MX" Q:'$D(^IBE(350.9,1,82,"B")) IBNEXT   ; no commercial set up in site parameters  ; Use form type not charge type 09/07/2016  ;S IBINSPRF=$$INSPRF^IBCEF(IBIFN)  S IBINSPRF=$$FT^IBCEF(+IBIFN)=3 ; set IBINST flag=1 if it is institutional,0 for professional.  ;  ; Institutional  I IBINSPRF=1 Q:'$D(^DIC(36,+IBINSDAT,15,"B")) IBNEXT   ; this insurance company has no institutional set up  ;  ; Professional  I IBINSPRF=0 Q:'$D(^DIC(36,+IBINSDAT,16,"B")) IBNEXT  ; this insurance company has no professional set up  ;  ; now it gets complicated :)  ; there needs to be one set up for this form type in the ins comp file  ; and also set up for medicare/commercial in the site parameter file  N IBTMPINS,IBTMPSP,IBLOOP,IBFOUND  M IBTMPINS=^DIC(36,+IBINSDAT,$S(IBINSPRF=1:15,1:16),"B")  M IBTMPSP=^IBE(350.9,1,$S(IBEPT="MX":81,1:82),"B")  S IBLOOP="",IBFOUND=0  F  S IBLOOP=$O(IBTMPINS(IBLOOP)) Q:IBLOOP=""  D  Q:IBFOUND  . Q:'$D(IBTMPSP(IBLOOP))  . S IBFOUND=1  I IBFOUND Q ""  Q IBNEXT  ;IBCSC10 | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCSC10H | | | | | | | | |
| **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** | | | | | | | | | |
| IBCSC10H ;ALB/ARH - MCCR SCREEN 10 (BILL SPECIFIC INFO) CMS-1500 ;4/21/92  ;;2.0;INTEGRATED BILLING;\*\*432,488,547\*\*;21-MAR-94;Build 119  ;;Per VA Directive 6402, this routine should not be modified.  ; CMS-1500 screen 10  ;  ; MAP TO DGCRSC8H  ;  ; DEM;432 - Moved IBCSC8\* billing screen routines to IBCSC10\* billing screen  ; routines and created a new billing screen 8 routine IBCSC8.  ; EN ;  N I,IB,Y,Z  D ^IBCSCU  ;  ;WCJ;IB\*2.0\*547  ;S IBSR=10,IBSR1="H",IBV1="000000000" S:IBV IBV1="111111111"  S IBSR=10,IBSR1="H",IBV1="0000000000" S:IBV IBV1="1111111111"  ;F I="U","U1","UF2","UF3","UF32","U2","M","TX",0,"U3" S IB(I)=$G(^DGCR(399,IBIFN,I))  F I="U","U1","UF2","UF3","UF32","U2","M","M2","TX",0,"U3" S IB(I)=$G(^DGCR(399,IBIFN,I))  ;  N IBZ,IBPRV,IBDATE,IBREQ,IBMRASEC,IBZ1,IBZCNT  ;  S IBDATE=$$BDATE^IBACSV(IBIFN) ; Date of service for the bill  S IBPRV=""  D GETPRV^IBCEU(IBIFN,"ALL",.IBPRV)  K IB("PRV")  S IBZ=0 F  S IBZ=$O(IBPRV(IBZ)) Q:'IBZ  I $O(IBPRV(IBZ,0))!$D(IBPRV(IBZ,"NOTOPT")) M IB("PRV",IBZ)=IBPRV(IBZ)  ;  D H^IBCSCU  ;  ; Section 1  S Z=1,IBW=1 X IBWW W " Unable To Work From: " S Y=$P(IB("U"),U,16) X ^DD("DD") W $S(Y'="":Y,1:IBUN)  W !?4,"Unable To Work To : " S Y=$P(IB("U"),U,17) X ^DD("DD") W $S(Y'="":Y,1:IBUN)  ;  ; Section 2  S Z=2,IBW=1 X IBWW I $$INPAT^IBCEF(IBIFN) W " Admitting Dx : " S IBZ=$$ICD9^IBACSV(+IB("U2"),IBDATE) W $S(IBZ'="":$P(IBZ,U)\_" - "\_$P(IBZ,U,3),1:IBUN),!  S IBZCNT=0,IBZ(IBZCNT)=""  I $P(IB("UF3"),U,4)]"" S IBZ(IBZCNT)="P: "\_$P(IB("UF3"),U,4),IBZCNT=IBZCNT+1  I $P(IB("UF3"),U,5)]"" S IBZ(IBZCNT)="S: "\_$P(IB("UF3"),U,5),IBZCNT=IBZCNT+1  I $P(IB("UF3"),U,6)]"" S IBZ(IBZCNT)="T: "\_$P(IB("UF3"),U,6)  S:IBZ(0)="" IBZ(0)=IBUN  W ?4,"ICN/DCN(s) : ",IBZ(0)  F IBZCNT=1:1 Q:'$D(IBZ(IBZCNT)) W !?25,IBZ(IBZCNT)  K IBZ  S IBZ=$$CKPROV^IBCEU(IBIFN,3)  S IBZCNT=0,IBZ(IBZCNT)=""  I $P(IB("U"),U,13)]"" S IBZ(IBZCNT)="P: "\_$P(IB("U"),U,13),IBZCNT=IBZCNT+1  I $P(IB("U2"),U,8)'="" S IBZ(IBZCNT)="S: "\_$P(IB("U2"),U,8),IBZCNT=IBZCNT+1  I $P(IB("U2"),U,9)'="" S IBZ(IBZCNT)="T: "\_$P(IB("U2"),U,9),IBZCNT=IBZCNT+1  I $P(IB("UF32"),U,1)'="" S IBZ(IBZCNT)="P: "\_$P(IB("UF32"),U,1),IBZCNT=IBZCNT+1  I $P(IB("UF32"),U,2)'="" S IBZ(IBZCNT)="S: "\_$P(IB("UF32"),U,2),IBZCNT=IBZCNT+1  I $P(IB("UF32"),U,3)'="" S IBZ(IBZCNT)="T: "\_$P(IB("UF32"),U,3)  S:IBZ(0)="" IBZ(0)=IBUN  W !,?3," Auth/Referral : ",IBZ(0)  F IBZCNT=1:1 Q:'$D(IBZ(IBZCNT)) W !?25,IBZ(IBZCNT)  K IBZ S IBZ=""  ;  ; Section 3  S Z=3,IBW=1 X IBWW  W " Providers : ",$S('$O(IB("PRV",0)):IBU,1:"")  I $D(IB("PRV")) D  ; at least 1 provider found  . N IBQ,A,A1,IBARR,IBTAX,IBNOTAX,IBSPEC,IBNOSPEC  . S IBZ=0  . D DEFSEC^IBCEF74(IBIFN,.IBARR)  . ; PRXM/KJH - Add Taxonomy code to display for patch 343. Moved secondary IDs slightly (below).  . S IBTAX=$$PROVTAX^IBCEF73A(IBIFN,.IBNOTAX)  . S IBSPEC=$$SPECTAX^IBCEF73A(IBIFN,.IBNOSPEC)  . F  S IBZ=$O(IB("PRV",IBZ)) Q:'IBZ  D  .. S IBQ=""  .. W !,?5,"- "  .. S A=$$EXPAND^IBTRE(399.0222,.01,IBZ)  .. I $P($G(IB("PRV",IBZ,1)),U,4)'="" S A1=" ("\_$E($P(IB("PRV",IBZ,1),U,4),1,3)\_")",A=$E(A,1,16-$L(A1))\_A1  .. W $E(A\_$J("",16),1,16),": "  .. I '$P($G(IB("PRV",IBZ,1)),U,3),$P($G(IB("PRV",IBZ,1)),U)="" W IBU Q  .. I $P($G(IB("PRV",IBZ,1)),U)'="" W:'$G(IB("PRV",IBZ)) $E($P(IB("PRV",IBZ,1),U)\_$J("",16),1,16) W:$G(IB("PRV",IBZ)) "(OLD BOX 31 DATA) "\_$P(IB("PRV",IBZ,1),U)  .. I $P($G(IB("PRV",IBZ,1)),U)="",$P($G(IB("PRV",IBZ)),U)'="" W $E($P(IB("PRV",IBZ),U)\_$J("",16),1,16)  .. W " Taxonomy: ",$S($P(IBTAX,U,IBZ)'="":$P(IBTAX,U,IBZ),1:IBU),$S($P(IBSPEC,U,IBZ)'="":" ("\_$P(IBSPEC,U,IBZ)\_")",1:"")  .. F A=1:1:3 I $G(IBARR(IBZ,A))'="" S IBQ=IBQ\_"["\_$E("PST",A)\_"]"\_IBARR(IBZ,A)\_" "  .. I $L(IBQ) W !,?30,$E(IBQ,1,49)  ;  K IB("PRV")  ;  ; Section 4  S Z=4,IBW=1 X IBWW  W " Other Facility (VA/non): " S IBZ=$$EXPAND^IBTRE(399,232,+$P(IB("U2"),U,10))  W $S(IBZ'="":$E(IBZ,1,23),$$PSRV^IBCEU(IBIFN):IBU,1:IBUN)  I IBZ'="" D  . ; PRXM/KJH - Add Taxonomy code to display for patch 343.  . W ?53,"Taxonomy: "  . S IBZ=$$GET1^DIQ(8932.1,+$P(IB("U3"),U,3),"X12 CODE") W $S(IBZ'="":IBZ,1:IBU)  . S IBZ=$$GET1^DIQ(8932.1,+$P(IB("U3"),U,3),"SPECIALTY CODE") W $S(IBZ'="":" ("\_IBZ\_")",1:"")  . Q  ;  ; clia# display - IB patch 320  S (IBZ,IBZ1)=$P(IB("U2"),U,13) ; retrieve CLIA# from database  ;  I IBZ="" D  . NEW CLIAREQ,DEFCLIA,DIE,DA,DR  . S CLIAREQ=$$CLIAREQ^IBCEP8A(IBIFN)  . I 'CLIAREQ S IBZ1=IBUN Q          ; clia# not needed  . S DEFCLIA=$$CLIA^IBCEP8A(IBIFN) ; default clia# for claim  . I DEFCLIA="" S IBZ1=IBU Q         ; no default found  . I $G(IBMDOTCN) K IBMDOTCN S IBZ1=IBU Q     ; user @-deleted clia#  . S IBZ1=DEFCLIA                    ; display and stuff default clia#  . S DIE=399,DA=IBIFN,DR="235///"\_DEFCLIA D ^DIE ; stuff in default  . Q  ;  W !,?4,"Lab CLIA # : ",IBZ1  ;  ; Mammo# display IB patch 320  S (IBZ,IBZ1)=$P(IB("U3"),U,1) ; retrieve mammo# from database  ;  ; If mammo# is there, but should not be, then blank it out  I IBZ'="",'$$XRAY^IBCEP8A(IBIFN) D  . NEW DIE,DA,DR  . S IBZ1=IBUN        ; mammo# not needed  . S DIE=399,DA=IBIFN,DR="242////@" D ^DIE  . Q  ;  I IBZ="" S IBZ1=IBUN  W !?4,"Mammography Cert # : ",IBZ1  ;  ; Section 5  S Z=5,IBW=1 X IBWW  W " Chiropractic Data : " S Y=$P(IB("U3"),U,5) X ^DD("DD") W $S(Y'="":"INITIAL TREATMENT ON "\_Y,1:IBUN)  ;  ; Section 6 -> changed prompt for \*488\* : baa  S Z=6,IBW=1 X IBWW  W " CMS-1500 Box 19 : " S IBZ=$P($G(^DGCR(399,IBIFN,"UF31")),U,3) W $S(IBZ'="":IBZ,1:IBUN)  ;/ Beginning of IB\*2.0\*488 - Moved the following lines of code to IBCSC8 (vd)  ;I $P(IB("U2"),U,14)'="" W !,?4,"Homebound : ",$$EXPAND^IBTRE(399,236,$P(IB("U2"),U,14))  ;I $P(IB("U2"),U,15)'="" W !,?4,"Date Last Seen : ",$$EXPAND^IBTRE(399,237,$P(IB("U2"),U,15))  ;I $P(IB("U2"),U,16)'="" W !,?4,"Spec Prog Indicator: " S IBZ=$$EXPAND^IBTRE(399,238,$P(IB("U2"),U,16)) W $S(IBZ'="":IBZ,$$WNRBILL^IBEFUNC(IBIFN):"31",1:"")  ;/ End of IB\*2.0\*488 (vd)  ;  ; Section 7  S Z=7,IBW=1 X IBWW  W " Billing Provider : "  K IBZ  D GETBP^IBCEF79(IBIFN,"",+$$B^IBCEF79(IBIFN),"CMS-1500 SCREEN 8",.IBZ)  S IBZ=$G(IBZ("CMS-1500 SCREEN 8","NAME"))  W $S(IBZ'="":IBZ,1:IBU) ; billing provider name  W !?3," Taxonomy Code : "  S IBZ=$$GET1^DIQ(8932.1,+$P(IB("U3"),U,11),"X12 CODE") W $S(IBZ'="":IBZ,1:IBU)  S IBZ=$$GET1^DIQ(8932.1,+$P(IB("U3"),U,11),"SPECIALTY CODE") W $S(IBZ'="":" ("\_IBZ\_")",1:"")  ;  ; Section 8  ;WCJ;IB\*2.0\*547  ;Adding ALT PRIMARY IDS and moving sections down to make room  S Z=8,IBW=1 X IBWW  W " Alt Prim Payer ID : "  K IBZ  S IBZCNT=0  I $P(IB("M2"),U,2)]"" S IBZCNT=IBZCNT+1,IBZ(IBZCNT)="P: "\_$P(IB("M2"),U,2)  I $P(IB("M2"),U,4)]"" S IBZCNT=IBZCNT+1,IBZ(IBZCNT)="S: "\_$P(IB("M2"),U,4)  I $P(IB("M2"),U,6)]"" S IBZCNT=IBZCNT+1,IBZ(IBZCNT)="T: "\_$P(IB("M2"),U,6)  I 'IBZCNT W ?23,IBUN  I IBZCNT F IBZ1=1:1:IBZCNT W ?23,IBZ(IBZ1) W:(IBZ1'=IBZCNT) !  K IBZ  ;  ; Section 9  S Z=9,IBW=1 X IBWW  S IBREQ=+$$REQMRA^IBEFUNC(IBIFN) S:IBREQ IBREQ=1  S IBMRASEC=$$MRASEC^IBCEF4(IBIFN)  W " ",$S('IBREQ:"Force To Print? : ",1:"Force MRA Sec Prt? : ")  S IBZ=$$EXTERNAL^DILFD(399,27+IBREQ,,+$P(IB("TX"),U,8+IBREQ))  I IBMRASEC,'$P(IB("TX"),U,8),$P(IB("TX"),U,9) S IBZ="FORCED TO PRINT BY MRA PRIMARY",$P(IB("TX"),U,8)=0  W $S(IBZ'=""&($P(IB("TX"),U,8+IBREQ)'=""):IBZ,'$$TXMT^IBCEF4(IBIFN):"[NOT APPLICABLE - NOT TRANSMITTABLE]",IBREQ:"NO FORCED PRINT",1:IBZ)  ;  ; Section 10  S Z=10,IBW=1 X IBWW  W " Provider ID Maint : (Edit Provider ID information)",!  G ^IBCSCP Q Q  ; WRT1(IBCRED) ; Write credentials mismatch  W !,\*7," \*\*Warning\*\* Credentials differ from those found in NEW PERSON or IB NON VA",!,$J("",14),"BILLING PROVIDER file (",$S(IBCRED="":"none",1:IBCRED),")"  W !,$J("",14),"Changes will print local, but only credentials on file transmit"  Q  ; NSAME(DA) ; Returns 1 if div on bill is not the default billing facility  Q ($P($G(^IBE(350.9,1,0)),U,2)'=$P($G(^DG(40.8,+$P(^DGCR(399,DA,0),U,22),0)),U,7))  ;  ;IBCSC10H | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCSC10H ;ALB/ARH - MCCR SCREEN 10 (BILL SPECIFIC INFO) CMS-1500 ;4/21/92  ;;2.0;INTEGRATED BILLING;\*\*432,488,547**,592**\*\*;21-MAR-94;Build 119  ;;Per VA Directive 6402, this routine should not be modified.  ; CMS-1500 screen 10  ;  ; MAP TO DGCRSC8H  ;  ; DEM;432 - Moved IBCSC8\* billing screen routines to IBCSC10\* billing screen  ; routines and created a new billing screen 8 routine IBCSC8.  ; EN ;  N I,IB,Y,Z  D ^IBCSCU  ;  ;WCJ;IB\*2.0\*547  ;S IBSR=10,IBSR1="H",IBV1="000000000" S:IBV IBV1="111111111"  S IBSR=10,IBSR1="H",IBV1="0000000000" S:IBV IBV1="1111111111" **;JWS;IB\*2.0\*592 US1108 - Dental form 7  I $$FT^IBCU3(IBIFN)=7 S IBV1="1000100010" S:IBV IBV1="11111111"**  ;F I="U","U1","UF2","UF3","UF32","U2","M","TX",0,"U3" S IB(I)=$G(^DGCR(399,IBIFN,I))  F I="U","U1","UF2","UF3","UF32","U2","M","M2","TX",0,"U3" S IB(I)=$G(^DGCR(399,IBIFN,I))  ;  N IBZ,IBPRV,IBDATE,IBREQ,IBMRASEC,IBZ1,IBZCNT  ;  S IBDATE=$$BDATE^IBACSV(IBIFN) ; Date of service for the bill  S IBPRV=""  D GETPRV^IBCEU(IBIFN,"ALL",.IBPRV)  K IB("PRV")  S IBZ=0 F  S IBZ=$O(IBPRV(IBZ)) Q:'IBZ  I $O(IBPRV(IBZ,0))!$D(IBPRV(IBZ,"NOTOPT")) M IB("PRV",IBZ)=IBPRV(IBZ)  ;  D H^IBCSCU  ;  ; Section 1  S Z=1,IBW=1 X IBWW W " Unable To Work From: " S Y=$P(IB("U"),U,16) X ^DD("DD") W $S(Y'="":Y,1:IBUN)  W !?4,"Unable To Work To : " S Y=$P(IB("U"),U,17) X ^DD("DD") W $S(Y'="":Y,1:IBUN)  ;  ; Section 2  S Z=2,IBW=1  X IBWW I $$INPAT^IBCEF(IBIFN) W " Admitting Dx : " S IBZ=$$ICD9^IBACSV(+IB("U2"),IBDATE) W $S(IBZ'="":$P(IBZ,U)\_" - "\_$P(IBZ,U,3),1:IBUN),!  S IBZCNT=0,IBZ(IBZCNT)=""  I $P(IB("UF3"),U,4)]"" S IBZ(IBZCNT)="P: "\_$P(IB("UF3"),U,4),IBZCNT=IBZCNT+1  I $P(IB("UF3"),U,5)]"" S IBZ(IBZCNT)="S: "\_$P(IB("UF3"),U,5),IBZCNT=IBZCNT+1  I $P(IB("UF3"),U,6)]"" S IBZ(IBZCNT)="T: "\_$P(IB("UF3"),U,6)  S:IBZ(0)="" IBZ(0)=IBUN  W ?4,"ICN/DCN(s) : ",IBZ(0)  F IBZCNT=1:1 Q:'$D(IBZ(IBZCNT)) W !?25,IBZ(IBZCNT)  K IBZ  S IBZ=$$CKPROV^IBCEU(IBIFN,3)  S IBZCNT=0,IBZ(IBZCNT)=""  I $P(IB("U"),U,13)]"" S IBZ(IBZCNT)="P: "\_$P(IB("U"),U,13),IBZCNT=IBZCNT+1  I $P(IB("U2"),U,8)'="" S IBZ(IBZCNT)="S: "\_$P(IB("U2"),U,8),IBZCNT=IBZCNT+1  I $P(IB("U2"),U,9)'="" S IBZ(IBZCNT)="T: "\_$P(IB("U2"),U,9),IBZCNT=IBZCNT+1  I $P(IB("UF32"),U,1)'="" S IBZ(IBZCNT)="P: "\_$P(IB("UF32"),U,1),IBZCNT=IBZCNT+1  I $P(IB("UF32"),U,2)'="" S IBZ(IBZCNT)="S: "\_$P(IB("UF32"),U,2),IBZCNT=IBZCNT+1  I $P(IB("UF32"),U,3)'="" S IBZ(IBZCNT)="T: "\_$P(IB("UF32"),U,3)  S:IBZ(0)="" IBZ(0)=IBUN  W !,?3," Auth/Referral : ",IBZ(0)  F IBZCNT=1:1 Q:'$D(IBZ(IBZCNT)) W !?25,IBZ(IBZCNT)  K IBZ S IBZ=""  ;  ; Section 3  S Z=3,IBW=1 X IBWW  W " Providers : ",$S('$O(IB("PRV",0)):IBU,1:"")  I $D(IB("PRV")) D  ; at least 1 provider found  . N IBQ,A,A1,IBARR,IBTAX,IBNOTAX,IBSPEC,IBNOSPEC  . S IBZ=0  . D DEFSEC^IBCEF74(IBIFN,.IBARR)  . ; PRXM/KJH - Add Taxonomy code to display for patch 343. Moved secondary IDs slightly (below).  . S IBTAX=$$PROVTAX^IBCEF73A(IBIFN,.IBNOTAX)  . S IBSPEC=$$SPECTAX^IBCEF73A(IBIFN,.IBNOSPEC)  . F  S IBZ=$O(IB("PRV",IBZ)) Q:'IBZ  D  .. S IBQ=""  .. W !,?5,"- "  .. S A=$$EXPAND^IBTRE(399.0222,.01,IBZ)  .. I $P($G(IB("PRV",IBZ,1)),U,4)'="" S A1=" ("\_$E($P(IB("PRV",IBZ,1),U,4),1,3)\_")",A=$E(A,1,16-$L(A1))\_A1  .. W $E(A\_$J("",16),1,16),": "  .. I '$P($G(IB("PRV",IBZ,1)),U,3),$P($G(IB("PRV",IBZ,1)),U)="" W IBU Q  .. I $P($G(IB("PRV",IBZ,1)),U)'="" W:'$G(IB("PRV",IBZ)) $E($P(IB("PRV",IBZ,1),U)\_$J("",16),1,16) W:$G(IB("PRV",IBZ)) "(OLD BOX 31 DATA) "\_$P(IB("PRV",IBZ,1),U)  .. I $P($G(IB("PRV",IBZ,1)),U)="",$P($G(IB("PRV",IBZ)),U)'="" W $E($P(IB("PRV",IBZ),U)\_$J("",16),1,16)  .. W " Taxonomy: ",$S($P(IBTAX,U,IBZ)'="":$P(IBTAX,U,IBZ),1:IBU),$S($P(IBSPEC,U,IBZ)'="":" ("\_$P(IBSPEC,U,IBZ)\_")",1:"")  .. F A=1:1:3 I $G(IBARR(IBZ,A))'="" S IBQ=IBQ\_"["\_$E("PST",A)\_"]"\_IBARR(IBZ,A)\_" "  .. I $L(IBQ) W !,?30,$E(IBQ,1,49)  ;  K IB("PRV")  ;  ; Section 4  S Z=4,IBW=1 X IBWW  W " Other Facility (VA/non): " S IBZ=$$EXPAND^IBTRE(399,232,+$P(IB("U2"),U,10))  W $S(IBZ'="":$E(IBZ,1,23),$$PSRV^IBCEU(IBIFN):IBU,1:IBUN)  I IBZ'="" D  . ; PRXM/KJH - Add Taxonomy code to display for patch 343.  . W ?53,"Taxonomy: "  . S IBZ=$$GET1^DIQ(8932.1,+$P(IB("U3"),U,3),"X12 CODE") W $S(IBZ'="":IBZ,1:IBU)  . S IBZ=$$GET1^DIQ(8932.1,+$P(IB("U3"),U,3),"SPECIALTY CODE") W $S(IBZ'="":" ("\_IBZ\_")",1:"")  . Q  ;  ; clia# display - IB patch 320  S (IBZ,IBZ1)=$P(IB("U2"),U,13) ; retrieve CLIA# from database  ;  I IBZ="" D  . NEW CLIAREQ,DEFCLIA,DIE,DA,DR  . S CLIAREQ=$$CLIAREQ^IBCEP8A(IBIFN)  . I 'CLIAREQ S IBZ1=IBUN Q          ; clia# not needed  . S DEFCLIA=$$CLIA^IBCEP8A(IBIFN) ; default clia# for claim  . I DEFCLIA="" S IBZ1=IBU Q         ; no default found  . I $G(IBMDOTCN) K IBMDOTCN S IBZ1=IBU Q     ; user @-deleted clia#  . S IBZ1=DEFCLIA                    ; display and stuff default clia#  . S DIE=399,DA=IBIFN,DR="235///"\_DEFCLIA D ^DIE ; stuff in default  . Q  ;  W !,?4,"Lab CLIA # : ",IBZ1  ;  ; Mammo# display IB patch 320  S (IBZ,IBZ1)=$P(IB("U3"),U,1) ; retrieve mammo# from database  ;  ; If mammo# is there, but should not be, then blank it out  I IBZ'="",'$$XRAY^IBCEP8A(IBIFN) D  . NEW DIE,DA,DR  . S IBZ1=IBUN        ; mammo# not needed  . S DIE=399,DA=IBIFN,DR="242////@" D ^DIE  . Q  ;  I IBZ="" S IBZ1=IBUN  W !?4,"Mammography Cert # : ",IBZ1  ;  ; Section 5  S Z=5,IBW=1 X IBWW  W " Chiropractic Data : " S Y=$P(IB("U3"),U,5) X ^DD("DD") W $S(Y'="":"INITIAL TREATMENT ON "\_Y,1:IBUN)  ;  ; Section 6 -> changed prompt for \*488\* : baa  S Z=6,IBW=1 X IBWW  **;JWS;IB\*2.0\*592 US1108 - Dental  I $$FT^IBCU3(IBIFN)'=7** W " CMS-1500 Box 19 : " S IBZ=$P($G(^DGCR(399,IBIFN,"UF31")),U,3) W $S(IBZ'="":IBZ,1:IBUN)  **E  W " Dental Claim Note : " S IBZ=$$GET1^DIQ(399,IBIFN\_",",97) W $S(IBZ'="":IBZ,1:IBUN)  ;end - JWS;IB\*2.0\*592 US1108 - Dental**  ;/ Beginning of IB\*2.0\*488 - Moved the following lines of code to IBCSC8 (vd)  ;I $P(IB("U2"),U,14)'="" W !,?4,"Homebound : ",$$EXPAND^IBTRE(399,236,$P(IB("U2"),U,14))  ;I $P(IB("U2"),U,15)'="" W !,?4,"Date Last Seen : ",$$EXPAND^IBTRE(399,237,$P(IB("U2"),U,15))  ;I $P(IB("U2"),U,16)'="" W !,?4,"Spec Prog Indicator: " S IBZ=$$EXPAND^IBTRE(399,238,$P(IB("U2"),U,16)) W $S(IBZ'="":IBZ,$$WNRBILL^IBEFUNC(IBIFN):"31",1:"")  ;/ End of IB\*2.0\*488 (vd)  ;  ; Section 7  S Z=7,IBW=1 X IBWW  W " Billing Provider : "  K IBZ  D GETBP^IBCEF79(IBIFN,"",+$$B^IBCEF79(IBIFN),"CMS-1500 SCREEN 8",.IBZ)  S IBZ=$G(IBZ("CMS-1500 SCREEN 8","NAME"))  W $S(IBZ'="":IBZ,1:IBU) ; billing provider name  W !?3," Taxonomy Code : "  S IBZ=$$GET1^DIQ(8932.1,+$P(IB("U3"),U,11),"X12 CODE") W $S(IBZ'="":IBZ,1:IBU)  S IBZ=$$GET1^DIQ(8932.1,+$P(IB("U3"),U,11),"SPECIALTY CODE") W $S(IBZ'="":" ("\_IBZ\_")",1:"")  ;  ; Section 8  ;WCJ;IB\*2.0\*5471  ;Adding ALT PRIMARY IDS and moving sections down to make room  S Z=8,IBW=1 X IBWW  W " Alt Prim Payer ID : "  K IBZ  S IBZCNT=0  I $P(IB("M2"),U,2)]"" S IBZCNT=IBZCNT+1,IBZ(IBZCNT)="P: "\_$P(IB("M2"),U,2)  I $P(IB("M2"),U,4)]"" S IBZCNT=IBZCNT+1,IBZ(IBZCNT)="S: "\_$P(IB("M2"),U,4)  I $P(IB("M2"),U,6)]"" S IBZCNT=IBZCNT+1,IBZ(IBZCNT)="T: "\_$P(IB("M2"),U,6)  I 'IBZCNT W ?23,IBUN  I IBZCNT F IBZ1=1:1:IBZCNT W ?23,IBZ(IBZ1) W:(IBZ1'=IBZCNT) !  K IBZ  ;  ; Section 9  S Z=9,IBW=1 X IBWW  S IBREQ=+$$REQMRA^IBEFUNC(IBIFN) S:IBREQ IBREQ=1  S IBMRASEC=$$MRASEC^IBCEF4(IBIFN)  W " ",$S('IBREQ:"Force To Print? : ",1:"Force MRA Sec Prt? : ")  S IBZ=$$EXTERNAL^DILFD(399,27+IBREQ,,+$P(IB("TX"),U,8+IBREQ))  I IBMRASEC,'$P(IB("TX"),U,8),$P(IB("TX"),U,9) S IBZ="FORCED TO PRINT BY MRA PRIMARY",$P(IB("TX"),U,8)=0  W $S(IBZ'=""&($P(IB("TX"),U,8+IBREQ)'=""):IBZ,'$$TXMT^IBCEF4(IBIFN):"[NOT APPLICABLE - NOT TRANSMITTABLE]",IBREQ:"NO FORCED PRINT",1:IBZ)  ;  ; Section 10  S Z=10,IBW=1 X IBWW  W " Provider ID Maint : (Edit Provider ID information)",!  G ^IBCSCP Q Q  ; WRT1(IBCRED) ; Write credentials mismatch  W !,\*7," \*\*Warning\*\* Credentials differ from those found in NEW PERSON or IB NON VA",!,$J("",14),"BILLING PROVIDER file (",$S(IBCRED="":"none",1:IBCRED),")"  W !,$J("",14),"Changes will print local, but only credentials on file transmit"  Q  ; NSAME(DA) ; Returns 1 if div on bill is not the default billing facility  Q ($P($G(^IBE(350.9,1,0)),U,2)'=$P($G(^DG(40.8,+$P(^DGCR(399,DA,0),U,22),0)),U,7))  ;  ;IBCSC10H | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCSC3 | | | | | | | | |
| **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** | | | | | | | | | |
| IBCSC3 ;ALB/MJB - MCCR SCREEN 3 (PAYER/MAILING ADDRESS) ;27 MAY 88 10:15  ;;2.0;INTEGRATED BILLING;\*\*8,43,52,80,82,51,137,232,320,377,516\*\*;21-MAR-94;Build 123  ;;Per VA Directive 6402, this routine should not be modified.  ;  ;MAP TO DGCRSC3  ; EN N IB,IBX,IBINS,Y,Z  I $D(DGRVRCAL) D ^IBCU6 K DGRVRCAL  D ^IBCSCU S IBSR=3,IBSR1="",IBV1="000" I IBV S IBV1="111"  D H^IBCSCU  D:$D(^DGCR(399,IBIFN,"AIC")) 3^IBCVA0  D:'$D(^DGCR(399,IBIFN,"AIC")) 123^IBCVA  D POL^IBCNSU41(DFN)  F I=0,"M","M1","U","U2" S IB(I)=$S($D(^DGCR(399,IBIFN,I)):(^(I)),1:"")  S IBOUTP=2,IBINDT=$S(+$G(IB("U")):+IB("U"),1:DT)  ;  S X=" Rate Type : "\_$S($P(IB(0),U,7)']"":IBU,$D(^DGCR(399.3,$P(IB(0),U,7),0)):$P(^(0),U),1:IBUN)  S Z=1,IBW=1 X IBWW W X  I +$P($G(^IBE(350.9,1,1)),U,22) W $J("",(42-$L(X))),"Form Type: ",$P($G(^IBE(353,+$P(IB(0),U,19),0)),U,1)  W !?4,"Responsible: ",$S($P(IB(0),U,11)']"":IBU,$P(IB(0),U,11)="p":"PATIENT",$P(IB(0),U,11)="i":"INSURER",1:"OTHER")  W ?45,"Payer Sequence: " S IBX=$P(IB(0),U,21) W $S(IBX="P":"Primary",IBX="S":"Secondary",IBX="T":"Tertiary",IBX="A":"Patient",1:"")  I $P(IB(0),U,11)="i" D  . W !?4,"Bill Payer : " S X=$G(^DGCR(399,IBIFN,"MP"))  . W $S(+X:$P($G(^DIC(36,+X,0)),U,1),$$MCRWNR^IBEFUNC($$CURR^IBCEF2(IBIFN)):"MRA NEEDED FROM MEDICARE",1:IBU)  . W ?45,"Transmit: " S Z=0,X=$$TXMT^IBCEF4(IBIFN,.Z)  . W $S(X:"Yes",1:"No-"\_$S(Z=1:"Forced to print local",Z=2&($$WNRBILL^IBEFUNC(IBIFN)):"MRA not active",Z=2:"EDI not active",Z=3:"Rate typ transmit off",Z=4:"Ins. co transmit off",Z=5:"Failed RULE #"\_$G(Z(0)),Z=6:"Invalid NDC code type",1:"??"))  I $P(IB(0),U,11)']"" G MAIL  I $P(IB(0),U,11)="p" G MAIL  I $P(IB(0),U,11)="o" W !?4,"Inst. Name : ",$S($P(IB("M"),U,11)']"":IBU,$D(^DIC(4,$P(IB("M"),U,11),0)):$P(^(0),U,1),1:"UNKNOWN INSTITUTION") G MAIL  I $P(IB(0),U,11)="i" I $D(IBDD)>1,$D(^DGCR(399,IBIFN,"AIC")) G SHW  D UP G LST:$D(IBDD)>1 W !?4,"Insurance : NO REIMBURSABLE INSURANCE INFORMATION ON FILE",!?17,"[Add Insurance Information by entering '1' at the prompt below]" G MAIL  ; LST N IBDTIN,IBICT  S IBDTIN=+$G(IB("U")),IBICT=0  W ! D HDR^IBCNS  S I=0 F  S I=$O(IBDD("S",I)) Q:'I  D  Q:IBICT'<5  .S IBX=0 F  S IBX=$O(IBDD("S",I,IBX)) Q:'IBX  S IBINS=$G(IBDD(IBX,0)) I IBINS'="" S IBICT=IBICT+1 D:IBICT<5 D1^IBCNS I IBICT'<5 W !,?1,"\*\*Patient has additional insurance - use ?INS to see the entire list" Q  G MAIL LST1 W !?4,$S($D(^DIC(36,+IBDD(IBX,0),0)):$E($P(^(0),"^",1),1,20),1:"UNKNOWN") S X=$P(IBDD(IBX,0),"^",6) W ?26,$S(X="v":"VETERAN",X="s":"SPOUSE",1:"OTHER") S X=$P(IBDD(IBX,0),"^",16)  S X=$S(+X=1:"PATIENT",+X=2:"SPOUSE",+X=3:"CHILD",+X=8:"EMPLOYEE",+X=11:"ORGAN DONOR",+X=18:"PARENT",+X=15:"PLANTIFF",1:"UNKNOWN")  I X="UNKNOWN" S X1=$S($D(IBDD(IBX,0)):$P(IBDD(IBX,0),"^",6),1:""),X=$S(X1="v":"PATIENT",X1="s":"SPOUSE",1:X)  W ?37,X,?49 S Y=$P(IBDD(IBX,0),"^",8) X ^DD("DD") W Y,?64 S Y=$P(IBDD(IBX,0),"^",4) X ^DD("DD") W Y  Q SHW I $D(IBDD) S I="" F  S I=$O(IBDD(I)) Q:'I  D SHW1 MAIL I $$BUFFER^IBCNBU1(DFN) W !!,?17,"\*\*\* Patient has Insurance Buffer entries \*\*\*"  ;  S IB("M")=$S($D(^DGCR(399,IBIFN,"M")):^("M"),1:""),IB("M1")=$S($D(^DGCR(399,IBIFN,"M1")):^("M1"),1:""),IB(0)=^DGCR(399,IBIFN,0)  S Z=2,IBW=1 W ! X IBWW  N IBRAMS S IBRAMS=4.06  I $$FT^IBCEF(IBIFN)=3 S IBRAMS=4.08  S IB("RAFLAG",1)=$S($P(IB("M"),U,1)="":0,1:$$GET1^DIQ(36,$P(IB("M"),U,1),IBRAMS,"I"))  S IB("RAFLAG",2)=$S($P(IB("M"),U,2)="":0,1:$$GET1^DIQ(36,$P(IB("M"),U,2),IBRAMS,"I"))  S IB("RAFLAG",3)=$S($P(IB("M"),U,3)="":0,1:$$GET1^DIQ(36,$P(IB("M"),U,3),IBRAMS,"I"))  S X=0  I $P(IB("M1"),U,2)="",'IB("RAFLAG",1),$P(IB("M1"),U,3)="",'IB("RAFLAG",2),$P(IB("M1"),U,4)="",'IB("RAFLAG",3) S X=1  W " Billing Provider Secondary IDs: "  I X W IBUN          ; no data found, unspecified not required  I 'X D              ; data found, display below  . W !?5,"Primary Payer: ",$S($P(IB("M1"),U,2)]"":$P(IB("M1"),U,2),IB("RAFLAG",1):"ATT/REND ID",1:"")  . W !?5,"Secondary Payer: ",$S($P(IB("M1"),U,3)]"":$P(IB("M1"),U,3),IB("RAFLAG",2):"ATT/REND ID",1:"")  . W ?46,"Tertiary Payer: ",$S($P(IB("M1"),U,4)]"":$P(IB("M1"),U,4),IB("RAFLAG",3):"ATT/REND ID",1:"")  . Q  ;  S Z=3,IBW=1 W ! X IBWW  W " Mailing Address : "  S X=+$G(^DGCR(399,IBIFN,"MP"))  I 'X,$$MCRWNR^IBEFUNC(+$$CURR^IBCEF2(IBIFN)) S X=+$$CURR^IBCEF2(IBIFN)  I X,+$G(^DIC(36,X,3)) S I=$P(^(3),U,$S($$FT^IBCEF(IBIFN)=2:2,1:4)) W ?56,"Electronic ID: ",$S(I'="":I,1:"<NONE>")  S X="" I IB("M")]"" F I=4:1:9 Q:X]""  S X=$P(IB("M"),"^",I)  I X']"" W !?4,"NO MAILING ADDRESS HAS BEEN SPECIFIED!",?45,$$UP1,!?4,"Send Bill to PAYER listed above." G ENDSCR  S X=IB("M") W !,?4,$S($P(X,"^",4)]"":$P(X,"^",4),1:"'MAIL TO' PERSON/PLACE UNSPECIFIED"),?45,$$UP1  W !?4,$S($P(X,"^",5)]"":$P(X,"^",5),1:"STREET ADDRESS UNSPECIFIED") W:$P(X,"^",6)]"" ", ",$P(X,"^",6)  W ! W:$P(IB("M1"),"^",1)]"" ?4,$P(IB("M1"),"^",1),", "  W ?4,$S($P(X,"^",7)]"":$P(X,"^",7),1:"CITY UNSPECIFIED"),", ",$S($D(^DIC(5,+$P(X,"^",8),0)):$P(^(0),"^",2),1:"STATE UNSPECIFIED")," ",$S($P(X,"^",9)]"":$P(X,"^",9),1:"ZIP UNSPECIFIED")  ; ENDSCR K IBADI,IBDD,IBOUTP,IBINDT,I,X,X1  G ^IBCSCP  ; SHW1 ; Display information for insurance I.  ; MRD;IB\*2.0\*516 - Rearranged some fields to allow more characters  ; to be displayed for Group #, Group Name, Policy #, Insured.  S X=IBDD(I,0),Z=$G(^DIC(36,+X,0))  W !!?4,"Ins ",I,": " W $E($S($P(Z,U,1)'="":$P(Z,U,1),1:IBU),1,16)  I $P(Z,U,2)="N" W ?30,"WILL NOT REIMBURSE"  W ?51,"Whose: ",$S($P(X,"^",6)="v":"VETERAN",$P(X,"^",6)="s":"SPOUSE",1:"OTHER")  W !?4,"Policy #: ",$E($S($P(X,"^",2)]"":$P(X,"^",2),1:IBU),1,34)  W ?51,"Rel to Insd: ",$E(IBIR(I),1,15)  W !?4,"Insured: ",$E($P(X,"^",17),1,35)  W ?51,"Insd Sex: ",$S($D(IBISEX(I)):IBISEX(I),1:IBU)  W !?4,"Grp #: ",$E($S($P(X,"^",3)]"":$P(X,"^",3),1:IBU),1,67)  W !?4,"Grp Nm: ",$E($S($P(X,"^",15)]"":$P(X,"^",15),1:IBU),1,66)  Q  ; UP K IBDD D ALL^IBCNS1(DFN,"IBDD",2,IBINDT,1)  I $D(IBDD("S",.5)) D  ; At least 1 MCR WNR insurance policy exists  . ;try to put correct part (A for institution and B for facility)  . N Z,IBAB  . S IBAB=$S($$FT^IBCEF(IBIFN)=3:"A",1:"B")  . S Z=0 F  S Z=$O(IBDD("S",.5,Z)) Q:'Z  D  .. I $P($G(IBDD(Z,355.3)),U,14)=IBAB S IBDD("S",.1,Z,0)="" K IBDD("S",.5,Z)  Q  ; UP1() ;check if patient has medicare so can print a flag for the user  N IBDD,IBX,IBY S IBY="" D ALL^IBCNS1(DFN,"IBDD",2,IBINDT)  S IBX=0 F  S IBX=$O(IBDD(IBX)) Q:'IBX  I $P($G(IBDD(IBX,355.3)),U,9)=33 S IBY="(Patient has Medicare)"  Q IBY  ;IBCSC3 | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCSC3 ;ALB/MJB - MCCR SCREEN 3 (PAYER/MAILING ADDRESS) ;27 MAY 88 10:15  ;;2.0;INTEGRATED BILLING;\*\*8,43,52,80,82,51,137,232,320,377,516,**592**\*\*;21-MAR-94;Build 123  ;;Per VA Directive 6402, this routine should not be modified.  ;  ;MAP TO DGCRSC3  ; EN N IB,IBX,IBINS,Y,Z  I $D(DGRVRCAL) D ^IBCU6 K DGRVRCAL  D ^IBCSCU S IBSR=3,IBSR1="",IBV1="000" I IBV S IBV1="111"  D H^IBCSCU  D:$D(^DGCR(399,IBIFN,"AIC")) 3^IBCVA0  D:'$D(^DGCR(399,IBIFN,"AIC")) 123^IBCVA  D POL^IBCNSU41(DFN)  F I=0,"M","M1","U","U2" S IB(I)=$S($D(^DGCR(399,IBIFN,I)):(^(I)),1:"")  S IBOUTP=2,IBINDT=$S(+$G(IB("U")):+IB("U"),1:DT)  ;  S X=" Rate Type : "\_$S($P(IB(0),U,7)']"":IBU,$D(^DGCR(399.3,$P(IB(0),U,7),0)):$P(^(0),U),1:IBUN)  S Z=1,IBW=1 X IBWW W X  I +$P($G(^IBE(350.9,1,1)),U,22) W $J("",(42-$L(X))),"Form Type: ",$P($G(^IBE(353,+$P(IB(0),U,19),0)),U,1)  W !?4,"Responsible: ",$S($P(IB(0),U,11)']"":IBU,$P(IB(0),U,11)="p":"PATIENT",$P(IB(0),U,11)="i":"INSURER",1:"OTHER")  W ?45,"Payer Sequence: " S IBX=$P(IB(0),U,21) W $S(IBX="P":"Primary",IBX="S":"Secondary",IBX="T":"Tertiary",IBX="A":"Patient",1:"")  I $P(IB(0),U,11)="i" D  . W !?4,"Bill Payer : " S X=$G(^DGCR(399,IBIFN,"MP"))  . W $S(+X:$P($G(^DIC(36,+X,0)),U,1),$$MCRWNR^IBEFUNC($$CURR^IBCEF2(IBIFN)):"MRA NEEDED FROM MEDICARE",1:IBU)  . W ?45,"Transmit: " S Z=0,X=$$TXMT^IBCEF4(IBIFN,.Z)  . W $S(X:"Yes",1:"No-"\_$S(Z=1:"Forced to print local",Z=2&($$WNRBILL^IBEFUNC(IBIFN)):"MRA not active",Z=2:"EDI not active",Z=3:"Rate typ transmit off",Z=4:"Ins. co transmit off",Z=5:"Failed RULE #"\_$G(Z(0)),Z=6:"Invalid NDC code type",1:"??"))  I $P(IB(0),U,11)']"" G MAIL  I $P(IB(0),U,11)="p" G MAIL  I $P(IB(0),U,11)="o" W !?4,"Inst. Name : ",$S($P(IB("M"),U,11)']"":IBU,$D(^DIC(4,$P(IB("M"),U,11),0)):$P(^(0),U,1),1:"UNKNOWN INSTITUTION") G MAIL  I $P(IB(0),U,11)="i" I $D(IBDD)>1,$D(^DGCR(399,IBIFN,"AIC")) G SHW  D UP G LST:$D(IBDD)>1 W !?4,"Insurance : NO REIMBURSABLE INSURANCE INFORMATION ON FILE",!?17,"[Add Insurance Information by entering '1' at the prompt below]" G MAIL  ; LST N IBDTIN,IBICT  S IBDTIN=+$G(IB("U")),IBICT=0  W ! D HDR^IBCNS  S I=0 F  S I=$O(IBDD("S",I)) Q:'I  D  Q:IBICT'<5  .S IBX=0 F  S IBX=$O(IBDD("S",I,IBX)) Q:'IBX  S IBINS=$G(IBDD(IBX,0)) I IBINS'="" S IBICT=IBICT+1 D:IBICT<5 D1^IBCNS I IBICT'<5 W !,?1,"\*\*Patient has additional insurance - use ?INS to see the entire list" Q  G MAIL LST1 W !?4,$S($D(^DIC(36,+IBDD(IBX,0),0)):$E($P(^(0),"^",1),1,20),1:"UNKNOWN") S X=$P(IBDD(IBX,0),"^",6) W ?26,$S(X="v":"VETERAN",X="s":"SPOUSE",1:"OTHER") S X=$P(IBDD(IBX,0),"^",16)  S X=$S(+X=1:"PATIENT",+X=2:"SPOUSE",+X=3:"CHILD",+X=8:"EMPLOYEE",+X=11:"ORGAN DONOR",+X=18:"PARENT",+X=15:"PLANTIFF",1:"UNKNOWN")  I X="UNKNOWN" S X1=$S($D(IBDD(IBX,0)):$P(IBDD(IBX,0),"^",6),1:""),X=$S(X1="v":"PATIENT",X1="s":"SPOUSE",1:X)  W ?37,X,?49 S Y=$P(IBDD(IBX,0),"^",8) X ^DD("DD") W Y,?64 S Y=$P(IBDD(IBX,0),"^",4) X ^DD("DD") W Y  Q SHW I $D(IBDD) S I="" F  S I=$O(IBDD(I)) Q:'I  D SHW1 MAIL I $$BUFFER^IBCNBU1(DFN) W !!,?17,"\*\*\* Patient has Insurance Buffer entries \*\*\*"  ;  S IB("M")=$S($D(^DGCR(399,IBIFN,"M")):^("M"),1:""),IB("M1")=$S($D(^DGCR(399,IBIFN,"M1")):^("M1"),1:""),IB(0)=^DGCR(399,IBIFN,0)  S Z=2,IBW=1 W ! X IBWW  N IBRAMS S IBRAMS=4.06  I $$FT^IBCEF(IBIFN)=3 S IBRAMS=4.08  S IB("RAFLAG",1)=$S($P(IB("M"),U,1)="":0,1:$$GET1^DIQ(36,$P(IB("M"),U,1),IBRAMS,"I"))  S IB("RAFLAG",2)=$S($P(IB("M"),U,2)="":0,1:$$GET1^DIQ(36,$P(IB("M"),U,2),IBRAMS,"I"))  S IB("RAFLAG",3)=$S($P(IB("M"),U,3)="":0,1:$$GET1^DIQ(36,$P(IB("M"),U,3),IBRAMS,"I"))  S X=0  I $P(IB("M1"),U,2)="",'IB("RAFLAG",1),$P(IB("M1"),U,3)="",'IB("RAFLAG",2),$P(IB("M1"),U,4)="",'IB("RAFLAG",3) S X=1  W " Billing Provider Secondary IDs: "  I X W IBUN          ; no data found, unspecified not required  I 'X D              ; data found, display below  . W !?5,"Primary Payer: ",$S($P(IB("M1"),U,2)]"":$P(IB("M1"),U,2),IB("RAFLAG",1):"ATT/REND ID",1:"")  . W !?5,"Secondary Payer: ",$S($P(IB("M1"),U,3)]"":$P(IB("M1"),U,3),IB("RAFLAG",2):"ATT/REND ID",1:"")  . W ?46,"Tertiary Payer: ",$S($P(IB("M1"),U,4)]"":$P(IB("M1"),U,4),IB("RAFLAG",3):"ATT/REND ID",1:"")  . Q  ;  S Z=3,IBW=1 W ! X IBWW  W " Mailing Address : "  S X=+$G(^DGCR(399,IBIFN,"MP"))  I 'X,$$MCRWNR^IBEFUNC(+$$CURR^IBCEF2(IBIFN)) S X=+$$CURR^IBCEF2(IBIFN)  **;JWS;IB\*2.0\*592 US1108 - Dental form #7**  I X,+$G(^DIC(36,X,3)) S I=$P(^(3),U,$S($$FT^IBCEF(IBIFN)=2:2**,$$FT^IBCEF(IBIFN)=7:15**,1:4)) W ?56,"Electronic ID: ",$S(I'="":I,1:"<NONE>")  S X="" I IB("M")]"" F I=4:1:9 Q:X]""  S X=$P(IB("M"),"^",I)  I X']"" W !?4,"NO MAILING ADDRESS HAS BEEN SPECIFIED!",?45,$$UP1,!?4,"Send Bill to PAYER listed above." G ENDSCR  S X=IB("M") W !,?4,$S($P(X,"^",4)]"":$P(X,"^",4),1:"'MAIL TO' PERSON/PLACE UNSPECIFIED"),?45,$$UP1  W !?4,$S($P(X,"^",5)]"":$P(X,"^",5),1:"STREET ADDRESS UNSPECIFIED") W:$P(X,"^",6)]"" ", ",$P(X,"^",6)  W ! W:$P(IB("M1"),"^",1)]"" ?4,$P(IB("M1"),"^",1),", "  W ?4,$S($P(X,"^",7)]"":$P(X,"^",7),1:"CITY UNSPECIFIED"),", ",$S($D(^DIC(5,+$P(X,"^",8),0)):$P(^(0),"^",2),1:"STATE UNSPECIFIED")," ",$S($P(X,"^",9)]"":$P(X,"^",9),1:"ZIP UNSPECIFIED")  ; ENDSCR K IBADI,IBDD,IBOUTP,IBINDT,I,X,X1  G ^IBCSCP  ; SHW1 ; Display information for insurance I.  ; MRD;IB\*2.0\*516 - Rearranged some fields to allow more characters  ; to be displayed for Group #, Group Name, Policy #, Insured.  S X=IBDD(I,0),Z=$G(^DIC(36,+X,0))  W !!?4,"Ins ",I,": " W $E($S($P(Z,U,1)'="":$P(Z,U,1),1:IBU),1,16)  I $P(Z,U,2)="N" W ?30,"WILL NOT REIMBURSE"  W ?51,"Whose: ",$S($P(X,"^",6)="v":"VETERAN",$P(X,"^",6)="s":"SPOUSE",1:"OTHER")  W !?4,"Policy #: ",$E($S($P(X,"^",2)]"":$P(X,"^",2),1:IBU),1,34)  W ?51,"Rel to Insd: ",$E(IBIR(I),1,15)  W !?4,"Insured: ",$E($P(X,"^",17),1,35)  W ?51,"Insd Sex: ",$S($D(IBISEX(I)):IBISEX(I),1:IBU)  W !?4,"Grp #: ",$E($S($P(X,"^",3)]"":$P(X,"^",3),1:IBU),1,67)  W !?4,"Grp Nm: ",$E($S($P(X,"^",15)]"":$P(X,"^",15),1:IBU),1,66)  Q  ; UP K IBDD D ALL^IBCNS1(DFN,"IBDD",2,IBINDT,1)  I $D(IBDD("S",.5)) D  ; At least 1 MCR WNR insurance policy exists  . ;try to put correct part (A for institution and B for facility)  . N Z,IBAB  . S IBAB=$S($$FT^IBCEF(IBIFN)=3:"A",1:"B")  . S Z=0 F  S Z=$O(IBDD("S",.5,Z)) Q:'Z  D  .. I $P($G(IBDD(Z,355.3)),U,14)=IBAB S IBDD("S",.1,Z,0)="" K IBDD("S",.5,Z)  Q  ; UP1() ;check if patient has medicare so can print a flag for the user  N IBDD,IBX,IBY S IBY="" D ALL^IBCNS1(DFN,"IBDD",2,IBINDT)  S IBX=0 F  S IBX=$O(IBDD(IBX)) Q:'IBX  I $P($G(IBDD(IBX,355.3)),U,9)=33 S IBY="(Patient has Medicare)"  Q IBY  ;IBCSC3 | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCSC5 | | | | | | | | |
| **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** | | | | | | | | | |
| IBCSC5 ;ALB/MJB - MCCR SCREEN 5 (OPT. EOC) ;27 MAY 88 10:15  ;;2.0;INTEGRATED BILLING;\*\*52,125,51,210,266,288,287,309,389,447,461\*\*;21-MAR-94;Build 58  ;;Per VHA Directive 2004-038, this routine should not be modified.  ;  ;MAP TO DGCRSC5  ; EN I $$INPAT^IBCEF(IBIFN) G ^IBCSC4  I $D(IBASKCOD) K IBASKCOD D CODMUL^IBCU7 I $$BILLCPT^IBCRU4(IBIFN) D ASK^IBCU7A(IBIFN) S DGRVRCAL=1  I $D(DGRVRCAL) D ^IBCU6 K DGRVRCAL  L ^DGCR(399,IBIFN):1  D ^IBCSCU S IBSR=5,IBSR1="",IBV1="10000000"\_$S($$FT^IBCEF(IBIFN)'=2:0,1:1) F I="U",0 S IB(I)=$S($D(^DGCR(399,IBIFN,I)):^(I),1:"") S:IBV IBV1="111111111"  D H^IBCSCU  S IBPTF=$P(IB(0),U,8),IBBT=$P(IB(0),"^",4)\_$P(IB(0),"^",5)\_$P(IB(0),"^",6)  D EN4^IBCVA1  S Z=1,IBW=1 X IBWW W " Event Date : " S Y=$P(IB(0),U,3) D DT^DIQ  N IBPOARR,IBDATE  D SET^IBCSC4D(IBIFN,"",.IBPOARR)  S IBDATE=$$BDATE^IBACSV(IBIFN) ; Statement To date  S Z=2,IBW=1 X IBWW W " Prin. Diag.: " S Y=$$DX^IBCSC4(0,IBDATE) W $S(Y'="":$E($P(Y,U,4),1,47)\_" - "\_$P(Y,U,2),$$DXREQ^IBCSC4(IBIFN):IBU,1:IBUN)  F I=1:1:4 S Y=$$DX^IBCSC4(+Y,IBDATE) Q:Y=""  W !?4,"Other Diag.: ",$E($P(Y,U,4),1,47)\_" - "\_$P(Y,U,2)  I +Y S Y=$$DX^IBCSC4(+Y,IBDATE) I +Y W !?4,"\*\*\*There are more diagnoses associated with this bill.\*\*\*" OP S Z=3,IBW=1 X IBWW W " OP Visits : " F I=0:0 S I=$O(^DGCR(399,IBIFN,"OP",I)) Q:'I  S Y=I X ^DD("DD") W:$X>67 !?17 W Y\_", "  S:$D(^DGCR(399,"OP")) DGOPV=1 I '$O(^DGCR(399,IBIFN,"OP",0)) W IBU  W !,?4,"Type : ",$$GET1^DIQ(399,IBIFN\_",",158) ; Added with IB\*2.0\*447 BI  S Z=4,IBW=1 X IBWW W " Cod. Method: ",$S($P(IB(0),U,9)="":IBUN,$P(IB(0),U,9)=9:"ICD",$P(IB(0),U,9)=4:"CPT-4",1:"HCPCS")  D WRT:$D(IBPROC)  S Z=5,IBW=1 X IBWW W " Rx. Refills: " S Y=$$RX I 'Y W IBUN OCC G OCC^IBCSC4  W !?4,"Opt. Code : ",IBUN  G OCC^IBCSC4  Q MORE W !?4,\*7,"\*\*\*There are more procedures associated with this bill.\*\*\*" S I=0  Q WRT ; -write out procedures codes on screen  N IBDATE  S J=0 F I=1:1 S J=$O(IBPROC(J)) Q:'J  D  I I>6 D MORE Q  .S IBDATE=$P(IBPROC(J),U,2) I 'IBDATE S IBDATE=$$BDATE^IBACSV($G(IBIFN))  .S X=$$PRCD^IBCEF1($P(IBPROC(J),U),1,IBDATE)  .I IBPROC(J)["ICD" W !?4,"ICD Code : ",$E($P(X,U,3),1,28)\_" - "\_$P(X,U,2)  .I IBPROC(J)["CPT" W !?4,"CPT Code : " D  .. N Z  .. S Z=$P(X,"^",3)\_" "\_$P(X,"^",2)\_$S($P(IBPROC(J),U,15):"-"\_$$MODLST^IBEFUNC2($P(IBPROC(J),U,15)),1:"")  .. I $L(Z)>40 S Z=" "\_$P(X,"^",2)\_$S($P(IBPROC(J),U,15):"-"\_$$MODLST^IBEFUNC2($P(IBPROC(J),U,15)),1:""),Z=$E($P(X,U,3),1,40-$L(Z))\_Z  .. W Z  .I $P(IB(0),U,19)=2 S Y=+$P(IBPROC(J),U,11) S:+Y Y=+$G(^IBA(362.3,+Y,0)) W ?58,$P($$ICD9^IBACSV(Y,IBDATE),U) S Y=$P(IBPROC(J),U,2) D D^DIQ W ?67,Y Q  .S Y=$P(IBPROC(J),"^",2) D D^DIQ W ?67,Y  Q  ; MOD(IBM,PUNC) ; Returns modifier list from comma delimited ien's in string IBM  ; PUNC = Punctuation to use as first character of output  N IBMOD,Q  S IBMOD=""  F Q=1:1:$L(IBM,",") I $P(IBM,",",Q)'="" S IBMOD=IBMOD\_$S(IBMOD'="":",",1:"")\_$P($$MOD^ICPTMOD($P(IBM,",",Q),"I"),U,2)  I IBMOD'="" S IBMOD=$G(PUNC)\_IBMOD  Q IBMOD  ; PD() ;prints prosthetic device in external form, returns 0 if there are none  N IBX,IBY,IBZ,IBN,X S X=0 S IBX=0 F  S IBX=$O(^IBA(362.5,"AIFN"\_IBIFN,IBX)) Q:'IBX  D  Q:X>5  . S IBY=0 F  S IBY=$O(^IBA(362.5,"AIFN"\_IBIFN,IBX,IBY)) Q:'IBY  S IBZ=$G(^IBA(362.5,IBY,0)) I IBZ'="" D  Q:X>5  .. S X=X+1 I X>5 W !,?17,"\*\*\* There are more Pros. Items associated with this bill.\*\*\*" Q  .. W:X'=1 ! W ?17,$E($P(IBZ,U,5),1,40),?67,$$FMTE^XLFDT(+IBZ)  Q X  ; RX() ;prints RX REFILLS in external form, returns 0 if there are none  N IBX,IBY,IBZ,IBN,X S X=0 S IBX="" F  S IBX=$O(^IBA(362.4,"AIFN"\_IBIFN,IBX)) Q:IBX=""  D  Q:X>5  . S IBY=0 F  S IBY=$O(^IBA(362.4,"AIFN"\_IBIFN,IBX,IBY)) Q:'IBY  S IBZ=$G(^IBA(362.4,IBY,0)) I IBZ'="" D  Q:X>5  .. S X=X+1 I X>5 W !,?17,"\*\*\* There are more Rx. Refills associated with this bill.\*\*\*" Q  ..D ZERO^IBRXUTL(+$P(IBZ,U,4))  .. S IBN=$G(^TMP($J,"IBDRUG",+$P(IBZ,U,4),.01)) W:X'=1 ! W ?17,IBN,?65,$$FMTE^XLFDT(+$P(IBZ,U,3))  K ^TMP($J,"IBDRUG")  Q X  ;  ;IBCSC5 | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCSC5 ;ALB/MJB - MCCR SCREEN 5 (OPT. EOC) ;27 MAY 88 10:15  ;;2.0;INTEGRATED BILLING;\*\*52,125,51,210,266,288,287,309,389,447,461,592\*\*;21-MAR-94;Build 58  ;;Per VHA Directive 2004-038, this routine should not be modified.  ;  ;MAP TO DGCRSC5  ; EN I $$INPAT^IBCEF(IBIFN) G ^IBCSC4  I $D(IBASKCOD) K IBASKCOD D CODMUL^IBCU7 I $$BILLCPT^IBCRU4(IBIFN) D ASK^IBCU7A(IBIFN) S DGRVRCAL=1  I $D(DGRVRCAL) D ^IBCU6 K DGRVRCAL  L ^DGCR(399,IBIFN):1  D ^IBCSCU S IBSR=5,IBSR1="",IBV1="10000000"\_$S($$FT^IBCEF(IBIFN)'=2:0,1:1)  ;JWS;IB\*2.0\*592 US1108 - Dental  I $$FT^IBCEF(IBIFN)=7 S IBV1=1000  F I="U",0 S IB(I)=$S($D(^DGCR(399,IBIFN,I)):^(I),1:"") S:IBV IBV1="111111111"  D H^IBCSCU  S IBPTF=$P(IB(0),U,8),IBBT=$P(IB(0),"^",4)\_$P(IB(0),"^",5)\_$P(IB(0),"^",6)  D EN4^IBCVA1  S Z=1,IBW=1 X IBWW W " Event Date : " S Y=$P(IB(0),U,3) D DT^DIQ  N IBPOARR,IBDATE  D SET^IBCSC4D(IBIFN,"",.IBPOARR)  S IBDATE=$$BDATE^IBACSV(IBIFN) ; Statement To date  S Z=2,IBW=1 X IBWW W " Prin. Diag.: " S Y=$$DX^IBCSC4(0,IBDATE) W $S(Y'="":$E($P(Y,U,4),1,47)\_" - "\_$P(Y,U,2),$$DXREQ^IBCSC4(IBIFN):IBU,1:IBUN)  F I=1:1:4 S Y=$$DX^IBCSC4(+Y,IBDATE) Q:Y=""  W !?4,"Other Diag.: ",$E($P(Y,U,4),1,47)\_" - "\_$P(Y,U,2)  I +Y S Y=$$DX^IBCSC4(+Y,IBDATE) I +Y W !?4,"\*\*\*There are more diagnoses associated with this bill.\*\*\*" OP S Z=3,IBW=1 X IBWW W " OP Visits : " F I=0:0 S I=$O(^DGCR(399,IBIFN,"OP",I)) Q:'I  S Y=I X ^DD("DD") W:$X>67 !?17 W Y\_", "  S:$D(^DGCR(399,"OP")) DGOPV=1 I '$O(^DGCR(399,IBIFN,"OP",0)) W IBU  W !,?4,"Type : ",$$GET1^DIQ(399,IBIFN\_",",158) ; Added with IB\*2.0\*447 BI  S Z=4,IBW=1 X IBWW W " Cod. Method: ",$S($P(IB(0),U,9)="":IBUN,$P(IB(0),U,9)=9:"ICD",$P(IB(0),U,9)=4:"CPT-4",1:"HCPCS")  D WRT:$D(IBPROC)  ;JWS;IB\*2.0\*592 US1108 - Dental  I $$FT^IBCEF(IBIFN)=7 D Q^IBCSC4B G ^IBCSCP  S Z=5,IBW=1 X IBWW W " Rx. Refills: " S Y=$$RX I 'Y W IBUN OCC G OCC^IBCSC4  W !?4,"Opt. Code : ",IBUN  G OCC^IBCSC4  Q MORE W !?4,\*7,"\*\*\*There are more procedures associated with this bill.\*\*\*" S I=0  Q WRT ; -write out procedures codes on screen  N IBDATE  S J=0 F I=1:1 S J=$O(IBPROC(J)) Q:'J  D  I I>6 D MORE Q  .S IBDATE=$P(IBPROC(J),U,2) I 'IBDATE S IBDATE=$$BDATE^IBACSV($G(IBIFN))  .S X=$$PRCD^IBCEF1($P(IBPROC(J),U),1,IBDATE)  .I IBPROC(J)["ICD" W !?4,"ICD Code : ",$E($P(X,U,3),1,28)\_" - "\_$P(X,U,2)  .I IBPROC(J)["CPT" W !?4,"CPT Code : " D  .. N Z  .. S Z=$P(X,"^",3)\_" "\_$P(X,"^",2)\_$S($P(IBPROC(J),U,15):"-"\_$$MODLST^IBEFUNC2($P(IBPROC(J),U,15)),1:"")  .. I $L(Z)>40 S Z=" "\_$P(X,"^",2)\_$S($P(IBPROC(J),U,15):"-"\_$$MODLST^IBEFUNC2($P(IBPROC(J),U,15)),1:""),Z=$E($P(X,U,3),1,40-$L(Z))\_Z  .. W Z  .;JWS;IB\*2.0\*592 US1108 - Dental form #7  .I $P(IB(0),U,19)=2!($P(IB(0),U,19)=7) S Y=+$P(IBPROC(J),U,11) S:+Y Y=+$G(^IBA(362.3,+Y,0)) W ?58,$P($$ICD9^IBACSV(Y,IBDATE),U) S Y=$P(IBPROC(J),U,2) D D^DIQ W ?67,Y Q  .S Y=$P(IBPROC(J),"^",2) D D^DIQ W ?67,Y  Q  ; MOD(IBM,PUNC) ; Returns modifier list from comma delimited ien's in string IBM  ; PUNC = Punctuation to use as first character of output  N IBMOD,Q  S IBMOD=""  F Q=1:1:$L(IBM,",") I $P(IBM,",",Q)'="" S IBMOD=IBMOD\_$S(IBMOD'="":",",1:"")\_$P($$MOD^ICPTMOD($P(IBM,",",Q),"I"),U,2)  I IBMOD'="" S IBMOD=$G(PUNC)\_IBMOD  Q IBMOD  ; PD() ;prints prosthetic device in external form, returns 0 if there are none  N IBX,IBY,IBZ,IBN,X S X=0 S IBX=0 F  S IBX=$O(^IBA(362.5,"AIFN"\_IBIFN,IBX)) Q:'IBX  D  Q:X>5  . S IBY=0 F  S IBY=$O(^IBA(362.5,"AIFN"\_IBIFN,IBX,IBY)) Q:'IBY  S IBZ=$G(^IBA(362.5,IBY,0)) I IBZ'="" D  Q:X>5  .. S X=X+1 I X>5 W !,?17,"\*\*\* There are more Pros. Items associated with this bill.\*\*\*" Q  .. W:X'=1 ! W ?17,$E($P(IBZ,U,5),1,40),?67,$$FMTE^XLFDT(+IBZ)  Q X  ; RX() ;prints RX REFILLS in external form, returns 0 if there are none  N IBX,IBY,IBZ,IBN,X S X=0 S IBX="" F  S IBX=$O(^IBA(362.4,"AIFN"\_IBIFN,IBX)) Q:IBX=""  D  Q:X>5  . S IBY=0 F  S IBY=$O(^IBA(362.4,"AIFN"\_IBIFN,IBX,IBY)) Q:'IBY  S IBZ=$G(^IBA(362.4,IBY,0)) I IBZ'="" D  Q:X>5  .. S X=X+1 I X>5 W !,?17,"\*\*\* There are more Rx. Refills associated with this bill.\*\*\*" Q  ..D ZERO^IBRXUTL(+$P(IBZ,U,4))  .. S IBN=$G(^TMP($J,"IBDRUG",+$P(IBZ,U,4),.01)) W:X'=1 ! W ?17,IBN,?65,$$FMTE^XLFDT(+$P(IBZ,U,3))  K ^TMP($J,"IBDRUG")  Q X  ;  ;IBCSC5 | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCSC8 | | | | | | | | |
| **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** | | | | | | | | | |
| IBCSC8 ;ALB/MJB/AAS - MCCR SCREEN 8 (BILLING - CLAIM INFORMATION SCREEN) ;27 MAY 88 10:15  ;;2.0;INTEGRATED BILLING;\*\*432,447,488\*\*;21-MAR-94;Build 184  ;;Per VHA Directive 2004-038, this routine should not be modified.  ;  ; EN D ^IBCSCU S IBSR=8,IBSR1="" S IB("U2")=$G(^DGCR(399,IBIFN,"U2")),IB("U4")=$G(^DGCR(399,IBIFN,"U4")),IB("U5")=$G(^DGCR(399,IBIFN,"U5")),IB("U6")=$G(^DGCR(399,IBIFN,"U6")),IB("U8")=$G(^DGCR(399,IBIFN,"U8"))  D H^IBCSCU  ; DEM - IBV is set in EDI^IBCB => S IBAC=1,IBV=0 D EN G Q:'IBAC1,EDI  ; IBV=0, or IBV=1 as a flag if field on screen is required  ; or not. <Field #> indicates field is not required.  ; [Field #]  ; Make some sections NOT available for UB04 form  S IBT=$P($G(^DGCR(399,IBIFN,0)),U,19)  ;S IBV1=$S(IBT=3:"001011",IBV:"111111",1:"000000")  S IBV1=$S(IBT=3:"001011111",IBV:"111111111",1:"000000000") ; IB\*2.0\*488 (vd)  ;  S Z=1,IBW=1 X IBWW W " COB Non-Covered Charge Amt: " S X=$P(IB("U4"),U),X2="2$" I X'="" D COMMA^%DTC W X  S Z=2 X IBWW W " Property Casualty Information"  W !,?4,"Claim Number: ",$P(IB("U4"),U,2),?41,"Contact Name: ",$P(IB("U4"),U,9)  W !,?4,"Date of 1st Contact: ",$$FMTE^XLFDT($P(IB("U4"),U,3)),?41,"Contact Phone: ",$P(IB("U4"),U,10)," ",$P(IB("U4"),U,11)  ; Start IB\*2.0\*447 BI  ;S Z=3 X IBWW W " Ambulance Information"  ;W !,?41,"D/O Location: ",$P(IB("U6"),U)  ;W !,?4,"P/U Address1: ",$P(IB("U5"),U,2),?41,"D/O Address1: ",$P(IB("U6"),U,2)  ;W !,?4,"P/U Address2: ",$P(IB("U5"),U,3),?41,"D/O Address2: ",$P(IB("U6"),U,3)  ;W !,?4,"P/U City: ",$P(IB("U5"),U,4),?41,"D/O City: ",$P(IB("U6"),U,4)  ;W !,?4,"P/U State/Zip: " W:$P(IB("U5"),U,5)'="" $P($G(^DIC(5,$P(IB("U5"),U,5),0)),U,2)  ;W:$P(IB("U5"),U,6)]"" "/"\_$P(IB("U5"),U,6)  ;W ?41,"D/O State/Zip: " W:$P(IB("U6"),U,5)'="" $P($G(^DIC(5,$P(IB("U6"),U,5),0)),U,2)  ;W:$P(IB("U6"),U,6)]"" "/"\_$P(IB("U6"),U,6)  ;;W !,?4,"P/U Country/SubDiv: ",$P(IB("U5"),U),?41,"D/O Country/SubDiv: "  S Z=3 X IBWW W " Surgical Codes for Anesthesia Claims"  W !,?4,"Primary Code: " W:$P(IB("U4"),U,7)'="" $P($G(^ICPT($P(IB("U4"),U,7),0)),U)  W ?41,"Secondary Code: " W:$P(IB("U4"),U,8)'="" $P($G(^ICPT($P(IB("U4"),U,8),0)),U)  S Z=4 X IBWW W " Paperwork Attachment Information"  W !,?4,"Report Type: " W:$P(IB("U8"),U,2)'="" $P($G(^IBE(353.3,$P(IB("U8"),U,2),0)),U)  W ?41,"Transmission Method: ",$P(IB("U8"),U,3)  W !,?4,"Attachment Control #: ",$P(IB("U8"),U)  S Z=5 X IBWW W " Disability Start Date: ",$$FMTE^XLFDT($P(IB("U4"),U,4)),?41,"Disability End Date: ",$$FMTE^XLFDT($P(IB("U4"),U,5))  S Z=6 X IBWW W " Assumed Care Date: ",$$FMTE^XLFDT($P(IB("U4"),U,13)),?41,"Relinquished Care Date: ",$$FMTE^XLFDT($P(IB("U4"),U,14))  ; End IB\*2.0\*447 BI  ;  ;/ Beginning of IB\*2.0\*488 - code moved from IBCSC10H (vd)  S Z=7 X IBWW W " Special Program: " I $P(IB("U2"),U,16)'="" S IBZ=$$EXPAND^IBTRE(399,238,$P(IB("U2"),U,16)) W $S(IBZ'="":IBZ,$$WNRBILL^IBEFUNC(IBIFN):"31",1:"")  S Z=8 X IBWW W " Homebound: ",$$EXPAND^IBTRE(399,236,$P(IB("U2"),U,14))  S Z=9 X IBWW W " Date Last Seen: ",$$EXPAND^IBTRE(399,237,$P(IB("U2"),U,15))  ;/ End of IB\*2.0\*488 (vd) REV G ^IBCSCP  ;IBCSC8 | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCSC8 ;ALB/MJB/AAS - MCCR SCREEN 8 (BILLING - CLAIM INFORMATION SCREEN) ;27 MAY 88 10:15  ;;2.0;INTEGRATED BILLING;\*\*432,447,488,577,592\*\*;21-MAR-94;Build 1  ;;Per VHA Directive 2004-038, this routine should not be modified.  ;  ; EN D ^IBCSCU S IBSR=8,IBSR1="" S IB("U2")=$G(^DGCR(399,IBIFN,"U2")),IB("U4")=$G(^DGCR(399,IBIFN,"U4")),IB("U5")=$G(^DGCR(399,IBIFN,"U5")),IB("U6")=$G(^DGCR(399,IBIFN,"U6")),IB("U8")=$G(^DGCR(399,IBIFN,"U8"))  D H^IBCSCU  ; DEM - IBV is set in EDI^IBCB => S IBAC=1,IBV=0 D EN G Q:'IBAC1,EDI  ; IBV=0, or IBV=1 as a flag if field on screen is required  ; or not. <Field #> indicates field is not required.  ; [Field #]  ; Make some sections NOT available for UB04 form  S IBT=$P($G(^DGCR(399,IBIFN,0)),U,19)  ;S IBV1=$S(IBT=3:"001011",IBV:"111111",1:"000000")  ;JWS;IB\*2.0\*592 US1108 - Dental  S IBV1=$S(IBT=3:"001011111",IBT=7:"000",IBV:"111111111",1:"000000000") ; IB\*2.0\*488 (vd)  I IBT=7 D IBTEETH,DENTAL K IBTEETH G REV  ;JWS;IB\*2.0\*592 -end  ;  S Z=1,IBW=1 X IBWW W " COB Non-Covered Charge Amt: " S X=$P(IB("U4"),U),X2="2$" I X'="" D COMMA^%DTC W X  S Z=2 X IBWW W " Property Casualty Information"  ;W !,?4,"Claim Number: ",$P(IB("U4"),U,2),?41,"Contact Name: ",$P(IB("U4"),U,9) ;JRA IB\*2.0\*577 ';'  W !,?4,"Claim Number: ",$P(IB("U4"),U,2) ;JRA IB\*2.0\*577  W !,?4,"Contact Name: ",$P(IB("U4"),U,9) ;JRA IB\*2.0\*577  W !,?4,"Date of 1st Contact: ",$$FMTE^XLFDT($P(IB("U4"),U,3)),?41,"Contact Phone: ",$P(IB("U4"),U,10)," ",$P(IB("U4"),U,11)  ; Start IB\*2.0\*447 BI  ;S Z=3 X IBWW W " Ambulance Information"  ;W !,?41,"D/O Location: ",$P(IB("U6"),U)  ;W !,?4,"P/U Address1: ",$P(IB("U5"),U,2),?41,"D/O Address1: ",$P(IB("U6"),U,2)  ;W !,?4,"P/U Address2: ",$P(IB("U5"),U,3),?41,"D/O Address2: ",$P(IB("U6"),U,3)  ;W !,?4,"P/U City: ",$P(IB("U5"),U,4),?41,"D/O City: ",$P(IB("U6"),U,4)  ;W !,?4,"P/U State/Zip: " W:$P(IB("U5"),U,5)'="" $P($G(^DIC(5,$P(IB("U5"),U,5),0)),U,2)  ;W:$P(IB("U5"),U,6)]"" "/"\_$P(IB("U5"),U,6)  ;W ?41,"D/O State/Zip: " W:$P(IB("U6"),U,5)'="" $P($G(^DIC(5,$P(IB("U6"),U,5),0)),U,2)  ;W:$P(IB("U6"),U,6)]"" "/"\_$P(IB("U6"),U,6)  ;;W !,?4,"P/U Country/SubDiv: ",$P(IB("U5"),U),?41,"D/O Country/SubDiv: "  S Z=3 X IBWW W " Surgical Codes for Anesthesia Claims"  W !,?4,"Primary Code: " W:$P(IB("U4"),U,7)'="" $P($G(^ICPT($P(IB("U4"),U,7),0)),U)  W ?41,"Secondary Code: " W:$P(IB("U4"),U,8)'="" $P($G(^ICPT($P(IB("U4"),U,8),0)),U)  S Z=4 X IBWW W " Paperwork Attachment Information"  W !,?4,"Report Type: " W:$P(IB("U8"),U,2)'="" $P($G(^IBE(353.3,$P(IB("U8"),U,2),0)),U)  W ?41,"Transmission Method: ",$P(IB("U8"),U,3)  W !,?4,"Attachment Control #: ",$P(IB("U8"),U)  S Z=5 X IBWW W " Disability Start Date: ",$$FMTE^XLFDT($P(IB("U4"),U,4)),?41,"Disability End Date: ",$$FMTE^XLFDT($P(IB("U4"),U,5))  S Z=6 X IBWW W " Assumed Care Date: ",$$FMTE^XLFDT($P(IB("U4"),U,13)),?41,"Relinquished Care Date: ",$$FMTE^XLFDT($P(IB("U4"),U,14))  ; End IB\*2.0\*447 BI  ;  ;/ Beginning of IB\*2.0\*488 - code moved from IBCSC10H (vd)  S Z=7 X IBWW W " Special Program: " I $P(IB("U2"),U,16)'="" S IBZ=$$EXPAND^IBTRE(399,238,$P(IB("U2"),U,16)) W $S(IBZ'="":IBZ,$$WNRBILL^IBEFUNC(IBIFN):"31",1:"")  S Z=8 X IBWW W " Homebound: ",$$EXPAND^IBTRE(399,236,$P(IB("U2"),U,14))  S Z=9 X IBWW W " Date Last Seen: ",$$EXPAND^IBTRE(399,237,$P(IB("U2"),U,15))  ;/ End of IB\*2.0\*488 (vd) REV G ^IBCSCP  ;JWS;IB\*2.0\*592 US1108 - Dental IBTEETH ;Create array of teeth status  N TH  K IBTEETH S IBTEETH=0  S IBTEETH(0)=+$P($G(^DGCR(399,IBIFN,"DEN1",0)),U,4)  S TH=0  F  S TH=$O(^DGCR(399,IBIFN,"DEN1",TH)) Q:'TH  S IBTEETH(TH)=$G(^DGCR(399,IBIFN,"DEN1",TH,0))  Q  ; DENTAL ;Dental Information for Form Type 7(J430D)  S IB("DEN")=$G(^DGCR(399,IBIFN,"DEN"))  S Z=1,IBW=1 X IBWW W "Tooth Status"  D WRT:$D(IBTEETH)  S Z=2,IBW=1 X IBWW W "Orthodontic Information"  W !?4,"Banding Date: " I $P(IB("DEN"),U)'="" W $$FMTE^XLFDT($P(IB("DEN"),U),2)  W !?4,"Treatment Indicator: ",$$GET1^DIQ(399,IBIFN\_",",95,"E")  W !?4,"Treatment Months Count: ",$P(IB("DEN"),U,2)  W !?4,"Treatment Months Remaining Count: ",$P(IB("DEN"),U,3)  S Z=3,IBW=1 X IBWW W "Dental Paperwork Attachment"  W !?4,"Report Type: " I $P(IB("U8"),U,2)'="" W $$GET1^DIQ(353.3,$P(IB("U8"),U,2)\_",",.01)," (",$E($$GET1^DIQ(353.3,$P(IB("U8"),U,2)\_",",1),1,18),")"  W ?41,"Attachment Control #: ",$P(IB("U8"),U)  Q  ; WRT ;write out teeth status on screen  N I,J  S J=0 F I=1:1 S J=$O(IBTEETH(J)) Q:'J  D  I I>10 D MORE Q  . W !?4,"Tooth Number: ",$P(IBTEETH(J),U),?41,"Status Code: ",$$GET1^DIQ(399.096,J\_","\_IBIFN\_",",.02)  Q  ; MORE ;  W !?4,"\*\*\*There are more teeth statuses associated with this bill.\*\*\*" S I=0  Q  ;end - JWS;IB\*2.0\*592 US1108 - Dental  ;IBCSC8 | | | | | | | | | |

IB,PATIENT MRA CM XX-XX-XXXX BILL#: K101XXX - Outpat/J430D SCREEN <8>

================================================================================

DENTAL - CLAIM INFORMATION

[1] Tooth Status 🡨 Loop may repeat 35 times

Tooth Number: Status Code:??

This code indicates whether a tooth will be extracted or is missing.

Select from:

E To Be Extracted

M Missing

[2] Orthodontic Information

Banding Date:

Treatment Indicator:

Treatment Months Count:

Treatment Months Remaining Count:

[3] Dental Paperwork Attachment Paperwork

Report Type: ?? Transmission Method:

Attachment Control #:

<RET> to CONTINUE, 1-3 to EDIT, '^N' for screen N, or '^' to QUIT:3

Report Type: ??

This is a Report Type to describe the type of documentation that

will provide additional information for this claim. This

applies to the entire claim.

Choose from: 🡨 Different code set than regular claims

B4 Referral Form

DA Dental Models

DG Diagnostic Report

EB EOB (COB or Medicare Secondary Payor)

OZ Support data for Claim

P6 Periodontal Charts

RB Radiology Films

RR Radiology Reports

Report Type: DA

Transmission Method:

Attachment Control #:

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCSC9 | | | | | | | | |
| **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** | | | | | | | | | |
| IBCSC9 ;ALB/BI - MCCR SCREEN 9 (AMBULANCE INFO) ;11 MAY 2011 10:20  ;;2.0;INTEGRATED BILLING;\*\*52,51,447,473\*\*;11-MAY-2011;Build 29  ;;Per VHA Directive 2004-038, this routine should not be modified.  ; EN ; Main Entry Point  N IBACI,IBACIX,IB,IBT  D ^IBCSCU  S IBT=$P($G(^DGCR(399,IBIFN,0)),U,19)  S IBSR=9,IBSR1="",IBV1=$S(IBT=3:"11",IBV:"11",1:"00")  S IB("U")=$G(^DGCR(399,IBIFN,"U"))  S IB("U1")=$G(^DGCR(399,IBIFN,"U1"))  S IB("U4")=$G(^DGCR(399,IBIFN,"U4"))  S IB("U5")=$G(^DGCR(399,IBIFN,"U5"))  S IB("U6")=$G(^DGCR(399,IBIFN,"U6"))  S IB("U7")=$G(^DGCR(399,IBIFN,"U7"))  S IB("U8")=$G(^DGCR(399,IBIFN,"U8"))  M IB("U9")=^DGCR(399,IBIFN,"U9")  D H^IBCSCU  S Z=1,IBW=1 X IBWW W " Ambulance Transport Data"  W !,?41,"D/O Location: ",$P(IB("U6"),U)  W !,?4,"P/U Address1: ",$P(IB("U5"),U,2),?41,"D/O Address1: ",$P(IB("U6"),U,2)  W !,?4,"P/U Address2: ",$P(IB("U5"),U,3),?41,"D/O Address2: ",$P(IB("U6"),U,3)  W !,?4,"P/U City: ",$P(IB("U5"),U,4),?41,"D/O City: ",$P(IB("U6"),U,4)  W !,?4,"P/U State/Zip: " W:$P(IB("U5"),U,5)'="" $P($G(^DIC(5,$P(IB("U5"),U,5),0)),U,2)  W:$P(IB("U5"),U,6)]"" "/"\_$P(IB("U5"),U,6)  W ?41,"D/O State/Zip: " W:$P(IB("U6"),U,5)'="" $P($G(^DIC(5,$P(IB("U6"),U,5),0)),U,2)  W:$P(IB("U6"),U,6)]"" "/"\_$P(IB("U6"),U,6)  W !,?4,"Patient Weight: ",$P(IB("U7"),U,1),?41,"Transport Distance: ",$P(IB("U7"),U,3)  W !,?4,"Transport Reason: " I $P(IB("U7"),U,2)'="" D IBWP($$GET1^DIQ(353.4,$P(IB("U7"),U,2)\_",",.02),22,55)  W !,?4,"R/T Purpose: " D IBWP($P(IB("U7"),U,4),17,60)  W !,?4,"Stretcher Purpose: " D IBWP($P(IB("U7"),U,5),23,54)  S Z=2,IBW=2 X IBWW W " Ambulance Certification Data"  W !,?4,"Condition Indicator:"  S IBACIX=0  F  S IBACIX=$O(IB("U9",IBACIX)) Q:+IBACIX=0 D  . S IBACI=IB("U9",IBACIX,0)  . W ?25,$$GET1^DIQ(353.5,IBACI\_",",.01)," - ",$$GET1^DIQ(353.5,IBACI\_",",.02),!  K IB("U9")  W !  G ^IBCSCP  Q  ; IBWP(IBX,IBLM,IBRM) ;  K ^UTILITY($J,"W")  N X,Y,DIWF,DIWL,DIWR S X=IBX  S DIWL=1,DIWR=IBRM,DIWF="" D ^DIWP  I $D(^UTILITY($J,"W")) S Y=0 F  S Y=$O(^UTILITY($J,"W",1,Y)) Q:'Y  W:Y>1 !,?(IBLM) W $G(^UTILITY($J,"W",1,Y,0))  K ^UTILITY($J,"W")  Q  ; SCREEN1(DA1) ;  N A,RESPONSE S RESPONSE=0  I +$P($G(^DGCR(399,DA1,"U9",0)),U,4)<5 S RESPONSE=1 Q RESPONSE  S A(1,"F")="!?35",A(1)="Maximum of 5 Condition Indicators allowed"  D EN^DDIOL(.A)  Q RESPONSE  ;IBCSC9 | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCSC9 ;ALB/BI - MCCR SCREEN 9 (AMBULANCE INFO) ;11 MAY 2011 10:20  ;;2.0;INTEGRATED BILLING;\*\*52,51,447,473,577,**592\***\*;11-MAY-2011;Build 1  ;;Per VHA Directive 2004-038, this routine should not be modified.  ; EN ; Main Entry Point  **;JWS;IB\*2.0\*592;skip screen 9 for Dental  I $$FT^IBCEF(IBIFN)=7 G EN^IBCSC10**  N IBACI,IBACIX,IB,IBT  D ^IBCSCU  S IBT=$P($G(^DGCR(399,IBIFN,0)),U,19)  S IBSR=9,IBSR1="",IBV1=$S(IBT=3:"11",IBV:"11",1:"00")  S IB("U")=$G(^DGCR(399,IBIFN,"U"))  S IB("U1")=$G(^DGCR(399,IBIFN,"U1"))  S IB("U4")=$G(^DGCR(399,IBIFN,"U4"))  S IB("U5")=$G(^DGCR(399,IBIFN,"U5"))  S IB("U6")=$G(^DGCR(399,IBIFN,"U6"))  S IB("U7")=$G(^DGCR(399,IBIFN,"U7"))  S IB("U8")=$G(^DGCR(399,IBIFN,"U8"))  M IB("U9")=^DGCR(399,IBIFN,"U9")  D H^IBCSCU  S Z=1,IBW=1 X IBWW W " Ambulance Transport Data"  ;JRA IB\*2.0\*577 Rearrange Field order so that expanded 55 char PU/DO Address1 & PU/DO Address2 can be displayed  ;W !,?41,"D/O Location: ",$P(IB("U6"),U) ;JRA ';' IB\*2.0\*577  ;W !,?4,"P/U Address1: ",$P(IB("U5"),U,2),?41,"D/O Address1: ",$P(IB("U6"),U,2) ;JRA IB\*2.0\*577 ';'  ;W !,?4,"P/U Address2: ",$P(IB("U5"),U,3),?41,"D/O Address2: ",$P(IB("U6"),U,3) ;JRA IB\*2.0\*577 ';'  ;W !,?4,"P/U City: ",$P(IB("U5"),U,4),?41,"D/O City: ",$P(IB("U6"),U,4) ;JRA IB\*2.0\*577 ';'  ;W !,?4,"P/U State/Zip: " W:$P(IB("U5"),U,5)'="" $P($G(^DIC(5,$P(IB("U5"),U,5),0)),U,2) ;JRA IB\*2.0\*577 ';'  ;W:$P(IB("U5"),U,6)]"" "/"\_$P(IB("U5"),U,6) ;JRA IB\*2.0\*577 ';'  ;W ?41,"D/O State/Zip: " W:$P(IB("U6"),U,5)'="" $P($G(^DIC(5,$P(IB("U6"),U,5),0)),U,2) ;JRA IB\*2.0\*577 ';'  W !,?4,"P/U Address1: ",$P(IB("U5"),U,2) ;JRA IB\*2.0\*577  W !,?4,"P/U Address2: ",$P(IB("U5"),U,3) ;JRA IB\*2.0\*577  W !,?4,"P/U City: ",$P(IB("U5"),U,4) ;JRA IB\*2.0\*577  W ?41,"P/U State/Zip: " W:$P(IB("U5"),U,5)'="" $P($G(^DIC(5,$P(IB("U5"),U,5),0)),U,2) ;JRA IB\*2.0\*577  W:$P(IB("U5"),U,6)]"" "/"\_$P(IB("U5"),U,6) ;JRA IB\*2.0\*577  W !,?4,"D/O Location: ",$P(IB("U6"),U) ;JRA IB\*2.0\*577  W !,?4,"D/O Address1: ",$P(IB("U6"),U,2) ;JRA IB\*2.0\*577  W !,?4,"D/O Address2: ",$P(IB("U6"),U,3) ;JRA IB\*2.0\*577  W !,?4,"D/O City: ",$P(IB("U6"),U,4) ;JRA IB\*2.0\*577  W ?41,"D/O State/Zip: " W:$P(IB("U6"),U,5)'="" $P($G(^DIC(5,$P(IB("U6"),U,5),0)),U,2) ;JRA IB\*2.0\*577  W:$P(IB("U6"),U,6)]"" "/"\_$P(IB("U6"),U,6)  W !,?4,"Patient Weight: ",$P(IB("U7"),U,1),?41,"Transport Distance: ",$P(IB("U7"),U,3)  W !,?4,"Transport Reason: " I $P(IB("U7"),U,2)'="" D IBWP($$GET1^DIQ(353.4,$P(IB("U7"),U,2)\_",",.02),22,55)  W !,?4,"R/T Purpose: " D IBWP($P(IB("U7"),U,4),17,60)  W !,?4,"Stretcher Purpose: " D IBWP($P(IB("U7"),U,5),23,54)  S Z=2,IBW=2 X IBWW W " Ambulance Certification Data"  W !,?4,"Condition Indicator:"  S IBACIX=0  F  S IBACIX=$O(IB("U9",IBACIX)) Q:+IBACIX=0 D  . S IBACI=IB("U9",IBACIX,0)  . W ?25,$$GET1^DIQ(353.5,IBACI\_",",.01)," - ",$$GET1^DIQ(353.5,IBACI\_",",.02),!  K IB("U9")  W !  G ^IBCSCP  Q  ; IBWP(IBX,IBLM,IBRM) ;  K ^UTILITY($J,"W")  N X,Y,DIWF,DIWL,DIWR S X=IBX  S DIWL=1,DIWR=IBRM,DIWF="" D ^DIWP  I $D(^UTILITY($J,"W")) S Y=0 F  S Y=$O(^UTILITY($J,"W",1,Y)) Q:'Y  W:Y>1 !,?(IBLM) W $G(^UTILITY($J,"W",1,Y,0))  K ^UTILITY($J,"W")  Q  ; SCREEN1(DA1) ;  N A,RESPONSE S RESPONSE=0  I +$P($G(^DGCR(399,DA1,"U9",0)),U,4)<5 S RESPONSE=1 Q RESPONSE  S A(1,"F")="!?35",A(1)="Maximum of 5 Condition Indicators allowed"  D EN^DDIOL(.A)  Q RESPONSE  ;IBCSC9 | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCSCE | | | | | | | | |
| **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** | | | | | | | | | |
| IBCSCE ;ALB/MRL,MJB - MCCR SCREEN EDITS ;07 JUN 88 14:35  ;;2.0;INTEGRATED BILLING;\*\*52,80,91,106,51,137,236,245,287,349,371,400,432,447,547\*\*;21-MAR-94;Build 119  ;;Per VA Directive 6402, this routine should not be modified.  ;  ;MAP TO DGCRSCE  ; always do procedures last because they are edited upon return to screen routine  I IBDR20["54," S IBDR20=$P(IBDR20,"54,",1)\_$P(IBDR20,"54,",2)\_"54,"  I IBDR20["44," S IBDR20=$P(IBDR20,"44,",1)\_$P(IBDR20,"44,",2)\_"44," LOOP N IBDRLP,IBDRL S IBDRLP=IBDR20 F IBDRL=1:1 S IBDR20=$P(IBDRLP,",",IBDRL) Q:IBDR20=""  D EDIT  Q EDIT N IBQUERY  I (IBDR20["31") D MCCR^IBCNSP2 G ENQ  I (IBDR20["43")!(IBDR20["52") D ^IBCSC4D G ENQ  I (IBDR20["74")!(IBDR20["53") K DR N I D ^IBCOPV S (DA,Y)=IBIFN G TMPL  I (IBDR20["54"),$P($G(^IBE(350.9,1,1)),"^",17) K DR N I D EN1^IBCCPT(.IBQUERY) D CLOSE^IBSDU(.IBQUERY) G TMPL ;  I (IBDR20["55") D ^IBCSC5A G ENQ  I (IBDR20["45")!(IBDR20["56") D ^IBCSC5B G ENQ  I (IBDR20["66")!(IBDR20["76") D EDIT^IBCRBE(IBIFN) D ASKCMB^IBCU65(IBIFN) G ENQ  I IBDR20["102",$$FT^IBCEF(IBIFN)=3 D EN^IBCSC10B G ENQ   ; UB-04 patient reason for visit (screen 10, section 2)  I IBDR20["105",$$FT^IBCEF(IBIFN)=2 D ^IBCSC10A G ENQ     ; cms-1500 chiropractic data (screen 10, section 5)  ;  ;WCJ;IB\*2.0\*547  ;I IBDR20["107",$$FT^IBCEF(IBIFN)=3 D EN1^IBCEP6 G ENQ ; UB-04 provider ID maintenance (screen 10, section 7)  I IBDR20["108",$$FT^IBCEF(IBIFN)=3 D EN1^IBCEP6 G ENQ   ; UB-04 provider ID maintenance (screen 10, section 8)  ;  ;WCJ;IB\*2.0\*547  ;I IBDR20["109",$$FT^IBCEF(IBIFN)=2 D EN1^IBCEP6 G ENQ ; cms-1500 provider ID maintenance (screen 10, section 9)  I IBDR20["110",$$FT^IBCEF(IBIFN)=2 D EN1^IBCEP6 G ENQ   ; cms-1500 provider ID maintenance (screen 10, section 10); not a misprint it is screen \*10 +section which is 110  ;  F Q=1:1:9 I IBDR20[("11"\_Q) D EDIT^IBCSC11 G ENQ     ; IB\*2.0\*447 BI TMPL N IBFLIAE S IBFLIAE=1 ;to invoke EN^DGREGAED from [IB SCREEN1]  S DR="[IB SCREEN"\_IBSR\_IBSR1\_"]",(DA,Y)=IBIFN,DIE="^DGCR(399,"  D ^DIE K DIE,DR,DLAYGO  I (IBDR20["61")!(IBDR20["71") I +$G(DGRVRCAL) D PROC^IBCU7A(IBIFN,1)  ; ENQ ;  K DIE,DR,IBDR1,IBDR20,DGDRD,DGDRS,DGDRS1,DA  Q  ;  ;called by screen 3 (input template) UPDT F IBDD=0:0 S IBDD=$O(^DPT(DFN,.312,IBDD)) Q:IBDD'>0 S IBI1=^DPT(DFN,.312,IBDD,0) I $D(^DIC(36,+IBI1,0)),$P(^(0),"^",2)'="N" S IBDD(+IBI1)=IBI1  F IBAIC=0:0 S IBAIC=$O(^DGCR(399,IBIFN,"AIC",IBAIC)) Q:IBAIC'>0 I $D(IBDD(IBAIC)) F IBI1="I1","I2","I3" I $D(^DGCR(399,IBIFN,IBI1)),+^(IBI1)=IBAIC,^(IBI1)'=IBDD(IBAIC) S ^DGCR(399,IBIFN,IBI1)=IBDD(IBAIC)  K IBAIC,IBDD,IBI1 Q  ;  ;Edit patient's address using DGREGAED API EDADDR(IBDFN) ;  I $G(IBFLIAE)'=1!(IBDFN=0) Q 0  N IBFL S IBFL(1)=1  N X,Y,DIE,DA,DR,DIDEL,DIW,DIEDA,DG,DICR  D EN^DGREGAED(IBDFN,.IBFL)  Q 1  ;IBCSCE | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCSCE ;ALB/MRL,MJB - MCCR SCREEN EDITS ;07 JUN 88 14:35  ;;2.0;INTEGRATED BILLING;\*\*52,80,91,106,51,137,236,245,287,349,371,400,432,447,547,592\*\*;21-MAR-94;Build 119  ;;Per VA Directive 6402, this routine should not be modified.  ;  ;MAP TO DGCRSCE  ; always do procedures last because they are edited upon return to screen routine  I IBDR20["54," S IBDR20=$P(IBDR20,"54,",1)\_$P(IBDR20,"54,",2)\_"54,"  I IBDR20["44," S IBDR20=$P(IBDR20,"44,",1)\_$P(IBDR20,"44,",2)\_"44," LOOP N IBDRLP,IBDRL S IBDRLP=IBDR20 F IBDRL=1:1 S IBDR20=$P(IBDRLP,",",IBDRL) Q:IBDR20=""  D EDIT  Q EDIT N IBQUERY  I (IBDR20["31") D MCCR^IBCNSP2 G ENQ  I (IBDR20["43")!(IBDR20["52") D ^IBCSC4D G ENQ  I (IBDR20["74")!(IBDR20["53") K DR N I D ^IBCOPV S (DA,Y)=IBIFN G TMPL  I (IBDR20["54"),$P($G(^IBE(350.9,1,1)),"^",17) K DR N I D EN1^IBCCPT(.IBQUERY) D CLOSE^IBSDU(.IBQUERY) G TMPL ;  I (IBDR20["55") D ^IBCSC5A G ENQ  I (IBDR20["45")!(IBDR20["56") D ^IBCSC5B G ENQ  I (IBDR20["66")!(IBDR20["76") D EDIT^IBCRBE(IBIFN) D ASKCMB^IBCU65(IBIFN) G ENQ  I IBDR20["102",$$FT^IBCEF(IBIFN)=3 D EN^IBCSC10B G ENQ   ; UB-04 patient reason for visit (screen 10, section 2)  I IBDR20["105",$$FT^IBCEF(IBIFN)=2 D ^IBCSC10A G ENQ     ; cms-1500 chiropractic data (screen 10, section 5)  ;  ;WCJ;IB\*2.0\*547  ;I IBDR20["107",$$FT^IBCEF(IBIFN)=3 D EN1^IBCEP6 G ENQ ; UB-04 provider ID maintenance (screen 10, section 7)  I IBDR20["108",$$FT^IBCEF(IBIFN)=3 D EN1^IBCEP6 G ENQ   ; UB-04 provider ID maintenance (screen 10, section 8)  ;  ;WCJ;IB\*2.0\*547  ;I IBDR20["109",$$FT^IBCEF(IBIFN)=2 D EN1^IBCEP6 G ENQ ; cms-1500 provider ID maintenance (screen 10, section 9)  ;JWS;IB\*2.0\*592 US1108 - Dental form 7  I IBDR20["110",$$FT^IBCEF(IBIFN)=2!($$FT^IBCEF(IBIFN)=7) D EN1^IBCEP6 G ENQ   ; cms-1500 provider ID maintenance (screen 10, section 10); not a misprint it is screen \*10 +section which is 110  ;  F Q=1:1:9 I IBDR20[("11"\_Q) D EDIT^IBCSC11 G ENQ     ; IB\*2.0\*447 BI TMPL N IBFLIAE S IBFLIAE=1 ;to invoke EN^DGREGAED from [IB SCREEN1]  S DR="[IB SCREEN"\_IBSR\_IBSR1\_"]",(DA,Y)=IBIFN,DIE="^DGCR(399,"  D ^DIE K DIE,DR,DLAYGO  I (IBDR20["61")!(IBDR20["71") I +$G(DGRVRCAL) D PROC^IBCU7A(IBIFN,1)  ; ENQ ;  K DIE,DR,IBDR1,IBDR20,DGDRD,DGDRS,DGDRS1,DA  Q  ;  ;called by screen 3 (input template) UPDT F IBDD=0:0 S IBDD=$O(^DPT(DFN,.312,IBDD)) Q:IBDD'>0 S IBI1=^DPT(DFN,.312,IBDD,0) I $D(^DIC(36,+IBI1,0)),$P(^(0),"^",2)'="N" S IBDD(+IBI1)=IBI1  F IBAIC=0:0 S IBAIC=$O(^DGCR(399,IBIFN,"AIC",IBAIC)) Q:IBAIC'>0 I $D(IBDD(IBAIC)) F IBI1="I1","I2","I3" I $D(^DGCR(399,IBIFN,IBI1)),+^(IBI1)=IBAIC,^(IBI1)'=IBDD(IBAIC) S ^DGCR(399,IBIFN,IBI1)=IBDD(IBAIC)  K IBAIC,IBDD,IBI1 Q  ;  ;Edit patient's address using DGREGAED API EDADDR(IBDFN) ;  I $G(IBFLIAE)'=1!(IBDFN=0) Q 0  N IBFL S IBFL(1)=1  N X,Y,DIE,DA,DR,DIDEL,DIW,DIEDA,DG,DICR  D EN^DGREGAED(IBDFN,.IBFL)  Q 1  ;IBCSCE | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCSCU | | | | | | | | |
| **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** | | | | | | | | | |
| IBCSCU ;ALB/MJB - MCCR SCREEN UTILITY ROUTINE ;27 MAY 88 11:09  ;;2.0;INTEGRATED BILLING;\*\*52,51,348,432,447\*\*;21-MAR-94;Build 80  ;;Per VHA Directive 2004-038, this routine should not be modified.  ;  ;MAP TO DGCRSCU  ;  S IBW=1,IBU="UNSPECIFIED",IBUN=IBU\_" [NOT REQUIRED]",IBV=$S($D(IBV):IBV,1:1) D HOME^%ZIS  ;S IBWW1="X ""F Z2=1:1:(Z1-$L(Z)) S Z=Z\_"""" """""" W Z Q"  S (IBVO,IBVI)="" I $S('$D(IOST(0)):1,'$D(^DG(43,1,0)):1,'$P(^DG(43,1,0),"^",36):1,$D(^DG(43,1,"TERM",IOST(0))):1,1:0) G M  ;  I $D(IOST(0)) S X="IOINHI;IOINLOW;IOINORM" D ENDR^%ZISS  I $L(IOINHI),$L(IOINLOW) S IBVI=IOINHI,IBVO=$S(IOINORM]"":IOINORM,1:IBINLOW)  D KILL^%ZISS  ;I $D(^%ZIS(2,IOST(0),7)) S I=^(7) I $L($P(I,"^",1)),$L($P(I,"^",2)) S IBVI=$P(I,"^",1),IBVO=$S($P(I,"^",3)]"":$P(I,"^",3),1:$P(I,"^",2))  ; M ;I $L(IBVI\_IBVO)>4 S X=80 X ^%ZOSF("RM")  S IBWW="W:IBW ! S Z=$S(IBV:""<""\_Z\_"">"",$E(IBV1,Z):""<""\_Z\_"">"",1:""[""\_Z\_""]"") W:$E(Z)=""["" IBVI,Z,IBVO W:$E(Z)'=""["" Z Q"  ;S IBWW="W:IBW ! S Z=$S(IOST=""C-QUME""&($L(IBVI)'=2):Z,IBV:""<""\_Z\_"">"",$E(IBV1,Z):""<""\_Z\_"">"",1:""[""\_Z\_""]"") W:$E(Z)=""["" @IBVI,Z,@IBVO W:$E(Z)'=""["" Z Q"  I $D(IBPAR) S IBV=0,IBVV="00000" Q  S IBBNO=$P(^DGCR(399,IBIFN,0),"^",1)  S IBVV=$S('$$INPAT^IBCEF(IBIFN):"00010100001",1:"00001010001"),X="63266556"      ; IB\*2.0\*447 BI  I $P($G(^IBE(353,+$P($G(^DGCR(399,IBIFN,0)),U,19),2)),U,9)'="",$S($D(^DGCR(399,IBIFN,"I1")):1,1:$P($G(^DGCR(399,IBIFN,"M")),U,11)) S $E(IBVV,11)="0"  Q  ; H ;Screen Header  S L="",$P(L,"=",81)=""  I $D(IBH("HELP")) S X="HELP SCREEN" W @IOF,!?(40-($L(X)\2)),IBVI,X,IBVO,!,L G HQ  ; IB\*2.0\*447 BI Start  S X=$P("DEMOGRAPHIC^EMPLOYMENT^PAYER^EVENT - INPATIENT^EVENT - OUTPATIENT^BILLING - GENERAL^BILLING - GENERAL^BILLING - CLAIM^AMBULANCE^BILLING - SPECIFIC^LOCALLY DEFINED","^",IBSR)\_" INFORMATION",X1="SCREEN <"\_+IBSR\_">"  ; IB\*2.0\*447 BI End  N IB0,IBT S IB0=$G(^DGCR(399,IBIFN,0)),IBT=$P(IB0,U,19),DGINPT=$S($$INPAT^IBCEF(IBIFN):"Inpat",1:"Outpat")  ;  W @IOF                                          ; clear screen  W !,VADM(1) ; name  W " ",$P(VADM(2),"^",2) ; ssn  W " BILL#: ",IBBNO\_" - "\_DGINPT,"/"           ; claim# - type  I IBT=2 W "1500"                                ; form type 2  I IBT=3 W $TR($P($G(^IBE(353,3,0)),U,1),"-") ; form type 3  W ?(80-$L(X1)),X1                               ; screen#  W !,L                                           ; separator line  W !?(40-($L(X)\2)),IBVI,X,IBVO                  ; screen description HQ ;  K L,DGINPT  Q  ; A ;Format Address(es)  N Y F I=IBA1:1:IBA1+2 I $P(IB(IBAD),U,I)]"" S IBA(IBA2)=$P(IB(IBAD),U,I),IBA2=IBA2+2  I IBA2=1 S IBA(1)="STREET ADDRESS UNKNOWN",IBA2=IBA2+2  S J=$S($D(^DIC(5,+$P(IB(IBAD),U,IBA1+4),0)):$P(^(0),U,2),1:""),J(1)=$P(IB(IBAD),U,IBA1+3),J(2)=$P(IB(IBAD),U,IBA1+11),IBA(IBA2)=$S(J(1)]""&(J]""):J(1)\_", "\_J,J(1)]"":J(1),J]"":J,1:"CITY/STATE UNKNOWN")  S Y=$S(IBAD=.11!(IBAD=.121):$P(IB(IBAD),U,IBA1+11),IBAD=.25:$P($G(^DPT(+$G(DFN),.22)),U,6),IBAD=.311:$P($G(^DPT(+$G(DFN),.22)),U,5),1:"") D ZIPOUT^VAFADDR  S IBA(IBA2)=IBA(IBA2)\_" "\_Y F I=0:0 S I=$O(IBA(I)) Q:I=""  S IBA(I)=$E(IBA(I),1,25)  K IBA1,I,J Q | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCSCU ;ALB/MJB - MCCR SCREEN UTILITY ROUTINE ;27 MAY 88 11:09  ;;2.0;INTEGRATED BILLING;\*\*52,51,348,432,447,**592**\*\*;21-MAR-94;Build 80  ;;Per VHA Directive 2004-038, this routine should not be modified.  ;  ;MAP TO DGCRSCU  ;  S IBW=1,IBU="UNSPECIFIED",IBUN=IBU\_" [NOT REQUIRED]",IBV=$S($D(IBV):IBV,1:1) D HOME^%ZIS  ;S IBWW1="X ""F Z2=1:1:(Z1-$L(Z)) S Z=Z\_"""" """""" W Z Q"  S (IBVO,IBVI)="" I $S('$D(IOST(0)):1,'$D(^DG(43,1,0)):1,'$P(^DG(43,1,0),"^",36):1,$D(^DG(43,1,"TERM",IOST(0))):1,1:0) G M  ;  I $D(IOST(0)) S X="IOINHI;IOINLOW;IOINORM" D ENDR^%ZISS  I $L(IOINHI),$L(IOINLOW) S IBVI=IOINHI,IBVO=$S(IOINORM]"":IOINORM,1:IBINLOW)  D KILL^%ZISS  ;I $D(^%ZIS(2,IOST(0),7)) S I=^(7) I $L($P(I,"^",1)),$L($P(I,"^",2)) S IBVI=$P(I,"^",1),IBVO=$S($P(I,"^",3)]"":$P(I,"^",3),1:$P(I,"^",2))  ; M ;I $L(IBVI\_IBVO)>4 S X=80 X ^%ZOSF("RM")  S IBWW="W:IBW ! S Z=$S(IBV:""<""\_Z\_"">"",$E(IBV1,Z):""<""\_Z\_"">"",1:""[""\_Z\_""]"") W:$E(Z)=""["" IBVI,Z,IBVO W:$E(Z)'=""["" Z Q"  ;S IBWW="W:IBW ! S Z=$S(IOST=""C-QUME""&($L(IBVI)'=2):Z,IBV:""<""\_Z\_"">"",$E(IBV1,Z):""<""\_Z\_"">"",1:""[""\_Z\_""]"") W:$E(Z)=""["" @IBVI,Z,@IBVO W:$E(Z)'=""["" Z Q"  I $D(IBPAR) S IBV=0,IBVV="00000" Q  S IBBNO=$P(^DGCR(399,IBIFN,0),"^",1)  S IBVV=$S('$$INPAT^IBCEF(IBIFN):"00010100001",1:"00001010001"),X="63266556"      ; IB\*2.0\*447 BI  **;JWS;IB\*2.0\*592;skip screen 9 for Dental  I $$FT^IBCEF(IBIFN)=7 S IBVV="00010100101"**  I $P($G(^IBE(353,+$P($G(^DGCR(399,IBIFN,0)),U,19),2)),U,9)'="",$S($D(^DGCR(399,IBIFN,"I1")):1,1:$P($G(^DGCR(399,IBIFN,"M")),U,11)) S $E(IBVV,11)="0"  Q  ; H ;Screen Header  S L="",$P(L,"=",81)=""  I $D(IBH("HELP")) S X="HELP SCREEN" W @IOF,!?(40-($L(X)\2)),IBVI,X,IBVO,!,L G HQ  ; IB\*2.0\*447 BI Start  S X=$P("DEMOGRAPHIC^EMPLOYMENT^PAYER^EVENT - INPATIENT^EVENT - OUTPATIENT^BILLING - GENERAL^BILLING - GENERAL^BILLING - CLAIM^AMBULANCE^BILLING - SPECIFIC^LOCALLY DEFINED","^",IBSR)\_" INFORMATION",X1="SCREEN <"\_+IBSR\_">"  **;JWS;IB\*2.0\*592; Dental  I $$FT^IBCEF(IBIFN)=7,IBSR=8 S X="DENTAL - CLAIM INFORMATION"**  ; IB\*2.0\*447 BI End  N IB0,IBT S IB0=$G(^DGCR(399,IBIFN,0)),IBT=$P(IB0,U,19),DGINPT=$S($$INPAT^IBCEF(IBIFN):"Inpat",1:"Outpat")  ;  W @IOF                                          ; clear screen  W !,VADM(1) ; name  W " ",$P(VADM(2),"^",2) ; ssn  W " BILL#: ",IBBNO\_" - "\_DGINPT,"/"           ; claim# - type  I IBT=2 W "1500"                                ; form type 2  I IBT=3 W $TR($P($G(^IBE(353,3,0)),U,1),"-") ; form type 3 **;JWS;IB\*2.0\*592 US1108 - Dental form 7  I IBT=7 W $$GET1^DIQ(353,"7,",.01) ; form type 7 - dental**  W ?(80-$L(X1)),X1                               ; screen#  W !,L                                           ; separator line  W !?(40-($L(X)\2)),IBVI,X,IBVO                  ; screen description HQ ;  K L,DGINPT  Q  ; A ;Format Address(es)  N Y F I=IBA1:1:IBA1+2 I $P(IB(IBAD),U,I)]"" S IBA(IBA2)=$P(IB(IBAD),U,I),IBA2=IBA2+2  I IBA2=1 S IBA(1)="STREET ADDRESS UNKNOWN",IBA2=IBA2+2  S J=$S($D(^DIC(5,+$P(IB(IBAD),U,IBA1+4),0)):$P(^(0),U,2),1:""),J(1)=$P(IB(IBAD),U,IBA1+3),J(2)=$P(IB(IBAD),U,IBA1+11),IBA(IBA2)=$S(J(1)]""&(J]""):J(1)\_", "\_J,J(1)]"":J(1),J]"":J,1:"CITY/STATE UNKNOWN")  S Y=$S(IBAD=.11!(IBAD=.121):$P(IB(IBAD),U,IBA1+11),IBAD=.25:$P($G(^DPT(+$G(DFN),.22)),U,6),IBAD=.311:$P($G(^DPT(+$G(DFN),.22)),U,5),1:"") D ZIPOUT^VAFADDR  S IBA(IBA2)=IBA(IBA2)\_" "\_Y F I=0:0 S I=$O(IBA(I)) Q:I=""  S IBA(I)=$E(IBA(I),1,25)  K IBA1,I,J Q | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCU7 | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** |  | | | | | | | | |
| **Related Options** | None | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  |  | | | | |  | | | |
| **Data Dictionary (DD) References** |  | | | | | | | | |
| **Related Protocols** | None | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Current Logic** | | | | | | | | | |
| IBCU7 ;ALB/AAS - INTERCEPT SCREEN INPUT OF PROCEDURE CODES ;29-OCT-91  ;;2.0;INTEGRATED BILLING;\*\*62,52,106,125,51,137,210,245,228,260,348,371,432,447,488,461,516,522\*\*;21-MAR-94;Build 11  ;;Per VA Directive 6402, this routine should not be modified.  ;  ;MAP TO DGCRU7  ; CHKX ; -interception of input x from Additional Procedure input  G:X=" " CHKXQ  I $$INPAT^IBCEF(DA(1)),'$P($G(^IBE(350.9,1,1)),"^",15),X'?1A1.2N D  G CHKXQ  . K X  . D EN^DDIOL("Site param does not allow entry of non-PTF procedures") ;Fileman error here will be: The previous error occurred when performing an action specified in a Pre-lookup transform (7.5 node).  G:'$D(^UTILITY($J,"IB")) CHKXQ  ;S M=($A($E(X,1))-64),S=+$E(X,2) Q:'$G(^UTILITY($J,"IB",M,S)) S X="`"\_+^(S)  S M=0 I X?1A1.2N S N=$G(^UTILITY($J,"IB","B",X)) S M=+N,S=+$P(N,U,2),P=X S S=$G(^UTILITY($J,"IB",M,S)) I +S S X="`"\_+S I $P(N,U,3)="N" S X=""""\_X\_"""" S $P(^UTILITY($J,"IB","B",P),U,3)="Y"  I +M,$D(DGPROCDT),DGPROCDT'=$P($G(^UTILITY($J,"IB",M,1)),"^",2) S DGPROCDT=$P(^(1),"^",2) W !!,"Procedure Date: " S Y=DGPROCDT X ^DD("DD") W Y,! CHKXQ Q  ; CODMUL ;Date oriented entry of procedure DELASK I $D(IBZ20),IBZ20,IBZ20'=$P(^DGCR(399,IBIFN,0),U,9) S %=2 W !,"SINCE THE PROCEDURE CODING METHOD HAS BEEN CHANGED, DO YOU WANT TO DELETE ALL",!,"PROCEDURE CODES IN THIS BILL"  I  D YN^DICN Q:%=-1 D:%=1 DELADD I %Y?1."?" W !!,"If you answer 'Yes', all procedure codes will be DELETED from this bill.",! G DELASK  K %,%Y,DA,IBZ20,DIK ;W !,"Procedure Entry:"  ; CODDT I $D(IBIFN),$D(^DGCR(399,IBIFN,0)),$P(^(0),U,9) S DIC("V")=$S($P(^(0),U,9)=9:"I +Y(0)=80.1",$P(^(0),U,9)=4!($P(^(0),U,9)=5):"I +Y(0)=81",1:"")  I $P($G(^DGCR(399,IBIFN,0)),"^",5)<3 S IBZTYPE=1 I $P($G(^UTILITY($J,"IB",1,1)),"^",2) S DGPROCDT=$P(^(1),"^",2) D ASKCOD  S X=$$PRCDIV^IBCU71(IBIFN) I +X W !!,$P(X,U,2),!  N Z,Z0 S Z=$G(^DGCR(399,IBIFN,"U")),Z0=$$FMTE^XLFDT($P(Z,U),"2D")\_"-"\_$$FMTE^XLFDT($P(Z,U,2),"2D")  W !,"Select PROCEDURE DATE"\_$S($TR(Z0,"-")'="":" ("\_Z0\_")",1:"")\_": " R X:DTIME G:'$T!("^"[X) CODQ D:X["?" CODHLP  S IBEX=0 D  ; Get procedure date  . I X=" ",$D(DGPROCDT),DGPROCDT?7N S Y=DGPROCDT D D^DIQ W " (",Y,")" Q  . I X=" ",+$P($G(^DGCR(399,IBIFN,"OP",0)),"^",4) S (DGPROCDT,Y)=$O(^DGCR(399,IBIFN,"OP",0)) D D^DIQ W " (",Y,")" Q  . S %DT="EXP",%DT(0)=-DT D ^%DT K %DT I Y<1 S IBEX=1 Q  . I '$$OPV2^IBCU41(Y,IBIFN,1) S IBEX=1 Q  . S:'$G(IBZTYPE) X=$$OPV^IBCU41(Y,IBIFN) S DGPROCDT=Y  I 'IBEX D ASKCOD,ADDCPT^IBCU71:$D(DGCPT)  K IBEX  G CODDT  ; ASKCOD N Z,Z0,DA,IBACT,IBQUIT,IBLNPRV  ;WCJ;2.0\*432  N IBPOPOUT  S IBPOPOUT=0 ; IB\*2.0\*447 BI  K DGCPT  S DGCPT=0,DGCPTUP=$P($G(^IBE(350.9,1,1)),"^",19),DGADDVST=0,IBFT=$P($G(^DGCR(399,IBIFN,0)),"^",19)  I '$D(^DGCR(399,IBIFN,"CP",0)) S ^DGCR(399,IBIFN,"CP",0)=U\_$$GETSPEC^IBEFUNC(399,304)  ;  F  S IBQUIT=0 D  Q:IBQUIT  . S IBPOPOUT=0  . D DICV ; restrict code type to PCM  . S DIC("A")=" Select PROCEDURE: "  . S DIC="^DGCR(399,"\_IBIFN\_",""CP"","  . S DIC(0)="AEQMNL"  . S DIC("S")="I '$D(DIV(""S""))&($P(^(0),U,2)=DGPROCDT)"  . S DIC("DR")="1///^S X=DGPROCDT"  . S DA(1)=IBIFN,DLAYGO=399  . W ! D ^DIC I Y<1 S IBQUIT=1 Q  . S IBPROCP=+Y  . ; If we just added inactive code - it must be deleted.  . S IBACT=0 ; Active flag  . I Y["ICD0" S IBACT=$$ICD0ACT^IBACSV(+$P(Y,U,2),$$BDATE^IBACSV(IBIFN))  . I Y["ICPT" S IBACT=$$CPTACT^IBACSV(+$P(Y,U,2),DGPROCDT)  . S DGCPTNEW=$P(Y,"^",3); Was the procedure just added?  . I DGCPTNEW,'IBACT D DELPROC Q  . I 'IBACT W !,\*7,"Warning: Procedure code is inactive on this date",!  . I DGCPTNEW,$D(^UTILITY($J,"IB")),$$INPAT^IBCEF(IBIFN),Y["ICPT(" D DATA^IBCU74(Y,.IBLNPRV)  . S DGADDVST=$S(DGCPTNEW:1,$D(DGADDVST):DGADDVST,1:0)  . N IBPRV,IBPRVO,IBPRVN  . ;  . ; Line level provider function by form type.  . ; CMS-1500 (FORM TYPE=2)  . ; RENDERING PROVIDER, REFERRING PROVIDER,  . ; and SUPERVISING PROVIDER.  . ; UB-04 (FORM TYPE=3)  . ; RENDERING PROVIDER, REFERRING PROVIDER,  . ; OPERATING PROVIDER, and OTHER OPERATING  . ; PROVIDER.  . ;  . ; Removed: Call to $$MAINPRV^IBCEU(IBIFN) is for claim  . ; level provider defaults.  . ; 1. For new line level providers we don't need  . ; or want default claim level provider  . ; (requirement).  . ; 2. We don't want to default claim level to  . ; line level provider (requirement).  . ;  . K DIC("V") ; DEM;432 - KILL DIC("V") because this was for previous variable pointer use.  . ;  . N IBPROCSV  ; DEM;432 - Variable IBPROCSV is variable to preserve value of 'Y', which is procedure code info returned by call to ^DIC.  . S IBPROCSV=Y  ; DEM;432 - Preserve value of Y for after calls to FileMan (Y = procedure code info returned by call to ^DIC).  . K DR   ;WCJ;IB\*2.0\*432  . ;  . I IBPROCSV["ICD0" S DR=".01",DIE=DIC,(IBPROCP,DA)=+Y D ^DIE Q:'$D(DA)!($D(Y)) K DR ; IB\*2.0\*461  . I IBPROCSV["ICPT" S DR=".01;16",DIE=DIC,(IBPROCP,DA)=+Y D ^DIE Q:'$D(DA)!($D(Y)) K DR ; IB\*2.0\*447 BI  . ;  . S DR=""  . ;  . ; MRD;IB\*2.0\*516 - Added line level PROCEDURE DESCRIPTION field,  . ; asked only if the procedure is an "NOC".  . I IBPROCSV["ICPT",$$NOCPROC(IBPROCSV) D  . . S DA=$P(IBPROCSV,"^") ; The line# on the bill/claim.  . . S DR=51 ; Field# for PROCEDURE DESCRIPTION  . . D ^DIE  . . Q  . ;  . D EN^IBCU7B ; DEM;432 - Call to line level provider user input.  . S Y=IBPROCSV  ; DEM;432 - Restore value of Y after calls to FileMan  . K IBPROCSV  . K DR   ;WCJ;IB\*2.0\*432  . I IBPOPOUT Q   ; IB\*2.0\*447 BI  . S DR="" I Y["ICPT" S DR="6;5//"\_$$DEFDIV(IBIFN)\_";"  . S DR=DR\_$S(IBFT=2:"8;9;17//NO;",1:"")\_3,DIE=DIC,(IBPROCP,DA)=+Y D ^DIE Q:'$D(DA)!($E($G(Y))=U)  . K DR   ;WCJ;IB\*2.0\*432  . ;  . ; MRD;IB\*2.0\*516 - Allow user to add an NDC and Units. Ask only if  . ; coding system is not ICD and this is not a prescription claim. If  . ; an NDC is entered, prompt for Units.  . I $P($G(^DGCR(399,IBIFN,0)),U,9)'=9,'$$RXLINK^IBCSC5C(IBIFN,IBPROCP) D  . . K DA  . . S DA=IBPROCP,DA(1)=IBIFN,DIE="^DGCR(399,"\_IBIFN\_",""CP"","  . . S DR="53NDC NUMBER;I X="""" S Y="""";54//1"  . . D ^DIE  . . Q  . ;  . I IBFT=3 D:'$$INPAT^IBCEF(IBIFN) ATTACH  ; DEM;432 - Prompt for Attachment Control Number.  . ; DEM;432 - Add Additional OB Minutes to DR string for call to DIE.  . S DR=$$SPCUNIT(IBIFN,IBPROCP) S:DR["15;" DR=DR\_"74Additional OB Minutes" D ^DIE ; miles/minutes/hours  . ;  . I IBFT=2 D  .. D DX^IBCU72(IBIFN,IBPROCP)  .. S X=$$ADDTNL(IBIFN,.DA)  . Q:$$INPAT^IBCEF(IBIFN) ;only outpatient bills  . ;add procedures to array for download to PCE: dgcpt(assoc clinic,cpt,'provider^first dx^modifiers',cnt)=""  . S DGPROC=$G(^DGCR(399,IBIFN,"CP",+DA,0))  . S X=$P(DGPROC,U,18)\_U\_+$G(^IBA(362.3,+$P(DGPROC,U,11),0))\_U\_$P(DGPROC,U,15)  . I 'DGCPTNEW,$P(DGPROC,"^",7)="" S DGCPTNEW=2  . I DGCPTUP,DGCPTNEW S DGCPT=DGCPT+1 I $P(DGPROC,"^",7) S DGCPT($P(DGPROC,"^",7),+DGPROC,X,DGCPT)=""  . ; add visit date to bill  . I DGADDVST S (X,DINUM)=DGPROCDT D VFILE1^IBCOPV1 K DINUM,X,DGNOADD,DGADDVST  ; Delete modifiers with only a sequence #, no code  S Z=0 F  S Z=$O(^DGCR(399,IBIFN,"CP",Z)) Q:'Z  S Z0=0 F  S Z0=$O(^DGCR(399,IBIFN,"CP",Z,"MOD",Z0)) Q:'Z0  I $P($G(^(Z0,0)),U,2)="" S DA(2)=IBIFN,DA(1)=Z,DA=Z0,DIK="^DGCR(399,"\_DA(2)\_",""CP"","\_DA(1)\_",""MOD""," D ^DIK  Q CODQ K %DT,DGPROC,DIC,DIE,DR,DGPROCDT,IBPROCP,DLAYGO  K IBFT,DGNOADD,DGADDVST,DGCPT,DGCPTUP,IBZTYPE,DGCPTNEW  Q  ; DELPROC ; Remove the selected procedure, because of inactive status (cancel selection)  W !!,\*7,"The Procedure code is inactive on ",$$DAT1^IBOUTL(DGPROCDT),"."  W !,"Please select another Procedure."  S DA(1)=IBIFN,DA=+Y,DIK="^DGCR(399,"\_IBIFN\_",""CP"","  D ^DIK  Q  ; DELADD N Z,Z0,DA,DIK,X,Y  S DA(1)=IBIFN  ;Delete references to proc on rev codes  S Z=0 F  S Z=$O(^DGCR(399,IBIFN,"RC",Z)) Q:'Z  S Z0=$G(^(Z,0)) I Z0'="",$P(Z0,U,15)!$S($P(Z0,U,10)=3:$P(Z0,U,11),1:0) S DIE="^DGCR(399,"\_DA(1)\_",""RC"",",DA=Z,DR=".11///@;.15///@"\_$S($P(Z0,U,8):"",1:";.08////1") D ^DIE  S DIK="^DGCR(399,"\_DA(1)\_",""CP""," F DA=0:0 S DA=$O(^DGCR(399,DA(1),"CP",DA)) Q:'DA  D ^DIK  S DGRVRCAL=1  Q  ; DTMES ;Message if procedure date not in date range  Q:'$D(IBIFN) Q:'$D(^DGCR(399,IBIFN,"U")) S DGNODUU=^("U")  G:X'<$P(DGNODUU,"^")&(X'>$P(DGNODUU,"^",2)) DTMESQ  W \*7,!!?3,"Date must be within STATEMENT COVERS FROM and STATEMENT COVERS TO period."  S Y=$P(DGNODUU,"^") X ^DD("DD")  W !?3,"Enter a date between ",Y," and " S Y=$P(DGNODUU,"^",2) X ^DD("DD") W Y,!  K X,Y DTMESQ K DGNODUU Q  ; CODHLP ;Display Additional Procedure codes  N I,J,Y,IBMOD  I '$O(^DGCR(399,IBIFN,"CP",0)) W !!?5,"No Codes Entered!",! Q  W ! F I=0:0 S I=$O(^DGCR(399,IBIFN,"CP",I)) Q:'I  S Y=$G(^(I,0)) S Z=$$PRCNM^IBCSCH1($P(Y,"^",1),$P(Y,"^",2)) W !?5,$E($P(Z,"^",2),1,33),?40,"- ",$P(Z,"^") D  . N IBY  . S IBY=$P(Y,U,2)  . S IBMOD=$$GETMOD^IBEFUNC(IBIFN,I,1)  . I IBMOD'="" S IBMOD="/"\_IBMOD W IBMOD  . W ?60,"Date: " S Y=IBY D DT^DIQ  W !  ;  K Z Q  ; DICV I $D(IBIFN),$D(^DGCR(399,IBIFN,0)),$P(^(0),U,9) S DIC("V")=$S($P(^(0),U,9)=9:"I +Y(0)=80.1",$P(^(0),U,9)=4!($P(^(0),U,9)=5):"I +Y(0)=81",1:"")  Q  ; DEFDIV(IBIFN) ; Find default division for bill IBIFN  Q $P($G(^DG(40.8,+$P($G(^DGCR(399,IBIFN,0)),U,22),0)),U)  ; ADDTNL(IBIFN,DA) ;  N DR,IBOK,X,Y,DIR  S IBOK=1  S DR="19T;50.09T;50.08T" D ^DIE ; WCJ;IB\*2.0\*488 Added Ts  ;I '($$FT^IBCEF(IBIFN)'=3&($$INPAT^IBCEF(IBIFN))) D ATTACH ; DEM;432 - Prompt for Attachment Control Number.  I '($$FT^IBCEF(IBIFN)=3&($$INPAT^IBCEF(IBIFN))) D ATTACH  ; DEM;432 - Prompt for Attachment Control Number.  I $D(Y) S IBOK=0 G ADDTNLQ  ;/Beginning of IB\*2.0\*488 (vd)  ;S DIR("B")="NO",DIR("A")="EDIT CMS-1500 SPECIAL PROGRAM FIELDS and BOX 19?: ",DIR("A",1)=" ",DIR(0)="YA"  ;S DIR("?",1)="Respond YES only if you need to add/edit data for chiropractic visits,"  ;S DIR("?")="EPSDT care, or if billing for HOSPICE and attending is not a hospice employee."  ;D ^DIR K DIR  ;I Y'=1 S IBOK=0 G ADDTNLQ  ;S DR="W !,"" <<EPSDT>>"";50.07;W !!,"" <<HOSPICE>>"";50.03"  S DR="50.07T;50.03T"   ;WCJ;IB\*2.0\*488 added Ts  ;/End of IB\*2.0\*488 (vd)  D ^DIE  W ! ADDTNLQ Q IBOK  ; XTRA1(Y) ;  K Y  Q  ; SPCUNIT(IBIFN,DA) ; return fields for special units if applicable, in DR form  N IB0,IBCPT,IBDR,IBCT,IBFT,DFN S IBDR=""  S IB0=$G(^DGCR(399,+$G(IBIFN),0)),IBCT=$P(IB0,U,27),IBFT=$P(IB0,U,19),DFN=$P(IB0,U,2)  S IBCPT=$G(^DGCR(399,+$G(IBIFN),"CP",+$G(DA),0)) I IBCPT'["ICPT" G SPCUNTQ  I +$$ITMUNIT^IBCRU4(+IBCPT,5,IBCT) S IBDR="15;" D SROMIN^IBCU74(IBIFN,DA) G SPCUNTQ ; minutes  I +$$ITMUNIT^IBCRU4(+IBCPT,4,IBCT) S IBDR="21;" G SPCUNTQ ; miles  I +$$ITMUNIT^IBCRU4(+IBCPT,6,IBCT) S IBDR="22//"\_$$OBSHOUR^IBCU74(DFN,$P(IBCPT,U,2))\_";" G SPCUNTQ ; hours  I +IBFT=2,$P($G(^IBE(353.2,+$P(IBCPT,U,10),0)),U,2)="ANESTHESIA" S IBDR="15;" ; minutes SPCUNTQ Q IBDR  ; ATTACH ; DEM;432 - Attachment control number.  ; Ask if user wants to enter Attachment Control Number.  N DIR,X,Y,DA,DIE,DR  S DIR("A")="Enter Attachment Control Number"  S DIR(0)="Y",DIR("B")="NO"  D ^DIR  Q:'Y  ; User chose to enter Attachment Control Number.  ; User enters Attachment Control fields.  S DA(1)=IBIFN,DA=IBPROCP  S DIE="^DGCR(399,"\_DA(1)\_",""CP"","  S DR="71Report Type;72Report Transmission Method;70Attachment Control Number"  D ^DIE  Q  ; NOCPROC(IBPROCSV) ; MRD;IB\*2.0\*516 - Function to determine if procedure is an  ; "NOC". Returns '1' if "NOC" procedure, otherwise '0'.  ;  N IBNOC,IBPROCEX,IBPROCIN,IBPROCNM,IBX  S IBNOC=0  I $G(IBPROCSV)="" G NOCPROCQ  S IBPROCIN=$P($P(IBPROCSV,U,2),";")  I IBPROCIN="" G NOCPROCQ  ;  ; If procedure code ends in '99', quit with a '1'.  ;  S IBPROCEX=$P($G(^ICPT(IBPROCIN,0)),U,1)  I $E(IBPROCEX,$L(IBPROCEX)-1,$L(IBPROCEX))=99 S IBNOC=1 G NOCPROCQ  ;  ; Pull procedure name, then check to see if it contains one of the  ; specified strings.  ;  S IBPROCNM=$P($G(^ICPT(IBPROCIN,0)),U,2)  I IBPROCNM'="",$$NOC(IBPROCNM) S IBNOC=1 G NOCPROCQ  ;  S IBX=0  F  S IBX=$O(^ICPT(IBPROCIN,"D",IBX)) Q:'IBX  D  I IBNOC=1 Q  . S IBTEXT=$G(^ICPT(IBPROCIN,"D",IBX,0))  . I $G(^ICPT(IBPROCIN,"D",IBX+1,0))'="" S IBTEXT=IBTEXT\_" "\_$G(^ICPT(IBPROCIN,"D",IBX+1,0))  . S IBNOC=$$NOC(IBTEXT)  . Q  ; NOCPROCQ ; Quit out.  Q IBNOC  ; NOC(IBTEXT) ; Quit with '1' if IBTEXT contains one of the specified strings.  ;  S IBTEXT=$TR(IBTEXT,"abcdefghijklmnopqrstuvwxyz","ABCDEFGHIJKLMNOPQRSTUVWXYZ")  ;  I IBTEXT["NOT OTHERWISE" Q 1  I IBTEXT["NOT ELSEWHERE" Q 1  I IBTEXT["NOT LISTED" Q 1  I IBTEXT["UNLISTED" Q 1  I IBTEXT["UNSPECIFIED" Q 1  I IBTEXT["UNCLASSIFIED" Q 1  I IBTEXT["NON-SPECIFIED" Q 1  I IBTEXT["NOS " Q 1  I IBTEXT["NOS;" Q 1  I IBTEXT["NOS." Q 1  I IBTEXT["NOS," Q 1  I IBTEXT["NOS/" Q 1  I IBTEXT["(NOS)" Q 1  I IBTEXT["NOC " Q 1  I IBTEXT["NOC;" Q 1  I IBTEXT["NOC." Q 1  I IBTEXT["NOC," Q 1  I IBTEXT["NOC/" Q 1  I IBTEXT["(NOC)" Q 1  ;  ; Check if last three characters are 'NOC' or 'NOS'.  ;  S IBTEXT=$E(IBTEXT,$L(IBTEXT)-2,$L(IBTEXT))  I IBTEXT="NOC" Q 1  I IBTEXT="NOS" Q 1  ;  Q 0  ; | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCU7 ;ALB/AAS - INTERCEPT SCREEN INPUT OF PROCEDURE CODES ;29-OCT-91  ;;2.0;INTEGRATED BILLING;\*\*62,52,106,125,51,137,210,245,228,260,348,371,432,447,488,461,516,522,577,592\*\*;21-MAR-94;Build 1  ;;Per VA Directive 6402, this routine should not be modified.  ;  ;MAP TO DGCRU7  ; CHKX ; -interception of input x from Additional Procedure input  G:X=" " CHKXQ  I $$INPAT^IBCEF(DA(1)),'$P($G(^IBE(350.9,1,1)),"^",15),X'?1A1.2N D  G CHKXQ  . K X  . D EN^DDIOL("Site param does not allow entry of non-PTF procedures") ;Fileman error here will be: The previous error occurred when performing an action specified in a Pre-lookup transform (7.5 node).  G:'$D(^UTILITY($J,"IB")) CHKXQ  ;S M=($A($E(X,1))-64),S=+$E(X,2) Q:'$G(^UTILITY($J,"IB",M,S)) S X="`"\_+^(S)  S M=0 I X?1A1.2N S N=$G(^UTILITY($J,"IB","B",X)) S M=+N,S=+$P(N,U,2),P=X S S=$G(^UTILITY($J,"IB",M,S)) I +S S X="`"\_+S I $P(N,U,3)="N" S X=""""\_X\_"""" S $P(^UTILITY($J,"IB","B",P),U,3)="Y"  I +M,$D(DGPROCDT),DGPROCDT'=$P($G(^UTILITY($J,"IB",M,1)),"^",2) S DGPROCDT=$P(^(1),"^",2) W !!,"Procedure Date: " S Y=DGPROCDT X ^DD("DD") W Y,! CHKXQ Q  ; CODMUL ;Date oriented entry of procedure DELASK I $D(IBZ20),IBZ20,IBZ20'=$P(^DGCR(399,IBIFN,0),U,9) S %=2 W !,"SINCE THE PROCEDURE CODING METHOD HAS BEEN CHANGED, DO YOU WANT TO DELETE ALL",!,"PROCEDURE CODES IN THIS BILL"  I  D YN^DICN Q:%=-1 D:%=1 DELADD I %Y?1."?" W !!,"If you answer 'Yes', all procedure codes will be DELETED from this bill.",! G DELASK  K %,%Y,DA,IBZ20,DIK ;W !,"Procedure Entry:"  ; CODDT I $D(IBIFN),$D(^DGCR(399,IBIFN,0)),$P(^(0),U,9) S DIC("V")=$S($P(^(0),U,9)=9:"I +Y(0)=80.1",$P(^(0),U,9)=4!($P(^(0),U,9)=5):"I +Y(0)=81",1:"")  I $P($G(^DGCR(399,IBIFN,0)),"^",5)<3 S IBZTYPE=1 I $P($G(^UTILITY($J,"IB",1,1)),"^",2) S DGPROCDT=$P(^(1),"^",2) D ASKCOD  S X=$$PRCDIV^IBCU71(IBIFN) I +X W !!,$P(X,U,2),!  N Z,Z0 S Z=$G(^DGCR(399,IBIFN,"U")),Z0=$$FMTE^XLFDT($P(Z,U),"2D")\_"-"\_$$FMTE^XLFDT($P(Z,U,2),"2D")  W !,"Select PROCEDURE DATE"\_$S($TR(Z0,"-")'="":" ("\_Z0\_")",1:"")\_": " R X:DTIME G:'$T!("^"[X) CODQ D:X["?" CODHLP  S IBEX=0 D  ; Get procedure date  . I X=" ",$D(DGPROCDT),DGPROCDT?7N S Y=DGPROCDT D D^DIQ W " (",Y,")" Q  . I X=" ",+$P($G(^DGCR(399,IBIFN,"OP",0)),"^",4) S (DGPROCDT,Y)=$O(^DGCR(399,IBIFN,"OP",0)) D D^DIQ W " (",Y,")" Q  . S %DT="EXP",%DT(0)=-DT D ^%DT K %DT I Y<1 S IBEX=1 Q  . I '$$OPV2^IBCU41(Y,IBIFN,1) S IBEX=1 Q  . S:'$G(IBZTYPE) X=$$OPV^IBCU41(Y,IBIFN) S DGPROCDT=Y  I 'IBEX D ASKCOD,ADDCPT^IBCU71:$D(DGCPT)  K IBEX  G CODDT  ; ASKCOD N Z,Z0,DA,IBACT,IBQUIT,IBLNPRV  ;WCJ;2.0\*432  N IBPOPOUT  S IBPOPOUT=0 ; IB\*2.0\*447 BI  K DGCPT  S DGCPT=0,DGCPTUP=$P($G(^IBE(350.9,1,1)),"^",19),DGADDVST=0,IBFT=$P($G(^DGCR(399,IBIFN,0)),"^",19)  I '$D(^DGCR(399,IBIFN,"CP",0)) S ^DGCR(399,IBIFN,"CP",0)=U\_$$GETSPEC^IBEFUNC(399,304)  ;  F  S IBQUIT=0 D  Q:IBQUIT  . S IBPOPOUT=0  . D DICV ; restrict code type to PCM  . S DIC("A")=" Select PROCEDURE: "  . S DIC="^DGCR(399,"\_IBIFN\_",""CP"","  . S DIC(0)="AEQMNL"  . S DIC("S")="I '$D(DIV(""S""))&($P(^(0),U,2)=DGPROCDT)"  . S DIC("DR")="1///^S X=DGPROCDT"  . S DA(1)=IBIFN,DLAYGO=399  . W ! D ^DIC I Y<1 S IBQUIT=1 Q  . S IBPROCP=+Y  . ; If we just added inactive code - it must be deleted.  . S IBACT=0 ; Active flag  . I Y["ICD0" S IBACT=$$ICD0ACT^IBACSV(+$P(Y,U,2),$$BDATE^IBACSV(IBIFN))  . I Y["ICPT" S IBACT=$$CPTACT^IBACSV(+$P(Y,U,2),DGPROCDT)  . S DGCPTNEW=$P(Y,"^",3) ;Was the procedure just added?  . I DGCPTNEW,'IBACT D DELPROC Q  . I 'IBACT W !,\*7,"Warning: Procedure code is inactive on this date",!  . I DGCPTNEW,$D(^UTILITY($J,"IB")),$$INPAT^IBCEF(IBIFN),Y["ICPT(" D DATA^IBCU74(Y,.IBLNPRV)  . S DGADDVST=$S(DGCPTNEW:1,$D(DGADDVST):DGADDVST,1:0)  . N IBPRV,IBPRVO,IBPRVN  . ;  . ; Line level provider function by form type.  . ; CMS-1500 (FORM TYPE=2)  . ; RENDERING PROVIDER, REFERRING PROVIDER,  . ; and SUPERVISING PROVIDER.  . ; UB-04 (FORM TYPE=3)  . ; RENDERING PROVIDER, REFERRING PROVIDER,  . ; OPERATING PROVIDER, and OTHER OPERATING  . ; PROVIDER.  . ;  . ; Removed: Call to $$MAINPRV^IBCEU(IBIFN) is for claim  . ; level provider defaults.  . ; 1. For new line level providers we don't need  . ; or want default claim level provider  . ; (requirement).  . ; 2. We don't want to default claim level to  . ; line level provider (requirement).  . ;  . K DIC("V") ; DEM;432 - KILL DIC("V") because this was for previous variable pointer use.  . ;  . N IBPROCSV  ; DEM;432 - Variable IBPROCSV is variable to preserve value of 'Y', which is procedure code info returned by call to ^DIC.  . S IBPROCSV=Y  ; DEM;432 - Preserve value of Y for after calls to FileMan (Y = procedure code info returned by call to ^DIC).  . K DR   ;WCJ;IB\*2.0\*432  . ;  . I IBPROCSV["ICD0" S DR=".01",DIE=DIC,(IBPROCP,DA)=+Y D ^DIE Q:'$D(DA)!($D(Y)) K DR ; IB\*2.0\*461  . I IBPROCSV["ICPT" S DR=".01;16",DIE=DIC,(IBPROCP,DA)=+Y D ^DIE Q:'$D(DA)!($D(Y)) K DR ; IB\*2.0\*447 BI  . ;  . S DR=""  . ;  . ; MRD;IB\*2.0\*516 - Added line level PROCEDURE DESCRIPTION field,  . ; asked only if the procedure is an "NOC".  . I IBPROCSV["ICPT",$$NOCPROC(IBPROCSV) D  . . S DA=$P(IBPROCSV,"^") ; The line# on the bill/claim.  . . S DR=51 ; Field# for PROCEDURE DESCRIPTION  . . D ^DIE  . . Q  . ;  . D EN^IBCU7B ; DEM;432 - Call to line level provider user input.  . S Y=IBPROCSV  ; DEM;432 - Restore value of Y after calls to FileMan  . K IBPROCSV  . K DR   ;WCJ;IB\*2.0\*432  . I IBPOPOUT Q   ; IB\*2.0\*447 BI  . S DR="" I Y["ICPT" S DR="6;5//"\_$$DEFDIV(IBIFN)\_";"  . ;JWS;IB\*2.0\*592 US1108 - Dental  . S DR=DR\_$S(IBFT=7:"8;9//;",IBFT=2:"8;9;17//NO;",1:"")\_3,DIE=DIC,(IBPROCP,DA)=+Y D ^DIE Q:'$D(DA)!($E($G(Y))=U)  . K DR   ;WCJ;IB\*2.0\*432  . ;  . ; MRD;IB\*2.0\*516 - Allow user to add an NDC and Units. Ask only if  . ; coding system is not ICD and this is not a prescription claim. If  . ; an NDC is entered, prompt for Units.  . I $P($G(^DGCR(399,IBIFN,0)),U,9)'=9,'$$RXLINK^IBCSC5C(IBIFN,IBPROCP) D  . . ;JWS;IB\*2.0\*592 US1108 - Dental  . . I IBFT=7 Q  . . K DA  . . S DA=IBPROCP,DA(1)=IBIFN,DIE="^DGCR(399,"\_IBIFN\_",""CP"","  . . ; vd/Beginning IB\*2\*577 - Added the prompt for Unit/Basis of Measurement.  . . ; S DR="53NDC NUMBER;I X="""" S Y="""";54//1"  . . S DR="53NDC NUMBER;I X="""" S Y="""";52//UN;54QUANTITY//1"  ;Prompt for NDC, UN & amt.  . . ; vd/Ending IB\*2\*577  . . D ^DIE  . . Q  . ;  . I IBFT=3 D:'$$INPAT^IBCEF(IBIFN) ATTACH  ; DEM;432 - Prompt for Attachment Control Number.  . ; DEM;432 - Add Additional OB Minutes to DR string for call to DIE.  . S DR=$$SPCUNIT(IBIFN,IBPROCP) S:DR["15;" DR=DR\_"74Additional OB Minutes" D ^DIE ; miles/minutes/hours  . ;JWS;IB\*2.0\*592 US1108 - Dental  . I IBFT=2!(IBFT=7) D  .. D DX^IBCU72(IBIFN,IBPROCP)  .. ;JWS;IB\*2.0\*592 US1108 - Dental  .. I IBFT'=7 S X=$$ADDTNL(IBIFN,.DA)  . Q:$$INPAT^IBCEF(IBIFN) ;only outpatient bills  . ;JWS;IB\*2.0\*592 US1108 - Dental input fields  . I $$FT^IBCEF(IBIFN)=7 D ORAL^IBCU72  . ;add procedures to array for download to PCE: dgcpt(assoc clinic,cpt,'provider^first dx^modifiers',cnt)=""  . S DGPROC=$G(^DGCR(399,IBIFN,"CP",+DA,0))  . S X=$P(DGPROC,U,18)\_U\_+$G(^IBA(362.3,+$P(DGPROC,U,11),0))\_U\_$P(DGPROC,U,15)  . I 'DGCPTNEW,$P(DGPROC,"^",7)="" S DGCPTNEW=2  . I DGCPTUP,DGCPTNEW S DGCPT=DGCPT+1 I $P(DGPROC,"^",7) S DGCPT($P(DGPROC,"^",7),+DGPROC,X,DGCPT)=""  . ; add visit date to bill  . I DGADDVST S (X,DINUM)=DGPROCDT D VFILE1^IBCOPV1 K DINUM,X,DGNOADD,DGADDVST  ; Delete modifiers with only a sequence #, no code  S Z=0 F  S Z=$O(^DGCR(399,IBIFN,"CP",Z)) Q:'Z  S Z0=0 F  S Z0=$O(^DGCR(399,IBIFN,"CP",Z,"MOD",Z0)) Q:'Z0  I $P($G(^(Z0,0)),U,2)="" S DA(2)=IBIFN,DA(1)=Z,DA=Z0,DIK="^DGCR(399,"\_DA(2)\_",""CP"","\_DA(1)\_",""MOD""," D ^DIK  Q CODQ K %DT,DGPROC,DIC,DIE,DR,DGPROCDT,IBPROCP,DLAYGO  K IBFT,DGNOADD,DGADDVST,DGCPT,DGCPTUP,IBZTYPE,DGCPTNEW  Q  ; DELPROC ; Remove the selected procedure, because of inactive status (cancel selection)  W !!,\*7,"The Procedure code is inactive on ",$$DAT1^IBOUTL(DGPROCDT),"."  W !,"Please select another Procedure."  S DA(1)=IBIFN,DA=+Y,DIK="^DGCR(399,"\_IBIFN\_",""CP"","  D ^DIK  Q  ; DELADD N Z,Z0,DA,DIK,X,Y  S DA(1)=IBIFN  ;Delete references to proc on rev codes  S Z=0 F  S Z=$O(^DGCR(399,IBIFN,"RC",Z)) Q:'Z  S Z0=$G(^(Z,0)) I Z0'="",$P(Z0,U,15)!$S($P(Z0,U,10)=3:$P(Z0,U,11),1:0) S DIE="^DGCR(399,"\_DA(1)\_",""RC"",",DA=Z,DR=".11///@;.15///@"\_$S($P(Z0,U,8):"",1:";.08////1") D ^DIE  S DIK="^DGCR(399,"\_DA(1)\_",""CP""," F DA=0:0 S DA=$O(^DGCR(399,DA(1),"CP",DA)) Q:'DA  D ^DIK  S DGRVRCAL=1  Q  ; DTMES ;Message if procedure date not in date range  Q:'$D(IBIFN) Q:'$D(^DGCR(399,IBIFN,"U")) S DGNODUU=^("U")  G:X'<$P(DGNODUU,"^")&(X'>$P(DGNODUU,"^",2)) DTMESQ  W \*7,!!?3,"Date must be within STATEMENT COVERS FROM and STATEMENT COVERS TO period."  S Y=$P(DGNODUU,"^") X ^DD("DD")  W !?3,"Enter a date between ",Y," and " S Y=$P(DGNODUU,"^",2) X ^DD("DD") W Y,!  K X,Y DTMESQ K DGNODUU Q  ; CODHLP ;Display Additional Procedure codes  N I,J,Y,IBMOD  I '$O(^DGCR(399,IBIFN,"CP",0)) W !!?5,"No Codes Entered!",! Q  W ! F I=0:0 S I=$O(^DGCR(399,IBIFN,"CP",I)) Q:'I  S Y=$G(^(I,0)) S Z=$$PRCNM^IBCSCH1($P(Y,"^",1),$P(Y,"^",2)) W !?5,$E($P(Z,"^",2),1,33),?40,"- ",$P(Z,"^") D  . N IBY  . S IBY=$P(Y,U,2)  . S IBMOD=$$GETMOD^IBEFUNC(IBIFN,I,1)  . I IBMOD'="" S IBMOD="/"\_IBMOD W IBMOD  . W ?60,"Date: " S Y=IBY D DT^DIQ  W !  ;  K Z Q  ; DICV I $D(IBIFN),$D(^DGCR(399,IBIFN,0)),$P(^(0),U,9) S DIC("V")=$S($P(^(0),U,9)=9:"I +Y(0)=80.1",$P(^(0),U,9)=4!($P(^(0),U,9)=5):"I +Y(0)=81",1:"")  Q  ; DEFDIV(IBIFN) ; Find default division for bill IBIFN  Q $P($G(^DG(40.8,+$P($G(^DGCR(399,IBIFN,0)),U,22),0)),U)  ; ADDTNL(IBIFN,DA) ;  N DR,IBOK,X,Y,DIR  S IBOK=1  S DR="19T;50.09T;50.08T" D ^DIE ; WCJ;IB\*2.0\*488 Added Ts  ;I '($$FT^IBCEF(IBIFN)'=3&($$INPAT^IBCEF(IBIFN))) D ATTACH ; DEM;432 - Prompt for Attachment Control Number.  I '($$FT^IBCEF(IBIFN)=3&($$INPAT^IBCEF(IBIFN))) D ATTACH  ; DEM;432 - Prompt for Attachment Control Number.  I $D(Y) S IBOK=0 G ADDTNLQ  ;/Beginning of IB\*2.0\*488 (vd)  ;S DIR("B")="NO",DIR("A")="EDIT CMS-1500 SPECIAL PROGRAM FIELDS and BOX 19?: ",DIR("A",1)=" ",DIR(0)="YA"  ;S DIR("?",1)="Respond YES only if you need to add/edit data for chiropractic visits,"  ;S DIR("?")="EPSDT care, or if billing for HOSPICE and attending is not a hospice employee."  ;D ^DIR K DIR  ;I Y'=1 S IBOK=0 G ADDTNLQ  ;S DR="W !,"" <<EPSDT>>"";50.07;W !!,"" <<HOSPICE>>"";50.03"  S DR="50.07T;50.03T"   ;WCJ;IB\*2.0\*488 added Ts  ;/End of IB\*2.0\*488 (vd)  D ^DIE  W ! ADDTNLQ Q IBOK  ; XTRA1(Y) ;  K Y  Q  ; SPCUNIT(IBIFN,DA) ; return fields for special units if applicable, in DR form  N IB0,IBCPT,IBDR,IBCT,IBFT,DFN S IBDR=""  S IB0=$G(^DGCR(399,+$G(IBIFN),0)),IBCT=$P(IB0,U,27),IBFT=$P(IB0,U,19),DFN=$P(IB0,U,2)  S IBCPT=$G(^DGCR(399,+$G(IBIFN),"CP",+$G(DA),0)) I IBCPT'["ICPT" G SPCUNTQ  I +$$ITMUNIT^IBCRU4(+IBCPT,5,IBCT) S IBDR="15;" D SROMIN^IBCU74(IBIFN,DA) G SPCUNTQ ; minutes  I +$$ITMUNIT^IBCRU4(+IBCPT,4,IBCT) S IBDR="21;" G SPCUNTQ ; miles  I +$$ITMUNIT^IBCRU4(+IBCPT,6,IBCT) S IBDR="22//"\_$$OBSHOUR^IBCU74(DFN,$P(IBCPT,U,2))\_";" G SPCUNTQ ; hours  I +IBFT=2,$P($G(^IBE(353.2,+$P(IBCPT,U,10),0)),U,2)="ANESTHESIA" S IBDR="15;" ; minutes SPCUNTQ Q IBDR  ; ATTACH ; DEM;432 - Attachment control number.  ; Ask if user wants to enter Attachment Control Number.  N DIR,X,Y,DA,DIE,DR  S DIR("A")="Enter Attachment Control Number"  S DIR(0)="Y",DIR("B")="NO"  D ^DIR  Q:'Y  ; User chose to enter Attachment Control Number.  ; User enters Attachment Control fields.  S DA(1)=IBIFN,DA=IBPROCP  S DIE="^DGCR(399,"\_DA(1)\_",""CP"","  S DR="71Report Type;72Report Transmission Method;70Attachment Control Number"  D ^DIE  Q  ; NOCPROC(IBPROCSV) ; MRD;IB\*2.0\*516 - Function to determine if procedure is an  ; "NOC". Returns '1' if "NOC" procedure, otherwise '0'.  ;  N IBNOC,IBPROCEX,IBPROCIN,IBPROCNM,IBX  S IBNOC=0  I $G(IBPROCSV)="" G NOCPROCQ  S IBPROCIN=$P($P(IBPROCSV,U,2),";")  I IBPROCIN="" G NOCPROCQ  ;  ; If procedure code ends in '99', quit with a '1'.  ;  S IBPROCEX=$P($G(^ICPT(IBPROCIN,0)),U,1)  I $E(IBPROCEX,$L(IBPROCEX)-1,$L(IBPROCEX))=99 S IBNOC=1 G NOCPROCQ  ;  ; Pull procedure name, then check to see if it contains one of the  ; specified strings.  ;  S IBPROCNM=$P($G(^ICPT(IBPROCIN,0)),U,2)  I IBPROCNM'="",$$NOC(IBPROCNM) S IBNOC=1 G NOCPROCQ  ;  S IBX=0  F  S IBX=$O(^ICPT(IBPROCIN,"D",IBX)) Q:'IBX  D  I IBNOC=1 Q  . S IBTEXT=$G(^ICPT(IBPROCIN,"D",IBX,0))  . I $G(^ICPT(IBPROCIN,"D",IBX+1,0))'="" S IBTEXT=IBTEXT\_" "\_$G(^ICPT(IBPROCIN,"D",IBX+1,0))  . S IBNOC=$$NOC(IBTEXT)  . Q  ; NOCPROCQ ; Quit out.  Q IBNOC  ; NOC(IBTEXT) ; Quit with '1' if IBTEXT contains one of the specified strings.  ;  S IBTEXT=$TR(IBTEXT,"abcdefghijklmnopqrstuvwxyz","ABCDEFGHIJKLMNOPQRSTUVWXYZ")  ;  I IBTEXT["NOT OTHERWISE" Q 1  I IBTEXT["NOT ELSEWHERE" Q 1  I IBTEXT["NOT LISTED" Q 1  I IBTEXT["UNLISTED" Q 1  I IBTEXT["UNSPECIFIED" Q 1  I IBTEXT["UNCLASSIFIED" Q 1  I IBTEXT["NON-SPECIFIED" Q 1  I IBTEXT["NOS " Q 1  I IBTEXT["NOS;" Q 1  I IBTEXT["NOS." Q 1  I IBTEXT["NOS," Q 1  I IBTEXT["NOS/" Q 1  I IBTEXT["(NOS)" Q 1  I IBTEXT["NOC " Q 1  I IBTEXT["NOC;" Q 1  I IBTEXT["NOC." Q 1  I IBTEXT["NOC," Q 1  I IBTEXT["NOC/" Q 1  I IBTEXT["(NOC)" Q 1  ;  ; Check if last three charcters are 'NOC' or 'NOS'.  ;  S IBTEXT=$E(IBTEXT,$L(IBTEXT)-2,$L(IBTEXT))  I IBTEXT="NOC" Q 1  I IBTEXT="NOS" Q 1  ;  Q 0  ; ORALCAV(FLD) ;EP  ; Dictionary Screen function called from Procedures Oral Cavity Fields:  ; 399.0304.90.01, 399.0304.90.02, 399.0304.90.03, 399.0304.90.04, 399.0304.90.05   ; Prevents the same Oral Cavity from being selected more than once.  ; Input: FLD - Field # of the field being checked  ; DA - IEN of the Service Line Multiple being edited  ; DA(1) - IEN of the 356.22 entry being edited  ; Y - Internal Value of the user response  ; Returns: 1 - Data input by the user is valid, 0 otherwise  N NDE,RTN  S NDE=$G(^DGCR(399,DA(1),"CP",DA,"DEN"))  S RTN=1 ; Assume Valid Input  Q:Y="" 1 ; No value entered  ;  ; Make sure there are no duplicates  I FLD=90.01 D  Q RTN  . I $P(NDE,"^",2)=Y S RTN=0 Q  . I $P(NDE,"^",3)=Y S RTN=0 Q  . I $P(NDE,"^",4)=Y S RTN=0 Q  . I $P(NDE,"^",5)=Y S RTN=0 Q  I FLD=90.02 D  Q RTN  . I $P(NDE,"^",1)=Y S RTN=0 Q  . I $P(NDE,"^",3)=Y S RTN=0 Q  . I $P(NDE,"^",4)=Y S RTN=0 Q  . I $P(NDE,"^",5)=Y S RTN=0 Q  I FLD=90.03 D  Q RTN  . I $P(NDE,"^",1)=Y S RTN=0 Q  . I $P(NDE,"^",2)=Y S RTN=0 Q  . I $P(NDE,"^",4)=Y S RTN=0 Q  . I $P(NDE,"^",5)=Y S RTN=0 Q  I FLD=90.04 D  Q RTN  . I $P(NDE,"^",1)=Y S RTN=0 Q  . I $P(NDE,"^",2)=Y S RTN=0 Q  . I $P(NDE,"^",3)=Y S RTN=0 Q  . I $P(NDE,"^",5)=Y S RTN=0 Q  I FLD=90.05 D  Q RTN  . I $P(NDE,"^",1)=Y S RTN=0 Q  . I $P(NDE,"^",2)=Y S RTN=0 Q  . I $P(NDE,"^",3)=Y S RTN=0 Q  . I $P(NDE,"^",4)=Y S RTN=0 Q  Q RTN  ; TOOTHS(FLD) ;EP  ; Dictionary Screen function called from Dental Service Line Tooth fields:  ; 399,91,.02, 399,91,.03, 399,91,.04, 399,91,.05, 399,91,.06. Prevents the   ; same Tooth Surface from being selected more than once.  ; Input: FLD - Field # of the field being checked  ; DA - Tooth Surface multiple IEN  ; DA(1) - Service Line multiple IEN  ; DA(2) - IEN of the 356.22 entry being edited  ; Y - Internal Value of the user response  ; Returns: 1 - Data input by the user is valid, 0 otherwise  N NDE,RTN  S NDE=$G(^DGCR(399,DA(2),"CP",DA(1),"DEN1",DA,0))  S RTN=1 ; Assume Valid Input  Q:Y="" 1 ; No value entered  ;  ; Make sure there are no duplicates  I FLD=.02 D  Q RTN  . I $P(NDE,"^",3)=Y S RTN=0 Q  . I $P(NDE,"^",4)=Y S RTN=0 Q  . I $P(NDE,"^",5)=Y S RTN=0 Q  . I $P(NDE,"^",6)=Y S RTN=0 Q  I FLD=.03 D  Q RTN  . I $P(NDE,"^",2)=Y S RTN=0 Q  . I $P(NDE,"^",4)=Y S RTN=0 Q  . I $P(NDE,"^",5)=Y S RTN=0 Q  . I $P(NDE,"^",6)=Y S RTN=0 Q  I FLD=.04 D  Q RTN  . I $P(NDE,"^",2)=Y S RTN=0 Q  . I $P(NDE,"^",3)=Y S RTN=0 Q  . I $P(NDE,"^",5)=Y S RTN=0 Q  . I $P(NDE,"^",6)=Y S RTN=0 Q  I FLD=.05 D  Q RTN  . I $P(NDE,"^",2)=Y S RTN=0 Q  . I $P(NDE,"^",3)=Y S RTN=0 Q  . I $P(NDE,"^",4)=Y S RTN=0 Q  . I $P(NDE,"^",6)=Y S RTN=0 Q  I FLD=.06 D  Q RTN  . I $P(NDE,"^",2)=Y S RTN=0 Q  . I $P(NDE,"^",3)=Y S RTN=0 Q  . I $P(NDE,"^",4)=Y S RTN=0 Q  . I $P(NDE,"^",5)=Y S RTN=0 Q  Q RTN  ; | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCU7B | | | | | | | | |
| **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** | | | | | | | | | |
| IBCU7B ;ALB/DEM - LINE LEVEL PROVIDER USER INPUT ;27-SEP-2010  ;;2.0;INTEGRATED BILLING;\*\*432,447\*\*;21-MAR-94;Build 80  ;;Per VHA Directive 2004-038, this routine should not be modified.  Q  ; EN ;  ;  N X,DIC,DIE,DR,DA,DLAYGO,PRVFUN,DIPA,Y,DO,DD,I  ; ,IBPOPOUT IB\*2.0\*447 BI  I '$D(IBLNPRV("IBCCPT")) N IBLNPRV  ; DEM;432 - Coming from routine IBCCPT.  S:'$G(IBFT) IBFT=$$FT^IBCEF(IBIFN) ;DEM;432 - Form Type for claim.  I IBFT=3,$$INPAT^IBCEF(IBIFN) Q   ;WCJ\*2.0\*432 Don't ask line level providers if INPAT UB  Q:(IBFT'=2)&(IBFT'=3) ;DEM;432 - Must be CMS-1500 (2) or UB-04 (3) Form Type.  S:IBFT=2 PRVFUN(2)="Rendering,Referring,Supervising"  ;DEM;432 - Allowable provider functions for CMS-1500.  S:IBFT=3 PRVFUN(3)="Rendering,Referring,Operating,Other Operating"  ;DEM;432 - Allowable provider functions for UB-04.  ; IB\*2.0\*447 BI  ; F PRVFUN("CNT")=1:1:$L(PRVFUN(IBFT),",") S PRVFUN=$P(PRVFUN(IBFT),",",PRVFUN("CNT")) D I $G(IBPOPOUT) K IBPOPOUT Q  F PRVFUN("CNT")=1:1:$L(PRVFUN(IBFT),",") S PRVFUN=$P(PRVFUN(IBFT),",",PRVFUN("CNT")) D  I $G(IBPOPOUT) Q  . S X=$S(PRVFUN="Rendering":3,PRVFUN="Referring":1,PRVFUN="Supervising":5,PRVFUN="Operating":2,1:9) ;DEM;432 - X=Provider Function Code Number.  . ;I $D(IBLNPRV("IBCCPT")),X'=3 Q ; DEM;432 - Coming from routine IBCCPT, only interested in RENDERING PROVIDER.  . K DA,DO,DD  . S DA(2)=IBIFN,DA(1)=IBPROCP  ;DEM;432 - Set up DA array for call to FILE^DICN.  . S DIC="^DGCR(399,"\_DA(2)\_",""CP"","\_DA(1)\_",""LNPRV"","  ;DEM;432 - Global root of Line Provider multiple.  . S DIC(0)="L"  . S DIC("DR")=".01////"\_X  ;DEM;432 - Stuff X (provider function) into new entry.  . I '$D(^DGCR(399,DA(2),"CP",DA(1),"LNPRV","B",X)) D FILE^DICN ; DEM;432 - Add new entry.  . S DA=+$O(^DGCR(399,DA(2),"CP",DA(1),"LNPRV","B",X,0)) ;DEM;432 - Get DA of line provider entry.  . S DIPA("RF")=X  ;DEM;432 - Save provider function in DIPA("RF") for later use in call to DIE.  . S DIE=DIC  . K DIC,DO,DD,DR,X,Y  . D DRARRY  ;DEM;432 - Set up DR array for call to DIE.  . ;  . ; DEM;432 - Variable IBLNPRV is a flag for called code  . ; that we are coming from line level provider  . ; user input (example, EXTCR^IBCEU5).  . ;  . S IBLNPRV=1  . ; preserve DA values  . S IBLNPRV("LNPRVIEN")=DA  ;DEM;432 - DA of line provider entry to edit.  . S IBLNPRV("PROCIEN")=DA(1) ;DEM;432 - DA(1) is procedure code multiple IEN.  . S DLAYGO=399 ;DEM;432 - Set DLAYGO.  . D ^DIE  . ; IB\*2.0\*447 BI Changed to correct for empty provider types in global.  . ;I ($G(Y)="^")!($G(Y)=-1) S IBPOPOUT=1 Q ; User entered caret ("^"), so exit line provider entry.  . I ($D(Y)) S IBPOPOUT=1 ; User entered caret ("^"), so exit line provider entry.  . ; DEM;432 - If line provider zero node exist, and no provider, then delete entry. Reset DA  . S DA=IBLNPRV("LNPRVIEN"),DA(1)=IBLNPRV("PROCIEN")  . I $D(^DGCR(399,IBIFN,"CP",IBLNPRV("PROCIEN"),"LNPRV",IBLNPRV("LNPRVIEN"),0))#10,'$P(^DGCR(399,IBIFN,"CP",IBLNPRV("PROCIEN"),"LNPRV",IBLNPRV("LNPRVIEN"),0),U,2) S DR=".01///@" D ^DIE  . K DIC,DIE,DR,DA,X,Y,DO,DD,DLAYGO,DIPA  ;DEM;432 - Clean up.  . Q  ;  K IBLNPRV,PRVFUN  ; END ;  Q  ; DRARRY ; Set of DR array for user input.  ;  ; DEM;432 - DIE uses DR to execute individual DR array elements, so  ; need to leave DR(1,399.0404) undefined for DIE to move  ; DR string into DR(1,399.0404).  ;  ; Note: 'B' line tags represent DR string branching.  ;  ; 399.0404,.01 LINE FUNCTION.  ; Stuff value from FILE^DICN add above (DIPA("RF")) into .01 field.  ; Also, need to set up DIPA("I#") array from claim level for later reference in DR array.  S DR=".01///^S X=DIPA(""RF"");K DIPA S DIPA(""RF"")=X,DIPA(""I1"")=$D(^DGCR(399,DA(2),""I1"")),DIPA(""I2"")=$D(^(""I2"")),DIPA(""I3"")=$D(^(""I3""))"  ;  ; 399.0404,.02 LINE PERFORMED BY.  ; If no provider entered by user, then delete entry (accomplished by  ; deleting .01 field, LINE FUNCTION field).  ; Branch to end (@499) if no provider entered.  ;S:'$D(IBLNPRV("IBCCPT")) DR(1,399.0404,1)=".02"\_PRVFUN\_$S(PRVFUN'["Operating":" Provider",1:" Physician")\_";S:X DIPA(""PRF"")=X,Y=""@4"";.01///@;S Y=""@499"""  ;S:$D(IBLNPRV("IBCCPT")) DR(1,399.0404,1)=".02///"\_IBLNPRV("IBCCPT")\_";.02Rendering;S:X DIPA(""PRF"")=X,Y=""@4"";.01///@;S Y=""@499"""  S DR(1,399.0404,1)=""  S:$D(IBLNPRV("IBCCPT"))&(PRVFUN["Rendering") DR(1,399.0404,1)=".02///"\_IBLNPRV("IBCCPT")\_";"  S DR(1,399.0404,1)=DR(1,399.0404,1)\_".02"\_PRVFUN\_$S(PRVFUN'["Operating":" Provider",1:" Physician")\_";S:X DIPA(""PRF"")=X,Y=""@4"";.01///@;S Y=""@499"""  ; Branch to @48 if VA PROVIDER.  ; IF Non-VA PROVIDER, then file changes to IB NON/OTHER VA BILLING PROVIDER File (#355.93) for user input.  ; DR string syntax ";^355.93^IBA(355.93," accomplishes variable pointer file change.  ; See DR array DR(2,355.93) and DR(2,355.93,SEQ #) below for details.  ;  S DR(1,399.0404,2)="@4;N Z1 S Z1=$P($G(^DGCR(399,DA(2),""CP"",DA(1),""LNPRV"",DA,0)),U,2) S DIPA(""NVA\_PRV"")=$S(Z1[""IBA(355.93"":+Z1,1:0) S X=+X I DIPA(""NVA\_PRV"")=0 S Y=""@48"""  S DR(1,399.0404,3)="S:$D(^XUSEC(""IB PROVIDER EDIT"",DUZ)) DLAYGO=355.93;^355.93^IBA(355.93,"  ; NVAPRV ; Start of user input into IB NON/OTHER VA BILLING PROVIDER File (#355.93).  ;  S DR(2,355.93)="S DIPA(""NVA\_PRV-0"")=$G(^IBA(355.93,DIPA(""NVA\_PRV""),0))"  ;  ; Branch to @42 if PROVIDER TYPE equals '1' FOR FACILITY/GROUP.  ; Branch to @41 if CREDENTIALS are not NULL.  S DR(2,355.93,1)="S:$P(DIPA(""NVA\_PRV-0""),U,2)=1 Y=""@42"";S:$P(DIPA(""NVA\_PRV-0""),U,3)'="""" Y=""@41"""  ;  ; 355.93,.03 CREDENTIALS.  S DR(2,355.93)="S DIPA(""NVA\_PRV-0"")=$G(^IBA(355.93,DIPA(""NVA\_PRV""),0))"  ;  ; Branch to @42 if PROVIDER TYPE equals '1' FOR FACILITY/GROUP.  ; Branch to @41 if CREDENTIALS are not NULL.  S DR(2,355.93,1)="S:$P(DIPA(""NVA\_PRV-0""),U,2)=1 Y=""@42"";S:$P(DIPA(""NVA\_PRV-0""),U,3)'="""" Y=""@41"""  ;  ; 355.93,.03 CREDENTIALS.  S DR(2,355.93,2)=".03" B41 ;  ; 355.93,.04 SPECIALTY.  ; Branch to @45 if CREDENTIALS are not NULL.  S DR(2,355.93,3)="@41;S:$P(DIPA(""NVA\_PRV-0""),U,3)'="""" Y=""@45"";.04;S Y=""@45""" B42 ;  ; 355.93,.05 STREET ADDRESS.  ; 355.93,.06 CITY.  ; 355.93,.07 STATE.  ; Branch to @43 if there is an STREET ADDRESS, CITY, and STATE.  S DR(2,355.93,4)="@42;S:$P(DIPA(""NVA\_PRV-0""),U,5)'=""""&($P(DIPA(""NVA\_PRV-0""),U,6)'="""")&($P(DIPA(""NVA\_PRV-0""),U,7)'="""") Y=""@43"""  ; 355.93,.05 STREET ADDRESS.  ; 355.93,.1 STREET ADDRESS LINE 2.  ; 355.93,.06 CITY.  ; 355.93,.07 STATE.  ; 355.93,.08 ZIP CODE.  S DR(2,355.93,5)=".05;.1;.06;.07;.08" B43 ;  ; 355.93,.09 FACILITY DEFAULT ID NUMBER.  ; Branch to @44 if there is a FACILITY DEFAULT ID NUMBER.  S DR(2,355.93,6)="@43;S:$P(DIPA(""NVA\_PRV-0""),U,9)'="""" Y=""@44"";.09LAB OR FACILITY PRIMARY ID" B44 ;  ; 355.93,.11 X12 TYPE OF FACILITY.  ; Branch to @45 if there is a X12 TYPE OF FACILITY.  S DR(2,355.93,7)="@44;S:$P(DIPA(""NVA\_PRV-0""),U,11)'="""" Y=""@45"";.11" B45 ;  ; 355.93,41.01 NPI.  ; Branch to @46 if there is an NPI.  S DR(2,355.93,8)="@45;S:$P(DIPA(""NVA\_PRV-0""),U,14)'="""" Y=""@46"";D EN2^IBCEP82(DIPA(""NVA\_PRV""),4)" B46 ;  ; 355.93,42 TAXONOMY CODE.  ; Branch to @47 if there is TAXONOMY data.  ; 355.93,42 TAXONOMY CODE is a multiple (Sub-File 355.9342). We want 'ALL'  ; fields from TAXONOMY CODE Sub-File 355.9342. Thus,  ; DR string S DR(4,355.9342)=".01:.03" below.  S DR(2,355.93,9)="@46;S:$D(^IBA(355.93,DIPA(""NVA\_PRV""),""TAXONOMY""))>0 Y=""@47"";42"  S DR(3,355.9342)=".01:.03" B47 ;  ; End of data entry for IB NON/OTHER VA BILLING PROVIDER File (#399.53).  S DR(2,355.93,10)="@47"  ; B48 ;  ; LNPRV ; User input into LINE PROVIDER Sub-File 399.0404.  ;  S DR(1,399.0404,4)="@48"  S DR(1,399.0404,5)="S DIK=""^DGCR(399,""\_DA(2)\_"",""""CP"""",""\_DA(1)\_"",""""LNPRV"""","",DIK(1)="".02"" D EN1^DIK K DIK"  ; 399.0404,.15 LINE TAXONOMY.  S DR(1,399.0404,6)=".15Line Level Taxonomy"  S DR(1,399.0404,7)="D DISPTAX^IBCEP81($P($G(^DGCR(399,DA(2),""CP"",DA(1),""LNPRV"",DA,0)),U,15),"""")"  S DR(1,399.0404,8)="N Z S Z=$$EXPAND^IBTRE(399.0404,.08,$P($G(^DGCR(399,DA(2),""CP"",DA(1),""LNPRV"",DA,0)),U,8)) S DIPA(""SPC"")=$S(Z'="""":Z,1:""UNSPECIFIED"")"  S DR(1,399.0404,9)="W !,"" Prov Specialty On File: "",DIPA(""SPC"")"  S DR(1,399.0404,10)="S DIPA(""CRD"")=$$CRED^IBCEU($P($G(^DGCR(399,DA(2),""CP"",DA(1),""LNPRV"",DA,0)),U,2))"  ; 399.0404,.03 LINE CREDENTIALS  S DR(1,399.0404,11)=".03;K DIPA(""W1"") S:$G(DIPA(""CRD""))'=$P($G(^DGCR(399,DA(2),""CP"",DA(1),""LNPRV"",DA,0)),U,3) DIPA(""W1"")=1"  S DR(1,399.0404,12)="I $G(DIPA(""W1"")) D WRT1^IBCSC10H($G(DIPA(""CRD"")))"  ; Branch to @405 if File #399 PRIMARY NODE is non numeric.  S DR(1,399.0404,13)="K DIPA(""W1"") I '$G(DIPA(""I1"")) S Y=""@405"""  ; Branching based on DIPA("EDIT") - DIPA("EDIT") set in PROVID^IBCEP2B call  S DR(1,399.0404,14)="D PROVID^IBCEP2B(DA(2),DA,1,.DIPA) S Y=$S(DIPA(""EDIT"")<0:""@482"",DIPA(""EDIT"")=1:""@491"",DIPA(""EDIT"")=2:""@471"",1:"""")" B482 ;  ; Branch to @405 if File #399 SECORDARY NODE is non numeric.  S DR(1,399.0404,15)="@482;I '$G(DIPA(""I2"")) S Y=""@405"""  S DR(1,399.0404,16)="D PROVID^IBCEP2B(DA(2),DA,2,.DIPA)"  ; Branching based on DIPA("EDIT") - DIPA("EDIT") set in PROVID^IBCEP2B call.  S DR(1,399.0404,17)="S Y=$S(DIPA(""EDIT"")<0:""@483"",DIPA(""EDIT"")=1:""@492"",DIPA(""EDIT"")=2:""@472"",1:"""")" B483 ;  ; Branch to @405 if File #399 TERTIARY NODE is non numeric.  S DR(1,399.0404,18)="@483;I '$G(DIPA(""I3"")) S Y=""@405"""  S DR(1,399.0404,19)="D PROVID^IBCEP2B(DA(2),DA,3,.DIPA)"  ; Branching based on DIPA("EDIT") - DIPA("EDIT") set in PROVID^IBCEP2B call.  S DR(1,399.0404,20)="S Y=$S(DIPA(""EDIT"")<0:""@405"",DIPA(""EDIT"")=1:""@493"",DIPA(""EDIT"")=2:""@473"",1:"""");S Y=""@405""" B491 ;  ; 399.0404,.12 LINE PRIM INS PROVIDER ID TYPE.  ; 399.0404,.05 LINE PRIMARY INS CO ID NUMBER.  ; Branch to @482.  S DR(1,399.0404,21)="@491;.12R~T;.05T;S Y=""@482""" B492 ;  ; 399.0404,.13 LINE SEC INS PROVIDER ID TYPE.  ; 399.0404,.06 LINE SECONDARY INS CO ID NUMBER.  ; Branch to @483.  S DR(1,399.0404,22)="@492;.13R~T;.06T;S Y=""@483""" B493 ;  ; 399.0404,.14 LINE TERT INS PROVIDER ID TYPE.  ; 399.0404,.07 LINE TERTIARY INS CO ID NUMBER.  ; Branch to @405.  S DR(1,399.0404,23)="@493;.14R~T;.07T;S Y=""@405""" B471 ;  ; 399.0404,.12 LINE PRIM INS PROVIDER ID TYPE.  ; 399.0404,.05 LINE PRIMARY INS CO ID NUMBER.  ; Branch to @482.  S DR(1,399.0404,24)="@471;.12////^S X=DIPA(""PRIDT"");.05////^S X=DIPA(""PRID"");S Y=""@482""" B472 ;  ; 399.0404,.13 LINE SEC INS PROVIDER ID TYPE.  ; 399.0404,.06 LINE SECONDARY INS CO ID NUMBER.  ; Branch to @483.  S DR(1,399.0404,25)="@472;.13////^S X=DIPA(""PRIDT"");.06////^S X=DIPA(""PRID"");S Y=""@483""" B473 ;  ; 399.0404,.14 LINE TERT INS PROVIDER ID TYPE.  ; 399.0404,.07 LINE TERTIARY INS CO ID NUMBER.  ; Branch to @405.  S DR(1,399.0404,26)="@473;.14////^S X=DIPA(""PRIDT"");.07////^S X=DIPA(""PRID"");S Y=""@405""" B405 ;  S DR(1,399.0404,27)="@405"  ; B499 ;  ; End of user input @499 and W @IOF.  S DR(1,399.0404,28)="@499;W @IOF"  Q | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCU7B ;ALB/DEM - LINE LEVEL PROVIDER USER INPUT ;27-SEP-2010  ;;2.0;INTEGRATED BILLING;\*\*432,447,592\*\*;21-MAR-94;Build 80  ;;Per VHA Directive 2004-038, this routine should not be modified.  Q  ; EN ;  ;  N X,DIC,DIE,DR,DA,DLAYGO,PRVFUN,DIPA,Y,DO,DD,I  ; ,IBPOPOUT IB\*2.0\*447 BI  I '$D(IBLNPRV("IBCCPT")) N IBLNPRV  ; DEM;432 - Coming from routine IBCCPT.  S:'$G(IBFT) IBFT=$$FT^IBCEF(IBIFN) ;DEM;432 - Form Type for claim.  I IBFT=3,$$INPAT^IBCEF(IBIFN) Q   ;WCJ\*2.0\*432 Don't ask line level providers if INPAT UB  ;JWS;IB\*2.0\*592;Dental form 7  Q:(IBFT'=2)&(IBFT'=3)&(IBFT'=7) ;DEM;432 - Must be CMS-1500 (2) or UB-04 (3) Form Type or J430D Dental  S:IBFT=2 PRVFUN(2)="Rendering,Referring,Supervising"  ;DEM;432 - Allowable provider functions for CMS-1500.  S:IBFT=3 PRVFUN(3)="Rendering,Referring,Operating,Other Operating"  ;DEM;432 - Allowable provider functions for UB-04.  ;JWS;IB\*2.0\*592;Dental form 7  S:IBFT=7 PRVFUN(7)="Rendering,Referring,Supervising,Assistant Surgeon"  ; IB\*2.0\*447 BI  ; F PRVFUN("CNT")=1:1:$L(PRVFUN(IBFT),",") S PRVFUN=$P(PRVFUN(IBFT),",",PRVFUN("CNT")) D I $G(IBPOPOUT) K IBPOPOUT Q  F PRVFUN("CNT")=1:1:$L(PRVFUN(IBFT),",") S PRVFUN=$P(PRVFUN(IBFT),",",PRVFUN("CNT")) D  I $G(IBPOPOUT) Q  . ;JWS;IB\*2.0\*592;Dental form 7 add Assistant Surgeon  . S X=$S(PRVFUN="Rendering":3,PRVFUN="Referring":1,PRVFUN="Supervising":5,PRVFUN="Operating":2,PRVFUN="Assistant Surgeon":6,1:9) ;DEM;432 - X=Provider Function Code Number.  . ;I $D(IBLNPRV("IBCCPT")),X'=3 Q ; DEM;432 - Coming from routine IBCCPT, only interested in RENDERING PROVIDER.  . K DA,DO,DD  . S DA(2)=IBIFN,DA(1)=IBPROCP  ;DEM;432 - Set up DA array for call to FILE^DICN.  . S DIC="^DGCR(399,"\_DA(2)\_",""CP"","\_DA(1)\_",""LNPRV"","  ;DEM;432 - Global root of Line Provider multiple.  . S DIC(0)="L"  . S DIC("DR")=".01////"\_X  ;DEM;432 - Stuff X (provider function) into new entry.  . I '$D(^DGCR(399,DA(2),"CP",DA(1),"LNPRV","B",X)) D FILE^DICN ; DEM;432 - Add new entry.  . S DA=+$O(^DGCR(399,DA(2),"CP",DA(1),"LNPRV","B",X,0)) ;DEM;432 - Get DA of line provider entry.  . S DIPA("RF")=X  ;DEM;432 - Save provider function in DIPA("RF") for later use in call to DIE.  . S DIE=DIC  . K DIC,DO,DD,DR,X,Y  . D DRARRY  ;DEM;432 - Set up DR array for call to DIE.  . ;  . ; DEM;432 - Variable IBLNPRV is a flag for called code  . ; that we are coming from line level provider  . ; user input (example, EXTCR^IBCEU5).  . ;  . S IBLNPRV=1  . ; preserve DA values  . S IBLNPRV("LNPRVIEN")=DA  ;DEM;432 - DA of line provider entry to edit.  . S IBLNPRV("PROCIEN")=DA(1) ;DEM;432 - DA(1) is procedure code multiple IEN.  . S DLAYGO=399 ;DEM;432 - Set DLAYGO.  . D ^DIE  . ; IB\*2.0\*447 BI Changed to correct for empty provider types in global.  . ;I ($G(Y)="^")!($G(Y)=-1) S IBPOPOUT=1 Q ; User entered caret ("^"), so exit line provider entry.  . I ($D(Y)) S IBPOPOUT=1 ; User entered caret ("^"), so exit line provider entry.  . ; DEM;432 - If line provider zero node exist, and no provider, then delete entry. Reset DA  . S DA=IBLNPRV("LNPRVIEN"),DA(1)=IBLNPRV("PROCIEN")  . I $D(^DGCR(399,IBIFN,"CP",IBLNPRV("PROCIEN"),"LNPRV",IBLNPRV("LNPRVIEN"),0))#10,'$P(^DGCR(399,IBIFN,"CP",IBLNPRV("PROCIEN"),"LNPRV",IBLNPRV("LNPRVIEN"),0),U,2) S DR=".01///@" D ^DIE  . K DIC,DIE,DR,DA,X,Y,DO,DD,DLAYGO,DIPA  ;DEM;432 - Clean up.  . Q  ;  K IBLNPRV,PRVFUN  ; END ;  Q  ; DRARRY ; Set of DR array for user input.  ;  ; DEM;432 - DIE uses DR to execute individual DR array elements, so  ; need to leave DR(1,399.0404) undefined for DIE to move  ; DR string into DR(1,399.0404).  ;  ; Note: 'B' line tags represent DR string branching.  ;  ; 399.0404,.01 LINE FUNCTION.  ; Stuff value from FILE^DICN add above (DIPA("RF")) into .01 field.  ; Also, need to set up DIPA("I#") array from claim level for later reference in DR array.  S DR=".01///^S X=DIPA(""RF"");K DIPA S DIPA(""RF"")=X,DIPA(""I1"")=$D(^DGCR(399,DA(2),""I1"")),DIPA(""I2"")=$D(^(""I2"")),DIPA(""I3"")=$D(^(""I3""))"  ;  ; 399.0404,.02 LINE PERFORMED BY.  ; If no provider entered by user, then delete entry (accomplished by  ; deleting .01 field, LINE FUNCTION field).  ; Branch to end (@499) if no provider entered.  ;S:'$D(IBLNPRV("IBCCPT")) DR(1,399.0404,1)=".02"\_PRVFUN\_$S(PRVFUN'["Operating":" Provider",1:" Physician")\_";S:X DIPA(""PRF"")=X,Y=""@4"";.01///@;S Y=""@499"""  ;S:$D(IBLNPRV("IBCCPT")) DR(1,399.0404,1)=".02///"\_IBLNPRV("IBCCPT")\_";.02Rendering;S:X DIPA(""PRF"")=X,Y=""@4"";.01///@;S Y=""@499"""  S DR(1,399.0404,1)=""  S:$D(IBLNPRV("IBCCPT"))&(PRVFUN["Rendering") DR(1,399.0404,1)=".02///"\_IBLNPRV("IBCCPT")\_";"  ;JWS;IB\*2.0\*592;Dental - added Surgeon for Dental  S DR(1,399.0404,1)=DR(1,399.0404,1)\_".02"\_PRVFUN\_$S(PRVFUN["Surgeon":"",PRVFUN'["Operating":" Provider",1:" Physician")\_";S:X DIPA(""PRF"")=X,Y=""@4"";.01///@;S Y=""@499"""  ; Branch to @48 if VA PROVIDER.  ; IF Non-VA PROVIDER, then file changes to IB NON/OTHER VA BILLING PROVIDER File (#355.93) for user input.  ; DR string syntax ";^355.93^IBA(355.93," accomplishes variable pointer file change.  ; See DR array DR(2,355.93) and DR(2,355.93,SEQ #) below for details.  ;  S DR(1,399.0404,2)="@4;N Z1 S Z1=$P($G(^DGCR(399,DA(2),""CP"",DA(1),""LNPRV"",DA,0)),U,2) S DIPA(""NVA\_PRV"")=$S(Z1[""IBA(355.93"":+Z1,1:0) S X=+X I DIPA(""NVA\_PRV"")=0 S Y=""@48"""  S DR(1,399.0404,3)="S:$D(^XUSEC(""IB PROVIDER EDIT"",DUZ)) DLAYGO=355.93;^355.93^IBA(355.93,"  ; NVAPRV ; Start of user input into IB NON/OTHER VA BILLING PROVIDER File (#355.93).  ;  S DR(2,355.93)="S DIPA(""NVA\_PRV-0"")=$G(^IBA(355.93,DIPA(""NVA\_PRV""),0))"  ;  ; Branch to @42 if PROVIDER TYPE equals '1' FOR FACILITY/GROUP.  ; Branch to @41 if CREDENTIALS are not NULL.  S DR(2,355.93,1)="S:$P(DIPA(""NVA\_PRV-0""),U,2)=1 Y=""@42"";S:$P(DIPA(""NVA\_PRV-0""),U,3)'="""" Y=""@41"""  ;  ; 355.93,.03 CREDENTIALS.  S DR(2,355.93)="S DIPA(""NVA\_PRV-0"")=$G(^IBA(355.93,DIPA(""NVA\_PRV""),0))"  ;  ; Branch to @42 if PROVIDER TYPE equals '1' FOR FACILITY/GROUP.  ; Branch to @41 if CREDENTIALS are not NULL.  S DR(2,355.93,1)="S:$P(DIPA(""NVA\_PRV-0""),U,2)=1 Y=""@42"";S:$P(DIPA(""NVA\_PRV-0""),U,3)'="""" Y=""@41"""  ;  ; 355.93,.03 CREDENTIALS.  S DR(2,355.93,2)=".03" B41 ;  ; 355.93,.04 SPECIALTY.  ; Branch to @45 if CREDENTIALS are not NULL.  S DR(2,355.93,3)="@41;S:$P(DIPA(""NVA\_PRV-0""),U,3)'="""" Y=""@45"";.04;S Y=""@45""" B42 ;  ; 355.93,.05 STREET ADDRESS.  ; 355.93,.06 CITY.  ; 355.93,.07 STATE.  ; Branch to @43 if there is an STREET ADDRESS, CITY, and STATE.  S DR(2,355.93,4)="@42;S:$P(DIPA(""NVA\_PRV-0""),U,5)'=""""&($P(DIPA(""NVA\_PRV-0""),U,6)'="""")&($P(DIPA(""NVA\_PRV-0""),U,7)'="""") Y=""@43"""  ; 355.93,.05 STREET ADDRESS.  ; 355.93,.1 STREET ADDRESS LINE 2.  ; 355.93,.06 CITY.  ; 355.93,.07 STATE.  ; 355.93,.08 ZIP CODE.  S DR(2,355.93,5)=".05;.1;.06;.07;.08" B43 ;  ; 355.93,.09 FACILITY DEFAULT ID NUMBER.  ; Branch to @44 if there is a FACILITY DEFAULT ID NUMBER.  S DR(2,355.93,6)="@43;S:$P(DIPA(""NVA\_PRV-0""),U,9)'="""" Y=""@44"";.09LAB OR FACILITY PRIMARY ID" B44 ;  ; 355.93,.11 X12 TYPE OF FACILITY.  ; Branch to @45 if there is a X12 TYPE OF FACILITY.  S DR(2,355.93,7)="@44;S:$P(DIPA(""NVA\_PRV-0""),U,11)'="""" Y=""@45"";.11" B45 ;  ; 355.93,41.01 NPI.  ; Branch to @46 if there is an NPI.  S DR(2,355.93,8)="@45;S:$P(DIPA(""NVA\_PRV-0""),U,14)'="""" Y=""@46"";D EN2^IBCEP82(DIPA(""NVA\_PRV""),4)" B46 ;  ; 355.93,42 TAXONOMY CODE.  ; Branch to @47 if there is TAXONOMY data.  ; 355.93,42 TAXONOMY CODE is a multiple (Sub-File 355.9342). We want 'ALL'  ; fields from TAXONOMY CODE Sub-File 355.9342. Thus,  ; DR string S DR(4,355.9342)=".01:.03" below.  S DR(2,355.93,9)="@46;S:$D(^IBA(355.93,DIPA(""NVA\_PRV""),""TAXONOMY""))>0 Y=""@47"";42"  S DR(3,355.9342)=".01:.03" B47 ;  ; End of data entry for IB NON/OTHER VA BILLING PROVIDER File (#399.53).  S DR(2,355.93,10)="@47"  ; B48 ;  ; LNPRV ; User input into LINE PROVIDER Sub-File 399.0404.  ;  S DR(1,399.0404,4)="@48"  S DR(1,399.0404,5)="S DIK=""^DGCR(399,""\_DA(2)\_"",""""CP"""",""\_DA(1)\_"",""""LNPRV"""","",DIK(1)="".02"" D EN1^DIK K DIK"  ; 399.0404,.15 LINE TAXONOMY.  S DR(1,399.0404,6)=".15Line Level Taxonomy"  S DR(1,399.0404,7)="D DISPTAX^IBCEP81($P($G(^DGCR(399,DA(2),""CP"",DA(1),""LNPRV"",DA,0)),U,15),"""")"  S DR(1,399.0404,8)="N Z S Z=$$EXPAND^IBTRE(399.0404,.08,$P($G(^DGCR(399,DA(2),""CP"",DA(1),""LNPRV"",DA,0)),U,8)) S DIPA(""SPC"")=$S(Z'="""":Z,1:""UNSPECIFIED"")"  S DR(1,399.0404,9)="W !,"" Prov Specialty On File: "",DIPA(""SPC"")"  S DR(1,399.0404,10)="S DIPA(""CRD"")=$$CRED^IBCEU($P($G(^DGCR(399,DA(2),""CP"",DA(1),""LNPRV"",DA,0)),U,2))"  ; 399.0404,.03 LINE CREDENTIALS  S DR(1,399.0404,11)=".03;K DIPA(""W1"") S:$G(DIPA(""CRD""))'=$P($G(^DGCR(399,DA(2),""CP"",DA(1),""LNPRV"",DA,0)),U,3) DIPA(""W1"")=1"  S DR(1,399.0404,12)="I $G(DIPA(""W1"")) D WRT1^IBCSC10H($G(DIPA(""CRD"")))"  ; Branch to @405 if File #399 PRIMARY NODE is non numeric.  S DR(1,399.0404,13)="K DIPA(""W1"") I '$G(DIPA(""I1"")) S Y=""@405"""  ; Branching based on DIPA("EDIT") - DIPA("EDIT") set in PROVID^IBCEP2B call  S DR(1,399.0404,14)="D PROVID^IBCEP2B(DA(2),DA,1,.DIPA) S Y=$S(DIPA(""EDIT"")<0:""@482"",DIPA(""EDIT"")=1:""@491"",DIPA(""EDIT"")=2:""@471"",1:"""")" B482 ;  ; Branch to @405 if File #399 SECORDARY NODE is non numeric.  S DR(1,399.0404,15)="@482;I '$G(DIPA(""I2"")) S Y=""@405"""  S DR(1,399.0404,16)="D PROVID^IBCEP2B(DA(2),DA,2,.DIPA)"  ; Branching based on DIPA("EDIT") - DIPA("EDIT") set in PROVID^IBCEP2B call.  S DR(1,399.0404,17)="S Y=$S(DIPA(""EDIT"")<0:""@483"",DIPA(""EDIT"")=1:""@492"",DIPA(""EDIT"")=2:""@472"",1:"""")" B483 ;  ; Branch to @405 if File #399 TERTIARY NODE is non numeric.  S DR(1,399.0404,18)="@483;I '$G(DIPA(""I3"")) S Y=""@405"""  S DR(1,399.0404,19)="D PROVID^IBCEP2B(DA(2),DA,3,.DIPA)"  ; Branching based on DIPA("EDIT") - DIPA("EDIT") set in PROVID^IBCEP2B call.  S DR(1,399.0404,20)="S Y=$S(DIPA(""EDIT"")<0:""@405"",DIPA(""EDIT"")=1:""@493"",DIPA(""EDIT"")=2:""@473"",1:"""");S Y=""@405""" B491 ;  ; 399.0404,.12 LINE PRIM INS PROVIDER ID TYPE.  ; 399.0404,.05 LINE PRIMARY INS CO ID NUMBER.  ; Branch to @482.  S DR(1,399.0404,21)="@491;.12R~T;.05T;S Y=""@482""" B492 ;  ; 399.0404,.13 LINE SEC INS PROVIDER ID TYPE.  ; 399.0404,.06 LINE SECONDARY INS CO ID NUMBER.  ; Branch to @483.  S DR(1,399.0404,22)="@492;.13R~T;.06T;S Y=""@483""" B493 ;  ; 399.0404,.14 LINE TERT INS PROVIDER ID TYPE.  ; 399.0404,.07 LINE TERTIARY INS CO ID NUMBER.  ; Branch to @405.  S DR(1,399.0404,23)="@493;.14R~T;.07T;S Y=""@405""" B471 ;  ; 399.0404,.12 LINE PRIM INS PROVIDER ID TYPE.  ; 399.0404,.05 LINE PRIMARY INS CO ID NUMBER.  ; Branch to @482.  S DR(1,399.0404,24)="@471;.12////^S X=DIPA(""PRIDT"");.05////^S X=DIPA(""PRID"");S Y=""@482""" B472 ;  ; 399.0404,.13 LINE SEC INS PROVIDER ID TYPE.  ; 399.0404,.06 LINE SECONDARY INS CO ID NUMBER.  ; Branch to @483.  S DR(1,399.0404,25)="@472;.13////^S X=DIPA(""PRIDT"");.06////^S X=DIPA(""PRID"");S Y=""@483""" B473 ;  ; 399.0404,.14 LINE TERT INS PROVIDER ID TYPE.  ; 399.0404,.07 LINE TERTIARY INS CO ID NUMBER.  ; Branch to @405.  S DR(1,399.0404,26)="@473;.14////^S X=DIPA(""PRIDT"");.07////^S X=DIPA(""PRID"");S Y=""@405""" B405 ;  S DR(1,399.0404,27)="@405"  ; B499 ;  ; End of user input @499 and W @IOF.  S DR(1,399.0404,28)="@499;W @IOF"  Q | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCU82 | | | | | | | | |
| **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** | | | | | | | | | |
| IBCU82 ;ALB/ARH - THIRD PARTY BILLING UTILITIES (AUTOMATED BILLER) ;02 JUL 93  ;;2.0;INTEGRATED BILLING;\*\*43,55,91,124,160,304,347,432\*\*;21-MAR-94;Build 192  ;;Per VHA Directive 2004-038, this routine should not be modified.  ;  ; EVNTCHK(IBTRN) ;special checks to determine if event should be auto billed  ;checks for INS, non-veteran patient, possible workers comp and tort feasor, admitted for sc cond., outp dental stop, optv while inpt, category covered by ins, non-billable stop or clinic  ;(assumes that Claims Tracking does the SC check for Outpatients)  ;input: IBTRN - claims tracking event  ; DISP - if true then any error message will be displayed on exit.  ;output: returns "1^error message" if one of the checks failed, 0 otherwise  ;  N X,IBX,IBY,IBZ,IBTRND,IBCAT,IBCOV,DFN,IBEVDT,VAEL,VADMVT,VAINDT S X=0,IBTRND=$G(^IBT(356,+$G(IBTRN),0)) G:IBTRND="" EVNTCQ  I +$P(IBTRND,U,18)=1,'+$P(IBTRND,U,5) S X="1^Claims Tracking event does not have an associated Inpatient Admission." G EVNTCQ  I +$P(IBTRND,U,18)=2,'+$P(IBTRND,U,4) S X="1^Claims Tracking event does not have an associated Outpatient Visit." G EVNTCQ  I +$P(IBTRND,U,18)=4,'+$P(IBTRND,U,8) S X="1^Claims Tracking event does not have an associated prescription in Pharmacy." G EVNTCQ  I +$P(IBTRND,U,18)=4,$P(IBTRND,U,10)="" S X="1^Claims Tracking event does not have an associated prescription refill in Pharmacy." G EVNTCQ  ;  S DFN=+$P(IBTRND,U,2),IBEVDT=$P(IBTRND,U,6) I '$$INSURED^IBCNS1(DFN,IBEVDT) S X="1^Patient not insured for event date." G EVNTCQ  ; Check filing timeframe  I '$$PTFTF^IBCNSU31(DFN,IBEVDT) S X="1^Filing timeframe not met" G EVNTCQ  S IBCAT=$S($P(IBTRND,U,18)=1!($P(IBTRND,U,18)=5):"INPATIENT",$P(IBTRND,U,18)=2:"OUTPATIENT",$P(IBTRND,U,18)=4:"PHARMACY",1:"")  I IBCAT'="",'$$PTCOV^IBCNSU3(DFN,IBEVDT,IBCAT) S X="1^Patient insurance does not cover "\_IBCAT\_"." G EVNTCQ  D ELIG^VADPT S X=0 I 'VAEL(4) S X="1^Patient is not a veteran." G EVNTCQ  ;  ;check the last disposition before the episode to see if maybe workers comp or tort feasor  S IBX=9999999-(IBEVDT\1+1),IBX=$O(^DPT(+DFN,"DIS",IBX)) I +IBX S IBY=$$DT(IBX),IBX=$G(^DPT(DFN,"DIS",IBX,2)) D  G:+X EVNTCQ  . I $P(IBX,U,1)="Y" S X="1^Need may be related to occupation, check "\_IBY\_" disposition." Q  . I $P(IBX,U,4)="Y" S X="1^Need may be related to an accident, check "\_IBY\_" disposition." Q  ;  I +$P(IBTRND,U,5) S IBX=$G(^DGPM(+$P(IBTRND,U,5),0)) D  G EVNTCQ ; inpatient specific  . I IBX="" S X="1^Inpatient admission movement not found." Q  . I +$P(IBX,U,11) S X="1^Admitted for an SC condition." Q  ;  I +$P(IBTRND,U,4) S IBX=$$SCE^IBSDU(+$P(IBTRND,U,4)) D  G EVNTCQ ; outpatient specific  . I IBX="" S X="1^Outpatient Encounter not found." Q  . S IBY=$$NBOE^IBCU81(+$P(IBTRND,U,4),IBX) I +IBY D  Q:+X  .. ;I +IBY=1 S X="1^Service Connected visit." Q  .. I +IBY=2 S X="1^Non-billable Stop Code." Q  .. I +IBY=3 S X="1^Non-billable Clinic." Q  .. I +IBY=4 S X="1^Non-billable Status: "\_$P(IBY,U,2) Q  . ; dental is generally billed differently  . I $P($G(^DIC(40.7,+$P(IBX,U,3),0)),U,1)["DENTAL" S X="1^Outpatient visit contains a dental stop code." Q  . ;outpatient visit was a disposition: application without exam is not billable  . I $P(IBX,U,8)=3 D  Q:X  .. S IBY=$$DISND^IBSDU(+$P(IBTRND,U,4),IBX) ; 0-node of "DIS"  .. I $P(IBY,U,2)=2 S X="1^Disposition was Application Without Exam." Q  .. I $P($G(^DIC(37,+$P(IBY,U,7),0)),U,1)="CANCEL WITHOUT EXAM" S X="1^Disposition was Cancel Without Exam." Q  . ;can not bill twice for same day so ignore outpatient visits if patient was an inpatient at end of day (this means that outpatient visits on the date of discharge will be billed)  . I $$ADM^IBCU64(DFN,IBEVDT) S X="1^Not Billable: Patient was an inpatient on this visit date."  ;  I +$P(IBTRND,U,8) S IBX=$$RXZERO^IBRXUTL(+$P(IBTRND,U,2),+$P(IBTRND,U,8)) D  G EVNTCQ ; rx refills  . I IBX="" S X="1^Prescription not found in Pharmacy." Q  . I +$P(IBTRND,U,10)>0 S IBY=$$ZEROSUB^IBRXUTL(+$P(IBTRND,U,2),+$P(IBTRND,U,8),+$P(IBTRND,U,10)) I IBY="" S X="1^Prescription refill not found in Pharmacy." Q  . S IBZ=$$DBLCHK^IBTRKR31(IBTRN) I 'IBZ S X="1^Can not auto bill this refill, check Claims Tracking." Q EVNTCQ Q X  ; DT(X) ;convert disposition type date/time to external format (9999999-date)  N Y S Y=0 I +X S Y=9999999-X X ^DD("DD")  Q Y | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCU82 ;ALB/ARH - THIRD PARTY BILLING UTILITIES (AUTOMATED BILLER) ;02 JUL 93  ;;2.0;INTEGRATED BILLING;\*\*43,55,91,124,160,304,347,432,**592**\*\*;21-MAR-94;Build 192  ;;Per VHA Directive 2004-038, this routine should not be modified.  ;  ; EVNTCHK(IBTRN) ;special checks to determine if event should be auto billed  ;checks for INS, non-veteran patient, possible workers comp and tort feasor, admitted for sc cond., outp dental stop, optv while inpt, category covered by ins, non-billable stop or clinic  ;(assumes that Claims Tracking does the SC check for Outpatients)  ;input: IBTRN - claims tracking event  ; DISP - if true then any error message will be displayed on exit.  ;output: returns "1^error message" if one of the checks failed, 0 otherwise  ;  N X,IBX,IBY,IBZ,IBTRND,IBCAT,IBCOV,DFN,IBEVDT,VAEL,VADMVT,VAINDT S X=0,IBTRND=$G(^IBT(356,+$G(IBTRN),0)) G:IBTRND="" EVNTCQ  I +$P(IBTRND,U,18)=1,'+$P(IBTRND,U,5) S X="1^Claims Tracking event does not have an associated Inpatient Admission." G EVNTCQ  I +$P(IBTRND,U,18)=2,'+$P(IBTRND,U,4) S X="1^Claims Tracking event does not have an associated Outpatient Visit." G EVNTCQ  I +$P(IBTRND,U,18)=4,'+$P(IBTRND,U,8) S X="1^Claims Tracking event does not have an associated prescription in Pharmacy." G EVNTCQ  I +$P(IBTRND,U,18)=4,$P(IBTRND,U,10)="" S X="1^Claims Tracking event does not have an associated prescription refill in Pharmacy." G EVNTCQ  ;  S DFN=+$P(IBTRND,U,2),IBEVDT=$P(IBTRND,U,6) I '$$INSURED^IBCNS1(DFN,IBEVDT) S X="1^Patient not insured for event date." G EVNTCQ  ; Check filing timeframe  I '$$PTFTF^IBCNSU31(DFN,IBEVDT) S X="1^Filing timeframe not met" G EVNTCQ  S IBCAT=$S($P(IBTRND,U,18)=1!($P(IBTRND,U,18)=5):"INPATIENT",$P(IBTRND,U,18)=2:"OUTPATIENT",$P(IBTRND,U,18)=4:"PHARMACY",1:"")  I IBCAT'="",'$$PTCOV^IBCNSU3(DFN,IBEVDT,IBCAT) S X="1^Patient insurance does not cover "\_IBCAT\_"." G EVNTCQ  D ELIG^VADPT S X=0 I 'VAEL(4) S X="1^Patient is not a veteran." G EVNTCQ  ;  ;check the last disposition before the episode to see if maybe workers comp or tort feasor  S IBX=9999999-(IBEVDT\1+1),IBX=$O(^DPT(+DFN,"DIS",IBX)) I +IBX S IBY=$$DT(IBX),IBX=$G(^DPT(DFN,"DIS",IBX,2)) D  G:+X EVNTCQ  . I $P(IBX,U,1)="Y" S X="1^Need may be related to occupation, check "\_IBY\_" disposition." Q  . I $P(IBX,U,4)="Y" S X="1^Need may be related to an accident, check "\_IBY\_" disposition." Q  ;  I +$P(IBTRND,U,5) S IBX=$G(^DGPM(+$P(IBTRND,U,5),0)) D  G EVNTCQ ; inpatient specific  . I IBX="" S X="1^Inpatient admission movement not found." Q  . I +$P(IBX,U,11) S X="1^Admitted for an SC condition." Q  ;  I +$P(IBTRND,U,4) S IBX=$$SCE^IBSDU(+$P(IBTRND,U,4)) D  G EVNTCQ ; outpatient specific  . I IBX="" S X="1^Outpatient Encounter not found." Q  . S IBY=$$NBOE^IBCU81(+$P(IBTRND,U,4),IBX) I +IBY D  Q:+X  .. ;I +IBY=1 S X="1^Service Connected visit." Q  .. I +IBY=2 S X="1^Non-billable Stop Code." Q  .. I +IBY=3 S X="1^Non-billable Clinic." Q  .. I +IBY=4 S X="1^Non-billable Status: "\_$P(IBY,U,2) Q  . ; dental is generally billed differently **. ;JWS;IB\*2.0\*592;US1109;allow dental events to be processed and billed.**  . **;;**I $P($G(^DIC(40.7,+$P(IBX,U,3),0)),U,1)["DENTAL" S X="1^Outpatient visit contains a dental stop code." Q  **. ;JWS;IB\*2.0\*592;USXXXX;added ability to turn off Dental Claims processing in site parameters**  . **I $P(^IBE(350.9,1,8),U,20)=0 S X="1^Dental Claims processing is disabled in IB Site Parameters." Q**  . ;outpatient visit was a disposition: application without exam is not billable  . I $P(IBX,U,8)=3 D  Q:X  .. S IBY=$$DISND^IBSDU(+$P(IBTRND,U,4),IBX) ; 0-node of "DIS"  .. I $P(IBY,U,2)=2 S X="1^Disposition was Application Without Exam." Q  .. I $P($G(^DIC(37,+$P(IBY,U,7),0)),U,1)="CANCEL WITHOUT EXAM" S X="1^Disposition was Cancel Without Exam." Q  . ;can not bill twice for same day so ignore outpatient visits if patient was an inpatient at end of day (this means that outpatient visits on the date of discharge will be billed)  . I $$ADM^IBCU64(DFN,IBEVDT) S X="1^Not Billable: Patient was an inpatient on this visit date."  ;  I +$P(IBTRND,U,8) S IBX=$$RXZERO^IBRXUTL(+$P(IBTRND,U,2),+$P(IBTRND,U,8)) D  G EVNTCQ ; rx refills  . I IBX="" S X="1^Prescription not found in Pharmacy." Q  . I +$P(IBTRND,U,10)>0 S IBY=$$ZEROSUB^IBRXUTL(+$P(IBTRND,U,2),+$P(IBTRND,U,8),+$P(IBTRND,U,10)) I IBY="" S X="1^Prescription refill not found in Pharmacy." Q  . S IBZ=$$DBLCHK^IBTRKR31(IBTRN) I 'IBZ S X="1^Can not auto bill this refill, check Claims Tracking." Q EVNTCQ Q X  ; DT(X) ;convert disposition type date/time to external format (9999999-date)  N Y S Y=0 I +X S Y=9999999-X X ^DD("DD")  Q Y | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCU9 | | | | | | | | |
| **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** | | | | | | | | | |
| IBCU9 ;ALB/BI - BILLING UTILITY ROUTINE (CONTINUED) ;01 JUL 2011 11:13  ;;2.0;INTEGRATED BILLING;\*\*447\*\*;01-JUL-2011;Build 80  ;;Per VHA Directive 2004-038, this routine should not be modified.  Q  ; CMAEDALL(IBIEN) ; Clear all manually edited flags for a claim.  N IBRCIEN S IBRCIEN=0  F  S IBRCIEN=$O(^DGCR(399,IBIEN,"RC",IBRCIEN)) Q:+IBRCIEN=0 D  . D CMAEDIND(IBIEN,IBRCIEN)  Q  ; CMAEDIND(IBIEN,IBRCIEN) ; Clear individual manually edited flags for a revenue code.  S $P(^DGCR(399,IBIEN,"RC",IBRCIEN,0),U,16)=""  Q  ; FROMPROC(IBIEN,IBCPIEN,IBFLG) ; Clear individual manually edited flag if procedures match.  I $G(IBIEN)="" Q  I $G(IBCPIEN)="" Q  I $G(IBFLG)="" Q  I IBFLG="E",IBCPIEN=$O(^DGCR(399,IBIEN,"CP",0)) D CMAEDALL(IBIEN) Q  I IBFLG="D",IBCPIEN=$O(^DGCR(399,IBIEN,"CP",0)) D PROC1DEL(IBIEN) Q  N IBRC0,IBRCPRSP  N IBRCIEN S IBRCIEN=0  F  S IBRCIEN=$O(^DGCR(399,IBIEN,"RC",IBRCIEN)) Q:+IBRCIEN=0 D  . S IBRC0=$G(^DGCR(399,IBIEN,"RC",IBRCIEN,0)),IBRCPRSP=$P(IBRC0,U,11)  . I IBRCPRSP=IBCPIEN D CMAEDIND(IBIEN,IBRCIEN)  Q  ; PROC1DEL(IBIEN) ; The first procedure was deleted, determine division change.  N IBCPIEN1,IBCPIEN2  S IBCPIEN1=$O(^DGCR(399,IBIEN,"CP",0)) I IBCPIEN1="" Q  S IBCPIEN2=$O(^DGCR(399,IBIEN,"CP",IBCPIEN1)) I IBCPIEN2="" D CMAEDALL(IBIEN) Q  I $P($G(^DGCR(399,IBIEN,"CP",IBCPIEN1,0)),U,6)'=$P($G(^DGCR(399,IBIEN,"CP",IBCPIEN2,0)),U,6) D CMAEDALL(IBIEN)  Q | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCU9 ;ALB/BI - BILLING UTILITY ROUTINE (CONTINUED) ;01 JUL 2011 11:13  ;;2.0;INTEGRATED BILLING;\*\*447,592\*\*;01-JUL-2011;Build 80  ;;Per VHA Directive 2004-038, this routine should not be modified.  Q  ; CMAEDALL(IBIEN) ; Clear all manually edited flags for a claim.  N IBRCIEN S IBRCIEN=0  F  S IBRCIEN=$O(^DGCR(399,IBIEN,"RC",IBRCIEN)) Q:+IBRCIEN=0 D  . D CMAEDIND(IBIEN,IBRCIEN)  Q  ; CMAEDIND(IBIEN,IBRCIEN) ; Clear individual manually edited flags for a revenue code.  S $P(^DGCR(399,IBIEN,"RC",IBRCIEN,0),U,16)=""  Q  ; FROMPROC(IBIEN,IBCPIEN,IBFLG) ; Clear individual manually edited flag if procedures match.  I $G(IBIEN)="" Q  I $G(IBCPIEN)="" Q  I $G(IBFLG)="" Q  I IBFLG="E",IBCPIEN=$O(^DGCR(399,IBIEN,"CP",0)) D CMAEDALL(IBIEN) Q  I IBFLG="D",IBCPIEN=$O(^DGCR(399,IBIEN,"CP",0)) D PROC1DEL(IBIEN) Q  N IBRC0,IBRCPRSP  N IBRCIEN S IBRCIEN=0  F  S IBRCIEN=$O(^DGCR(399,IBIEN,"RC",IBRCIEN)) Q:+IBRCIEN=0 D  . S IBRC0=$G(^DGCR(399,IBIEN,"RC",IBRCIEN,0)),IBRCPRSP=$P(IBRC0,U,11)  . I IBRCPRSP=IBCPIEN D CMAEDIND(IBIEN,IBRCIEN)  Q  ; PROC1DEL(IBIEN) ; The first procedure was deleted, determine division change.  N IBCPIEN1,IBCPIEN2  S IBCPIEN1=$O(^DGCR(399,IBIEN,"CP",0)) I IBCPIEN1="" Q  S IBCPIEN2=$O(^DGCR(399,IBIEN,"CP",IBCPIEN1)) I IBCPIEN2="" D CMAEDALL(IBIEN) Q  I $P($G(^DGCR(399,IBIEN,"CP",IBCPIEN1,0)),U,6)'=$P($G(^DGCR(399,IBIEN,"CP",IBCPIEN2,0)),U,6) D CMAEDALL(IBIEN)  Q  ;  **;JWS;IB\*2.0\*592;US1109 Dental FTINPUT(Y) ;SCREEN FOR 399, .19 FORM TYPE  N Z  I Y=7,$P($G(^IBE(350.9,1,8)),U,20)=0 Q 0  S Z=$G(^IBE(353,Y,2)) I $P(Z,U,2)="P",$P(Z,U,4) Q 1  Q 0**  ; | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBJPS | | | | | | | | |
| **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** | | | | | | | | | |
| IBJPS ;ALB/MAF,ARH - IBSP IB SITE PARAMETER SCREEN ;22-DEC-1995  ;;2.0;INTEGRATED BILLING;\*\*39,52,70,115,143,51,137,161,155,320,348,349,377,384,400,432,494,461,516,547\*\*;21-MAR-94;Build 119  ;;Per VA Directive 6402, this routine should not be modified.  ; EN ; -- main entry point for IBJP IB SITE PARAMETERS, display IB site parameters  D EN^VALM("IBJP IB SITE PARAMETERS")  Q  ; HDR ; -- header code  S VALMHDR(1)="Only authorized persons may edit this data."  Q  ; INIT ; -- init variables and list array  K ^TMP("IBJPS",$J),^TMP("IBJPSAX",$J)  D BLD^IBJPS1  Q  ; HELP ; -- help code  S X="?" D DISP^XQORM1 W !!  Q  ; EXIT ; -- exit code  K ^TMP("IBJPS",$J),^TMP("IBJPSAX",$J)  D CLEAR^VALM1  Q  ; NXEDIT ; -- IBJP IB SITE PARAMETER EDIT ACTION (EP): Select data set to edit, do edit  N VALMY,IBSELN,IBSET  D EN^VALM2($G(XQORNOD(0)))  I $D(VALMY) S IBSELN=0 F  S IBSELN=$O(VALMY(IBSELN)) Q:'IBSELN  D  . S IBSET=$P($G(^TMP("IBJPSAX",$J,IBSELN)),U,1) Q:'IBSET  . D EDIT(IBSET)  S VALMBCK="R"  Q  ; EDIT(IBSET) ; edit IB Site Parameters  D FULL^VALM1  N DR  I IBSET'="" D  . ; MRD;IB\*2.0\*516 - Added TRICARE Pay-To Providers.  . ; WCJ;IB\*2.0\*547 - shifted the numbers down to insert a new one  . I IBSET=8 D EN^IBJPS5 Q  . I IBSET=11 D EN^IBJPS3(0) Q  . I IBSET=12 D EN^IBJPS3(1) Q  . ;WCJ;IB\*2.0\*547 added default Administrative contractors for billing (medicare and commercial)  . I IBSET=17 D EN^IBJPS6(1) Q   ; medicare  . I IBSET=18 D EN^IBJPS6(2) Q   ; commercial  . S DR=$P($T(@IBSET),";;",2,999)  . Q  ; WCJ;IB\*2.0\*547 - shifted the number down to insert a new one  I IBSET=9,$$ICD9SYS^IBACSV(DT)=30 S $P(DR,";",1)=7.05  ;  I $G(DR)'="" S DIE="^IBE(350.9,",DA=1 D ^DIE K DA,DR,DIE,DIC,X,Y  D INIT^IBJPS S VALMBCK="R"  Q  ;  ;WCJ;IB\*2.0\*547 - cleared the spot for the new #8, added 17 & 18, move 16 to 19.  ;gef;IB\*2.0\*547 - added 20 1 ;;.09;.13;.14 2 ;;1.2;.15;.11;.12;7.04 3 ;;1.09;1.07;2.07 4 ;;4.04;6.25;6.24 5 ;;.02;1.14;1.25;1.08 6 ;;1.23;1.16;1.22;1.19;1.15;1.17 7 ;;1.33;1.32;1.31;1.27;8.14T;8.15T;8.16T;8.19T 9 ;;1.29;1.3;1.18;1.28 10 ;;1.01;1.02;1.05 13 ;;2.08;2.09 14 ;;11.01 15 ;;10.02;10.03;10.04;10.05;D INIT^IBATFILE 16 ;;2.11;8.01;8.09;8.03;8.06;8.04;8.07;8.02;8.12T;8.11T;8.17T 19 ;;50.01;50.02;50.05;50.06;50.03;50.04;50.07 20 ;;52.01;52.02  ; | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBJPS ;ALB/MAF,ARH - IBSP IB SITE PARAMETER SCREEN ;22-DEC-1995  ;;2.0;INTEGRATED BILLING;\*\*39,52,70,115,143,51,137,161,155,320,348,349,377,384,400,432,494,461,516,547,**592**\*\*;21-MAR-94;Build 119  ;;Per VA Directive 6402, this routine should not be modified.  ; EN ; -- main entry point for IBJP IB SITE PARAMETERS, display IB site parameters  D EN^VALM("IBJP IB SITE PARAMETERS")  Q  ; HDR ; -- header code  S VALMHDR(1)="Only authorized persons may edit this data."  Q  ; INIT ; -- init variables and list array  K ^TMP("IBJPS",$J),^TMP("IBJPSAX",$J)  D BLD^IBJPS1  Q  ; HELP ; -- help code  S X="?" D DISP^XQORM1 W !!  Q  ; EXIT ; -- exit code  K ^TMP("IBJPS",$J),^TMP("IBJPSAX",$J)  D CLEAR^VALM1  Q  ; NXEDIT ; -- IBJP IB SITE PARAMETER EDIT ACTION (EP): Select data set to edit, do edit  N VALMY,IBSELN,IBSET  D EN^VALM2($G(XQORNOD(0)))  I $D(VALMY) S IBSELN=0 F  S IBSELN=$O(VALMY(IBSELN)) Q:'IBSELN  D  . S IBSET=$P($G(^TMP("IBJPSAX",$J,IBSELN)),U,1) Q:'IBSET  . D EDIT(IBSET)  S VALMBCK="R"  Q  ; EDIT(IBSET) ; edit IB Site Parameters  D FULL^VALM1  N DR  I IBSET'="" D  . ; MRD;IB\*2.0\*516 - Added TRICARE Pay-To Providers.  . ; WCJ;IB\*2.0\*547 - shifted the numbers down to insert a new one  . I IBSET=8 D EN^IBJPS5 Q  . I IBSET=11 D EN^IBJPS3(0) Q  . I IBSET=12 D EN^IBJPS3(1) Q  . ;WCJ;IB\*2.0\*547 added default Administrative contractors for billing (medicare and commercial)  . I IBSET=17 D EN^IBJPS6(1) Q   ; medicare  . I IBSET=18 D EN^IBJPS6(2) Q   ; commercial  . S DR=$P($T(@IBSET),";;",2,999)  . Q  ; WCJ;IB\*2.0\*547 - shifted the number down to insert a new one  I IBSET=9,$$ICD9SYS^IBACSV(DT)=30 S $P(DR,";",1)=7.05  ;  I $G(DR)'="" S DIE="^IBE(350.9,",DA=1 D ^DIE K DA,DR,DIE,DIC,X,Y  D INIT^IBJPS S VALMBCK="R"  Q  ;  ;WCJ;IB\*2.0\*547 - cleared the spot for the new #8, added 17 & 18, move 16 to 19.  ;gef;IB\*2.0\*547 - added 20  **;JWS;IB\*2.0\*592 - added field 8.2 to 16** 1 ;;.09;.13;.14 2 ;;1.2;.15;.11;.12;7.04 3 ;;1.09;1.07;2.07 4 ;;4.04;6.25;6.24 5 ;;.02;1.14;1.25;1.08 6 ;;1.23;1.16;1.22;1.19;1.15;1.17 7 ;;1.33;1.32;1.31;1.27;8.14T;8.15T;8.16T;8.19T 9 ;;1.29;1.3;1.18;1.28 10 ;;1.01;1.02;1.05 13 ;;2.08;2.09 14 ;;11.01 15 ;;10.02;10.03;10.04;10.05;D INIT^IBATFILE 16 ;;2.11;8.01;8.09;8.03;8.06;8.04;8.07;8.02;8.12T;8.11T;8.17T;**8.2T** 19 ;;50.01;50.02;50.05;50.06;50.03;50.04;50.07 20 ;;52.01;52.02  ; | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBJPS2 | | | | | | | | |
| **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** | | | | | | | | | |
| IBJPS2 ;ALB/MAF,ARH - IBSP IB SITE PARAMETER BUILD (cont) ;22-DEC-1995  ;;2.0;INTEGRATED BILLING;\*\*39,52,115,143,51,137,161,155,320,348,349,377,384,400,432,494,461,516,547\*\*;21-MAR-94;Build 119  ;;Per VA Directive 6402, this routine should not be modified.  ; BLD2 ; - continue build screen array for IB parameters  ;  N Z,Z0,PTPSTR,BPZZ  D RIGHT(1,1,1) ; - facility/med center (new line for each)  S IBLN=$$SET("Medical Center",$$EXSET^IBJU1($P(IBPD0,U,2),350.9,.02),IBLN,IBLR,IBSEL)  S IBLN=$$SET("MAS Service",$$EXSET^IBJU1($P(IBPD1,U,14),350.9,1.14),IBLN,IBLR,IBSEL)  ;  D LEFT(2)  S IBLN=$$SET("Default Division",$$EXSET^IBJU1($P(IBPD1,U,25),350.9,1.25),IBLN,IBLR,IBSEL)  S IBLN=$$SET("Billing Supervisor",$$EXSET^IBJU1($P(IBPD1,U,8),350.9,1.08),IBLN,IBLR,IBSEL)  ;  D RIGHT(1,1,1)  S IBLN=$$SET("Initiator Authorize",$$YN(+$P(IBPD1,U,23)),IBLN,IBLR,IBSEL)  S IBLN=$$SET("Ask HINQ in MCCR",$$YN(+$P(IBPD1,U,16)),IBLN,IBLR,IBSEL)  S IBLN=$$SET("Multiple Form Types",$$YN(+$P(IBPD1,U,22)),IBLN,IBLR,IBSEL)  ;  D LEFT(2)  S IBLN=$$SET("Xfer Proc to Sched",$$YN(+$P(IBPD1,U,19)),IBLN,IBLR,IBSEL)  S IBLN=$$SET("Use Non-PTF Codes",$$YN(+$P(IBPD1,U,15)),IBLN,IBLR,IBSEL)  S IBLN=$$SET("Use OP CPT screen",$$YN(+$P(IBPD1,U,17)),IBLN,IBLR,IBSEL)  ;  ; IB patch 349 for UB-04 claim form and parameters  D RIGHT(1,1,1)  S IBLN=$$SET("UB-04 Print IDs",$$EXSET^IBJU1($P(IBPD1,U,33),350.9,1.33),IBLN,IBLR,IBSEL)  S IBLN=$$SET("CMS-1500 Print IDs",$$EXSET^IBJU1($P(IBPD1,U,32),350.9,1.32),IBLN,IBLR,IBSEL)  S IBLN=$$SET("CMS-1500 Auto Prter",$$EXSET^IBJU1($P(IBPD8,U,14),350.9,8.14),IBLN,IBLR,IBSEL)  S IBLN=$$SET("EOB Auto Prter",$$EXSET^IBJU1($P(IBPD8,U,16),350.9,8.16),IBLN,IBLR,IBSEL)  ;  D LEFT(2)  S IBLN=$$SET("UB-04 Address Col",$P(IBPD1,U,31),IBLN,IBLR,IBSEL)  S IBLN=$$SET("CMS-1500 Addr Col",$P(IBPD1,U,27),IBLN,IBLR,IBSEL)  S IBLN=$$SET("UB-04 Auto Prter",$$EXSET^IBJU1($P(IBPD8,U,15),350.9,8.15),IBLN,IBLR,IBSEL)  S IBLN=$$SET("MRA Auto Prter",$$EXSET^IBJU1($P(IBPD8,U,19),350.9,8.19),IBLN,IBLR,IBSEL)  ;  ; VAD - IB\*2.0\*547 - inserted a new section 8. Only count activated codes  D RIGHT(3,1,1)  S (Z,Z0)=0 F  S Z=$O(^IBE(350.9,1,15,"B",Z)) Q:'Z  I $P($G(^DGCR(399.2,Z,0)),U,3)=1 S Z0=Z0+1  S PTPSTR=Z0\_" Activated Codes Defined"  S IBLN=$$SET("Printed Claims Rev Code Excl",PTPSTR,IBLN,IBLR,IBSEL)  ;  D RIGHT(1,1,1)  S Z=$$ICD9SYS^IBACSV(DT)  I Z=1 S IBLN=$$SET("Default RX DX Cd",$$EXSET^IBJU1($P(IBPD1,U,29),350.9,1.29)\_" (ICD-9)",IBLN,IBLR,IBSEL)  I Z'=1 S IBLN=$$SET("Default RX DX Cd",$$EXSET^IBJU1($P(IBPD7,U,5),350.9,7.05)\_" (ICD-10)",IBLN,IBLR,IBSEL)  S IBLN=$$SET("Default RX CPT Cd",$$EXSET^IBJU1($P(IBPD1,U,30),350.9,1.30),IBLN,IBLR,IBSEL)  ;  D LEFT(2)  S IBLN=$$SET("Default ASC Rev Cd",$$EXSET^IBJU1($P(IBPD1,U,18),350.9,1.18),IBLN,IBLR,IBSEL)  S IBLN=$$SET("Default RX Rev Cd",$$EXSET^IBJU1($P(IBPD1,U,28),350.9,1.28),IBLN,IBLR,IBSEL)  ;  D RIGHT(1,1,1)  S IBLN=$$SET("Bill Signer Name","<No longer used>",IBLN,IBLR,IBSEL)  S IBLN=$$SET("Bill Signer Title","<No longer used>",IBLN,IBLR,IBSEL)  ;  D LEFT(2)  S IBLN=$$SET("Federal Tax #",$P(IBPD1,U,5),IBLN,IBLR,IBSEL)  ;  D RIGHT(3,1,1) ; - Pay-To Providers - section 11  S (Z,Z0)=0 F  S Z=$O(^IBE(350.9,1,19,Z)) Q:'Z  S:$P($G(^IBE(350.9,1,19,Z,0)),U,5)="" Z0=Z0+1  S Z=+$P($G(^IBE(350.9,1,11)),U,3),PTPSTR=Z0\_" defined"\_$S(Z>0:", default - "\_$P($$PTG^IBJPS3(Z,0),U),1:"")  S IBLN=$$SET("Pay-To Providers",PTPSTR,IBLN,IBLR,IBSEL)  ;  ; MRD;IB\*2.0\*516 - Added TRICARE Pay-To Providers.  D RIGHT(3,1,1) ; - TRICARE Pay-To Providers - section 12  S (Z,Z0)=0 F  S Z=$O(^IBE(350.9,1,29,Z)) Q:'Z  S:$P($G(^IBE(350.9,1,29,Z,0)),U,5)="" Z0=Z0+1  S Z=+$P($G(^IBE(350.9,1,11)),U,4),PTPSTR=Z0\_" defined"\_$S(Z>0:", default - "\_$P($$PTG^IBJPS3(Z,1),U),1:"")  S IBLN=$$SET("TRICARE Pay-To Providers",PTPSTR,IBLN,IBLR,IBSEL)  ;  D RIGHT(3,1,1)  S IBLN=$$SET("Inpt Health Summary",$$EXSET^IBJU1($P(IBPD2,U,8),350.9,2.08),IBLN,IBLR,IBSEL)  S IBLN=$$SET("Opt Health Summary",$$EXSET^IBJU1($P(IBPD2,U,9),350.9,2.09),IBLN,IBLR,IBSEL)  ;  ; ePharmacy parameters  D RIGHT(7,1,1)  S IBLN=$$SET("HIPPA NCPDP Active Flag",$S($P(IBPD11,U)=1:"Active",1:"Not Active"),IBLN,IBLR,IBSEL)  ;  ; transfer pricing  D RIGHT(1,1,1)  S IBLN=$$SET("Inpatient TP Active ",$$YN(+$P(IBPD10,U,2)),IBLN,IBLR,IBSEL)  S IBLN=$$SET("Outpatient TP Active",$$YN(+$P(IBPD10,U,3)),IBLN,IBLR,IBSEL)  S IBLN=$$SET("Pharmacy TP Active ",$$YN(+$P(IBPD10,U,4)),IBLN,IBLR,IBSEL)  S IBLN=$$SET("Prosthetic TP Active",$$YN(+$P(IBPD10,U,5)),IBLN,IBLR,IBSEL)  ;  ; EDI/MRA parameters  D RIGHT(7,1,1)  N IBZ S IBZ=$P(IBPD8,U,3)  S IBLN=$$SET(" EDI/MRA Activated",$$EXSET^IBJU1(+$P(IBPD8,U,10),350.9,8.1),IBLN,IBLR,IBSEL)  S IBLN=$$SET(" EDI Contact Phone",$P(IBPD2,U,11),IBLN,IBLR,IBSEL)  S IBLN=$$SET(" EDI 837 Live Transmit Queue",$P(IBPD8,U),IBLN,IBLR,IBSEL)  S IBLN=$$SET(" EDI 837 Test Transmit Queue",$P(IBPD8,U,9),IBLN,IBLR,IBSEL)  S IBLN=$$SET(" Auto-Txmt Bill Frequency",$S(IBZ:"Every"\_$S(IBZ>1:" "\_$P(IBPD8,U,3),1:""),1:"")\_$S(IBZ:" Day"\_$S(IBZ=1:"",1:"s"),1:"Never Run"),IBLN,IBLR,IBSEL)  S IBLN=$$SET(" Hours To Auto-Transmit",$P(IBPD8,U,6),IBLN,IBLR,IBSEL)  S IBLN=$$SET(" Max # Bills Per Batch",$P(IBPD8,U,4),IBLN,IBLR,IBSEL)  S IBLN=$$SET(" Only Allow 1 Ins Co/Claim Batch?",$$EXPAND^IBTRE(350.9,8.07,+$P(IBPD8,U,7)),IBLN,IBLR,IBSEL)  S IBLN=$$SET(" Last Auto-Txmt Run Date",$$DATE^IBJU1($P(IBPD8,U,5)),IBLN,IBLR,IBSEL)  S IBLN=$$SET(" Days To Wait To Purge Msgs",$P(IBPD8,U,2),IBLN,IBLR,IBSEL)  S IBLN=$$SET(" Allow MRA Processing?",$$YN(+$P(IBPD8,U,12)),IBLN,IBLR,IBSEL)  S IBLN=$$SET(" Enable Automatic MRA Processing?",$$YN(+$P(IBPD8,U,11)),IBLN,IBLR,IBSEL)  S IBLN=$$SET(" Enable Auto Reg EOB Processing?",$$YN(+$P(IBPD8,U,17)),IBLN,IBLR,IBSEL)  ;  ; WCJ;IB\*2.0\*547;administrative contractors medicare  D RIGHT(3,1,1)  S Z=+$P($G(^IBE(350.9,1,81,0)),U,4)\_" defined"  S IBLN=$$SET("Alt Prim Payer ID Typ-Medicare",Z,IBLN,IBLR,IBSEL)  ;  ; WCJ;IB\*2.0\*547;administrative contractors commercial  D RIGHT(3,1,1)  S Z=+$P($G(^IBE(350.9,1,82,0)),U,4)\_" defined"  S IBLN=$$SET("Alt Prim Payer ID Typ-Commercial",Z,IBLN,IBLR,IBSEL)  ;  ; Ingenix ClaimsManager Information  D RIGHT(9,1,1)  S IBLN=$$SET("Are we using ClaimsManager?",$$YN(+$P(IBPD50,U,1)),IBLN,IBLR,IBSEL)  S IBLN=$$SET("Is ClaimsManager working OK?",$$YN(+$P(IBPD50,U,2)),IBLN,IBLR,IBSEL)  S IBLN=$$SET("ClaimsManager TCP/IP Address",$P(IBPD50,U,5),IBLN,IBLR,IBSEL)  S IBCISOCK=$O(^IBE(350.9,1,50.06,"B",""))  S IBLN=$$SET("ClaimsManager TCP/IP Ports",IBCISOCK,IBLN,IBLR,IBSEL)  F  S IBCISOCK=$O(^IBE(350.9,1,50.06,"B",IBCISOCK)) Q:IBCISOCK=""  D  . S IBLN=$$SET("",IBCISOCK,IBLN,IBLR,IBSEL)  . Q  S IBLN=$$SET("General Error MailGroup",$$EXSET^IBJU1($P(IBPD50,U,3),350.9,50.03),IBLN,IBLR,IBSEL)  S IBLN=$$SET("Communication Error MailGroup",$$EXSET^IBJU1($P(IBPD50,U,4),350.9,50.04),IBLN,IBLR,IBSEL)  S IBCIMFLG=$$EXTERNAL^DILFD(350.9,50.07,"",$P(IBPD50,U,7))  I IBCIMFLG="" S IBCIMFLG="PRIORITY"  S IBLN=$$SET("MailMan Messages",IBCIMFLG,IBLN,IBLR,IBSEL)  ;  ; Request For Additional Info patch 547  D RIGHT(9,1,1)  S Z=$G(^IBE(350.9,1,52)) S:$P(Z,U)="" $P(Z,U)="No Purge"  S IBLN=$$SET("Days to store 277RFAI Transactions",$P(Z,U),IBLN,IBLR,IBSEL)  S IBLN=$$SET("Days to wait to purge entry on RFAI Management Worklist",$P(Z,U,2),IBLN,IBLR,IBSEL)  Q  ; SET(TTL,DATA,LN,LR,SEL,HDR) ;  N IBY,IBX,IBC S IBC=": " I TTL="" S IBC=" "  S IBY=TTL\_$J("",(IBTW(LR)-$L(TTL)-2))\_$S('$G(HDR):IBC\_DATA,1:""),IBX=$G(^TMP("IBJPS",$J,LN,0))  S IBX=$$SETSTR^VALM1(IBY,IBX,IBTC(LR),(IBTW(LR)+IBSW(LR)))  D SET1(IBX,LN,SEL)  S LN=LN+1  Q LN  ; SET1(STR,LN,SEL,HI) ; set up TMP array with screen data  S ^TMP("IBJPS",$J,LN,0)=STR  S ^TMP("IBJPS",$J,"IDX",LN,SEL)=""  S ^TMP("IBJPSAX",$J,SEL)=SEL  I $G(HI)'="" D CNTRL^VALM10(LN,1,4,IOINHI,IOINORM)  ;I $G(RV) D CNTRL^VALM10(LN,6,19,IOUON,IOUOFF)  Q  ; YN(X) Q $S(+X:"YES",1:"NO")  ; RIGHT(LR,SEL,BL) ; - reset control variables for right side of screen  S IBLN=$S(IBLN>IBGRPE:IBLN,1:IBGRPE) I $G(BL) S IBLN=$$SET("","",IBLN,IBLR,IBSEL)  S IBLR=$G(LR),IBGRPB=IBLN I +$G(SEL) S IBSEL=IBSEL+1 D SET1("["\_IBSEL\_"]",IBLN,IBSEL,1)  Q  ; LEFT(LR) ; - reset control variables for left side of screen  S IBLR=$G(LR),IBGRPE=IBLN,IBLN=IBGRPB  Q | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBJPS2 ;ALB/MAF,ARH - IBSP IB SITE PARAMETER BUILD (cont) ;22-DEC-1995  ;;2.0;INTEGRATED BILLING;\*\*39,52,115,143,51,137,161,155,320,348,349,377,384,400,432,494,461,516,547,**592**\*\*;21-MAR-94;Build 119  ;;Per VA Directive 6402, this routine should not be modified.  ; BLD2 ; - continue build screen array for IB parameters  ;  N Z,Z0,PTPSTR,BPZZ  D RIGHT(1,1,1) ; - facility/med center (new line for each)  S IBLN=$$SET("Medical Center",$$EXSET^IBJU1($P(IBPD0,U,2),350.9,.02),IBLN,IBLR,IBSEL)  S IBLN=$$SET("MAS Service",$$EXSET^IBJU1($P(IBPD1,U,14),350.9,1.14),IBLN,IBLR,IBSEL)  ;  D LEFT(2)  S IBLN=$$SET("Default Division",$$EXSET^IBJU1($P(IBPD1,U,25),350.9,1.25),IBLN,IBLR,IBSEL)  S IBLN=$$SET("Billing Supervisor",$$EXSET^IBJU1($P(IBPD1,U,8),350.9,1.08),IBLN,IBLR,IBSEL)  ;  D RIGHT(1,1,1)  S IBLN=$$SET("Initiator Authorize",$$YN(+$P(IBPD1,U,23)),IBLN,IBLR,IBSEL)  S IBLN=$$SET("Ask HINQ in MCCR",$$YN(+$P(IBPD1,U,16)),IBLN,IBLR,IBSEL)  S IBLN=$$SET("Multiple Form Types",$$YN(+$P(IBPD1,U,22)),IBLN,IBLR,IBSEL)  ;  D LEFT(2)  S IBLN=$$SET("Xfer Proc to Sched",$$YN(+$P(IBPD1,U,19)),IBLN,IBLR,IBSEL)  S IBLN=$$SET("Use Non-PTF Codes",$$YN(+$P(IBPD1,U,15)),IBLN,IBLR,IBSEL)  S IBLN=$$SET("Use OP CPT screen",$$YN(+$P(IBPD1,U,17)),IBLN,IBLR,IBSEL)  ;  ; IB patch 349 for UB-04 claim form and parameters  D RIGHT(1,1,1)  S IBLN=$$SET("UB-04 Print IDs",$$EXSET^IBJU1($P(IBPD1,U,33),350.9,1.33),IBLN,IBLR,IBSEL)  S IBLN=$$SET("CMS-1500 Print IDs",$$EXSET^IBJU1($P(IBPD1,U,32),350.9,1.32),IBLN,IBLR,IBSEL)  S IBLN=$$SET("CMS-1500 Auto Prter",$$EXSET^IBJU1($P(IBPD8,U,14),350.9,8.14),IBLN,IBLR,IBSEL)  S IBLN=$$SET("EOB Auto Prter",$$EXSET^IBJU1($P(IBPD8,U,16),350.9,8.16),IBLN,IBLR,IBSEL)  ;  D LEFT(2)  S IBLN=$$SET("UB-04 Address Col",$P(IBPD1,U,31),IBLN,IBLR,IBSEL)  S IBLN=$$SET("CMS-1500 Addr Col",$P(IBPD1,U,27),IBLN,IBLR,IBSEL)  S IBLN=$$SET("UB-04 Auto Prter",$$EXSET^IBJU1($P(IBPD8,U,15),350.9,8.15),IBLN,IBLR,IBSEL)  S IBLN=$$SET("MRA Auto Prter",$$EXSET^IBJU1($P(IBPD8,U,19),350.9,8.19),IBLN,IBLR,IBSEL)  ;  ; VAD - IB\*2.0\*547 - inserted a new section 8. Only count activated codes  D RIGHT(3,1,1)  S (Z,Z0)=0 F  S Z=$O(^IBE(350.9,1,15,"B",Z)) Q:'Z  I $P($G(^DGCR(399.2,Z,0)),U,3)=1 S Z0=Z0+1  S PTPSTR=Z0\_" Activated Codes Defined"  S IBLN=$$SET("Printed Claims Rev Code Excl",PTPSTR,IBLN,IBLR,IBSEL)  ;  D RIGHT(1,1,1)  S Z=$$ICD9SYS^IBACSV(DT)  I Z=1 S IBLN=$$SET("Default RX DX Cd",$$EXSET^IBJU1($P(IBPD1,U,29),350.9,1.29)\_" (ICD-9)",IBLN,IBLR,IBSEL)  I Z'=1 S IBLN=$$SET("Default RX DX Cd",$$EXSET^IBJU1($P(IBPD7,U,5),350.9,7.05)\_" (ICD-10)",IBLN,IBLR,IBSEL)  S IBLN=$$SET("Default RX CPT Cd",$$EXSET^IBJU1($P(IBPD1,U,30),350.9,1.30),IBLN,IBLR,IBSEL)  ;  D LEFT(2)  S IBLN=$$SET("Default ASC Rev Cd",$$EXSET^IBJU1($P(IBPD1,U,18),350.9,1.18),IBLN,IBLR,IBSEL)  S IBLN=$$SET("Default RX Rev Cd",$$EXSET^IBJU1($P(IBPD1,U,28),350.9,1.28),IBLN,IBLR,IBSEL)  ;  D RIGHT(1,1,1)  S IBLN=$$SET("Bill Signer Name","<No longer used>",IBLN,IBLR,IBSEL)  S IBLN=$$SET("Bill Signer Title","<No longer used>",IBLN,IBLR,IBSEL)  ;  D LEFT(2)  S IBLN=$$SET("Federal Tax #",$P(IBPD1,U,5),IBLN,IBLR,IBSEL)  ;  D RIGHT(3,1,1) ; - Pay-To Providers - section 11  S (Z,Z0)=0 F  S Z=$O(^IBE(350.9,1,19,Z)) Q:'Z  S:$P($G(^IBE(350.9,1,19,Z,0)),U,5)="" Z0=Z0+1  S Z=+$P($G(^IBE(350.9,1,11)),U,3),PTPSTR=Z0\_" defined"\_$S(Z>0:", default - "\_$P($$PTG^IBJPS3(Z,0),U),1:"")  S IBLN=$$SET("Pay-To Providers",PTPSTR,IBLN,IBLR,IBSEL)  ;  ; MRD;IB\*2.0\*516 - Added TRICARE Pay-To Providers.  D RIGHT(3,1,1) ; - TRICARE Pay-To Providers - section 12  S (Z,Z0)=0 F  S Z=$O(^IBE(350.9,1,29,Z)) Q:'Z  S:$P($G(^IBE(350.9,1,29,Z,0)),U,5)="" Z0=Z0+1  S Z=+$P($G(^IBE(350.9,1,11)),U,4),PTPSTR=Z0\_" defined"\_$S(Z>0:", default - "\_$P($$PTG^IBJPS3(Z,1),U),1:"")  S IBLN=$$SET("TRICARE Pay-To Providers",PTPSTR,IBLN,IBLR,IBSEL)  ;  D RIGHT(3,1,1)  S IBLN=$$SET("Inpt Health Summary",$$EXSET^IBJU1($P(IBPD2,U,8),350.9,2.08),IBLN,IBLR,IBSEL)  S IBLN=$$SET("Opt Health Summary",$$EXSET^IBJU1($P(IBPD2,U,9),350.9,2.09),IBLN,IBLR,IBSEL)  ;  ; ePharmacy parameters  D RIGHT(7,1,1)  S IBLN=$$SET("HIPPA NCPDP Active Flag",$S($P(IBPD11,U)=1:"Active",1:"Not Active"),IBLN,IBLR,IBSEL)  ;  ; transfer pricing  D RIGHT(1,1,1)  S IBLN=$$SET("Inpatient TP Active ",$$YN(+$P(IBPD10,U,2)),IBLN,IBLR,IBSEL)  S IBLN=$$SET("Outpatient TP Active",$$YN(+$P(IBPD10,U,3)),IBLN,IBLR,IBSEL)  S IBLN=$$SET("Pharmacy TP Active ",$$YN(+$P(IBPD10,U,4)),IBLN,IBLR,IBSEL)  S IBLN=$$SET("Prosthetic TP Active",$$YN(+$P(IBPD10,U,5)),IBLN,IBLR,IBSEL)  ;  ; EDI/MRA parameters  D RIGHT(7,1,1)  N IBZ S IBZ=$P(IBPD8,U,3)  S IBLN=$$SET(" EDI/MRA Activated",$$EXSET^IBJU1(+$P(IBPD8,U,10),350.9,8.1),IBLN,IBLR,IBSEL)  S IBLN=$$SET(" EDI Contact Phone",$P(IBPD2,U,11),IBLN,IBLR,IBSEL)  S IBLN=$$SET(" EDI 837 Live Transmit Queue",$P(IBPD8,U),IBLN,IBLR,IBSEL)  S IBLN=$$SET(" EDI 837 Test Transmit Queue",$P(IBPD8,U,9),IBLN,IBLR,IBSEL)  S IBLN=$$SET(" Auto-Txmt Bill Frequency",$S(IBZ:"Every"\_$S(IBZ>1:" "\_$P(IBPD8,U,3),1:""),1:"")\_$S(IBZ:" Day"\_$S(IBZ=1:"",1:"s"),1:"Never Run"),IBLN,IBLR,IBSEL)  S IBLN=$$SET(" Hours To Auto-Transmit",$P(IBPD8,U,6),IBLN,IBLR,IBSEL)  S IBLN=$$SET(" Max # Bills Per Batch",$P(IBPD8,U,4),IBLN,IBLR,IBSEL)  S IBLN=$$SET(" Only Allow 1 Ins Co/Claim Batch?",$$EXPAND^IBTRE(350.9,8.07,+$P(IBPD8,U,7)),IBLN,IBLR,IBSEL)  S IBLN=$$SET(" Last Auto-Txmt Run Date",$$DATE^IBJU1($P(IBPD8,U,5)),IBLN,IBLR,IBSEL)  S IBLN=$$SET(" Days To Wait To Purge Msgs",$P(IBPD8,U,2),IBLN,IBLR,IBSEL)  S IBLN=$$SET(" Allow MRA Processing?",$$YN(+$P(IBPD8,U,12)),IBLN,IBLR,IBSEL)  S IBLN=$$SET(" Enable Automatic MRA Processing?",$$YN(+$P(IBPD8,U,11)),IBLN,IBLR,IBSEL)  S IBLN=$$SET(" Enable Auto Reg EOB Processing?",$$YN(+$P(IBPD8,U,17)),IBLN,IBLR,IBSEL) **;JWS;IB\*2.0\*592;add on/off for Dental**  **I $P(IBPD8,U,20)="" S $P(IBPD8,U,20)=1 ;default to yes  S IBLN=$$SET(" Allow Dental Claim Processing?",$$YN(+$P(IBPD8,U,20)),IBLN,IBLR,IBSEL)**  ;  ; WCJ;IB\*2.0\*547;administrative contractors medicare  D RIGHT(3,1,1)  S Z=+$P($G(^IBE(350.9,1,81,0)),U,4)\_" defined"  S IBLN=$$SET("Alt Prim Payer ID Typ-Medicare",Z,IBLN,IBLR,IBSEL)  ;  ; WCJ;IB\*2.0\*547;administrative contractors commercial  D RIGHT(3,1,1)  S Z=+$P($G(^IBE(350.9,1,82,0)),U,4)\_" defined"  S IBLN=$$SET("Alt Prim Payer ID Typ-Commercial",Z,IBLN,IBLR,IBSEL)  ;  ; Ingenix ClaimsManager Information  D RIGHT(9,1,1)  S IBLN=$$SET("Are we using ClaimsManager?",$$YN(+$P(IBPD50,U,1)),IBLN,IBLR,IBSEL)  S IBLN=$$SET("Is ClaimsManager working OK?",$$YN(+$P(IBPD50,U,2)),IBLN,IBLR,IBSEL)  S IBLN=$$SET("ClaimsManager TCP/IP Address",$P(IBPD50,U,5),IBLN,IBLR,IBSEL)  S IBCISOCK=$O(^IBE(350.9,1,50.06,"B",""))  S IBLN=$$SET("ClaimsManager TCP/IP Ports",IBCISOCK,IBLN,IBLR,IBSEL)  F  S IBCISOCK=$O(^IBE(350.9,1,50.06,"B",IBCISOCK)) Q:IBCISOCK=""  D  . S IBLN=$$SET("",IBCISOCK,IBLN,IBLR,IBSEL)  . Q  S IBLN=$$SET("General Error MailGroup",$$EXSET^IBJU1($P(IBPD50,U,3),350.9,50.03),IBLN,IBLR,IBSEL)  S IBLN=$$SET("Communication Error MailGroup",$$EXSET^IBJU1($P(IBPD50,U,4),350.9,50.04),IBLN,IBLR,IBSEL)  S IBCIMFLG=$$EXTERNAL^DILFD(350.9,50.07,"",$P(IBPD50,U,7))  I IBCIMFLG="" S IBCIMFLG="PRIORITY"  S IBLN=$$SET("MailMan Messages",IBCIMFLG,IBLN,IBLR,IBSEL)  ;  ; Request For Additional Info patch 547  D RIGHT(9,1,1)  S Z=$G(^IBE(350.9,1,52)) S:$P(Z,U)="" $P(Z,U)="No Purge"  S IBLN=$$SET("Days to store 277RFAI Transactions",$P(Z,U),IBLN,IBLR,IBSEL)  S IBLN=$$SET("Days to wait to purge entry on RFAI Management Worklist",$P(Z,U,2),IBLN,IBLR,IBSEL)  Q  ; SET(TTL,DATA,LN,LR,SEL,HDR) ;  N IBY,IBX,IBC S IBC=": " I TTL="" S IBC=" "  S IBY=TTL\_$J("",(IBTW(LR)-$L(TTL)-2))\_$S('$G(HDR):IBC\_DATA,1:""),IBX=$G(^TMP("IBJPS",$J,LN,0))  S IBX=$$SETSTR^VALM1(IBY,IBX,IBTC(LR),(IBTW(LR)+IBSW(LR)))  D SET1(IBX,LN,SEL)  S LN=LN+1  Q LN  ; SET1(STR,LN,SEL,HI) ; set up TMP array with screen data  S ^TMP("IBJPS",$J,LN,0)=STR  S ^TMP("IBJPS",$J,"IDX",LN,SEL)=""  S ^TMP("IBJPSAX",$J,SEL)=SEL  I $G(HI)'="" D CNTRL^VALM10(LN,1,4,IOINHI,IOINORM)  ;I $G(RV) D CNTRL^VALM10(LN,6,19,IOUON,IOUOFF)  Q  ; YN(X) Q $S(+X:"YES",1:"NO")  ; RIGHT(LR,SEL,BL) ; - reset control variables for right side of screen  S IBLN=$S(IBLN>IBGRPE:IBLN,1:IBGRPE) I $G(BL) S IBLN=$$SET("","",IBLN,IBLR,IBSEL)  S IBLR=$G(LR),IBGRPB=IBLN I +$G(SEL) S IBSEL=IBSEL+1 D SET1("["\_IBSEL\_"]",IBLN,IBSEL,1)  Q  ; LEFT(LR) ; - reset control variables for left side of screen  S IBLR=$G(LR),IBGRPE=IBLN,IBLN=IBGRPB  Q | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBJTBA | | | | | | | | |
| **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** | | | | | | | | | |
| IBJTBA ;ALB/ARH - TPI BILL CHARGE INFO SCREEN ;01-MAR-1995  ;;2.0;INTEGRATED BILLING;\*\*39,80,51,137,135,309,349,389\*\*;21-MAR-94;Build 6  ;;Per VHA Directive 2004-038, this routine should not be modified.  ; EN ; -- main entry point for IBJ TP BILL CHARGES  D EN^VALM("IBJT BILL CHARGES")  Q  ; HDR ; -- header code  D HDR^IBJTU1(+IBIFN,+DFN,12)  Q  ; INIT ; -- init variables and list array  N IBOK,IBEOBDET  K ^TMP("IBJTBA",$J) N IBFT  I '$G(DFN)!'$G(IBIFN) S VALMQUIT="" G INITQ  S IBFT=+$P($G(^DGCR(399,+IBIFN,0)),U,19),IBOK=1  I $D(^IBM(361.1,"B",IBIFN))!$D(^IBM(361.1,"C",IBIFN)) D  G:'IBOK INITQ  . S DIR("A")="DO YOU WANT ALL EEOB DETAILS?: ",DIR("B")="NO",DIR(0)="YA"  . D FULL^VALM1 W ! D ^DIR K DIR  . I $D(DTOUT)!$D(DUOUT) S IBOK=0 Q  . S IBEOBDET=+Y  D BLD INITQ Q  ; MRA ; -- mra/eob  N IBI,Z,IBSTR,IBSHEOB,IBCT  S IBCT=0  S IBI=0 F  S IBI=$O(^IBM(361.1,"B",IBIFN,IBI)) Q:'IBI  S Z=+$O(^IBM(361.1,IBI,8,0)) I '$O(^(Z)) S IBCT=IBCT+1,IBSHEOB(IBI)=0 ; Entire EOB belongs to the bill  S IBI=0 F  S IBI=$O(^IBM(361.1,"C",IBIFN,IBI)) Q:'IBI  S IBCT=IBCT+1,IBSHEOB(IBI)=1 ; EOB has been reapportioned at the site  I 'IBCT D  . S IBSTR=$$SETLN("No EEOB/MRA Information","",1,79)  . S IBLN=$$SET(IBSTR,IBLN)  I IBCT D  . S Z=0  . S IBI=0 F  S IBI=$O(IBSHEOB(IBI)) Q:'IBI  S Z=Z+1 D SHEOB^IBJTBA1(IBI,+IBSHEOB(IBI),Z,IBCT)  ;  Q  ; HELP ; -- help code  S X="?" D DISP^XQORM1 W !!  Q  ; EXIT ; -- exit code  K ^TMP("IBJTBA",$J)  D CLEAR^VALM1  Q  ; BLD ; charges, as they would display on the bill  N IBXDATA,IBXSAVE  I $P($G(^DGCR(399,+IBIFN,0)),U,19)=2 D H1500 Q  D UB04  K ^TMP("IBXSAVE",$J)  Q  ; H1500 ; block 24  N X,IBI,IBJ,IBLN,IBX,IBSTR,IBLKLN,IBPFORM,IBLIN  K ^TMP("IBXSAVE",$J)  S IBLIN=$$BOX24D^IBCEF11("",1),IBLKLN=0,IBLN=1  Q:'$G(IBIFN) K ^TMP("IBXDISP",$J)  S IBPFORM=$S($P($G(^IBE(353,2,2)),U,8):$P(^(2),U,8),1:2),IBLN=1  S IBX=$$BILLN^IBCEFG0(1,"1^99",IBLIN,+IBIFN,IBPFORM)  S IBI=$O(^TMP("IBXDISP",$J,""),-1)  S IBJ="" F  S IBJ=$O(^TMP("IBXDISP",$J,IBI,IBJ),-1) Q:$S('IBJ:1,1:$TR($G(^(IBJ))," ")'="") K ^TMP("IBXDISP",$J,IBI,IBJ)  I '$O(^TMP("IBXDISP",$J,IBI,0)) S VALMSG="No charges or procedures defined.",VALMQUIT="" G H1500Q  S IBI="" F  S IBI=$O(^TMP("IBXDISP",$J,IBI)) Q:'IBI  S IBJ=0 F  S IBJ=$O(^TMP("IBXDISP",$J,IBI,IBJ)) Q:'IBJ  D  . S IBX=$G(^TMP("IBXDISP",$J,IBI,IBJ)),IBLN=$$SET(IBX,IBLN)  K ^TMP("IBXDISP",$J)  D COB,MRA  I $$ISRX^IBCEF1(IBIFN) D RX  I $$ISPROS^IBCEF1(IBIFN) D PROS  S VALMCNT=IBLN-1 H1500Q Q  ; UB04 ;form locator 42-49, IBIFN required  N X,Y,DIR,IBI,IBJ,IBX,IBLN,IBLC,IBLIN,IBPFORM,IBSTATE,IBCBILL,IBINPAT,IBQ,Z,Z0  K ^TMP("IBXSAVE",$J)  S IBLIN=$$RCBOX^IBCEF11()  S IBQ=0,IBLC=9 Q:'$G(IBIFN) K ^TMP("IBXDISP",$J)  S IBPFORM=$S($P($G(^IBE(353,3,2)),U,8):$P(^(2),U,8),1:3)  S IBX=$$BILLN^IBCEFG0(1,"1^99",IBLIN,+IBIFN,IBPFORM)  I '$O(^TMP("IBXDISP",$J,0)) S VALMSG="No charges defined.",VALMQUIT="" G UB04Q  S Z="" F  S Z=$O(^TMP("IBXDISP",$J,1,Z),-1) Q:Z=""  S Z0=$G(^(Z)) Q:$TR(Z0," ")'=""  K ^(Z)  S:Z ^TMP("IBXDISP",$J,1,Z+1)=" "  S IBINPAT=$$INPAT^IBCEF(IBIFN,1)  S IBSTATE=$G(^DGCR(399,IBIFN,"U")),IBCBILL=$G(^DGCR(399,IBIFN,0))  ;  S (VALMCNT,IBLN)=1,IBLKLN=0  I +IBINPAT D  S IBLN=$$SET(IBSTR,IBLN)  . S IBX=$P(IBSTATE,U,15),IBSTR=+IBX\_" DAY"\_$S(IBX'=1:"S",1:"")\_" INPATIENT CARE"  . S IBX=$$LOS^IBCU64(+IBSTATE,+$P(IBSTATE,U,2),+$P(IBCBILL,U,6)),IBX=IBX-$$LOS1^IBCU64(IBIFN) I IBX>0 S IBSTR=IBSTR\_$J("Pass Days: "\_IBX,55)  ;  S IBI="" F  S IBI=$O(^TMP("IBXDISP",$J,IBI)) Q:'IBI  S IBJ=0 F  S IBJ=$O(^TMP("IBXDISP",$J,IBI,IBJ)) Q:'IBJ  D  . S IBX=$G(^TMP("IBXDISP",$J,IBI,IBJ)),IBLN=$$SET(IBX,IBLN)  . I $E(IBX,1,3)="001" D COB  ;  K ^TMP("IBXDISP",$J)  ;  D MRA  S VALMCNT=IBLN-1 UB04Q Q  ; SETLN(STR,IBX,COL,WD) ;  S IBX=$$SETSTR^VALM1(STR,IBX,COL,WD)  Q IBX  ; SET(STR,LN) ; set up TMP array with screen data (allows 2 blank lines, if not at end of array)  N IBX,IBI I STR?80" " S IBLKLN=IBLKLN+1 G SETQ  F IBI=1:1:IBLKLN D SET^VALM10(LN," ") S LN=LN+1 Q:IBI>1  D SET^VALM10(LN,STR)  S LN=LN+1,IBLKLN=0 SETQ Q LN  ; COB ; if there is an offset or a secondary/tertiary payer add it to the display, with ins co, and prior bill #  ; IBIFN and IBLN must exist upon entry, IBLN is updated with new line count  N IBM,IBM1,IBI,IBJ,IBD,IBSTR,IBCU2,IBCU1 Q:'$G(IBIFN)  S IBM=$G(^DGCR(399,IBIFN,"M")),IBM1=$G(^DGCR(399,IBIFN,"M1"))  S IBCU2=$G(^DGCR(399,IBIFN,"U2")),IBCU1=$G(^DGCR(399,IBIFN,"U1"))  S IBJ=$P($G(^DGCR(399,IBIFN,0)),U,21),IBJ=$S(IBJ="P":3,IBJ="S":3,IBJ="T":3,1:0),IBSTR=""  I +$P(IBM,U,2)!(+$P(IBM,U,3)) F IBI=1:1:IBJ I +$P(IBM,U,IBI) D  S IBLN=$$SET(IBSTR,IBLN)  . I IBSTR="" S IBLN=$$SET("",IBLN)  . S IBD=$S(IBI=1:"Primary",IBI=2:"Secondary",1:"Tertiary")\_": " S IBSTR=$$SETLN(IBD,"",5,11)  . S IBD=$P($G(^DIC(36,+$P(IBM,U,IBI),0)),U,1) S IBSTR=$$SETLN(IBD,IBSTR,17,25)  . I $P(IBCU2,U,(IBI+3))'="" S IBD=$J(+$P(IBCU2,U,(IBI+3)),9,2) S IBSTR=$$SETLN(IBD,IBSTR,44,11)  . I $P(IBM1,U,(IBI+4))'="" S IBD=$$BN1^PRCAFN(+$P(IBM1,U,(IBI+4))) S IBSTR=$$SETLN(IBD,IBSTR,60,11)  I +$P(IBCU1,U,2) D  S IBLN=$$SET(IBSTR,IBLN)  . I IBSTR="" S IBLN=$$SET("",IBLN)  . S IBD="Offset: " S IBSTR=$$SETLN(IBD,"",5,11)  . S IBD=$P(IBCU1,U,3) S IBSTR=$$SETLN(IBD,IBSTR,17,25)  . S IBD=$J($P(IBCU1,U,2),9,2) S IBSTR=$$SETLN(IBD,IBSTR,44,11)  . S IBD=$P(IBCU1,U,1)-$P(IBCU1,U,2),IBD="Billed: "\_$J(IBD,0,2) S IBSTR=$$SETLN(IBD,IBSTR,60,17)  Q  ; RX ;RX refill info for CMS-1500 TPJI display  N Z,Z0,Z1,IBSPC,IBD,IBI,IBSTR,IBARRAY,IBRXX  S IBLN=IBLN+1  S IBSPC=$J("",5)  D SET^IBCSC5A(IBIFN,.IBARRAY)  I $D(IBARRAY) D  . S (Z,Z0)=0 F  S Z0=$O(IBARRAY(Z0)) Q:Z0=""  S Z1=0 F  S Z1=$O(IBARRAY(Z0,Z1)) Q:'Z1  S Z=Z+1 S IBXDATA(Z)=$$DAT1^IBOUTL(Z1)\_U\_$G(IBARRAY(Z0,Z1))  S IBD=$$SET("",IBLN)  S IBD="PRESCRIPTION REFILLS: (For TPJI display only)"  S IBSTR=$$SETLN(IBD,"",1,79),IBLN=$$SET(IBSTR,IBLN)  S IBI=0 F  S IBI=$O(IBXDATA(IBI)) Q:IBI=""  D  . S IBRXX=$G(IBXDATA(IBI))  . D ZERO^IBRXUTL($P(IBRXX,U,3))  . S IBD=$J($P(IBRXX,U,7),9,2)\_IBSPC\_$P(IBRXX,U)\_IBSPC\_$G(^TMP($J,"IBDRUG",+$P(IBRXX,U,3),.01))  . K ^TMP($J,"IBDRUG")  . S IBSTR=$$SETLN(IBD,"",1,79),IBLN=$$SET(IBSTR,IBLN)  . S IBD="QTY: "\_$P(IBRXX,U,5)\_" for "\_$P(IBRXX,U,4)\_" days supply "\_"NDC# "\_$P(IBRXX,U,6)  . S IBSTR=$$SETLN(IBD,"",23,79),IBLN=$$SET(IBSTR,IBLN)  Q  ; PROS ;prosthetic info for CMS-1500 TPJI display  N Z,Z0,Z1,IBARRAY,IBSPC,IBD,IBI,IBSTR  S IBSPC=$J("",10),IBLN=IBLN+1  D SET^IBCSC5B(IBIFN,.IBARRAY)  I $D(IBARRAY) D  . S (Z,Z0)=0 F  S Z0=$O(IBARRAY(Z0)) Q:Z0=""  S Z1=0 F  S Z1=$O(IBARRAY(Z0,Z1)) Q:'Z1  S Z=Z+1,IBXDATA(Z)=$$DAT1^IBOUTL(Z0)\_U\_$E($$PINB^IBCSC5B(+IBARRAY(Z0,Z1)),1,39)  S IBD=$$SET("",IBLN)  S IBD="PROSTHETIC REFILLS: (For TPJI display only)"  S IBSTR=$$SETLN(IBD,"",1,79),IBLN=$$SET(IBSTR,IBLN)  S IBI=0 F  S IBI=$O(IBXDATA(IBI)) Q:IBI=""  D  . S IBD=$P(IBXDATA(IBI),U)\_IBSPC\_$P(IBXDATA(IBI),U,2)  . S IBSTR=$$SETLN(IBD,"",1,79),IBLN=$$SET(IBSTR,IBLN)  Q  ; | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBJTBA ;ALB/ARH - TPI BILL CHARGE INFO SCREEN ;01-MAR-1995  ;;2.0;INTEGRATED BILLING;\*\*39,80,51,137,135,309,349,389,**592**\*\*;21-MAR-94;Build 6  ;;Per VHA Directive 2004-038, this routine should not be modified.  ; EN ; -- main entry point for IBJ TP BILL CHARGES  D EN^VALM("IBJT BILL CHARGES")  Q  ; HDR ; -- header code  D HDR^IBJTU1(+IBIFN,+DFN,12)  Q  ; INIT ; -- init variables and list array  N IBOK,IBEOBDET  K ^TMP("IBJTBA",$J) N IBFT  I '$G(DFN)!'$G(IBIFN) S VALMQUIT="" G INITQ  S IBFT=+$P($G(^DGCR(399,+IBIFN,0)),U,19),IBOK=1  I $D(^IBM(361.1,"B",IBIFN))!$D(^IBM(361.1,"C",IBIFN)) D  G:'IBOK INITQ  . S DIR("A")="DO YOU WANT ALL EEOB DETAILS?: ",DIR("B")="NO",DIR(0)="YA"  . D FULL^VALM1 W ! D ^DIR K DIR  . I $D(DTOUT)!$D(DUOUT) S IBOK=0 Q  . S IBEOBDET=+Y  D BLD INITQ Q  ; MRA ; -- mra/eob  N IBI,Z,IBSTR,IBSHEOB,IBCT  S IBCT=0  S IBI=0 F  S IBI=$O(^IBM(361.1,"B",IBIFN,IBI)) Q:'IBI  S Z=+$O(^IBM(361.1,IBI,8,0)) I '$O(^(Z)) S IBCT=IBCT+1,IBSHEOB(IBI)=0 ; Entire EOB belongs to the bill  S IBI=0 F  S IBI=$O(^IBM(361.1,"C",IBIFN,IBI)) Q:'IBI  S IBCT=IBCT+1,IBSHEOB(IBI)=1 ; EOB has been reapportioned at the site  I 'IBCT D  . S IBSTR=$$SETLN("No EEOB/MRA Information","",1,79)  . S IBLN=$$SET(IBSTR,IBLN)  I IBCT D  . S Z=0  . S IBI=0 F  S IBI=$O(IBSHEOB(IBI)) Q:'IBI  S Z=Z+1 D SHEOB^IBJTBA1(IBI,+IBSHEOB(IBI),Z,IBCT)  ;  Q  ; HELP ; -- help code  S X="?" D DISP^XQORM1 W !!  Q  ; EXIT ; -- exit code  K ^TMP("IBJTBA",$J)  D CLEAR^VALM1  Q  ; BLD ; charges, as they would display on the bill  N IBXDATA,IBXSAVE  **;JWS:IB\*2.0\*592:Dental form#7 as professional**  I $P($G(^DGCR(399,+IBIFN,0)),U,19)=2**!($P($G(^(0)),U,19)=7)** D H1500 Q  D UB04  K ^TMP("IBXSAVE",$J)  Q  ; H1500 ; block 24  N X,IBI,IBJ,IBLN,IBX,IBSTR,IBLKLN,IBPFORM,IBLIN  K ^TMP("IBXSAVE",$J)  S IBLIN=$$BOX24D^IBCEF11("",1),IBLKLN=0,IBLN=1  Q:'$G(IBIFN) K ^TMP("IBXDISP",$J)  S IBPFORM=$S($P($G(^IBE(353,2,2)),U,8):$P(^(2),U,8),1:2),IBLN=1  S IBX=$$BILLN^IBCEFG0(1,"1^99",IBLIN,+IBIFN,IBPFORM)  S IBI=$O(^TMP("IBXDISP",$J,""),-1)  S IBJ="" F  S IBJ=$O(^TMP("IBXDISP",$J,IBI,IBJ),-1) Q:$S('IBJ:1,1:$TR($G(^(IBJ))," ")'="") K ^TMP("IBXDISP",$J,IBI,IBJ)  I '$O(^TMP("IBXDISP",$J,IBI,0)) S VALMSG="No charges or procedures defined.",VALMQUIT="" G H1500Q  S IBI="" F  S IBI=$O(^TMP("IBXDISP",$J,IBI)) Q:'IBI  S IBJ=0 F  S IBJ=$O(^TMP("IBXDISP",$J,IBI,IBJ)) Q:'IBJ  D  . S IBX=$G(^TMP("IBXDISP",$J,IBI,IBJ)),IBLN=$$SET(IBX,IBLN)  K ^TMP("IBXDISP",$J)  D COB,MRA  I $$ISRX^IBCEF1(IBIFN) D RX  I $$ISPROS^IBCEF1(IBIFN) D PROS  S VALMCNT=IBLN-1 H1500Q Q  ; UB04 ;form locator 42-49, IBIFN required  N X,Y,DIR,IBI,IBJ,IBX,IBLN,IBLC,IBLIN,IBPFORM,IBSTATE,IBCBILL,IBINPAT,IBQ,Z,Z0  K ^TMP("IBXSAVE",$J)  S IBLIN=$$RCBOX^IBCEF11()  S IBQ=0,IBLC=9 Q:'$G(IBIFN) K ^TMP("IBXDISP",$J)  S IBPFORM=$S($P($G(^IBE(353,3,2)),U,8):$P(^(2),U,8),1:3)  S IBX=$$BILLN^IBCEFG0(1,"1^99",IBLIN,+IBIFN,IBPFORM)  I '$O(^TMP("IBXDISP",$J,0)) S VALMSG="No charges defined.",VALMQUIT="" G UB04Q  S Z="" F  S Z=$O(^TMP("IBXDISP",$J,1,Z),-1) Q:Z=""  S Z0=$G(^(Z)) Q:$TR(Z0," ")'=""  K ^(Z)  S:Z ^TMP("IBXDISP",$J,1,Z+1)=" "  S IBINPAT=$$INPAT^IBCEF(IBIFN,1)  S IBSTATE=$G(^DGCR(399,IBIFN,"U")),IBCBILL=$G(^DGCR(399,IBIFN,0))  ;  S (VALMCNT,IBLN)=1,IBLKLN=0  I +IBINPAT D  S IBLN=$$SET(IBSTR,IBLN)  . S IBX=$P(IBSTATE,U,15),IBSTR=+IBX\_" DAY"\_$S(IBX'=1:"S",1:"")\_" INPATIENT CARE"  . S IBX=$$LOS^IBCU64(+IBSTATE,+$P(IBSTATE,U,2),+$P(IBCBILL,U,6)),IBX=IBX-$$LOS1^IBCU64(IBIFN) I IBX>0 S IBSTR=IBSTR\_$J("Pass Days: "\_IBX,55)  ;  S IBI="" F  S IBI=$O(^TMP("IBXDISP",$J,IBI)) Q:'IBI  S IBJ=0 F  S IBJ=$O(^TMP("IBXDISP",$J,IBI,IBJ)) Q:'IBJ  D  . S IBX=$G(^TMP("IBXDISP",$J,IBI,IBJ)),IBLN=$$SET(IBX,IBLN)  . I $E(IBX,1,3)="001" D COB  ;  K ^TMP("IBXDISP",$J)  ;  D MRA  S VALMCNT=IBLN-1 UB04Q Q  ; SETLN(STR,IBX,COL,WD) ;  S IBX=$$SETSTR^VALM1(STR,IBX,COL,WD)  Q IBX  ; SET(STR,LN) ; set up TMP array with screen data (allows 2 blank lines, if not at end of array)  N IBX,IBI I STR?80" " S IBLKLN=IBLKLN+1 G SETQ  F IBI=1:1:IBLKLN D SET^VALM10(LN," ") S LN=LN+1 Q:IBI>1  D SET^VALM10(LN,STR)  S LN=LN+1,IBLKLN=0 SETQ Q LN  ; COB ; if there is an offset or a secondary/tertiary payer add it to the display, with ins co, and prior bill #  ; IBIFN and IBLN must exist upon entry, IBLN is updated with new line count  N IBM,IBM1,IBI,IBJ,IBD,IBSTR,IBCU2,IBCU1 Q:'$G(IBIFN)  S IBM=$G(^DGCR(399,IBIFN,"M")),IBM1=$G(^DGCR(399,IBIFN,"M1"))  S IBCU2=$G(^DGCR(399,IBIFN,"U2")),IBCU1=$G(^DGCR(399,IBIFN,"U1"))  S IBJ=$P($G(^DGCR(399,IBIFN,0)),U,21),IBJ=$S(IBJ="P":3,IBJ="S":3,IBJ="T":3,1:0),IBSTR=""  I +$P(IBM,U,2)!(+$P(IBM,U,3)) F IBI=1:1:IBJ I +$P(IBM,U,IBI) D  S IBLN=$$SET(IBSTR,IBLN)  . I IBSTR="" S IBLN=$$SET("",IBLN)  . S IBD=$S(IBI=1:"Primary",IBI=2:"Secondary",1:"Tertiary")\_": " S IBSTR=$$SETLN(IBD,"",5,11)  . S IBD=$P($G(^DIC(36,+$P(IBM,U,IBI),0)),U,1) S IBSTR=$$SETLN(IBD,IBSTR,17,25)  . I $P(IBCU2,U,(IBI+3))'="" S IBD=$J(+$P(IBCU2,U,(IBI+3)),9,2) S IBSTR=$$SETLN(IBD,IBSTR,44,11)  . I $P(IBM1,U,(IBI+4))'="" S IBD=$$BN1^PRCAFN(+$P(IBM1,U,(IBI+4))) S IBSTR=$$SETLN(IBD,IBSTR,60,11)  I +$P(IBCU1,U,2) D  S IBLN=$$SET(IBSTR,IBLN)  . I IBSTR="" S IBLN=$$SET("",IBLN)  . S IBD="Offset: " S IBSTR=$$SETLN(IBD,"",5,11)  . S IBD=$P(IBCU1,U,3) S IBSTR=$$SETLN(IBD,IBSTR,17,25)  . S IBD=$J($P(IBCU1,U,2),9,2) S IBSTR=$$SETLN(IBD,IBSTR,44,11)  . S IBD=$P(IBCU1,U,1)-$P(IBCU1,U,2),IBD="Billed: "\_$J(IBD,0,2) S IBSTR=$$SETLN(IBD,IBSTR,60,17)  Q  ; RX ;RX refill info for CMS-1500 TPJI display  N Z,Z0,Z1,IBSPC,IBD,IBI,IBSTR,IBARRAY,IBRXX  S IBLN=IBLN+1  S IBSPC=$J("",5)  D SET^IBCSC5A(IBIFN,.IBARRAY)  I $D(IBARRAY) D  . S (Z,Z0)=0 F  S Z0=$O(IBARRAY(Z0)) Q:Z0=""  S Z1=0 F  S Z1=$O(IBARRAY(Z0,Z1)) Q:'Z1  S Z=Z+1 S IBXDATA(Z)=$$DAT1^IBOUTL(Z1)\_U\_$G(IBARRAY(Z0,Z1))  S IBD=$$SET("",IBLN)  S IBD="PRESCRIPTION REFILLS: (For TPJI display only)"  S IBSTR=$$SETLN(IBD,"",1,79),IBLN=$$SET(IBSTR,IBLN)  S IBI=0 F  S IBI=$O(IBXDATA(IBI)) Q:IBI=""  D  . S IBRXX=$G(IBXDATA(IBI))  . D ZERO^IBRXUTL($P(IBRXX,U,3))  . S IBD=$J($P(IBRXX,U,7),9,2)\_IBSPC\_$P(IBRXX,U)\_IBSPC\_$G(^TMP($J,"IBDRUG",+$P(IBRXX,U,3),.01))  . K ^TMP($J,"IBDRUG")  . S IBSTR=$$SETLN(IBD,"",1,79),IBLN=$$SET(IBSTR,IBLN)  . S IBD="QTY: "\_$P(IBRXX,U,5)\_" for "\_$P(IBRXX,U,4)\_" days supply "\_"NDC# "\_$P(IBRXX,U,6)  . S IBSTR=$$SETLN(IBD,"",23,79),IBLN=$$SET(IBSTR,IBLN)  Q  ; PROS ;prosthetic info for CMS-1500 TPJI display  N Z,Z0,Z1,IBARRAY,IBSPC,IBD,IBI,IBSTR  S IBSPC=$J("",10),IBLN=IBLN+1  D SET^IBCSC5B(IBIFN,.IBARRAY)  I $D(IBARRAY) D  . S (Z,Z0)=0 F  S Z0=$O(IBARRAY(Z0)) Q:Z0=""  S Z1=0 F  S Z1=$O(IBARRAY(Z0,Z1)) Q:'Z1  S Z=Z+1,IBXDATA(Z)=$$DAT1^IBOUTL(Z0)\_U\_$E($$PINB^IBCSC5B(+IBARRAY(Z0,Z1)),1,39)  S IBD=$$SET("",IBLN)  S IBD="PROSTHETIC REFILLS: (For TPJI display only)"  S IBSTR=$$SETLN(IBD,"",1,79),IBLN=$$SET(IBSTR,IBLN)  S IBI=0 F  S IBI=$O(IBXDATA(IBI)) Q:IBI=""  D  . S IBD=$P(IBXDATA(IBI),U)\_IBSPC\_$P(IBXDATA(IBI),U,2)  . S IBSTR=$$SETLN(IBD,"",1,79),IBLN=$$SET(IBSTR,IBLN)  Q  ; | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBTRH5D | | | | | | | | |
| **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** | | | | | | | | | |
| IBTRH5D ;ALB/FA - HCSR Create 278 Request ;12-AUG-2014  ;;2.0;INTEGRATED BILLING;\*\*517\*\*;21-MAR-94;Build 240  ;;Per VA Directive 6402, this routine should not be modified.  ;;  ; Contains Functions used in creating a 278 request from a  ; selected entry in the HCSR Response worklist  ;  ; -------------------------- Entry Points --------------------------------  ; SELAPI - Allows the user to see a quick view of the currently entered  ; Additional Patient Information lines and either pick one to   ; edit, enter a new one or skip.  ; SELDX - Allows the user to see a quick view of the currently entered  ; Diagnoses and either pick one to edit, enter a new one or  ; skip.  ; SELPD - Allows the user to see a quick view of the currently entered  ; Patient Event Provider Data Lines and either pick one to   ; edit, enter a new one or skip.  ;-----------------------------------------------------------------------------  ; SELAPI(IBTRIEN) ;EP  ; Called from within Input template IB CREATE 278 REQUEST  ; Provides the user with a quick view of currently entered Additional Patient  ; Information multiples and allows them to select one to edit or enter a new   ; one.  ; Input: IBTRIEN - IEN of the 356.22 entry being edited  ; Returns: Value of the .01 field of the multiple to edit  ; "" if creating a new multiple, -2 to exit template  ; IBNEW - 1 if creating a new entry  N AIDATA,CNT,ENTNUM,FDA,IEN,H1,H2,L1,L2,MAX,RETIEN,RTYPE,SECT,X,XX,Y,YY  S IBNEW=0,SECT="Additional Patient Information"  ;  ; First check for an empty Additional Patient Information Line to delete  D DELAPI(IBTRIEN)  ;  ; Next create an array of all current Additional Patient Information lines to  ; display  S XX=+$P($G(^IBT(356.22,IBTRIEN,11,0)),"^",4) ; Total # of API Lines  S MAX=$S(XX<10:"",1:"Additional Patient Information Lines")  S IEN=0,CNT=0  F  D  Q:+IEN=0  . S IEN=$O(^IBT(356.22,IBTRIEN,11,IEN))  . Q:+IEN=0  . S CNT=CNT+1  . S XX=$$LJ^XLFSTR(CNT,4) ; Selection #  . S YY=$$GET1^DIQ(356.2211,IEN\_","\_DA\_",",.01) ; Report Type Desc  . S YY=$E(YY,1,28)\_" "  . S XX=XX\_$$LJ^XLFSTR(YY,30)  . S YY=$$GET1^DIQ(356.2211,IEN\_","\_DA\_",",.02) ; Delivery Method  . S YY=$E(YY,1,20)\_" "  . S XX=XX\_$$LJ^XLFSTR(YY,23)  . S YY=$$GET1^DIQ(356.2211,IEN\_","\_DA\_",",.03) ; Attachment #  . S YY=$E(YY,1,22)  . S XX=XX\_$$LJ^XLFSTR(YY,22)  . S AIDATA(CNT)=IEN\_"^"\_XX  ;  I 'CNT D  Q $S($O(RETIEN(0)):RETIEN($O(RETIEN(0))),1:XX)  . W !!,"No Additional Patient Information is currently on file.",!  . S XX=$$ASKNEW("Add Additional Patient Information","NO")  . Q:XX<0  . S RTYPE=$$RTYPE(IBTRIEN) ; Get the .01 value  . I RTYPE="" S XX=-1 Q                         ; None entered  . S IBNEW=1,XX=RTYPE  . S FDA(356.2211,"+1,"\_IBTRIEN\_",",.01)=RTYPE  . D UPDATE^DIE("","FDA","RETIEN") ; File the new line  ;  ; Next display all of the current Additional Patient Information  S H1="# Report Type Delivery Method Attachment Control #"  S H2="-- ---------------------------- --------------------- ----------------------"  S L1="The following Additional Patient Information is currently on file."  S L2="Enter the # of an entry to edit, 'NEW' to add one or press Return to skip."  S XX=$$SELENT(.AIDATA,H1,H2,L1,L2,MAX,"",SECT)  I XX?1"D".N D  Q -3  . S (XX,ENTNUM)=$P(XX,"D",2)  . S XX=$P(AIDATA(XX),U)  . D DELAPI(IBTRIEN,XX)  . W !,"Entry #",ENTNUM," has been deleted."  I XX<0 Q XX  I XX=0 D  Q $S($O(RETIEN(0)):RETIEN($O(RETIEN(0))),1:XX)  . S RTYPE=$$RTYPE(IBTRIEN) ; Get the .01 value  . I RTYPE="" S XX=-1 Q                         ; None entered  . S IBNEW=1  . S XX=RTYPE  . S FDA(356.2211,"+1,"\_IBTRIEN\_",",.01)=RTYPE  . D UPDATE^DIE("","FDA","RETIEN") ; File the new line  Q $P(AIDATA(XX),"^",1)  ; DELAPI(IBTRIEN,IEN) ; Checks to see if the user entered 'NEW' to create a new   ; Additional Patient Information Line and didn't enter any data for it. Also  ; checks to see if user selected to delete a specified line. If so, the   ; Additional Patient Information line with no data (or selected) is deleted  ; Input: IBTRIEN - IEN of the 356.22 entry being edited  ; IEN - Optional, IEN of the multiple to be deleted if passed  ; defaults to ""  ; Output: Empty or selected Additional Patient Information line is deleted (Potentially)  N APIIEN,DA,DIK,X,XX,Y  S:'$D(IEN) IEN=""  I IEN'="" D  Q  . S DA(1)=IBTRIEN,DA=IEN  . S DIK="^IBT(356.22,DA(1),11,"  . D ^DIK ; Delete the multiple  ;  S APIIEN=+$P($G(^IBT(356.22,IBTRIEN,11,0)),"^",3) ; Last Multiple IEN  Q:'APIIEN  S XX=$G(^IBT(356.22,IBTRIEN,11,APIIEN,0))  S $P(XX,"^",1)=""                                  ; Remove .01 field  Q:$TR(XX,"^","")'=""                               ; 0 node data exists  S DA(1)=IBTRIEN,DA=APIIEN  S DIK="^IBT(356.22,DA(1),11,"  D ^DIK ; Delete the multiple  Q  ; RTYPE(IBTRIEN) ; Prompts the user to enter the .01 (Report Type) field of the  ; Additional Patient Information multiple  ; Input: IBTRIEN - IEN of the 356.22 entry being edited  ; Returns: IEN of the selected Report Type or "" of not entered  N DA,DIR,DIROUT,DIRUT,DTOUT,DUOUT,X,Y  S DA(1)=IBTRIEN  S DIR(0)="356.2211,.01",DIR("A")=" Report Type"  D ^DIR  Q:$D(DIRUT) ""  Q $P(Y,"^",1)  ; SELPD(IBTRIEN) ;EP  ; Called from within Input template IB CREATE 278 REQUEST  ; Provides the user with a quick view of currently entered Provider Data  ; multiples and allows them to select one to edit or enter a new one.  ; Input: IBTRIEN - IEN of the 356.22 entry being edited  ; IBTRBRF - 1 if this display is being used from the brief template  ; 0 or undefined otherwise  ; Returns: Value of the .01 field of the multiple to edit  ; "" if creating a new multiple, -2 to exit template  ; IBNEW=1 when creating a new entry  N CNT,ENTNUM,FDA,IEN,H1,H2,L1,L2,MAX,PDDATA,PTYPE,RETIEN,SECT,X,XX,Y,YY  S IBNEW=0,SECT="Provider Data Information"  ;  ; First check for an empty Provider Data Line to delete  D DELPD(IBTRIEN)  ;  ; Next create an array of all current Provider Data Information lines  S XX=+$P($G(^IBT(356.22,IBTRIEN,13,0)),"^",4) ; # of Multiples  S MAX=$S(XX<14:"",1:"Provider Data Lines")  S IEN=0,CNT=0  F  D  Q:+IEN=0  . S IEN=$O(^IBT(356.22,IBTRIEN,13,IEN))  . Q:+IEN=0  . S CNT=CNT+1  . S XX=$$LJ^XLFSTR(CNT,4) ; Selection #  . S YY=$$GET1^DIQ(356.2213,IEN\_","\_DA\_",",.01) ; Prov Type Desc  . S YY=$E(YY,1,30)\_" "  . S XX=XX\_$$LJ^XLFSTR(YY,32)  . ;  . ; IBTRBRF is defined in IB CREATE 278 REQUEST SHORT input template  . I $G(IBTRBRF)'=1 D  . . S YY=$$GET1^DIQ(356.2213,IEN\_","\_DA\_",",.02) ; Person/Non-Person  . . S XX=XX\_$$LJ^XLFSTR(YY,12)  . S YY=$$GET1^DIQ(356.2213,IEN\_","\_DA\_",",.03)  . S XX=XX\_$$LJ^XLFSTR(YY,"28T")  . S PDDATA(CNT)=IEN\_"^"\_XX  ;   I 'CNT D  Q $S($O(RETIEN(0)):RETIEN($O(RETIEN(0))),1:XX)  .I $G(IBTRBRF)'=1 D  ..W !!,"No Provider Data Information is currently on file.",!  ..S XX=$$ASKNEW("Add Provider Data Information")  ..Q  .I $G(IBTRBRF)=1 S XX=0  .Q:XX<0  .S PTYPE=$$PTYPE(IBTRIEN) ; Get the .01 value  .I PTYPE="" S XX=-1 Q                         ; None entered  .S IBNEW=1,XX=PTYPE  .S FDA(356.2213,"+1,"\_IBTRIEN\_",",.01)=PTYPE  .D UPDATE^DIE("","FDA","RETIEN") ; File the new line  .Q  ;  ; Next display all of the current Provider Data lines  S H1="# Provider Type "  I $G(IBTRBRF)'=1 S H1=H1\_" Per/Non"  S H1=H1\_" Provider"  S H2="-- ------------------------------"  I $G(IBTRBRF)'=1 S H2=H2\_" ----------"  S H2=H2\_" ------------------------------"  S L1="The following Provider Data Information is currently on file."  S L2="Enter the # of an entry to edit, 'NEW' to add one or press Return to skip."  S XX=$$SELENT(.PDDATA,H1,H2,L1,L2,MAX,"",SECT)  I XX?1"D".N D  Q -3  . S (XX,ENTNUM)=$P(XX,"D",2)  . S XX=$P(PDDATA(XX),U)  . D DELPD(IBTRIEN,XX)  . W !,"Entry #",ENTNUM," has been deleted."  I XX<0 Q XX  I XX=0 D  Q $S($O(RETIEN(0)):RETIEN($O(RETIEN(0))),1:XX)  . S PTYPE=$$PTYPE(IBTRIEN) ; Get the .01 value  . I PTYPE="" S XX=-1 Q                         ; None entered  . S XX=PTYPE  . S IBNEW=1  . S FDA(356.2213,"+1,"\_IBTRIEN\_",",.01)=PTYPE  . D UPDATE^DIE("","FDA","RETIEN") ; File the new line  Q $P(PDDATA(XX),"^",1)  ; DELPD(IBTRIEN,IEN) ; Checks to see if the user entered 'NEW' to create a new   ; Provider Data Line and didn't enter any data for it or selected a line to   ; be deleted. If so, the empty or selected Provider Data line is deleted  ; Input: IBTRIEN - IEN of the 356.22 entry being edited  ; IEN - Optional, IEN of the multiple to be deleted if passed  ; defaults to ""  ; Output: Empty or selected Provider Data line is deleted (Potentially)  N PDIEN,DA,DIK,X,XX,Y  S:'$D(IEN) IEN=""  I IEN'="" D  Q  . S DA(1)=IBTRIEN,DA=IEN  . S DIK="^IBT(356.22,DA(1),13,"  . D ^DIK ; Delete the multiple  ;  S PDIEN=+$P($G(^IBT(356.22,IBTRIEN,13,0)),"^",3) ; Last Multiple IEN  Q:'PDIEN  S XX=$G(^IBT(356.22,IBTRIEN,13,PDIEN,0))  S $P(XX,"^",1)=""                                  ; Remove .01 field  Q:$TR(XX,"^","")'=""                               ; 0 node data exists  S DA(1)=IBTRIEN,DA=PDIEN  S DIK="^IBT(356.22,DA(1),13,"  D ^DIK ; Delete the multiple  Q  ; PTYPE(IBTRIEN) ; Prompts the user to enter the .01 (Provider Type) field of the  ; Provider Data multiple  ; Input: IBTRIEN - IEN of the 356.22 entry being edited  ; Returns: IEN of the selected Provider Type or "" of not entered  N DA,DIR,DIROUT,DIRUT,DTOUT,DUOUT,X,Y  S DA(1)=IBTRIEN  S DIR(0)="356.2213,.01",DIR("A")=" Provider Type"  D ^DIR  Q:$D(DIRUT) ""  Q $P(Y,"^",1)  ; SELDX(IBTRIEN) ;EP  ; Called from within Input template IB CREATE 278 REQUEST  ; Provides the user with a quick view of currently entered Diagnoses and  ; allows them to select one to edit or enter a new diagnosis.  ; Input: IBTRIEN - IEN of the 356.22 entry being edited  ; IBTRBRF - 1 if this display is being used from the brief template  ; 0 or undefined other otherwise  ; Returns: Value of the .01 field of the multiple to edit  ; "" if creating a new multiple, -2 to exit template  ; -3 if a if a line was deleted  ; IBNEW=1 when creating a new entry  N CNT,DXDATA,DXTYPE,ENTNUM,FDA,IEN,H1,H2,L1,L2,MAX,RETIEN,SECT,X,XX,Y,YY  S IBNEW=0,SECT="Diagnosis Information"  ;  ; First check for an empty Diagnosis Line to delete  D DELDX(IBTRIEN)  ;  ; Next create an array of all current Diagnoses lines  S XX=+$P($G(^IBT(356.22,IBTRIEN,3,0)),"^",4) ; Total # of Dx Lines  S MAX=$S(XX<12:"",1:"Diagnosis Lines")  S IEN=0,CNT=0  F  D  Q:+IEN=0  . S IEN=$O(^IBT(356.22,IBTRIEN,3,IEN))  . Q:+IEN=0  . S CNT=CNT+1  . S XX=$$LJ^XLFSTR(CNT,4) ; Selection #  . S YY=$$GET1^DIQ(356.223,IEN\_","\_DA\_",",.01,"I") ; Diagnosis Type  . S YY=$$GET1^DIQ(356.006,YY\_",",.01)  . S XX=XX\_$$LJ^XLFSTR(YY,7)  . S YY=$$GET1^DIQ(356.223,IEN\_","\_DA\_",",.02) ; Diagnosis  . S XX=XX\_$$LJ^XLFSTR(YY,11)  . I $G(IBTRBRF)'=1 D  . . S YY=$$GET1^DIQ(356.223,IEN\_","\_DA\_",",.03) ; Date Known  . . S XX=XX\_$$LJ^XLFSTR(YY,14)  . S DXDATA(CNT)=IEN\_"^"\_XX  ;  ; Creating 1st Diagnosis Line?  I 'CNT D  Q $S($O(RETIEN(0)):RETIEN($O(RETIEN(0))),1:XX)  .I $G(IBTRBRF)'=1 D  ..W !!,"No Diagnosis Information is currently on file.",!  ..S XX=$$ASKNEW("Add a new Diagnosis")  ..Q  .I $G(IBTRBRF)=1 S XX=0  .Q:XX<0  .S DXTYPE=$$DXTYPE(IBTRIEN) ; Get the .01 value  .I DXTYPE="" S XX=-1 Q                        ; None entered  .S IBNEW=1,XX=DXTYPE  .S FDA(356.223,"+1,"\_IBTRIEN\_",",.01)=DXTYPE  .D UPDATE^DIE("","FDA","RETIEN") ; File the new line  .Q  ;  ; Next display all of the current Diagnoses and let the user select one  S H1="# Type Diagnosis"  I $G(IBTRBRF)'=1 S H1=H1\_" Date DX Known"  S H2="-- ----- ---------"  I $G(IBTRBRF)'=1 S H2=H2\_" -------------"  S L1="The following Diagnoses are currently on file."  S L2="Enter the # of a Diagnosis to edit, 'NEW' to add one or press Return to skip."  S XX=$$SELENT(.DXDATA,H1,H2,L1,L2,MAX,"",SECT)  I XX?1"D".N D  Q -3  . S (XX,ENTNUM)=$P(XX,"D",2)  . S XX=$P(DXDATA(XX),U)  . D DELDX(IBTRIEN,XX)  . W !,"Entry #",ENTNUM," has been deleted."  I XX<0 Q XX  I XX=0 D  Q $S($O(RETIEN(0)):RETIEN($O(RETIEN(0))),1:XX)  . S DXTYPE=$$DXTYPE(IBTRIEN) ; Get the .01 value  . I DXTYPE="" S XX=-1 Q                        ; None entered  . S XX=DXTYPE  . S IBNEW=1  . S FDA(356.223,"+1,"\_IBTRIEN\_",",.01)=DXTYPE  . D UPDATE^DIE("","FDA","RETIEN") ; File the new line  Q $P(DXDATA(XX),"^",1)  ; DXTYPE(IBTRIEN) ; Prompts the user to enter the .01 (Diagnosis Type) field of  ; the diagnosis multiple  ; Input: IBTRIEN - IEN of the 356.22 entry being edited  ; Returns: IEN of the selected Diagnosis Type or "" of not entered  N DA,DIR,DIROUT,DIRUT,DTOUT,DUOUT,X,Y  S DA(1)=IBTRIEN,DA=$P($G(^IBT(356.22,IBTRIEN,3,0)),"^",3)+1  S DIR(0)="356.223,.01",DIR("A")=" Diagnosis Qualifier"  D ^DIR  Q:$D(DIRUT) ""  Q $P(Y,"^",1)  ; DELDX(IBTRIEN,IEN) ; Checks to see if the user entered 'NEW' to create a new   ; Diagnosis Line and didn't enter any data for it or selected a multiple to  ; to be deleted. If so, the empty or selected multiple is deleted  ; Input: IBTRIEN - IEN of the 356.22 entry being edited  ; IEN - Optional, IEN of the multiple to be deleted if passed  ; defaults to ""  ; Output: Empty or selected Diagnosis line is deleted (Potentially)  N DA,DIK,DXIEN,X,XX,Y  S:'$D(IEN) IEN=""  I IEN'="" D  Q  . S DA(1)=IBTRIEN,DA=IEN  . S DIK="^IBT(356.22,DA(1),3,"  . D ^DIK ; Delete the multiple  ;  S DXIEN=+$P($G(^IBT(356.22,IBTRIEN,3,0)),"^",3) ; Last Multiple IEN  Q:'DXIEN  S XX=$G(^IBT(356.22,IBTRIEN,3,DXIEN,0))  S $P(XX,"^",1)=""                                  ; Remove .01 field  Q:$TR(XX,"^","")'=""                               ; 0 node data exists  S DA(1)=IBTRIEN,DA=DXIEN  S DIK="^IBT(356.22,DA(1),3,"  D ^DIK ; Delete the multiple  Q  ; ASKNEW(PROMPT,DEFAULT) ;EP  ; Ask if user wants to create a new entry  ; Input: PROMPT - Yes/No question to ask the user  ; DEFALT - Default Answer  ; Optional, if not passed, set to 'YES'  ; Returns: 0 - User wants to add a new Entry  ; -1 - User doesn't want to add a new entry  ; -2 - User wants to exit template  N DIR,DIROUT,DIRUT,DTOUT,DUOUT,X,XX,Y  S:'$D(DEFAULT) DEFAULT="YES"  S XX=$P(PROMPT,"Add ",2)  S DIR("?")="Select NO to skip this section. Select YES to enter "\_XX\_"."  S DIR(0)="Y",DIR("A")=PROMPT,DIR("B")=DEFAULT A1 ;  D ^DIR  I Y?1"^"1.E D JUMPERR^IBTRH5H G A1  Q:$D(DUOUT) -2 ; User Pressed ^  Q:$D(DTOUT) -1 ; User timed out  I Y=0 Q -1  Q 1  ; SELENT(ARRAY,H1,H2,L1,L2,MAX,INDENT,SECT) ; Select an entry to add/edit from a list  ; Input: ARRAY() - Array of multiple lines to be displayed  ; H1 - 1st line of Header Information  ; H2 - 2nd line of Header Information  ; L1 - 1st line of DIR display  ; L2 - Selection line text  ; MAX - Multiple Description  ; If passed, entering a new line is not allowed  ; Optional, defaults to "" if not passed  ; INDENT - 1 to indent 2 spaces  ; Optional, defaults to 0  ; SECT - Section Header  ; Returns: # - User wants to edit Entry #  ; 0 - User wants to Add a new Entry  ; -1 - User wants to skip this section  ; -2 - User wants to exit template  N DEL,DIR,DIROUT,DIRUT,DOK,DTOUT,DUOUT,IX,LN,X,XX,Y,YY  S:'$D(MAX) MAX=""  S:'$D(INDENT) INDENT=0  S:'$D(SECT) SECT=""  S DIR(0)="FO",LN=0  S LN=LN+1,DIR("A",LN)=L1  S LN=LN+1,DIR("A",LN)=" "  S LN=LN+1,DIR("A",LN)=H1  S LN=LN+1,DIR("A",LN)=H2  S IX=""  F  D  Q:IX=""  . S IX=$O(ARRAY(IX))  . Q:IX=""  . S LN=LN+1,DIR("A",LN)=$P(ARRAY(IX),"^",2)  S LN=LN+1,DIR("A",LN)=" "  S LN=LN+1,DIR("A",LN)=L2  S DIR("A")=$S(INDENT:" ",1:"")\_"Selection #"  W !! SELE1 ;  S XX="Select NO to skip this section. Select YES to enter "\_SECT\_"."  S XX=XX\_" To delete an entry from the list, select D followed by the "  S XX=XX\_"number of the entry you wish to delete."  S DIR("?")=XX  D ^DIR  S DOK=1  S Y=$$UP^XLFSTR(Y) ; Convert to Upper  I Y?1"D".N D  Q:DOK Y  . S XX=$P(Y,"D",2)  . I XX>0,XX'>CNT,XX?.N Q                       ; Selected Entry to delete  . S DOK=0  . D SELERR(INDENT)  G:'DOK SELE1  I Y?1"^"1.E D JUMPERR^IBTRH5H G SELE1  I $D(DUOUT) Q -2 ; User pressed ^  I $D(DTOUT) Q -1 ; User timed out  I Y="" Q -1 ; User pressed return  S XX=$$UP^XLFSTR(Y)  S YY=$S((XX="NEW")!(XX="N")!(XX="NE"):1,1:0) ; User wants to enter a new one  I MAX'="",YY D  G SELE1  . W \*7,!!,$S(INDENT:" ",1:"")  . W "The maximum Number of "\_MAX\_" have already been entered.",!  Q:YY 0 ; Creating a new one  I XX>0,XX'>CNT,XX?.N Q XX                      ; Selected Entry  D SELERR(INDENT)  G SELE1  ; SELERR(INDENT) ; Multiple Selection error  ; Input: INDENT - 1 to indent error message display  W !!,\*7,$S(INDENT:" ",1:"")  W "Enter a number from 1-",CNT,". Enter NEW to enter a new entry."  W !,$S(INDENT:" ",1:"")  W "To delete an entry from the list, select D followed by the "  W !,$S(INDENT:" ",1:"")  W "number of the entry you wish to remove. Press return to skip selection."  W !!  Q | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBTRH5D ;ALB/FA - HCSR Create 278 Request ;12-AUG-2014  ;;2.0;INTEGRATED BILLING;\*\*517,**592**\*\*;21-MAR-94;Build 240  ;;Per VA Directive 6402, this routine should not be modified.  ;;  ; Contains Functions used in creating a 278 request from a  ; selected entry in the HCSR Response worklist  ;  ; -------------------------- Entry Points --------------------------------  ; SELAPI - Allows the user to see a quick view of the currently entered  ; Additional Patient Information lines and either pick one to   ; edit, enter a new one or skip.  ; SELDX - Allows the user to see a quick view of the currently entered  ; Diagnoses and either pick one to edit, enter a new one or  ; skip.  ; SELPD - Allows the user to see a quick view of the currently entered  ; Patient Event Provider Data Lines and either pick one to   ; edit, enter a new one or skip.  ;-----------------------------------------------------------------------------  ; SELAPI(IBTRIEN) ;EP  ; Called from within Input template IB CREATE 278 REQUEST  ; Provides the user with a quick view of currently entered Additional Patient  ; Information multiples and allows them to select one to edit or enter a new   ; one.  ; Input: IBTRIEN - IEN of the 356.22 entry being edited  ; Returns: Value of the .01 field of the multiple to edit  ; "" if creating a new multiple, -2 to exit template  ; IBNEW - 1 if creating a new entry  N AIDATA,CNT,ENTNUM,FDA,IEN,H1,H2,L1,L2,MAX,RETIEN,RTYPE,SECT,X,XX,Y,YY  S IBNEW=0,SECT="Additional Patient Information"  ;  ; First check for an empty Additional Patient Information Line to delete  D DELAPI(IBTRIEN)  ;  ; Next create an array of all current Additional Patient Information lines to  ; display  S XX=+$P($G(^IBT(356.22,IBTRIEN,11,0)),"^",4) ; Total # of API Lines  S MAX=$S(XX<10:"",1:"Additional Patient Information Lines")  S IEN=0,CNT=0  F  D  Q:+IEN=0  . S IEN=$O(^IBT(356.22,IBTRIEN,11,IEN))  . Q:+IEN=0  . S CNT=CNT+1  . S XX=$$LJ^XLFSTR(CNT,4) ; Selection #  . S YY=$$GET1^DIQ(356.2211,IEN\_","\_DA\_",",.01) ; Report Type Desc  . S YY=$E(YY,1,28)\_" "  . S XX=XX\_$$LJ^XLFSTR(YY,30)  . S YY=$$GET1^DIQ(356.2211,IEN\_","\_DA\_",",.02) ; Delivery Method  . S YY=$E(YY,1,20)\_" "  . S XX=XX\_$$LJ^XLFSTR(YY,23)  . S YY=$$GET1^DIQ(356.2211,IEN\_","\_DA\_",",.03) ; Attachment #  . S YY=$E(YY,1,22)  . S XX=XX\_$$LJ^XLFSTR(YY,22)  . S AIDATA(CNT)=IEN\_"^"\_XX  ;  I 'CNT D  Q $S($O(RETIEN(0)):RETIEN($O(RETIEN(0))),1:XX)  . W !!,"No Additional Patient Information is currently on file.",!  . S XX=$$ASKNEW("Add Additional Patient Information","NO")  . Q:XX<0  . S RTYPE=$$RTYPE(IBTRIEN) ; Get the .01 value  . I RTYPE="" S XX=-1 Q                         ; None entered  . S IBNEW=1,XX=RTYPE  . S FDA(356.2211,"+1,"\_IBTRIEN\_",",.01)=RTYPE  . D UPDATE^DIE("","FDA","RETIEN") ; File the new line  ;  ; Next display all of the current Additional Patient Information  S H1="# Report Type Delivery Method Attachment Control #"  S H2="-- ---------------------------- --------------------- ----------------------"  S L1="The following Additional Patient Information is currently on file."  S L2="Enter the # of an entry to edit, 'NEW' to add one or press Return to skip."  S XX=$$SELENT(.AIDATA,H1,H2,L1,L2,MAX,"",SECT)  I XX?1"D".N D  Q -3  . S (XX,ENTNUM)=$P(XX,"D",2)  . S XX=$P(AIDATA(XX),U)  . D DELAPI(IBTRIEN,XX)  . W !,"Entry #",ENTNUM," has been deleted."  I XX<0 Q XX  I XX=0 D  Q $S($O(RETIEN(0)):RETIEN($O(RETIEN(0))),1:XX)  . S RTYPE=$$RTYPE(IBTRIEN) ; Get the .01 value  . I RTYPE="" S XX=-1 Q                         ; None entered  . S IBNEW=1  . S XX=RTYPE  . S FDA(356.2211,"+1,"\_IBTRIEN\_",",.01)=RTYPE  . D UPDATE^DIE("","FDA","RETIEN") ; File the new line  Q $P(AIDATA(XX),"^",1)  ; DELAPI(IBTRIEN,IEN) ; Checks to see if the user entered 'NEW' to create a new   ; Additional Patient Information Line and didn't enter any data for it. Also  ; checks to see if user selected to delete a specified line. If so, the   ; Additional Patient Information line with no data (or selected) is deleted  ; Input: IBTRIEN - IEN of the 356.22 entry being edited  ; IEN - Optional, IEN of the multiple to be deleted if passed  ; defaults to ""  ; Output: Empty or selected Additional Patient Information line is deleted (Potentially)  N APIIEN,DA,DIK,X,XX,Y  S:'$D(IEN) IEN=""  I IEN'="" D  Q  . S DA(1)=IBTRIEN,DA=IEN  . S DIK="^IBT(356.22,DA(1),11,"  . D ^DIK ; Delete the multiple  ;  S APIIEN=+$P($G(^IBT(356.22,IBTRIEN,11,0)),"^",3) ; Last Multiple IEN  Q:'APIIEN  S XX=$G(^IBT(356.22,IBTRIEN,11,APIIEN,0))  S $P(XX,"^",1)=""                                  ; Remove .01 field  Q:$TR(XX,"^","")'=""                               ; 0 node data exists  S DA(1)=IBTRIEN,DA=APIIEN  S DIK="^IBT(356.22,DA(1),11,"  D ^DIK ; Delete the multiple  Q  ; RTYPE(IBTRIEN) ; Prompts the user to enter the .01 (Report Type) field of the  ; Additional Patient Information multiple  ; Input: IBTRIEN - IEN of the 356.22 entry being edited  ; Returns: IEN of the selected Report Type or "" of not entered  N DA,DIR,DIROUT,DIRUT,DTOUT,DUOUT,X,Y  S DA(1)=IBTRIEN  S DIR(0)="356.2211,.01",DIR("A")=" Report Type"  D ^DIR  Q:$D(DIRUT) ""  Q $P(Y,"^",1)  ; SELPD(IBTRIEN) ;EP  ; Called from within Input template IB CREATE 278 REQUEST  ; Provides the user with a quick view of currently entered Provider Data  ; multiples and allows them to select one to edit or enter a new one.  ; Input: IBTRIEN - IEN of the 356.22 entry being edited  ; IBTRBRF - 1 if this display is being used from the brief template  ; 0 or undefined otherwise  ; Returns: Value of the .01 field of the multiple to edit  ; "" if creating a new multiple, -2 to exit template  ; IBNEW=1 when creating a new entry  N CNT,ENTNUM,FDA,IEN,H1,H2,L1,L2,MAX,PDDATA,PTYPE,RETIEN,SECT,X,XX,Y,YY  S IBNEW=0,SECT="Provider Data Information"  ;  ; First check for an empty Provider Data Line to delete  D DELPD(IBTRIEN)  ;  ; Next create an array of all current Provider Data Information lines  S XX=+$P($G(^IBT(356.22,IBTRIEN,13,0)),"^",4) ; # of Multiples  S MAX=$S(XX<14:"",1:"Provider Data Lines")  S IEN=0,CNT=0  F  D  Q:+IEN=0  . S IEN=$O(^IBT(356.22,IBTRIEN,13,IEN))  . Q:+IEN=0  . S CNT=CNT+1  . S XX=$$LJ^XLFSTR(CNT,4) ; Selection #  . S YY=$$GET1^DIQ(356.2213,IEN\_","\_DA\_",",.01) ; Prov Type Desc  . S YY=$E(YY,1,30)\_" "  . S XX=XX\_$$LJ^XLFSTR(YY,32)  . ;  . ; IBTRBRF is defined in IB CREATE 278 REQUEST SHORT input template  . I $G(IBTRBRF)'=1 D  . . S YY=$$GET1^DIQ(356.2213,IEN\_","\_DA\_",",.02) ; Person/Non-Person  . . S XX=XX\_$$LJ^XLFSTR(YY,12)  . S YY=$$GET1^DIQ(356.2213,IEN\_","\_DA\_",",.03)  . S XX=XX\_$$LJ^XLFSTR(YY,"28T")  . S PDDATA(CNT)=IEN\_"^"\_XX  ;   I 'CNT D  Q $S($O(RETIEN(0)):RETIEN($O(RETIEN(0))),1:XX)  .I $G(IBTRBRF)'=1 D  ..W !!,"No Provider Data Information is currently on file.",!  ..S XX=$$ASKNEW("Add Provider Data Information")  ..Q  .I $G(IBTRBRF)=1 S XX=0  .Q:XX<0  .S PTYPE=$$PTYPE(IBTRIEN) ; Get the .01 value  .I PTYPE="" S XX=-1 Q                         ; None entered  .S IBNEW=1,XX=PTYPE  .S FDA(356.2213,"+1,"\_IBTRIEN\_",",.01)=PTYPE  .D UPDATE^DIE("","FDA","RETIEN") ; File the new line  .Q  ;  ; Next display all of the current Provider Data lines  S H1="# Provider Type "  I $G(IBTRBRF)'=1 S H1=H1\_" Per/Non"  S H1=H1\_" Provider"  S H2="-- ------------------------------"  I $G(IBTRBRF)'=1 S H2=H2\_" ----------"  S H2=H2\_" ------------------------------"  S L1="The following Provider Data Information is currently on file."  S L2="Enter the # of an entry to edit, 'NEW' to add one or press Return to skip."  S XX=$$SELENT(.PDDATA,H1,H2,L1,L2,MAX,"",SECT)  I XX?1"D".N D  Q -3  . S (XX,ENTNUM)=$P(XX,"D",2)  . S XX=$P(PDDATA(XX),U)  . D DELPD(IBTRIEN,XX)  . W !,"Entry #",ENTNUM," has been deleted."  I XX<0 Q XX  I XX=0 D  Q $S($O(RETIEN(0)):RETIEN($O(RETIEN(0))),1:XX)  . S PTYPE=$$PTYPE(IBTRIEN) ; Get the .01 value  . I PTYPE="" S XX=-1 Q                         ; None entered  . S XX=PTYPE  . S IBNEW=1  . S FDA(356.2213,"+1,"\_IBTRIEN\_",",.01)=PTYPE  . D UPDATE^DIE("","FDA","RETIEN") ; File the new line  Q $P(PDDATA(XX),"^",1)  ; DELPD(IBTRIEN,IEN) ; Checks to see if the user entered 'NEW' to create a new   ; Provider Data Line and didn't enter any data for it or selected a line to   ; be deleted. If so, the empty or selected Provider Data line is deleted  ; Input: IBTRIEN - IEN of the 356.22 entry being edited  ; IEN - Optional, IEN of the multiple to be deleted if passed  ; defaults to ""  ; Output: Empty or selected Provider Data line is deleted (Potentially)  N PDIEN,DA,DIK,X,XX,Y  S:'$D(IEN) IEN=""  I IEN'="" D  Q  . S DA(1)=IBTRIEN,DA=IEN  . S DIK="^IBT(356.22,DA(1),13,"  . D ^DIK ; Delete the multiple  ;  S PDIEN=+$P($G(^IBT(356.22,IBTRIEN,13,0)),"^",3) ; Last Multiple IEN  Q:'PDIEN  S XX=$G(^IBT(356.22,IBTRIEN,13,PDIEN,0))  S $P(XX,"^",1)=""                                  ; Remove .01 field  Q:$TR(XX,"^","")'=""                               ; 0 node data exists  S DA(1)=IBTRIEN,DA=PDIEN  S DIK="^IBT(356.22,DA(1),13,"  D ^DIK ; Delete the multiple  Q  ; PTYPE(IBTRIEN) ; Prompts the user to enter the .01 (Provider Type) field of the  ; Provider Data multiple  ; Input: IBTRIEN - IEN of the 356.22 entry being edited  ; Returns: IEN of the selected Provider Type or "" of not entered  N DA,DIR,DIROUT,DIRUT,DTOUT,DUOUT,X,Y  S DA(1)=IBTRIEN  S DIR(0)="356.2213,.01",DIR("A")=" Provider Type"  D ^DIR  Q:$D(DIRUT) ""  Q $P(Y,"^",1)  ; SELDX(IBTRIEN) ;EP  ; Called from within Input template IB CREATE 278 REQUEST  ; Provides the user with a quick view of currently entered Diagnoses and  ; allows them to select one to edit or enter a new diagnosis.  ; Input: IBTRIEN - IEN of the 356.22 entry being edited  ; IBTRBRF - 1 if this display is being used from the brief template  ; 0 or undefined other otherwise  ; Returns: Value of the .01 field of the multiple to edit  ; "" if creating a new multiple, -2 to exit template  ; -3 if a if a line was deleted  ; IBNEW=1 when creating a new entry  N CNT,DXDATA,DXTYPE,ENTNUM,FDA,IEN,H1,H2,L1,L2,MAX,RETIEN,SECT,X,XX,Y,YY  S IBNEW=0,SECT="Diagnosis Information"  ;  ; First check for an empty Diagnosis Line to delete  D DELDX(IBTRIEN)  ;  ; Next create an array of all current Diagnoses lines  S XX=+$P($G(^IBT(356.22,IBTRIEN,3,0)),"^",4) ; Total # of Dx Lines  S MAX=$S(XX<12:"",1:"Diagnosis Lines")  S IEN=0,CNT=0  F  D  Q:+IEN=0  . S IEN=$O(^IBT(356.22,IBTRIEN,3,IEN))  . Q:+IEN=0  . S CNT=CNT+1  . S XX=$$LJ^XLFSTR(CNT,4) ; Selection #  . S YY=$$GET1^DIQ(356.223,IEN\_","\_DA\_",",.01,"I") ; Diagnosis Type  . S YY=$$GET1^DIQ(356.006,YY\_",",.01)  . S XX=XX\_$$LJ^XLFSTR(YY,7)  . S YY=$$GET1^DIQ(356.223,IEN\_","\_DA\_",",.02) ; Diagnosis  . S XX=XX\_$$LJ^XLFSTR(YY,11)  . I $G(IBTRBRF)'=1 D  . . S YY=$$GET1^DIQ(356.223,IEN\_","\_DA\_",",.03) ; Date Known  . . S XX=XX\_$$LJ^XLFSTR(YY,14)  . S DXDATA(CNT)=IEN\_"^"\_XX  ;  ; Creating 1st Diagnosis Line?  I 'CNT D  Q $S($O(RETIEN(0)):RETIEN($O(RETIEN(0))),1:XX)  .I $G(IBTRBRF)'=1 D  ..W !!,"No Diagnosis Information is currently on file.",!  ..S XX=$$ASKNEW("Add a new Diagnosis")  ..Q  .I $G(IBTRBRF)=1 S XX=0  .Q:XX<0  .S DXTYPE=$$DXTYPE(IBTRIEN) ; Get the .01 value  .I DXTYPE="" S XX=-1 Q                        ; None entered  .S IBNEW=1,XX=DXTYPE  .S FDA(356.223,"+1,"\_IBTRIEN\_",",.01)=DXTYPE  .D UPDATE^DIE("","FDA","RETIEN") ; File the new line  .Q  ;  ; Next display all of the current Diagnoses and let the user select one  S H1="# Type Diagnosis"  I $G(IBTRBRF)'=1 S H1=H1\_" Date DX Known"  S H2="-- ----- ---------"  I $G(IBTRBRF)'=1 S H2=H2\_" -------------"  S L1="The following Diagnoses are currently on file."  S L2="Enter the # of a Diagnosis to edit, 'NEW' to add one or press Return to skip."  S XX=$$SELENT(.DXDATA,H1,H2,L1,L2,MAX,"",SECT)  I XX?1"D".N D  Q -3  . S (XX,ENTNUM)=$P(XX,"D",2)  . S XX=$P(DXDATA(XX),U)  . D DELDX(IBTRIEN,XX)  . W !,"Entry #",ENTNUM," has been deleted."  I XX<0 Q XX  I XX=0 D  Q $S($O(RETIEN(0)):RETIEN($O(RETIEN(0))),1:XX)  . S DXTYPE=$$DXTYPE(IBTRIEN) ; Get the .01 value  . I DXTYPE="" S XX=-1 Q                        ; None entered  . S XX=DXTYPE  . S IBNEW=1  . S FDA(356.223,"+1,"\_IBTRIEN\_",",.01)=DXTYPE  . D UPDATE^DIE("","FDA","RETIEN") ; File the new line  Q $P(DXDATA(XX),"^",1)  ; DXTYPE(IBTRIEN) ; Prompts the user to enter the .01 (Diagnosis Type) field of  ; the diagnosis multiple  ; Input: IBTRIEN - IEN of the 356.22 entry being edited  ; Returns: IEN of the selected Diagnosis Type or "" of not entered  N DA,DIR,DIROUT,DIRUT,DTOUT,DUOUT,X,Y  S DA(1)=IBTRIEN,DA=$P($G(^IBT(356.22,IBTRIEN,3,0)),"^",3)+1  S DIR(0)="356.223,.01",DIR("A")=" Diagnosis Qualifier"  D ^DIR  Q:$D(DIRUT) ""  Q $P(Y,"^",1)  ; DELDX(IBTRIEN,IEN) ; Checks to see if the user entered 'NEW' to create a new   ; Diagnosis Line and didn't enter any data for it or selected a multiple to  ; to be deleted. If so, the empty or selected multiple is deleted  ; Input: IBTRIEN - IEN of the 356.22 entry being edited  ; IEN - Optional, IEN of the multiple to be deleted if passed  ; defaults to ""  ; Output: Empty or selected Diagnosis line is deleted (Potentially)  N DA,DIK,DXIEN,X,XX,Y  S:'$D(IEN) IEN=""  I IEN'="" D  Q  . S DA(1)=IBTRIEN,DA=IEN  . S DIK="^IBT(356.22,DA(1),3,"  . D ^DIK ; Delete the multiple  ;  S DXIEN=+$P($G(^IBT(356.22,IBTRIEN,3,0)),"^",3) ; Last Multiple IEN  Q:'DXIEN  S XX=$G(^IBT(356.22,IBTRIEN,3,DXIEN,0))  S $P(XX,"^",1)=""                                  ; Remove .01 field  Q:$TR(XX,"^","")'=""                               ; 0 node data exists  S DA(1)=IBTRIEN,DA=DXIEN  S DIK="^IBT(356.22,DA(1),3,"  D ^DIK ; Delete the multiple  Q  ; ASKNEW(PROMPT,DEFAULT) ;EP  ; Ask if user wants to create a new entry  ; Input: PROMPT - Yes/No question to ask the user  ; DEFALT - Default Answer  ; Optional, if not passed, set to 'YES'  ; Returns: 0 - User wants to add a new Entry  ; -1 - User doesn't want to add a new entry  ; -2 - User wants to exit template  N DIR,DIROUT,DIRUT,DTOUT,DUOUT,X,XX,Y  S:'$D(DEFAULT) DEFAULT="YES"  S XX=$P(PROMPT,"Add ",2)  S DIR("?")="Select NO to skip this section. Select YES to enter "\_XX\_"."  S DIR(0)="Y",DIR("A")=PROMPT,DIR("B")=DEFAULT A1 ;  D ^DIR  I Y?1"^"1.E D JUMPERR^IBTRH5H G A1  Q:$D(DUOUT) -2 ; User Pressed ^  Q:$D(DTOUT) -1 ; User timed out  I Y=0 Q -1  Q 1  ; SELENT(ARRAY,H1,H2,L1,L2,MAX,INDENT,SECT) ; Select an entry to add/edit from a list  ; Input: ARRAY() - Array of multiple lines to be displayed  ; H1 - 1st line of Header Information  ; H2 - 2nd line of Header Information  ; L1 - 1st line of DIR display  ; L2 - Selection line text  ; MAX - Multiple Description  ; If passed, entering a new line is not allowed  ; Optional, defaults to "" if not passed  ; INDENT - 1 to indent 2 spaces  ; Optional, defaults to 0  ; SECT - Section Header  ; Returns: # - User wants to edit Entry #  ; 0 - User wants to Add a new Entry  ; -1 - User wants to skip this section  ; -2 - User wants to exit template  N DEL,DIR,DIROUT,DIRUT,DOK,DTOUT,DUOUT,IX,LN,X,XX,Y,YY  S:'$D(MAX) MAX=""  S:'$D(INDENT) INDENT=0  S:'$D(SECT) SECT=""  S DIR(0)="FO",LN=0  S LN=LN+1,DIR("A",LN)=L1  S LN=LN+1,DIR("A",LN)=" "  S LN=LN+1,DIR("A",LN)=H1  S LN=LN+1,DIR("A",LN)=H2  S IX=""  F  D  Q:IX=""  . S IX=$O(ARRAY(IX))  . Q:IX=""  . S LN=LN+1,DIR("A",LN)=$P(ARRAY(IX),"^",2)  S LN=LN+1,DIR("A",LN)=" "  S LN=LN+1,DIR("A",LN)=L2  S DIR("A")=$S(INDENT:" ",1:"")\_"Selection #"  W !! SELE1 ;  ;S XX="Select NO to skip this section. Select YES to enter "\_SECT\_"."  S XX="To delete an entry from the list, select D followed by the "  S XX=XX\_"number of the entry you wish to delete."  S DIR("?")=XX  D ^DIR  S DOK=1  S Y=$$UP^XLFSTR(Y) ; Convert to Upper  I Y?1"D".N D  Q:DOK Y  . S XX=$P(Y,"D",2)  . I XX>0,XX'>CNT,XX?.N Q                       ; Selected Entry to delete  . S DOK=0  . D SELERR(INDENT)  G:'DOK SELE1  I Y?1"^"1.E D JUMPERR^IBTRH5H G SELE1  I $D(DUOUT) Q -2 ; User pressed ^  I $D(DTOUT) Q -1 ; User timed out  I Y="" Q -1 ; User pressed return  S XX=$$UP^XLFSTR(Y)  S YY=$S((XX="NEW")!(XX="N")!(XX="NE"):1,1:0) ; User wants to enter a new one  I MAX'="",YY D  G SELE1  . W \*7,!!,$S(INDENT:" ",1:"")  . W "The maximum Number of "\_MAX\_" have already been entered.",!  **. ;JWS;IB\*2.0\*592**  **. I +CNT>21 R !!,"Press <ENTER> to continue",X:30**  Q:YY 0 ; Creating a new one  I XX>0,XX'>CNT,XX?.N Q XX                      ; Selected Entry  D SELERR(INDENT)  G SELE1  ; SELERR(INDENT) ; Multiple Selection error  ; Input: INDENT - 1 to indent error message display  W !!,\*7,$S(INDENT:" ",1:"")  W "Enter a number from 1-",CNT,". Enter NEW to enter a new entry."  W !,$S(INDENT:" ",1:"")  W "To delete an entry from the list, select D followed by the "  W !,$S(INDENT:" ",1:"")  W "number of the entry you wish to remove. Press return to skip selection."  W !!  Q | | | | | | | | | |

1. The following input template needs to be modified to handle the changes to IB SCREEN8.

NUMBER: 1513 NAME: IB SCREEN8

DATE CREATED: MAR 13, 2014@09:53 READ ACCESS: @

FILE: BILL/CLAIMS WRITE ACCESS: @

DATE LAST USED: MAR 20, 2017

ROUTINE INVOKED: ^IBXS8 PREVIOUS ROUTINE INVOKED: IBXS8

EDIT FIELDS (c)

: @81

: **I $$FT^IBCU3(IBIFN)=7 S Y="@801"**

: S:IBDR20'["81" Y="@82"

: COB TOTAL NON-COVERED AMOUNT;"COB Non-Covered Charge Amt"

: @82

: S:IBDR20'["82" Y="@83"

: PROPERTY/CASUALTY CLAIM NUMBER;"Claim Number"

: S:IBT=3 Y="@84"

: PROP/CAS DATE OF 1ST CONTACT;"Date of 1st Contact"

: PROPERTY/CASUALTY CONTACT NAME;"Contact Name"

: PROP/CAS COMMUNICATION NUMBER;"Contact Phone"

: PROP/CAS EXTENSION NUMBER;"Contact Phone Extension"

: @83

: S:IBDR20'["83" Y="@84"

: PRIMARY SURGICAL PROC CODE;"Primary Code"

: SECONDARY SURGICAL PROC CODE;"Secondary Code"

: @84

: S:IBDR20'["84" Y="@85"

: ATTACHMENT REPORT TYPE;"Report Type"

: ATTACHMENT REPORT TRANS CODE;"Transmission Method"

: ATTACHMENT CONTROL NUMBER;"Attachment Control #"

: @85

: S:IBDR20'["85" Y="@86"

: DISABILITY START DATE;"Disability Start Date"

: DISABILITY END DATE;"Disability End Date"

: @86

: S:IBDR20'["86" Y="@87"

: S:$P($G(^DGCR(IBIFN,0)),U,19)=3 Y="@87"

: ASSUMED CARE DATE;"Assumed Care Date"

: RELINQUISHED CARE DATE;"Relinquished Care Date"

: @87

: S:IBDR20'["87" Y="@88"

: SPECIAL PROGRAM INDICATOR////^S X=$S($P($G(^DGCR(399,DA,

"U2")),U,16)'="":$P($G(^DGCR(399,DA,"U2")),U,16),

$$WNRBILL^IBEFUNC(DA):"31",1:"")

: SPECIAL PROGRAM INDICATOR;"Special Program"

: @88

: S:IBDR20'["88" Y="@89"

: HOMEBOUND;"Homebound"

: @89

: S:IBDR20'["89" Y="@899"

: DATE LAST SEEN;"Date Last Seen"

: @899 COMPILED (c): YES

EDIT FIELDS (c) : @81

: S:IBDR20'["81" Y="@82"

: COB TOTAL NON-COVERED AMOUNT;"COB Non-Covered Charge Amt"

: @82

: S:IBDR20'["82" Y="@83"

: PROPERTY/CASUALTY CLAIM NUMBER;"Claim Number"

: S:IBT=3 Y="@84"

: PROP/CAS DATE OF 1ST CONTACT;"Date of 1st Contact"

: PROPERTY/CASUALTY CONTACT NAME;"Contact Name"

: PROP/CAS COMMUNICATION NUMBER;"Contact Phone"

: PROP/CAS EXTENSION NUMBER;"Contact Phone Extension"

: @83

: S:IBDR20'["83" Y="@84"

: PRIMARY SURGICAL PROC CODE;"Primary Code"

: SECONDARY SURGICAL PROC CODE;"Secondary Code"

: @84

: S:IBDR20'["84" Y="@85"

: ATTACHMENT REPORT TYPE;"Report Type"

: ATTACHMENT REPORT TRANS CODE;"Transmission Method"

: ATTACHMENT CONTROL NUMBER;"Attachment Control #"

: @85

: S:IBDR20'["85" Y="@86"

: DISABILITY START DATE;"Disability Start Date"

: DISABILITY END DATE;"Disability End Date"

: @86

: S:IBDR20'["86" Y="@87"

: S:$P($G(^DGCR(IBIFN,0)),U,19)=3 Y="@87"

: ASSUMED CARE DATE;"Assumed Care Date"

: RELINQUISHED CARE DATE;"Relinquished Care Date"

: @87

: S:IBDR20'["87" Y="@88"

: 238////^S X=$S($P($G(^DGCR(399,DA,"U2")),U,16)'="":

$P($G(^DGCR(399,DA,"U2")),U,16),$$WNRBILL^IBEFUNC(DA):"31",1:"")

: SPECIAL PROGRAM INDICATOR;"Special Program"

: @88

: S:IBDR20'["88" Y="@89"

: HOMEBOUND;"Homebound"

: @89

: S:IBDR20'["89" Y="@899"

: DATE LAST SEEN;"Date Last Seen"

**: Y=”@899”**

**: @801**

**: S:IBDR20'["81" Y="@802"**

**: TOOTH STATUS**

**: TOOTH NUMBER;"Tooth Number"**

**: STATUS CODE;”Status Code”**

**; @802**

**: S:IBDR20'["82" Y="@803"**

**; BANDING DATE;”Banding Date”**

**: TREATMENT INDICATOR;"Treatment Indicator"**

**: TREATMENT MONTHS COUNT;"Treatment Months Count"**

**: TREATMENT MONTHS REMAINING;"Treatment Months**

**Remaining Count"**

**; @803**

**: S:IBDR20'["83" Y="@899"**

**: ATTACHMENT REPORT TYPE;"Report Type"**

**: ATTACHMENT REPORT TRANS CODE;"Transmission Method"**

**: ATTACHMENT CONTROL NUMBER;"Attachment Control #"**

: @899

1. The following input template needs to be modified to handle the changes to IB SCREEN 10.

NUMBER: 2787 NAME: IB SCREEN10H

DATE CREATED: MAR 07, 2017@11:12 READ ACCESS: @

FILE: BILL/CLAIMS USER #: 520824637

WRITE ACCESS: @ DATE LAST USED: MAR 07, 2017

ROUTINE INVOKED: ^IBXSAH PREVIOUS ROUTINE INVOKED:IBXSAH

EDIT FIELDS (c)

: K DIPA S DIPA("I1")=$G(^DGCR(399,DA,

"I1")),DIPA("I2")=$G(^("I2")),

DIPA("I3")=$G(^("I3"))

: S:IBDR20'["101" Y="@102"

: UNABLE TO WORK FROM

: UNABLE TO WORK TO

: @102

: S:IBDR20'["102" Y="@103"

: S:'$$INPAT^IBCEF(DA) Y="@1021"

: ADMITTING DIAGNOSIS

: @1021

: FORM LOCATOR 64A;T

: S:'DIPA("I2") Y="@1025"

: FORM LOCATOR 64B;T

: S:'DIPA("I3") Y="@1025"

: FORM LOCATOR 64C

: @1025

: TREATMENT AUTHORIZATION CODE;"PRIMARY AUTHORIZATION

CODE"

: PRIMARY REFERRAL NUMBER

: S:'DIPA("I2") Y="@1029"

: SECONDARY AUTHORIZATION CODE

: SECONDARY REFERRAL NUMBER

: S:'DIPA("I3") Y="@1029"

: TERTIARY AUTHORIZATION CODE

: TERTIARY REFERRAL NUMBER

: @1029

: @103

: S:IBDR20'["103" Y="@104"

: PROVIDER

: FUNCTION

: S DIPA("RF")=X S:$D(^XUSEC("IB PROVIDER EDIT",DUZ))

DLAYGO=355.93

: PERFORMED BY

: K DLAYGO S DIPA("PRF")=X S:X="" Y="@10399"

: N Z1 S Z1=$P($G(^DGCR(399,DA(1),"PRV",DA,0)),U,2)

S DIPA("NVA\_PRV")=$S(Z1["IBA(355.93":+Z1,1:0)

: S:DIPA("NVA\_PRV")=0 Y="@1038"

: PERFORMED BY:355.93:

: S DIPA("NVA\_PRV-

0")=$G(^IBA(355.93,DIPA("NVA\_PRV"),0))

: S:$P(DIPA("NVA\_PRV-0"),U,2)=1 Y="@1032"

: S:$P(DIPA("NVA\_PRV-0"),U,3)'="" Y="@1031"

: CREDENTIALS

: @1031

: S:$P(DIPA("NVA\_PRV-0"),U,3)'="" Y="@1035"

: SPECIALTY

: S Y="@1035"

: @1032

: S:$P(DIPA("NVA\_PRV-

0"),U,5)'=""&($P(DIPA("NVA\_PRV-0"),U,6

)'="")&($P(DIPA("NVA\_PRV-0"),U,7)'="")

Y="@1033"

: STREET ADDRESS

: STREET ADDRESS LINE 2

: CITY

: STATE

: ZIP CODE

: @1033

: S:$P(DIPA("NVA\_PRV-0"),U,9)'="" Y="@1034"

: FACILITY DEFAULT ID NUMBER;"LAB OR FACILITY

PRIMARY ID"

: @1034

: S:$P(DIPA("NVA\_PRV-0"),U,11)'="" Y="@1035"

: X12 TYPE OF FACILITY

: @1035

: S:$P(DIPA("NVA\_PRV-0"),U,14)'="" Y="@1036"

: D EN2^IBCEP82(DIPA("NVA\_PRV"),4)

: @1036

: S:$D(^IBA(355.93,DIPA("NVA\_PRV"),"TAXONOMY"))>0

Y="@1037"

: TAXONOMY CODE

: ALL

: @1037

: @1038

: S DIK="^DGCR(399,"\_DA(1)\_",""PRV"",",DIK(1)=".02" D

EN1^DIK K DIK

: TAXONOMY

: D DISPTAX^IBCEP81($P($G(^DGCR(399,DA(1),"PRV"

,DA,0)),U,15),"")

: N Z S Z=$$EXPAND^IBTRE(399.0222,.08,$P($G(^DGCR(399

,DA(1),"PRV",DA,0)),U,8)),DIPA("SPC")=$S(Z'="":Z,1:"UNSPECIFIED") W !," Prov Specialty On File: ",DIPA("SPC")

: S DIPA("CRD")=$$CRED^IBCEU($P(^DGCR(399

,DA(1),"PRV",DA,0),U,2))

: CREDENTIALS

: K DIPA("W1") S:$G(DIPA("CRD"))'=$P(^DGCR(399

,DA(1),"PRV",DA,0),U,3) DIPA("W1")=1

: I $G(DIPA("W1")) D WRT1^IBCSC10H($G(DIPA("CRD")))

: K DIPA("W1")

: I '$G(DIPA("I1")) S Y="@10305"

: D PROVID^IBCEP2B(DA(1),DA,1,.DIPA) S

Y=$S(DIPA("EDIT")<0:"@10382",

DIPA("EDIT")=1:"@10391",DIPA("EDIT")

=2:"@10371",1:"")

: @10382

: I '$G(DIPA("I2")) S Y="@10305"

: D PROVID^IBCEP2B(DA(1),DA,2,.DIPA) S

Y=$S(DIPA("EDIT")<0:"@10383",

DIPA("EDIT")=1:"@10392",DIPA("EDIT")

=2:"@10372",1:"")

: @10383

: I '$G(DIPA("I3")) S Y="@10305"

: D PROVID^IBCEP2B(DA(1),DA,3,.DIPA) S

Y=$S(DIPA("EDIT")<0:"@10305",

DIPA("EDIT")=1:"@10393",DIPA("EDIT")

=2:"@10373",1:"")

: S Y="@10305"

: @10391

: PRIM INS PROVIDER ID TYPE;T;REQ

: PRIMARY INS CO ID NUMBER;T

: S Y="@10382"

: @10392

: SEC INS PROVIDER ID TYPE;T;REQ

: SECONDARY INS CO ID NUMBER;T

: S Y="@10383"

: @10393

: TERT INS PROVIDER ID TYPE;T;REQ

: TERTIARY INS CO ID NUMBER;T

: S Y="@10305"

: @10371

: PRIM INS PROVIDER ID TYPE////^S X=DIPA("PRIDT")

: PRIMARY INS CO ID NUMBER////^S X=DIPA("PRID")

: S Y="@10382"

: @10372

: SEC INS PROVIDER ID TYPE////^S X=DIPA("PRIDT")

: SECONDARY INS CO ID NUMBER////^S X=DIPA("PRID")

: S Y="@10383"

: @10373

: TERT INS PROVIDER ID TYPE////^S X=DIPA("PRIDT")

: TERTIARY INS CO ID NUMBER////^S X=DIPA("PRID")

: S Y="@10305"

: @10305

: @10399

: W @IOF

: @104

: S:IBDR20'["104" Y="@106"

: NON-VA FACILITY

: S DIPA("NVA\_FC")=X S:X="" Y="@1046"

: NON-VA FACILITY:

: S DIPA("NVA\_FC-0")=$G(^IBA(355.93,+DIPA

("NVA\_FC"),0)) S:$P(DIPA("NVA\_FC

-0"),U,5)'=""&($P(DIPA("NVA\_FC

-0"),U,6)'="")&($P(DIPA("NVA\_FC-0")

,U,7)'="") Y="@1041"

: STREET ADDRESS

: STREET ADDRESS LINE 2

: CITY

: STATE

: ZIP CODE

: @1041

: S:$P(DIPA("NVA\_FC-0"),U,9)'="" Y="@1042"

: FACILITY DEFAULT ID NUMBER;"LAB OR FACILITY

PRIMARY ID"

: @1042

: S:$P(DIPA("NVA\_FC-0"),U,11)'="" Y="@1043"

: X12 TYPE OF FACILITY

: @1043

: S:$P(DIPA("NVA\_FC-0"),U,14)'="" Y="@1044"

: D EN2^IBCEP82(+DIPA("NVA\_FC"),2)

: @1044

: S:$D(^IBA(355.93,+DIPA("NVA\_FC"),

"TAXONOMY"))>0 Y="@1045"

: TAXONOMY CODE

: ALL

: @1045

: S DIK="^DGCR(399,",DIK(1)="232" D EN1^DIK

: K DIK

: NON-VA FACILITY TAXONOMY

: D DISPTAX^IBCEP81($P($G(^DGCR(399,DA,"U3"))

,U,3),"Non-VA Facility")

: NON-VA CARE TYPE

: @1046

: S DIPA("OLDCLIA")=$P($G(^DGCR(399,DA,"U2")),U,13)

: LAB CLIA NUMBER

: I X="",$G(DIPA("OLDCLIA"))'="" S IBMDOTCN=1

: @1047

: I '$$XRAY^IBCEP8A(DA) S Y="@1048"

: D MAMMODP^IBCEP8A(DA)

: MAMMOGRAPHY CERT NUMBER

: @1048

: @106

: S:IBDR20'["106" Y="@107"

: **I $$FT^IBCU3(IBIFN)=7 S Y="@1061"**

: FORM LOC 19-UNSPECIFIED DATA;T

: D ASK19^IBCEU3(DA)

: S Y="@107"

: **@1061**

**: DENTAL CLAIM NOTE**

: @107

: S:IBDR20'["107" Y="@108"

: BILLING PROVIDER TAXONOMY

: D DISPTAX^IBCEP81($P($G(^DGCR(399,

DA,"U3")),U,11),"Billing Provider")

: I $P($G(^DGCR(399,DA,"U3")),U,11) N X,Y,DIR

S DIR(0)="EA",DIR("A")="Press

Return to continue" D ^DIR K DIR

: @108

: S:IBDR20'["108" Y="@109"

: I '$G(DIPA("I1")) S Y="@109"

: K DIPA("BRANCH") S DIPA("BRANCH")=$$ACINTEL

^IBCSC10(DIPA("I1"),"@1081")

: S:DIPA("BRANCH")]"" Y=DIPA("BRANCH") K DIPA("BRANCH")

: S DIPA("OLDALTT1")=$P($G(^DGCR(399,DA,"M2")),U)

: PRIMARY PAYER-ALT ID TYPE;"Primary Payer – Alt

Prof Prim Payer ID Type"

: I X="",$G(DIPA("OLDALTT1"))="" S Y="@1081"

: I $P($G(^DGCR(399,DA,"M2")),U)="" S Y="@1081"

: S DIPA("OLDALTI1")=$P($G(^DGCR(399,DA,"M2")),U,2)

: PRIMARY PAYER-ALT ID;"Primary Payer - Alt Prof Prim

Payer ID"

: I X="",$G(DIPA("OLDALTI1"))="" S Y="@10811"

: S Y="@1081"

: @10811

: PRIMARY PAYER-ALT ID TYPE////@

: @1081

: I '$G(DIPA("I2")) S Y="@109"

: K DIPA("BRANCH") S DIPA("BRANCH")=$$ACINTEL

^IBCSC10(DIPA("I2"),"@1082")

: S:DIPA("BRANCH")]"" Y=DIPA("BRANCH") K DIPA("BRANCH")

: S DIPA("OLDALTT2")=$P($G(^DGCR(399,DA,"M2")),U,3)

: SECONDARY PAYER-ALT ID TYPE;"Secondary Payer – Alt

Prof Prim Payer ID Type"

: I X="",$G(DIPA("OLDALTT2"))="" S Y="@1082"

: I $P($G(^DGCR(399,DA,"M2")),U,3)="" S Y="@1082"

: S DIPA("OLDALTI2")=$P($G(^DGCR(399,DA,"M2")),U,4)

: SECONDARY PAYER-ALT ID;"Secondary Payer - Alt Prof

Prim Payer ID"

: I X="",$G(DIPA("OLDALTI2"))="" S Y="@10821"

: S Y="@1082"

: @10821

: SECONDARY PAYER-ALT ID TYPE////@

: @1082

: I '$G(DIPA("I3")) S Y="@109"

: K DIPA("BRANCH") S DIPA("BRANCH")=$$ACINTEL

^IBCSC10(DIPA("I3"),"@109")

: S:DIPA("BRANCH")]"" Y=DIPA("BRANCH") K DIPA("BRANCH")

: S DIPA("OLDALTT2")=$P($G(^DGCR(399,DA,"M2")),U,5)

: TERTIARY PAYER-ALT ID TYPE;"Tertiary Payer

- Alt Prof Prim Payer ID Type"

: I X="",$G(DIPA("OLDALTT3"))="" S Y="@1083"

: I $P($G(^DGCR(399,DA,"M2")),U,5)="" S Y="@1083"

: S DIPA("OLDALTI3")=$P($G(^DGCR(399,DA,"M2")),U,6)

: TERTIARY PAYER-ALT ID;"Tertiary Payer –

Alt Prof Prim Payer ID"

: I X="",$G(DIPA("OLDALTI3"))="" S Y="@10831"

: S Y="@1083"

: @10831

: TERTIARY PAYER-ALT ID////@

: @1083

: @109

: S:IBDR20'["109" Y="@1010"

: I $$TEST^IBCEF84(DA) S Y="@1090"

: I '$P($G(^DGCR(399,DA,"TX")),U,8),'$$TXMT^IBCEF4(DA)

S Y="@1092"

: I $$REQMRA^IBEFUNC(DA) S Y="@10911"

: FORCE CLAIM TO PRINT//NO FORCED PRINT

: S Y="@1092"

: @1090

: D MESSAGE^IBCEF84

: S Y="@1092"

: @10911

: FORCE PRINT MRA SECONDARY//NO FORCED PRINT

: @1092

: @1010

1. The following Post Install routine will be run after installation of patch IB\*2.0\*592 in order to perform the following funtions:
   1. Create the following new IB ERROR codes:
      1. IB357 – Rendering Provider or Assistant Surgeon required on Dental Claims.
      2. IB358 – Assistant Surgeon’s NPI is required.
      3. IB256 – Assistant Surgeon taxonomy missing.
      4. IB335 – Claim Level Assistant Surgeon differs from all Line Level Assistant Surgeons.
      5. IB359 – Medicare (WNR) does not accept Dental claims.
      6. IB362 – Insurance Company does not have Dental Coverage.
   2. Create the following new Type of Service entries for Dental:
      1. 23 DIAGNOSTIC DENTAL
      2. 24 PERIODONTICS
      3. 25 RESTORATIVE
      4. 26 ENDODONTICS
      5. 27 MAXILLOFACIAL PROSTHETICS
      6. 28 ADJUNCTIVE DENTAL SERVICES
      7. 35 DENTAL CARE
      8. 36 DENTAL CROWNS
      9. 37 DENTAL ACCIDENT
      10. 38 ORTHODONTICS
      11. 39 PROSTHODONTICS
      12. 40 ORAL SURGERY
      13. 41 PREVENTIVE DENTAL
      14. E12 BASIC RESTORATIVE – DENTAL
      15. E13 MAJOR RESTORATIVE – DENTAL
      16. E14 FIXED PROSTHODONTICS
      17. E15 REMOVABLE PROSTHODONTICS
      18. E16 INTRAORAL IMAGES - COMPLETE SERIES
      19. E17 ORAL EVALUATION
      20. E18 DENTAL PROPHYLAXIS
      21. E19 PANORAMIC IMAGES
      22. E20 SEALANTS
      23. E21 FLOURIDE TREATMENTS
      24. E22 DENTAL IMPLANTS
      25. E23 TEMPOROMANDIBULAR JOINT DYSFUNCTION
      26. F3 DENTAL COVERAGE
      27. F7 ORTHODONTIA COVERAGE
   3. Set the new Dental Claims processing flag to YES in IB Site Parameters.

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBY592PO | | | | | | | | |
| **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** | | | | | | | | | |
| N/A | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| **IBY592PO ;EDE/JWS - POST-INSTALL FOR IB\*2.0\*592 ;22-FEB-2017  ;;2.0;INTEGRATED BILLING;\*\*592\*\*;21-MAR-94;Build 6  ;;Per VA Directive 6402, this routine should not be modified.  ; EN ;Entry Point  S IBA(2)="IB\*2\*592 Post-Install...",(IBA(1),IBA(3))=" " D MES^XPDUTL(.IBA) K IBA  D UPDERR,UPDTOS  ; set default processing of Dental Claims to YES in Site Parameters  S DIE="^IBE(350.9,",DA=1,DR="8.2////1" D ^DIE  S IBA(2)="IB\*2\*592 Post-Install Complete.",(IBA(1),IBA(3))=" " D MES^XPDUTL(.IBA) K IBA  Q  ; UPDERR ; Update existing error code message for 350.8  N IBCODE,IBMESN,IBIEN,DIE,DIC,DA,DR,X,Y  S IBCODE="IB357",IBMESN="Rendering Provider or Assistant Surgeon required on Dental Claims."  S IBIEN=$O(^IBE(350.8,"C",IBCODE,0)) I 'IBIEN D CREATE  S IBCODE="IB358",IBMESN="Assistant Surgeon's NPI is required."  S IBIEN=$O(^IBE(350.8,"C",IBCODE,0)) I 'IBIEN D CREATE  S IBCODE="IB256",IBMESN="Assistant Surgeon taxonomy missing."  S IBIEN=$O(^IBE(350.8,"C",IBCODE,0)) I 'IBIEN D CREATE  S IBCODE="IB335",IBMESN="Claim Level Assistant Surgeon differs from all Line Level Assistant Surgeons."  S IBIEN=$O(^IBE(350.8,"C",IBCODE,0)) I 'IBIEN D CREATE  S IBCODE="IB359",IBMESN="Medicare (WNR) does not accept Dental claims."  S IBIEN=$O(^IBE(350.8,"C",IBCODE,0)) I 'IBIEN D CREATE  S IBCODE="IB362",IBMESN="Insurance Company does not have Dental Coverage."  S IBIEN=$O(^IBE(350.8,"C",IBCODE,0)) I 'IBIEN D CREATE  Q  ; CREATE ;Create entry for 'IB357' in D350.8 if not there  S DIC="^IBE(350.8,",DIC(0)="",X=IBCODE D FILE^DICN K DIC,X  I Y=-1 D MES^XPDUTL(">> IB ERROR - Entry '"\_IBCODE\_"' was unable to be created <<") Q  S IBIEN=+Y  S DIE="^IBE(350.8,",DA=IBIEN,DR=".02////"\_IBMESN\_";.03////"\_IBCODE\_";.04////1;.05////1" D ^DIE K DIE,DIC,DA,DR  Q  ; UPDTOS ;Create Type of Service entries for Dental file 353.2  N IBFDA,I,IBIEN,ERROR  F I=23,24,25,26,27,28,35,36,37,38,39,40,41,"E12","E13","E14","E15","E16","E17","E18","E19","E20","E21","E22","E23","F3","F7" D  . I $O(^IBE(353.2,"B",I,0)) Q  ;already exists  . S IBFDA(353.2,"+1,",.01)=I  . I +I<29 D  .. S IBFDA(353.2,"+1,",.02)=$P("DIAGNOSTIC DENTAL,PERIODONTICS,RESTORATIVE,ENDODONTICS,MAXILLOFACIAL PROSTHETICS,ADJUNCTIVE DENTAL SERVICES",",",I-22)  .. S IBFDA(353.2,"+1,",.03)=$P("DIAGNOSTIC DENTAL,PERIODONTICS,RESTORATIVE,ENDODONTICS,MAXILLOFACIAL PRO,ADJUNCTIVE SERVICES",",",I-22)  . I +I>34,+I<42 D  .. S IBFDA(353.2,"+1,",.02)=$P("DENTAL CARE,DENTAL CROWNS,DENTAL ACCIDENT,ORTHODONTICS,PROSTHODONTICS,ORAL SURGERY,PREVENTIVE DENTAL",",",I-34)  .. S IBFDA(353.2,"+1,",.03)=$P("DENTAL CARE,DENTAL CROWNS,DENTAL ACCIDENT,ORTHODONTICS,PROSTHODONTICS,ORAL SURGERY,PREVENTIVE DENTAL",",",I-34)  . I $E(I)="E" D  .. S CT=$E(I,2,3)  .. I CT<18 S IBFDA(353.2,"+1,",.02)=$P("BASIC RESTORATIVE - DENTAL,MAJOR RESTORATIVE - DENTAL,FIXED PROSTHODONTICS,REMOVABLE PROSTHODONTICS,INTRAORAL IMAGES - COMPLETE SERIES,ORAL EVALUATION",",",CT-11)  .. I CT>17 S IBFDA(353.2,"+1,",.02)=$P("DENTAL PROPHYLAXIS,PANORAMIC IMAGES,SEALANTS,FLOURIDE TREATMENTS,DENTAL IMPLANTS,TEMPOROMANDIBULAR JOINT DYSFUNCTION",",",CT-17)  .. S IBFDA(353.2,"+1,",.03)=$P("BASIC RESTORATIVE,MAJOR RESTORATIVE,FIXED PROSTH,REMOVABLE PROSTH,IMAGES - COMPLETE,ORAL EVALUATION,PROPHYLAXIS,PANORAMIC IMAGES,SEALANTS,FLOURIDE,DENTAL IMPLANTS,JOINT DYSFUNCTION",",",CT-11)  . I I="F3" D  .. S IBFDA(353.2,"+1,",.02)="DENTAL COVERAGE"  .. S IBFDA(353.2,"+1,",.03)="DENTAL COVERAGE"  . I I="F7" D  .. S IBFDA(353.2,"+1,",.02)="ORTHODONTIA COVERAGE"  .. S IBFDA(353.2,"+1,",.03)="ORTHODONTIA COVERAGE"  . D UPDATE^DIE("","IBFDA","IBIEN","ERROR")  . I $D(ERROR) D MES^XPDUTL(">> IB ERROR - IB\*2.0\*592 Post Install - "\_$G(ERROR("DIERR",1,"TEXT",1))\_" <<")  . K IBIEN,ERROR  Q** | | | | | | | | | |