**TAS eBill SDD US3 v2.0**

System Design Document

IB\*2.0\*608



Department of Veterans Affairs

**March 2018**

Version 2.0

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Description | Author |
| 10/17 | 1.0 | Initial submittal prior to development | PII |
| 3/18 | 1.1 | Updated after completion of development | PII |
| 5/18 | 2.0 | Updated after completion of additional and final development | PII |

**User Story Number:** NO ID

**User Story Name:** CMN Oxygen and EPN Nutrition

**Product Backlog ID:** n/a

**Rally ID:** US-3

# Design/Assumption:

The design for this user story is going on the assumption that FILEMAN 22.2 (patch DI\*22.2\*3) will have been installed at all sites.

# Resolution Summary:

To resolve this request, the following items will need to be accomplished. Note that the user will be prompted for CMN data only if the claim is Professional (Institutional & Dental excluded):

1. Create section 21 in the IB System Parameters called “CMN CPT Code Inclusion” which contains a list of CPT codes requiring CMN prompting (routines IBJPS & IBJPS8). There is also a function in this routine to determine if a particular code is in the list.
2. Create a new Procedures sub-field 399.304,23 to store the response to the “CMN Required?” prompt (Yes or No). CMN prompting will only occur for procedures in the “CMN CPT Code Inclusion” list.
3. Create new file CMN FORM TYPES (399.6) that will be pointed to from the new CMN FORM TYPE field 399.304,24 (IV below). This new file will store an entry for each of the CMN Form Types. At this time there are only two CMN Form Types referenced in this SDD, namely, the CMS-484 Form for Oxygen, and the CMS-10126 form for Enteral and Parenteral Nutrition. Though this feature could have been handled by defining the new field in the 399.304 sub-file as a “SET”, there is a potential for up to 12 different Form Types so rather than possibly having to convert the new field to a pointer in the future, it was determined that a new file would be best to store the CMN Form Types.
4. Create a new Procedures sub-field CMN FORM TYPE (399.304,24) to store the CMN Form Type.
5. Create Procedures sub-fields (a.k.a. CMN fields) 24.01 thru 24.08 holding responses for questions which are the same for both forms, fields 24.1 thru 24.115 for questions specific to the CMN-484, and fields 24.201 thru 24.216 for questions specific to the CMN-10126. New nodes in ^DGCR will be created to hold the responses to the prompts for each of the new CMN forms: Node ‘CMN’ will hold responses to questions that are the same for both forms, node ‘CMN-484’ will hold responses specific to the CMN-484 form, and node ‘CMN-10126’ will hold responses specific to the CMN-10126 form.
6. Add entries to files 364.5 (IB DATA ELEMENT DEFINITION), 364.6 (IB FORM SKELETON DEFINITION) and 364.7 (IB FORM FIELD CONTENT) in order to include the new CMN fields in the 837 Transmission.
7. Add IB error messages for missing required CMN fields, fields missing a corresponding date, or CMN data irregularities (routines IBCBB1, IBCBB13). This involves adding 12 entries to file 350.8 (IB ERROR) to store the CMN data error messages.
8. Entries added to files 364.5, 364.6 and 364.7 done via the pre-install routine IBY608PR.
9. CMN CPT Code Inclusion list in IB System Parameters is populated via the post-install routine IBY608PO.
10. DR prompting used to collect the data for CMN fields 23 through 24.216 specified above (routine IBCU7 and IBCU75)
11. Develop Extract Code for to pull the values for the new CMN fields and related data (routine IBCEF31).
12. Add the CMN nodes to comparison code when rolling up procedures for the 837 transmission (routine IBCF23A)
13. Modify the cloning of a claim to include the new CMN fields (routine IBCCC2).
14. Modify the Interface Control Document (ICD) for the 837-P modifications by adding the CMN, FRM, LQ and MEA segments (separate document).

# Rules to be Applied during Design:

1. If user states no CMN is Required, then none of the subsequent CMN related prompts will appear. The system will automatically continue with the “Select CPT MODIFIER SEQUENCE:” prompt.
2. If the claim has a CMN then qualifier **UT** for CMN is required in 2440 LQ01 and the “CMN Form type” (399.0304,24) is Required in 2440 LQ02 Code Set.
3. “Certification Type” (399.0304,24.01) is Required.
4. “Patient Height (in)” (399.0304,24.02) is Not Required and needs modifier **TR** when populated. “Patient Weight (lbs)” (399.0304,24.03) field is Not Required, but when populated it needs modifier **01**.
5. The “Edema due to CHF Present?” (399.0304,24.104), “COR Pulmonale/Pulmonary Hypertension Present?” (399.0304,24.105) and “Hematocrit > 56%?” (399.0304,24.106) prompts should only appear if the “ABG PO2 (mmHg)” (399.0304,24.1) is between 56 and 59 or the “O2 Saturation (%)” (399.0304,24.102) value is equal to 89.
6. When “Certification Type” (399.0304,24.01) equals R or S, a date is required in “Recertification/Revision Date” (399.0304,24.07) and needs Qualifier **607**
7. “Date Therapy Started” (399.0304,24.05) is required and needs Qualifier **463**
8. “Last Certification Date” (399.0304,24.06) is required and needs Qualifier **461**
9. The only option for Units of Measurement for the length of time for equipment is months so the default for 2400 CR302 is equal to **MO** for Months.
10. The Supplier should be calculated to be the Billing Provider.
11. Oxygen flow rate can be a number or X if less than 1.
12. The “Latest 4 LPM ABG PO2 (mmHg)” (399.0304,24.111), “Date of Latest 4 LPM ABG Test” (399.0304,24.112), “Latest 4 LPM O2 Saturation (%)” (399.0304,24.113) and “Date of Latest 4 LPM O2 Saturation Test” (399.0304,24.114) fields should only have values when the “Highest O2 Flow Rate” (399.0304,24.11) field is greater than 4 LPM.
13. CRC Condition Indicator/Durable Medical Equipment should be populated with Replacement Item equal to **ZV**. For example: CRC\*09\*N\***ZV**

# Detailed Design:

**I) Create new Procedures sub-field 23 – “CMN REQUIRED?”:**

Create a new field **399.304,23** to store the response to the “**CMN Required?**” prompt. This is a **REQUIRED** field.

**399.0304,23 CMN REQUIRED?** CMN;1 SET (Required)

CMN Required?

'0' FOR NO;

'1' FOR YES;

LAST EDITED: NOV 15, 2017

HELP-PROMPT: Enter 'Yes' (1) if this procedure requires a Certificate of Medical Necessity, or 'No' (0) if it does not.

DESCRIPTION: This field indicates whether a Certificate of Medical Necessity must be submitted with this procedure.

**[NOTE: If CMN is NOT REQUIRED, then none of the subsequent CMN related prompts will appear. The system will automatically continue with the “Select CPT MODIFIER SEQUENCE:” prompt.]**

**II) Create new “CMN Form Types” file 399.6 containing 4 fields:**

STANDARD DATA DICTIONARY #399.6 -- CMN FORM TYPES FILE 5/23/18 PAGE 1

STORED IN ^IBE(399.6, (2 ENTRIES) SITE: TEST.CHEYENNE.MED.VA.GOV UCI: VISTA,ROU (VERSION 2.0)

DATA NAME GLOBAL DATA

ELEMENT TITLE LOCATION TYPE

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

The CMN FORM TYPES file was created to hold specifications for the various Certificate of Medical Necessity (CMN) form types and is

used in Enter/Edit Billing when the user specifies CMN information for an eligible procedure.

DD ACCESS: @

RD ACCESS: @

WR ACCESS: @

DEL ACCESS: @

LAYGO ACCESS: @

AUDIT ACCESS: @

POINTED TO BY: CMN FORM TYPE field (#24) of the PROCEDURES sub-field (#399.0304) of the BILL/CLAIMS File (#399)

CROSS

REFERENCED BY: NAME(B)

CREATED ON: JUN 2,2017 by PII LAST MODIFIED: MAY 22,2018@11:44:39

**399.6,.01 NAME** 0;1 FREE TEXT (Required)

INPUT TRANSFORM: K:$L(X)>50!($L(X)<3)!'(X'?1P.E) X

MAXIMUM LENGTH: 50

LAST EDITED: MAY 22, 2018

HELP-PROMPT: Enter the external name of the CMN form. Enter between 3 and 50 free-text characters.

DESCRIPTION: This is the external name of the Certificate of Medical Necessity (CMN) form. For example: 'OXYGEN (CMS-484)'

CROSS-REFERENCE: 399.6^B

1)= S ^IBE(399.6,"B",$E(X,1,30),DA)=""

2)= K ^IBE(399.6,"B",$E(X,1,30),DA)

**399.6,1 DESCRIPTION** 0;2 FREE TEXT

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

MAXIMUM LENGTH: 80

LAST EDITED: MAY 22, 2018

HELP-PROMPT: Enter a brief description (up to 80 characters) of the Form Type

DESCRIPTION: This is a free-text description of the CMN form type.

**399.6,2 INDUSTRY CODE** 0;3 FREE TEXT (Required)

INPUT TRANSFORM: K:$L(X)>15!($L(X)<1) X

MAXIMUM LENGTH: 15

LAST EDITED: MAY 22, 2018

HELP-PROMPT: Enter the Industry Code (up to 15 free-text characters) associated with this CMN form type.

DESCRIPTION: This is the industry code associated with this form. It is usually found in the upper-right corner of the printed form and is the number following 'DME'. For the 'E & P NUTRITION (CMS-10126)' form the industry code is 10.3 and for the 'OXYGEN (CMS-484)' it is 484.3.

**399.6,3 DATA NODE** 0;4 FREE TEXT (Required)

INPUT TRANSFORM: K:$L(X)>20!($L(X)<7) X

MAXIMUM LENGTH: 20

LAST EDITED: MAY 22, 2018

HELP-PROMPT: Enter the node in ^DGCR where data for this form is stored. Enter between 7 and 20 free-text characters.

DESCRIPTION: This is the node in ^DGCR where the CMN data for a particular form is stored. For the "484" form, the data node MUST be 'CMN-484' and for the "10126" form, the data node MUST be 'CMN-10126'.

**Create the 2 necessary D399.6 entries for the CMS-484 and CMS-10126 forms:**

**CMS-484-Oxygen**

NAME: **OXYGEN (CMS-484)**

DESCRIPTION: CERTIFICATE OF MEDICAL NECESSITY FORM 484.3 FOR OXYGEN

INDUSTRY CODE: 484.3

DATA NODE: CMN-484

**CMS-10126-Enteral & Parenteral Nutrition**

NAME: **ENTERAL & PARENTERAL NUTRITION (CMS-10126)**

DESCRIPTION: CERTIFICATE OF MEDICAL NECESSITY FORM 10126 FOR ENTERAL & PARENTERAL NUTRITION

INDUSTRY CODE: 10.03

DATA NODE: CMN-10126

**III) Create Procedures sub-field 24 – “CMN FORM TYPE”**

Create a new field **399.304,24** to store the **CMN Form Type**, which is a pointer to the new CMN FORM TYPES file (399.6). This is a **REQUIRED** field.

**399.0304,24 CMN FORM TYPE** CMN;2 POINTER TO CMN FORM TYPES FILE (#399.6)

CMN Form type

LAST EDITED: MAR 08, 2018

HELP-PROMPT: Select the REQUIRED CMN form type that will be sent with this procedure.

DESCRIPTION: This field indicates the Certificate of Medical Necessity form type that is to be submitted with this procedure.

TECHNICAL DESCR: If the CMN Required? field is set to "Y"es, this field must be an entry in the CMS FORM TYPES file #399.6.

**IV) Create Procedure sub-fields 24.01 thru 24.08, 24.1 thru 24.115, and 24.201 thru 24.219 to prompt the user for the new CMN form information.**

Creation of these Procedure sub-fields involves creating 3 new nodes in ^DGCR(399), namely nodes ‘CMN’, CMN-484’ and ‘CMN-10126’, which will contain the responses for BOTH forms, the CMN-484 form, and the CMN-10126 form, respectively.

**Fields 24.01 thru 24.09 are the same for both the CMN-484 and the CMN-10126:**

**399.0304,24.01** CMNCERTIFICATION TYPE **[REQUIRED]**

**399.0304,24.01 CMN CERTIFICATION TYPE** CMN;3 SET

Certification Type

'I' FOR INITIAL;

'R' FOR RENEWAL;

'S' FOR REVISED;

LAST EDITED: MAR 08, 2018

HELP-PROMPT: Select the REQUIRED Type of Certification requested.

DESCRIPTION: This field indicates the type of Certification that is being requested.

**[NOTE: When “CMN CERTIFICATION TYPE” equals ‘R’ or ‘S’, a date is required for the “CMN RECERTIFICATION/REVISION DATE” (399.0304,24.07)]**

**399.0304,24.02** CMN PATIENT HEIGHT (IN) **[Qualifier TR]**

**399.0304,24.02 CMN PATIENT HEIGHT (IN)** CMN;4 NUMBER

Patient Height (in)

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

LAST EDITED: APR 03, 2018

HELP-PROMPT: Enter the Patient's height in whole numbers representing inches.

DESCRIPTION: This field indicates the Patient's height in whole numbers representing inches.

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

**399.0304,24.03** CMN PATIENT WEIGHT (LBS)  **[Qualifier 01]**

**399.0304,24.03 CMN PATIENT WEIGHT (LBS)** CMN;5 NUMBER

Patient Weight (lbs)

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

LAST EDITED: MAR 02, 2018

HELP-PROMPT: Enter the Patient's weight in whole numbers representing pounds.

DESCRIPTION: This field indicates the Patient's weight in whole numbers representing pounds.

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

**399.0304,24.04** CMNMONTHS DME EQUIP NEEDED

**399.0304,24.04 CMN MONTHS DME EQUIP NEEDED** CMN;6 NUMBER

Months DME Equipment Needed

INPUT TRANSFORM: K:+X'=X!(X>99)!(X<1)!(X?.E1"."1N.N) X

LAST EDITED: NOV 14, 2017

HELP-PROMPT: Enter the number of MONTHS the patient will need the DME Equipment. Enter 1-99 with 99 equal to a lifetime.

DESCRIPTION: This field indicates the number of MONTHS that the Patient will need the DME Equipment. '99' represents a lifetime.

**399.0304,24.05** CMN DATE THERAPY STARTED **[REQUIRED; Qualifier is 463]**

**399.0304,24.05 CMN DATE THERAPY STARTED** CMN;7 DATE

Date Therapy Started

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

LAST EDITED: MAR 08, 2018

HELP-PROMPT: Enter the REQUIRED date the therapy began.

DESCRIPTION: This field indicates the date the therapy began.

**399.0304,24.06** CMN LAST CERTIFICATION DATE **[REQUIRED; Qualifier is 461]**

**399.0304,24.06 CMN LAST CERTIFICATION DATE** CMN;8 DATE

Last Certification Date

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

LAST EDITED: MAR 08, 2018

HELP-PROMPT: Enter the REQUIRED date the physician signed the Certificate of Medical Necessity.

DESCRIPTION: This field indicates the date the physician signed the Certificate of Medical Necessity.

**399.0304,24.07** CMN RECERTIFICATION/REVISN DT **[REQUIRED when “CMN CERTIFICATION TYPE” (399.0304,24.01) equals ‘R’ or ‘S’]**

**399.0304,24.07 CMN RECERTIFICATION/REVISN DT** CMN;9 DATE

Recertification/Revision Date

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

LAST EDITED: NOV 14, 2017

HELP-PROMPT: If the Certification Type is a Renewal or Revised, enter a REQUIRED Recertification/Revision date.

DESCRIPTION: If the Certification Type is a Renewal or Revised, this field is REQUIRED and indicates the date of the Recertification/Renewal.

**399.0304,24.08** CMN REPLACEMENT ITEM?

**399.0304,24.08 CMN REPLACEMENT ITEM?** CMN;10 SET

Replacement Item?

'0' FOR NO;

'1' FOR YES;

LAST EDITED: NOV 14, 2017

HELP-PROMPT: Enter 'Yes' (1) if this item is being billed as a replacement item, or 'No' (0) if it is not.

DESCRIPTION: This field indicates whether or not the item being billed is a Replacement item.

**Fields 24.1 thru 24.115 are specific to the CMN-484:**

**399.0304,24.1** CMNABG PO2 (MMHG)

**399.0304,24.1 CMN ABG PO2 (MMHG)** CMN-484;16 NUMBER

ABG PO2 (mmHg)

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

LAST EDITED: MAR 02, 2018

HELP-PROMPT: Enter the result of the most recent ABG test. Enter a whole Number which will be reported as mmHg.

DESCRIPTION: This field indicates the result of the most recent ABG test. The Number entered will be reported as mmHg.

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

**399.0304,24.102** CMN O2 SATURATION %

**399.0304,24.102 CMN O2 SATURATION %** CMN-484;2 NUMBER

O2 Saturation (%)

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

LAST EDITED: MAR 02, 2018

HELP-PROMPT: Enter the result of the most recent Oxygen saturation test. Enter a whole number which will be reported as %.

DESCRIPTION: This field indicates the result of the most recent Oxygen saturation test. The number entered will be reported as %.

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

**399.0304,24.103** CMN DT LAST ABG PO2 AND O2 SAT

**399.0304,24.103 CMN DT LAST ABG PO2 AND O2 SAT** CMN-484;3 DATE

Date of Last ABG PO2 and/or O2 Saturation Test(s)

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

LAST EDITED: MAR 14, 2018

HELP-PROMPT: Enter the REQUIRED date for the most recent ABG PO2 and/or O2 Saturation Test(s).

DESCRIPTION: This field indicates the Date for the most recent ABG PO2 and/or O2 Saturation test(s).

**399.0304,24.104** CMN EDEMA DUE TO CHF PRESENT?

**399.0304,24.104 CMN EDEMA DUE TO CHF PRESENT?** CMN-484;4 SET

Edema due to CHF Present?

'0' FOR NO;

'1' FOR YES;

LAST EDITED: NOV 14, 2017

HELP-PROMPT: Enter 'Yes' (1) if Edema being due to CHF being Present, or 'No' (0) if it is not.

DESCRIPTION: This field indicates whether or not the patient has dependent Edema due to Congestive Heart Failure.

**[NOTE: The “CMN EDEMA DUE TO CHF PRESENT?” prompt should only appear if the “CMN ABG PO2” (399.0304,24.1) value is between 56 and 59 or the “CMN O2 SATURATION %” (399.0304,24.102) value is equal to 89.]**

**399.0304,24.105** CMN COR PULMONARY HYPERTENSION?

**399.0304,24.105 CMN COR PULMONARY HYPERTENSN?** CMN-484;5 SET

COR Pulmonale/Pulmonary Hypertension Present?

'0' FOR NO;

'1' FOR YES;

LAST EDITED: NOV 14, 2017

HELP-PROMPT: Enter 'Yes' (1) if COR Pulmonale or Pulmonary Hypertension is Present, or 'No' (0) if it is not.

DESCRIPTION: This field indicates whether or not the patient has cor pulmonate or pulmonary hypertension documented by P pulmonale on an EKG or echocardiogram, gated blood pool scan or direct pulmonary artery pressure measurement.

**[NOTE: The “COR Pulmonary/Pulmonary Hypertension Present?” prompt should only appear if the “CMN ABG PO2 (MMHG)” (399.0304,24.1) value is between 56 and 59 or the “CMN O2 SATURATION %” (399.0304,24.102) value is equal to 89.]**

**399.0304,24.106** CMN HEMATOCRIT > 56%?

**399.0304,24.106 CMN HEMATOCRIT > 56%?** CMN-484;6 SET

Hematocrit > 56%?

'0' FOR NO;

'1' FOR YES;

LAST EDITED: NOV 14, 2017

HELP-PROMPT: Enter 'Yes' (1) if the patient has a Hematocrit level greater that 56% or 'No' (0) if not.

DESCRIPTION: This field indicates whether or not the patient has a Hematocrit level greater than 56%.

**[NOTE: The “Hematocrit > 56%?” prompt should only appear if the “CMN ABG PO2 (MMHG)” (399.0304,24.1) value is between 56 and 59 or the “CMN O2 SATURATION %” (399.0304,24.102) value is equal to 89.]**

**399.0304,24.107** CMN PT CONDITION AT TEST TIME

**399.0304,24.107 CMN PT CONDITION AT TEST TIME** CMN-484;7 SET

Patient Condition At Test Time

'1' FOR CHRONIC AND STABLE AS OUTPT;

'2' FOR W/I TWO DAYS PRIOR TO D/C FROM INPT FACILITY;

'3' FOR UNDER OTHER CIRCUMSTANCES;

LAST EDITED: NOV 14, 2017

HELP-PROMPT: Enter the patient's condition at the time of the ABG and/or O2 Saturation test(s).

DESCRIPTION: This field indicates the patient's condition at the time of the ABG and/or O2 Saturation test(s).

**399.0304,24.108** CMN TEST CONDITIONS

**399.0304,24.108 CMN TEST CONDITIONS** CMN-484;8 SET

Test Conditions

'1' FOR AT REST;

'2' FOR DURING EXERCISE;

'3' FOR DURING SLEEP;

LAST EDITED: NOV 14, 2017

HELP-PROMPT: Enter the conditions for the ABG and/or O2 Saturation test(s).

DESCRIPTION: This field indicates the condition for the ABG and/or O2 Saturation test(s).

**399.0304,24.109** CMN PORTABLE O2 INDICATOR

**399.0304,24.109** **CMN PORTABLE O2 INDICATOR** CMN-484;9 SET

Portable O2 Indicator

'Y' FOR PATIENT MOBILE WITHIN HOME;

'N' FOR PATIENT NOT MOBILE WITHIN HOME;

'D' FOR NOT ORDERING PORTABLE OXYGEN;

LAST EDITED: NOV 14, 2017

HELP-PROMPT: Enter the patient's mobility if ordering portable oxygen or indicate if not ordering portable oxygen.

DESCRIPTION: This field indicates the patient's mobility concerning the ordering of portable oxygen.

**399.0304,24.11** CMN HIGHEST O2 FLOW RATE

**399.0304,24.11 CMN HIGHEST O2 FLOW RATE CMN**-484;10 FREE TEXT

Highest O2 Flow Rate

INPUT TRANSFORM: K:$L(X)>50!($L(X)<1) X

MAXIMUM LENGTH: 50

LAST EDITED: MAR 02, 2018

HELP-PROMPT: Enter the highest oxygen flow rate ordered for this patient in liters per minute (LPM). Enter a number. If oxygen rate is less than 1 LPM, enter 'X'.

DESCRIPTION: This field indicates the highest oxygen flow rate ordered for this Patient in liters per minute (LPM). The value is either a number, or if the value is less than 1 LPM, it should be entered as an "X".

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

**399.0304,24.111** CMN LAST 4 LPM ABG PO2 (MMHG)

**399.0304,24.111 CMN LAST 4 LPM ABG PO2 (MMHG)** CMN-484;11 NUMBER

Latest 4 LPM ABG PO2 (mmHg)

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

LAST EDITED: MAR 02, 2018

HELP-PROMPT: Enter the result of the most recent ABG test taken on 4 LPM. Enter a whole number which will be reported as mmHg.

DESCRIPTION: This field indicates the result of the most recent ABG test taken on 4 LPM. The number entered will be reported as mmHg.

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

**399.0304,24.113** CMN LAST 4 LPM O2 SATURATION %

**399.0304,24.113 CMN LAST 4 LPM O2 SATURATION %** CMN-484;13 NUMBER

Latest 4 LPM O2 Saturation (%)

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

LAST EDITED: MAR 02, 2018

HELP-PROMPT: Enter the result of the most recent Oxygen saturation test. Enter a whole number which will be reported as %.

DESCRIPTION: This field indicates the result of the most recent Oxygen saturation test. The number entered will be reported as %.

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

**399.0304,24.114** CMN DATE OF LAST 4 LPM TESTS

**399.0304,24.114 CMN DATE OF LAST 4 LPM TESTS** CMN-484;14 DATE

Date of the Latest 4 LPM Test(s)

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

LAST EDITED: MAR 14, 2018

HELP-PROMPT: Enter the REQUIRED date for the most recent 4 LPM Test(s).

DESCRIPTION: This field indicates the Date for the most recent ABG PO2 and/or O2 Saturation test(s) taken on 4 LPM.

**399.0304,24.115** CMN EQUIPMENT/COST DESCRIPTION

**399.0304,24.115** **CMN EQUIPMENT/COST DESCRIPTION** CMN-484;15 FREE TEXT

Equipment/Cost Description

INPUT TRANSFORM: K:$L(X)>50!($L(X)<1) X

MAXIMUM LENGTH: 50

LAST EDITED: NOV 14, 2017

HELP-PROMPT: Enter a 1-50 character free text description of items, accessories, and options ordered, suppliers charge and Medicare Fee Schedule allowance for each item, accessory and option.

DESCRIPTION: This field indicates the description of the items, accessories, and options ordered, suppliers charge and Medicare Fee Schedule Allowance for each item, accessory and option.

**Fields 24.201 thru 24.219 are specific to the CMN-10126**

**399.0304,24.201** CMN SM BOWEL ABSORPTION DOC?

**399.0304,24.201 CMN SM BOWEL ABSORPTION DOC?** CMN-10126;1 SET

Small Bowel Absorption Documentation Present?

'0' FOR NO;

'1' FOR YES;

LAST EDITED: NOV 14, 2017

HELP-PROMPT: Enter 'Yes' (1) if there is documentation on file for Small Bowel Absorption, or 'No' (0) if there is not.

DESCRIPTION: This field indicates whether or not there is documentation in the medical record that supports the patient's permanent non-function or disease of the structures that permit food to reach or be absorbed from the small bowel.

**399.0304,24.202** CMN ENTERAL NUTRITION BY TUBE?

**399.0304,24.202 CMN ENTERAL NUTRITION BY TUBE?** CMN-10126;2 SET

Enteral Nutrition by Tube?

'0' FOR NO;

'1' FOR YES;

LAST EDITED: NOV 21, 2017

HELP-PROMPT: Enter 'Yes' (1) if the Enteral Nutrition is being administered by a tube, or 'No' (0) if it is not.

DESCRIPTION: This field indicates whether or not the Enteral Nutrition is being administered via a tube (Example: gastrostomy tube).

**399.0304,24.203** CMN PROCEDURE A CALORIES

**399.0304,24.203 CMN PROCEDURE A CALORIES** CMN-10126;3 NUMBER

Procedure A Calories

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

LAST EDITED: APR 20, 2018

HELP-PROMPT: Enter the calories per day associated with Procedure A.

DESCRIPTION:

This field indicates the calories per day associated with Procedure A.

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

**399.0304,24.204** CMN PROCEDURE A

**399.0304,24.204 CMN PROCEDURE A** CMN-10126;4 POINTER TO CPT FILE (#81)

Procedure A

LAST EDITED: APR 20, 2018

HELP-PROMPT: Enter first procedure with associated calories.

DESCRIPTION: This is the procedure code to which the "Procedure A Calories" field corresponds.

**399.0304,24.205** CMNMETHOD OF ADMINISTRATION

**399.0304,24.205 CMN METHOD OF ADMINISTRATION** CMN-10126;5 SET

Method of Administration

'1' FOR SYRINGE;

'2' FOR GRAVITY;

'3' FOR PUMP;

'4' FOR ORAL;

LAST EDITED: NOV 15, 2017

HELP-PROMPT: Select the appropriate method by which the service was administered.

DESCRIPTION: This field indicates the method by which the service was administered.

**399.030424,24.206** CMN DAYS PER WEEK ADMINISTERED

**399.0304,24.206 CMN DAYS PER WEEK ADMINISTERED** CMN-10126;6 NUMBER

Days/Week Administered

INPUT TRANSFORM: K:+X'=X!(X>7)!(X<1)!(X?.E1"."1N.N) X

LAST EDITED: NOV 14, 2017

HELP-PROMPT: Enter the number of days per week that the nutrition is administered or infused.

DESCRIPTION: This field indicates the number of days per week that the nutrition is administered or infused.

**399.030424,24.207** CMN SEVERE MALABSORPTION DOC?

**399.0304,24.207** **CMN SEVERE MALABSORPTION DOC?** CMN-10126;7 SET

Severe Malabsorption Documentation Present?

'0' FOR NO;

'1' FOR YES;

LAST EDITED: NOV 14, 2017

HELP-PROMPT: Enter 'Yes' (1) if there is documentation on file for Severe Malabsorption, or 'No' (0) if there is not.

DESCRIPTION: This field indicates whether or not there is documentation in the medical record that supports the patient having permanent disease of the gastrointestinal tract causing malabsorption severe enough to prevent maintenance of weight and strength.

**399.030424,24.208** CMN AMINO ACID (ML/DAY)

**399.0304,24.208 CMN AMINO ACID (ML/DAY)** CMN-10126;8 NUMBER

Amino Acid (ml/day)

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

LAST EDITED: MAR 02, 2018

HELP-PROMPT: Enter the number of milliliters of the component Amino Acid that are administered per day in this nutritional formula.

DESCRIPTION: This field indicates the number of milliliters of the component Amino Acid that are administered per day in this nutritional formula.

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

**399.030424,24.209** CMN AMINO ACID CONCENTRATION %

**399.0304,24.209 CMN AMINO ACID CONCENTRATION %** CMN-10126;9 NUMBER

Amino Acid Concentration (%)

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

LAST EDITED: MAR 02, 2018

HELP-PROMPT: Enter the percent concentration of Amino Acids in this nutritional formula.

DESCRIPTION: This field indicates the percent concentration of Amino Acids in this nutritional formula.

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

**399.030424,24.21** CMN AMINO ACID PROTEIN (GM/DY)

**399.0304,24.21 CMN AMINO ACID PROTEIN (GM/DY)** CMN-10126;10 NUMBER

Amino Acid Protein (gm/day)

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

LAST EDITED: MAR 02, 2018

HELP-PROMPT: Enter the amount of protein administered in grams/day in this nutritional formula.

DESCRIPTION: This field indicates the amount of protein administered in grams/day in this nutritional formula.

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

**399.030424,24.211** CMN DEXTROSE (ML/DAY)

**399.0304,24.211 CMN DEXTROSE (ML/DAY)** CMN-10126;11 NUMBER

Dextrose (ml/day)

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

LAST EDITED: MAR 02, 2018

HELP-PROMPT: Enter the number of milliliters of the component Dextrose that are administered per day in this nutritional formula.

DESCRIPTION: This field indicates the number of milliliters of the component Dextrose that are administered per day in this nutritional formula.

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

**399.030424,24.212** CMN DEXTROSE CONCENTRATE %

**399.0304,24.212 CMN DEXTROSE CONCENTRATE %** CMN-10126;12 NUMBER

Dextrose Concentrate (%)

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

LAST EDITED: MAR 02, 2018

HELP-PROMPT: Enter the percent concentration of Dextrose in this nutritional formula.

DESCRIPTION: This field indicates the percent concentration of Dextrose in this nutritional formula.

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

**399.030424,24.213** CMN LIPIDS (ML/DAY)

**399.0304,24.213 CMN LIPIDS (ML/DAY)** CMN-10126;13 NUMBER

Lipids (ml/day)

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

LAST EDITED: MAR 02, 2018

HELP-PROMPT: Enter the number of milliliters of the component Lipids that are administered per day in this nutritional formula.

DESCRIPTION: This field indicates the number of milliliters of the component Lipids that are administered per day in this nutritional formula.

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

**399.030424,24.214** ROUTE OF ADMINISTRATION

**399.0304,24.214 CMN ROUTE OF ADMINISTRATION** CMN-10126;14 SET

Route of Administration

'1' FOR CENTRAL LINE (INCLUDES PICC);

'2' FOR HEMODIALYSIS ACCESS LINE;

'3' FOR PERITONEAL CATHETER;

LAST EDITED: NOV 14, 2017

HELP-PROMPT: Enter the number that represents the appropriate route by which the nutrition was administered.

DESCRIPTION: This field indicates the route by which the nutrition was administered.

**399.030424,24.215** CMN LIPIDS (DAYS/WEEK)

**399.0304,24.215 CMN LIPIDS (DAYS/WEEK)** CMN-10126;15 NUMBER

Lipids (days/wk)

INPUT TRANSFORM: K:+X'=X!(X>7)!(X<1)!(X?.E1"."1N.N) X

LAST EDITED: NOV 14, 2017

HELP-PROMPT: Enter the number of days per week the component lipids are administered in this nutritional formula.

DESCRIPTION: This field indicates the number of days per week the component Lipids are administered in this nutritional formula.

**399.030424,24.216** CMN LIPIDS CONCENTRATE %

**399.0304,24.216 CMN LIPIDS CONCENTRATE %** CMN-10126;16 NUMBER

Lipids Concentrate (%)

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

LAST EDITED: MAR 02, 2018

HELP-PROMPT: Enter the percent concentration of Lipids in this nutritional formula.

DESCRIPTION: This field indicates the percent concentration of Lipids in this nutritional formula.

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

**399.030424,24.217** CMN PARENTERAL/ENTERAL/BOTH

**399.0304,24.217 CMN PARENTERAL/ENTERAL/BOTH** CMN-10126;17 SET

Is this for Parenteral nutrition, Enteral nutrition, or Both?

'P' FOR PARENTERAL;

'E' FOR ENTERAL;

'B' FOR BOTH;

LAST EDITED: APR 23, 2018

HELP-PROMPT: Is this CMN for Parenteral nutrition, enteral nutrition, or both?

DESCRIPTION: This field designates whether this CMN form is for Parenteral nutrition, enteral nutrition, or both.

**399.030424,24.218** CMN PROCEDURE B CALORIES

**399.0304,24.218 CMN PROCEDURE B CALORIES** CMN-10126;18 NUMBER

Procedure B Calories

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

LAST EDITED: APR 23, 2018

HELP-PROMPT: Enter the calories per day associated with Procedure B.

DESCRIPTION: This field indicates the calories per day associated with Procedure B.

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

**399.030424,24.219** CMN PROCEDURE B

**399.0304,24.219 CMN PROCEDURE B** CMN-10126;19 POINTER TO CPT FILE (#81)

Procedure B

LAST EDITED: APR 20, 2018

HELP-PROMPT: Enter second procedure with associated calories.

DESCRIPTION: This is the procedure code to which the "Procedure B Calories" field corresponds.

**V) Create new Entries in Files 364.5, 364.6 and 364.7 for the 837 Transmission of CMN Data via new segments CMN, FRM, LQ and MEA and 2 new pieces added to the existing PT1 segment.**

1. **File 364.5 Entries:**

**N-CMN RECORD ID ‘CMN ‘**

NAME: N-CMN RECORD ID 'CMN '

SECURITY LEVEL: NATIONAL,NO EDIT

TYPE OF ELEMENT: EXTRACTED VIA CODE

ELEMENT CATEGORY: INDIVIDUAL ELEMENT

**N-CMN RECORD ID ‘FRM ‘**

NAME: N-CMN RECORD ID 'FRM '

SECURITY LEVEL: NATIONAL,NO EDIT

TYPE OF ELEMENT: EXTRACTED VIA CODE

ELEMENT CATEGORY: INDIVIDUAL ELEMENT

**N-CMN RECORD ID ‘LQ ‘**

NAME: N-CMN RECORD ID 'LQ '

SECURITY LEVEL: NATIONAL,NO EDIT

TYPE OF ELEMENT: EXTRACTED VIA CODE

ELEMENT CATEGORY: INDIVIDUAL ELEMENT

**N-CMN RECORD ID ‘MEA ‘**

NAME: N-CMN RECORD ID 'MEA '

SECURITY LEVEL: NATIONAL,NO EDIT

TYPE OF ELEMENT: EXTRACTED VIA CODE

ELEMENT CATEGORY: INDIVIDUAL ELEMENT

1. **File 364.6 Entries**

**CMN Segment**

**CMN RECORD ID 'CMN '**

BILL FORM: IB 837 TRANSMISSION SECURITY LEVEL: NATIONAL,NO EDIT

PAGE OR SEQUENCE: 191.4 FIRST LINE NUMBER: 1

STARTING COLUMN OR PIECE: 1 LENGTH: 4

SHORT DESCRIPTION: **CMN RECORD ID 'CMN '**

CALCULATE ONLY OR OUTPUT: OUTPUT

**CMN DATA EXTRACT**

BILL FORM: IB 837 TRANSMISSION SECURITY LEVEL: NATIONAL,NO EDIT

PAGE OR SEQUENCE: 191.4 FIRST LINE NUMBER: 1

STARTING COLUMN OR PIECE: 1.5 LENGTH: 1

SHORT DESCRIPTION: **CMN DATA EXTRACT**

CALCULATE ONLY OR OUTPUT: CALCULATE ONLY

TRANSMIT IGNORES IF NULL: TRUE

**SERVICE LINE #**

BILL FORM: IB 837 TRANSMISSION SECURITY LEVEL: NATIONAL,NO EDIT

PAGE OR SEQUENCE: 191.4 FIRST LINE NUMBER: 1

STARTING COLUMN OR PIECE: 2 LENGTH: 6

SHORT DESCRIPTION: **SERVICE LINE #**

CALCULATE ONLY OR OUTPUT: OUTPUT

**CMN CERTIFICATION TYPE**

BILL FORM: IB 837 TRANSMISSION SECURITY LEVEL: NATIONAL,NO EDIT

PAGE OR SEQUENCE: 191.4 FIRST LINE NUMBER: 1

STARTING COLUMN OR PIECE: 3 LENGTH: 1

SHORT DESCRIPTION: **CMN CERTIFICATION TYPE**

CALCULATE ONLY OR OUTPUT: OUTPUT

DATA REQUIRED FOR FIELD: NO

**CMN UNIT OR BASIS FOR MEASUREMENT CODE**

BILL FORM: IB 837 TRANSMISSION SECURITY LEVEL: NATIONAL,NO EDIT

PAGE OR SEQUENCE: 191.4 FIRST LINE NUMBER: 1

STARTING COLUMN OR PIECE: 4 LENGTH: 2

SHORT DESCRIPTION: **CMN UNIT OR BASIS FOR MEASUREMENT CODE**

CALCULATE ONLY OR OUTPUT: OUTPUT

**CMN MONTHS DME EQUIPMENT NEEDED**

BILL FORM: IB 837 TRANSMISSION SECURITY LEVEL: NATIONAL,NO EDIT

PAGE OR SEQUENCE: 191.4 FIRST LINE NUMBER: 1

STARTING COLUMN OR PIECE: 5 LENGTH: 2

SHORT DESCRIPTION: **CMN MONTHS DME EQUIPMENT NEEDED**

CALCULATE ONLY OR OUTPUT: OUTPUT

**CMN CODE CATEGORY**

BILL FORM: IB 837 TRANSMISSION SECURITY LEVEL: NATIONAL,NO EDIT

PAGE OR SEQUENCE: 191.4 FIRST LINE NUMBER: 1

STARTING COLUMN OR PIECE: 6 LENGTH: 2

SHORT DESCRIPTION: **CMN CODE CATEGORY**

CALCULATE ONLY OR OUTPUT: OUTPUT

**CMN CERTIFICATION CONDITION INDICATOR**

BILL FORM: IB 837 TRANSMISSION SECURITY LEVEL: NATIONAL,NO EDIT

PAGE OR SEQUENCE: 191.4 FIRST LINE NUMBER: 1

STARTING COLUMN OR PIECE: 7 LENGTH: 1

SHORT DESCRIPTION: **CMN CERTIFICATION CONDITION INDICATOR**

CALCULATE ONLY OR OUTPUT: OUTPUT

**CMN CONDITION INDICATOR**

BILL FORM: IB 837 TRANSMISSION SECURITY LEVEL: NATIONAL,NO EDIT

PAGE OR SEQUENCE: 191.4 FIRST LINE NUMBER: 1

STARTING COLUMN OR PIECE: 8 LENGTH: 3

SHORT DESCRIPTION: **CMN CONDITION INDICATOR**

CALCULATE ONLY OR OUTPUT: OUTPUT

**CMN REPLACEMENT ITEM?**

BILL FORM: IB 837 TRANSMISSION SECURITY LEVEL: NATIONAL,NO EDIT

PAGE OR SEQUENCE: 191.4 FIRST LINE NUMBER: 1

STARTING COLUMN OR PIECE: 9 LENGTH: 3

SHORT DESCRIPTION: **CMN REPLACEMENT ITEM?**

CALCULATE ONLY OR OUTPUT: OUTPUT

**CMN DATE THERAPY STARTED QUALIFIER**

BILL FORM: IB 837 TRANSMISSION SECURITY LEVEL: NATIONAL,NO EDIT

PAGE OR SEQUENCE: 191.4 FIRST LINE NUMBER: 1

STARTING COLUMN OR PIECE: 10 LENGTH: 3

SHORT DESCRIPTION: **CMN DATE THERAPY STARTED QUALIFIER**

CALCULATE ONLY OR OUTPUT: OUTPUT

**CMN DATE THERAPY STARTED**

BILL FORM: IB 837 TRANSMISSION SECURITY LEVEL: NATIONAL,NO EDIT

PAGE OR SEQUENCE: 191.4 FIRST LINE NUMBER: 1

STARTING COLUMN OR PIECE: 11 LENGTH: 8

SHORT DESCRIPTION: **CMN DATE THERAPY STARTED**

CALCULATE ONLY OR OUTPUT: OUTPUT

**CMN LAST CERTIFICATION DATE QUALIFIER**

BILL FORM: IB 837 TRANSMISSION SECURITY LEVEL: NATIONAL,NO EDIT

PAGE OR SEQUENCE: 191.4 FIRST LINE NUMBER: 1

STARTING COLUMN OR PIECE: 12 LENGTH: 3

SHORT DESCRIPTION: **CMN LAST CERTIFICATION DATE QUALIFIER**

CALCULATE ONLY OR OUTPUT: OUTPUT

**CMN LAST CERTIFICATION DATE**

BILL FORM: IB 837 TRANSMISSION SECURITY LEVEL: NATIONAL,NO EDIT

PAGE OR SEQUENCE: 191.4 FIRST LINE NUMBER: 1

STARTING COLUMN OR PIECE: 13 LENGTH: 8

SHORT DESCRIPTION: **CMN LAST CERTIFICATION DATE**

CALCULATE ONLY OR OUTPUT: OUTPUT

**CMN CERTIFICATION TYPE QUAL**

BILL FORM: IB 837 TRANSMISSION SECURITY LEVEL: NATIONAL,NO EDIT

PAGE OR SEQUENCE: 191.4 FIRST LINE NUMBER: 1

STARTING COLUMN OR PIECE: 14 LENGTH: 3

SHORT DESCRIPTION: **CMN CERTIFICATION TYPE QUAL**

CALCULATE ONLY OR OUTPUT: OUTPUT

**CMN RECERTIFICATION/REVISION DATE**

BILL FORM: IB 837 TRANSMISSION SECURITY LEVEL: NATIONAL,NO EDIT

PAGE OR SEQUENCE: 191.4 FIRST LINE NUMBER: 1

STARTING COLUMN OR PIECE: 15 LENGTH: 8

SHORT DESCRIPTION: **CMN RECERTIFICATION/REVISION DATE**

CALCULATE ONLY OR OUTPUT: OUTPUT

**CMN ATTACHMENT REPORT TYPE CODE**

BILL FORM: IB 837 TRANSMISSION SECURITY LEVEL: NATIONAL,NO EDIT

PAGE OR SEQUENCE: 191.4 FIRST LINE NUMBER: 1

STARTING COLUMN OR PIECE: 16 LENGTH: 2

SHORT DESCRIPTION: **CMN ATTACHMENT REPORT TYPE CODE**

CALCULATE ONLY OR OUTPUT: OUTPUT

**CMN ATTACHMENT TRANSMISSION CODE**

BILL FORM: IB 837 TRANSMISSION SECURITY LEVEL: NATIONAL,NO EDIT

PAGE OR SEQUENCE: 191.4 FIRST LINE NUMBER: 1

STARTING COLUMN OR PIECE: 17 LENGTH: 2

SHORT DESCRIPTION: **CMN ATTACHMENT TRANSMISSION CODE**

CALCULATE ONLY OR OUTPUT: OUTPUT

**FRM Segment**

**CMN RECORD ID 'FRM '**

BILL FORM: IB 837 TRANSMISSION SECURITY LEVEL: NATIONAL,NO EDIT

PAGE OR SEQUENCE: 210 FIRST LINE NUMBER: 1

STARTING COLUMN OR PIECE: 1 LENGTH: 4

SHORT DESCRIPTION: **CMN RECORD ID 'FRM '**

CALCULATE ONLY OR OUTPUT: OUTPUT TRANSMIT IGNORES IF NULL: FALSE

DATA REQUIRED FOR FIELD: YES

**FRM DATA EXTRACT**

BILL FORM: IB 837 TRANSMISSION PAGE OR SEQUENCE: 210

FIRST LINE NUMBER: 1 LOCAL OVERRIDE ALLOWED: NO

STARTING COLUMN OR PIECE: 1.5 LENGTH: 1

SHORT DESCRIPTION: **FRM DATA EXTRACT**

CALCULATE ONLY OR OUTPUT: CALCULATE ONLY

TRANSMIT IGNORES IF NULL: TRUE

**SERVICE LINE #**

BILL FORM: IB 837 TRANSMISSION SECURITY LEVEL: NATIONAL,NO EDIT

PAGE OR SEQUENCE: 210 FIRST LINE NUMBER: 1

STARTING COLUMN OR PIECE: 2 LENGTH: 6

SHORT DESCRIPTION: **SERVICE LINE #**

CALCULATE ONLY OR OUTPUT: OUTPUT

**CMN QUESTION NUMBER/LETTER**

BILL FORM: IB 837 TRANSMISSION PAGE OR SEQUENCE: 210

FIRST LINE NUMBER: 1 STARTING COLUMN OR PIECE: 3

LENGTH: 20

SHORT DESCRIPTION: **CMN QUESTION NUMBER/LETTER**

CALCULATE ONLY OR OUTPUT: OUTPUT

TRANSMIT IGNORES IF NULL: FALSE

**CMN QUESTION RESPONSE Y/N**

BILL FORM: IB 837 TRANSMISSION SECURITY LEVEL: NATIONAL,NO EDIT

PAGE OR SEQUENCE: 210 FIRST LINE NUMBER: 1

STARTING COLUMN OR PIECE: 4 LENGTH: 1

SHORT DESCRIPTION: **CMN QUESTION RESPONSE Y/N**

CALCULATE ONLY OR OUTPUT: OUTPUT

**CMN QUESTION RESPONSE REF ID**

BILL FORM: IB 837 TRANSMISSION SECURITY LEVEL: NATIONAL,NO EDIT

PAGE OR SEQUENCE: 210 FIRST LINE NUMBER: 1

STARTING COLUMN OR PIECE: 5 LENGTH: 50

SHORT DESCRIPTION: **CMN QUESTION RESPONSE REF ID**

CALCULATE ONLY OR OUTPUT: OUTPUT

**CMN QUESTION RESPONSE DATE**

BILL FORM: IB 837 TRANSMISSION SECURITY LEVEL: NATIONAL,NO EDIT

PAGE OR SEQUENCE: 210 FIRST LINE NUMBER: 1

STARTING COLUMN OR PIECE: 6 LENGTH: 8

SHORT DESCRIPTION: **CMN QUESTION RESPONSE DATE**

CALCULATE ONLY OR OUTPUT: OUTPUT

**CMN QUESTION RESPONSE % & DECIMAL**

BILL FORM: IB 837 TRANSMISSION SECURITY LEVEL: NATIONAL,NO EDIT

PAGE OR SEQUENCE: 210 FIRST LINE NUMBER: 1

STARTING COLUMN OR PIECE: 7 LENGTH: 6

SHORT DESCRIPTION: **CMN QUESTION RESPONSE % & DECIMAL**

CALCULATE ONLY OR OUTPUT: OUTPUT

**LQ Segment**

**CMN RECORD ID 'LQ '**

BILL FORM: IB 837 TRANSMISSION SECURITY LEVEL: NATIONAL,NO EDIT

PAGE OR SEQUENCE: 205 FIRST LINE NUMBER: 1

STARTING COLUMN OR PIECE: 1 LENGTH: 4

SHORT DESCRIPTION: **CMN RECORD ID 'LQ '**

CALCULATE ONLY OR OUTPUT: OUTPUT TRANSMIT IGNORES IF NULL: FALSE

DATA REQUIRED FOR FIELD: YES

**LQ DATA EXTRACT**

BILL FORM: IB 837 TRANSMISSION PAGE OR SEQUENCE: 205

FIRST LINE NUMBER: 1 LOCAL OVERRIDE ALLOWED: NO

STARTING COLUMN OR PIECE: 1.5 LENGTH: 1

SHORT DESCRIPTION: **LQ DATA EXTRACT**

CALCULATE ONLY OR OUTPUT: CALCULATE ONLY

TRANSMIT IGNORES IF NULL: TRUE

**SERVICE LINE #**

BILL FORM: IB 837 TRANSMISSION SECURITY LEVEL: NATIONAL,NO EDIT

PAGE OR SEQUENCE: 205 FIRST LINE NUMBER: 1

STARTING COLUMN OR PIECE: 2 LENGTH: 6

SHORT DESCRIPTION: **SERVICE LINE #**

CALCULATE ONLY OR OUTPUT: OUTPUT

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

**CMN FORM TYPE QUALIFIER**

BILL FORM: IB 837 TRANSMISSION SECURITY LEVEL: NATIONAL,NO EDIT

PAGE OR SEQUENCE: 205 FIRST LINE NUMBER: 1

STARTING COLUMN OR PIECE: 3 LENGTH: 30

SHORT DESCRIPTION: **CMN FORM TYPE QUALIFIER**

CALCULATE ONLY OR OUTPUT: OUTPUT TRANSMIT IGNORES IF NULL: FALSE

DATA REQUIRED FOR FIELD: NO

**CMN INDUSTRY CODE**

BILL FORM: IB 837 TRANSMISSION SECURITY LEVEL: NATIONAL,NO EDIT

PAGE OR SEQUENCE: 205 FIRST LINE NUMBER: 1

STARTING COLUMN OR PIECE: 4 LENGTH: 10

SHORT DESCRIPTION: **CMN INDUSTRY CODE**

CALCULATE ONLY OR OUTPUT: OUTPUT

TRANSMIT IGNORES IF NULL: FALSE DATA REQUIRED FOR FIELD: NO

**MEA Segment**

**CMN RECORD ID 'MEA '**

BILL FORM: IB 837 TRANSMISSION SECURITY LEVEL: NATIONAL,NO EDIT

PAGE OR SEQUENCE: 191.7 FIRST LINE NUMBER: 1

STARTING COLUMN OR PIECE: 1 LENGTH: 4

SHORT DESCRIPTION: **CMN RECORD ID 'MEA '**

CALCULATE ONLY OR OUTPUT: OUTPUT

**MEA DATA EXTRACT**

BILL FORM: IB 837 TRANSMISSION

SECURITY LEVEL: NATIONAL,NO EDIT

PAGE OR SEQUENCE: 191.7 FIRST LINE NUMBER: 1

STARTING COLUMN OR PIECE: 1.5 LENGTH: 1

SHORT DESCRIPTION: **MEA DATA EXTRACT**

CALCULATE ONLY OR OUTPUT: CALCULATE ONLY

**SERVICE LINE #**

BILL FORM: IB 837 TRANSMISSION SECURITY LEVEL: NATIONAL,NO EDIT

PAGE OR SEQUENCE: 191.7 FIRST LINE NUMBER: 1

STARTING COLUMN OR PIECE: 2 LENGTH: 6

SHORT DESCRIPTION: **SERVICE LINE #**

CALCULATE ONLY OR OUTPUT: OUTPUT

**CMN MEASUREMENT REFERENCE ID CODE (Patient Height Modifier)**

BILL FORM: IB 837 TRANSMISSION SECURITY LEVEL: NATIONAL,NO EDIT

PAGE OR SEQUENCE: 191.7 FIRST LINE NUMBER: 1

STARTING COLUMN OR PIECE: 3 LENGTH: 2

SHORT DESCRIPTION: **CMN MEASUREMENT REFERENCE ID CODE**

CALCULATE ONLY OR OUTPUT: OUTPUT

**CMN MEASUREMENT QUALIFIER**

BILL FORM: IB 837 TRANSMISSION SECURITY LEVEL: NATIONAL,NO EDIT

PAGE OR SEQUENCE: 191.7 FIRST LINE NUMBER: 1

STARTING COLUMN OR PIECE: 4 LENGTH: 3

SHORT DESCRIPTION: **CMN MEASUREMENT QUALIFIER**

CALCULATE ONLY OR OUTPUT: OUTPUT

**CMN TEST RESULTS (Patient Height)**

BILL FORM: IB 837 TRANSMISSION SECURITY LEVEL: NATIONAL,NO EDIT

PAGE OR SEQUENCE: 191.7 FIRST LINE NUMBER: 1

STARTING COLUMN OR PIECE: 5 LENGTH: 20

SHORT DESCRIPTION: **CMN TEST RESULTS**

CALCULATE ONLY OR OUTPUT: OUTPUT

**PT1 Segment**

**CMN PATIENT WEIGHT MODIFIER**

BILL FORM: IB 837 TRANSMISSION SECURITY LEVEL: NATIONAL,NO EDIT

PAGE OR SEQUENCE: 40 FIRST LINE NUMBER: 1

STARTING COLUMN OR PIECE: 14 LENGTH: 2

SHORT DESCRIPTION: **CMN PATIENT WEIGHT MODIFIER**

CALCULATE ONLY OR OUTPUT: OUTPUT

**CMN PATIENT WEIGHT (LBS)**

BILL FORM: IB 837 TRANSMISSION SECURITY LEVEL: NATIONAL,NO EDIT

PAGE OR SEQUENCE: 40 FIRST LINE NUMBER: 1

STARTING COLUMN OR PIECE: 15 LENGTH: 4

SHORT DESCRIPTION: **CMN PATIENT WEIGHT (LBS)**

CALCULATE ONLY OR OUTPUT: OUTPUT

1. **File 364.7 Entries**

**CMN Segment**

**CMN RECORD ID 'CMN '**

FORM FIELD REFERENCE: IB 837 TRANSMISSION

SECURITY LEVEL: NATIONAL,NO EDIT

DATA ELEMENT: N-CMN RECORD ID 'CMN '

PAD CHARACTER: NO PAD REQUIRED

FORMAT CODE: K IBXDATA S IBXDATA="CMN "

**CMN DATA EXTRACT**

FORM FIELD REFERENCE: IB 837 TRANSMISSION

SECURITY LEVEL: NATIONAL,NO EDIT

DATA ELEMENT: N-GET FROM PREVIOUS EXTRACT

FORMAT CODE: D:$D(IBXSAVE("CMNDEX"))'>1 CMNDEX^IBCEF31(IBXIEN,.IBXSAVE)

**SERVICE LINE #**

FORM FIELD REFERENCE: IB 837 TRANSMISSION

SECURITY LEVEL: NATIONAL,NO EDIT

DATA ELEMENT: N-GET FROM PREVIOUS EXTRACT

FORMAT CODE: D:$D(IBXSAVE("CMNDEX"))'>1 CMNDEX^IBCEF31(IBXIEN,.IBXSAVE)

**CMN CERTIFICATION TYPE**

FORM FIELD REFERENCE: IB 837 TRANSMISSION

SECURITY LEVEL: NATIONAL,NO EDIT

DATA ELEMENT: N-GET FROM PREVIOUS EXTRACT

PAD CHARACTER: NO PAD REQUIRED REQUIRED: NO

EDIT STATUS: EDITABLE

FORMAT CODE: N Z N Z K IBXDATA S Z=0 F S Z=$O(IBXSAVE("CMNDEX",Z)) Q:'Z S IBXDATA(Z)=$$CMNDATA^IBCEF31(IBXIEN,+IBXSAVE("CMNDEX",Z),24.01,"I")

**CMN UNIT OR BASIS FOR MEASUREMENT CODE**

FORM FIELD REFERENCE: IB 837 TRANSMISSION

SECURITY LEVEL: NATIONAL,NO EDIT

DATA ELEMENT: N-GET FROM PREVIOUS EXTRACT

PAD CHARACTER: NO PAD REQUIRED

FORMAT CODE: N Z K IBXDATA S Z=0 F S Z=$O(IBXSAVE("CMNDEX",Z)) Q:'Z S IBXDATA(Z)="MO"

**CMN MONTHS DME EQUIPMENT NEEDED**

FORM FIELD REFERENCE: IB 837 TRANSMISSION

SECURITY LEVEL: NATIONAL,NO EDIT

DATA ELEMENT: N-GET FROM PREVIOUS EXTRACT

PAD CHARACTER: NO PAD REQUIRED REQUIRED: NO

EDIT STATUS: EDITABLE

FORMAT CODE: N Z K IBXDATA S Z=0 F S Z=$O(IBXSAVE("CMNDEX",Z)) Q:'Z S IBXDATA(Z)=$$CMNDATA^IBCEF31(IBXIEN,+IBXSAVE("CMNDEX",Z),24.04)

**CMN CODE CATEGORY**

FORM FIELD REFERENCE: IB 837 TRANSMISSION

SECURITY LEVEL: NATIONAL,NO EDIT

DATA ELEMENT: N-GET FROM PREVIOUS EXTRACT

PAD CHARACTER: NO PAD REQUIRED

FORMAT CODE: N Z K IBXDATA S Z=0 F S Z=$O(IBXSAVE("CMNDEX",Z)) Q:'Z S IBXDATA(Z)="09"

**CMN CERTIFICATION CONDITION INDICATOR**

FORM FIELD REFERENCE: IB 837 TRANSMISSION

SECURITY LEVEL: NATIONAL,NO EDIT

DATA ELEMENT: N-GET FROM PREVIOUS EXTRACT

PAD CHARACTER: NO PAD REQUIRED

FORMAT CODE: N Z K IBXDATA S Z=0 F S Z=$O(IBXSAVE("CMNDEX",Z)) Q:'Z S IBXDATA(Z)="Y"

**CMN CONDITION INDICATOR**

FORM FIELD REFERENCE: IB 837 TRANSMISSION

SECURITY LEVEL: NATIONAL,NO EDIT

DATA ELEMENT: N-GET FROM PREVIOUS EXTRACT

PAD CHARACTER: NO PAD REQUIRED

FORMAT CODE: N Z K IBXDATA S Z=0 F S Z=$O(IBXSAVE("CMNDEX",Z)) Q:'Z S IBXDATA(Z)="38"

**CMN REPLACEMENT ITEM?**

FORM FIELD REFERENCE: IB 837 TRANSMISSION

SECURITY LEVEL: NATIONAL,NO EDIT

DATA ELEMENT: N-GET FROM PREVIOUS EXTRACT

PAD CHARACTER: NO PAD REQUIRED REQUIRED: NO

EDIT STATUS: EDITABLE

FORMAT CODE: N Z K IBXDATA S Z=0 F S Z=$O(IBXSAVE("CMNDEX",Z)) Q:'Z I $$CMNDATA^IBCEF31(IBXIEN,+IBXSAVE("CMNDEX",Z),24.08,"I") S IBXDATA(Z)="ZV"

**CMN DATE THERAPY STARTED QUALIFIER**

FORM FIELD REFERENCE: IB 837 TRANSMISSION

SECURITY LEVEL: NATIONAL,NO EDIT

DATA ELEMENT: N-GET FROM PREVIOUS EXTRACT

PAD CHARACTER: NO PAD REQUIRED REQUIRED: NO

EDIT STATUS: DISPLAY ONLY

FORMAT CODE: N Z K IBXDATA S Z=0 F S Z=$O(IBXSAVE("CMNDEX",Z)) Q:'Z S IBXDATA(Z)=463

FORMAT CODE DESCRIPTION: The value of CMN DATE THERAPY STARTED QUALIFIER is always '463'.

**CMN DATE THERAPY STARTED**

FORM FIELD REFERENCE: IB 837 TRANSMISSION

SECURITY LEVEL: NATIONAL,NO EDIT

DATA ELEMENT: N-GET FROM PREVIOUS EXTRACT

PAD CHARACTER: NO PAD REQUIRED REQUIRED: NO

EDIT STATUS: EDITABLE

FORMAT CODE: N Z K IBXDATA S Z=0 F S Z=$O(IBXSAVE("CMNDEX",Z)) Q:'Z S IBXDATA(Z)=$$DT^IBCEFG1($$CMNDATA^IBCEF31(IBXIEN,+IBXSAVE("CMNDEX",Z),24.05,"I"),"","D8")

**CMN LAST CERTIFICATION DATE QUALIFIER**

FORM FIELD REFERENCE: IB 837 TRANSMISSION

SECURITY LEVEL: NATIONAL,NO EDIT

DATA ELEMENT: N-GET FROM PREVIOUS EXTRACT

PAD CHARACTER: NO PAD REQUIRED REQUIRED: NO

EDIT STATUS: EDITABLE

FORMAT CODE: N Z K IBXDATA S Z=0 F S Z=$O(IBXSAVE("CMNDEX",Z)) Q:'Z S IBXDATA(Z)=461

**CMN LAST CERTIFICATION DATE**

FORM FIELD REFERENCE: IB 837 TRANSMISSION

SECURITY LEVEL: NATIONAL,NO EDIT

DATA ELEMENT: N-GET FROM PREVIOUS EXTRACT

PAD CHARACTER: NO PAD REQUIRED REQUIRED: NO

EDIT STATUS: EDITABLE

FORMAT CODE: N Z K IBXDATA S Z=0 F S Z=$O(IBXSAVE("CMNDEX",Z)) Q:'Z S IBXDATA(Z)=$$DT^IBCEFG1($$CMNDATA^IBCEF31(IBXIEN,+IBXSAVE("CMNDEX",Z),24.06,"I"),"","D8")

**CMN CERTIFICATION TYPE QUAL**

FORM FIELD REFERENCE: IB 837 TRANSMISSION

SECURITY LEVEL: NATIONAL,NO EDIT

DATA ELEMENT: N-GET FROM PREVIOUS EXTRACT

PAD CHARACTER: NO PAD REQUIRED REQUIRED: NO

EDIT STATUS: DISPLAY ONLY

FORMAT CODE: N Z,CERTYP K IBXDATA S Z=0 F S Z=$O(IBXSAVE("CMNDEX",Z)) Q:'Z S CERTYP=$$CMNDATA^IBCEF31(IBXIEN,+IBXSAVE("CMNDEX",Z),24.01,"I") I CERTYP="R"!(CERTYP="S") S IBXDATA(Z)=607

**CMN RECERTIFICATION/REVISION DATE**

FORM FIELD REFERENCE: IB 837 TRANSMISSION

SECURITY LEVEL: NATIONAL,NO EDIT

DATA ELEMENT: N-GET FROM PREVIOUS EXTRACT

PAD CHARACTER: NO PAD REQUIRED REQUIRED: NO

EDIT STATUS: EDITABLE

FORMAT CODE: N Z K IBXDATA S Z=0 F S Z=$O(IBXSAVE("CMNDEX",Z)) Q:'Z S IBXDATA(Z)=$$DT^IBCEFG1($$CMNDATA^IBCEF31(IBXIEN,+IBXSAVE("CMNDEX",Z),24.07,"I"),"","D8")

**CMN ATTACHMENT REPORT TYPE CODE**

FORM FIELD REFERENCE: IB 837 TRANSMISSION

SECURITY LEVEL: NATIONAL,NO EDIT

DATA ELEMENT: N-GET FROM PREVIOUS EXTRACT

PAD CHARACTER: NO PAD REQUIRED

FORMAT CODE: N Z K IBXDATA S Z=0 F S Z=$O(IBXSAVE("CMNDEX",Z)) Q:'Z S IBXDATA(Z)="CT"

**CMN ATTACHMENT TRANSMISSION CODE**

FORM FIELD REFERENCE: IB 837 TRANSMISSION

SECURITY LEVEL: NATIONAL,NO EDIT

DATA ELEMENT: N-GET FROM PREVIOUS EXTRACT

PAD CHARACTER: NO PAD REQUIRED

FORMAT CODE: N Z K IBXDATA S Z=0 F S Z=$O(IBXSAVE("CMNDEX",Z)) Q:'Z S IBXDATA(Z)="AD"

**FRM Segment**

**CMN RECORD ID 'FRM '**

FORM FIELD REFERENCE: IB 837 TRANSMISSION

SECURITY LEVEL: NATIONAL,NO EDIT DATA ELEMENT: N-CMN RECORD ID 'FRM '

PAD CHARACTER: NO PAD REQUIRED EDIT STATUS: DISPLAY ONLY

FORMAT CODE: K IBXDATA S IBXDATA="FRM "

**FRM DATA EXTRACT**

FORM FIELD REFERENCE: IB 837 TRANSMISSION

SECURITY LEVEL: NATIONAL,NO EDIT

DATA ELEMENT: N-GET FROM PREVIOUS EXTRACT

PAD CHARACTER: NO PAD REQUIRED

FORMAT CODE: K IBXSAVE("FRM") D FRM^IBCEF31(IBXIEN,.IBXSAVE)

**SERVICE LINE #**

FORM FIELD REFERENCE: IB 837 TRANSMISSION

SECURITY LEVEL: NATIONAL,NO EDIT

DATA ELEMENT: N-GET FROM PREVIOUS EXTRACT

PAD CHARACTER: NO PAD REQUIRED EDIT STATUS: DISPLAY ONLY

FORMAT CODE: N Z K IBXDATA S Z=0 F S Z=$O(IBXSAVE("FRM",Z)) Q:'Z S IBXDATA(Z)=$P(IBXSAVE("FRM",Z),U,6) D:Z>1 ID^IBCEF2(Z,"FRM")

**CMN QUESTION NUMBER/LETTER**

FORM FIELD REFERENCE: IB 837 TRANSMISSION

SECURITY LEVEL: NATIONAL,NO EDIT

DATA ELEMENT: N-GET FROM PREVIOUS EXTRACT

PAD CHARACTER: NO PAD REQUIRED

FORMAT CODE: N Z K IBXDATA S Z=0 F S Z=$O(IBXSAVE("FRM",Z)) Q:'Z S IBXDATA(Z)=$P(IBXSAVE("FRM",Z),U)

**CMN QUESTION RESPONSE Y/N**

FORM FIELD REFERENCE: IB 837 TRANSMISSION

SECURITY LEVEL: NATIONAL,NO EDIT

DATA ELEMENT: N-GET FROM PREVIOUS EXTRACT

PAD CHARACTER: NO PAD REQUIRED

FORMAT CODE: N Z K IBXDATA S Z=0 F S Z=$O(IBXSAVE("FRM",Z)) Q:'Z S IBXDATA(Z)=$P(IBXSAVE("FRM",Z),U,2)

**CMN QUESTION RESPONSE REF ID**

FORM FIELD REFERENCE: IB 837 TRANSMISSION

SECURITY LEVEL: NATIONAL,NO EDIT

DATA ELEMENT: N-GET FROM PREVIOUS EXTRACT

PAD CHARACTER: NO PAD REQUIRED

FORMAT CODE: N Z K IBXDATA S Z=0 F S Z=$O(IBXSAVE("FRM",Z)) Q:'Z S IBXDATA(Z)=$P(IBXSAVE("FRM",Z),U,3)

**CMN QUESTION RESPONSE DATE**

FORM FIELD REFERENCE: IB 837 TRANSMISSION

SECURITY LEVEL: NATIONAL,NO EDIT

DATA ELEMENT: N-GET FROM PREVIOUS EXTRACT

PAD CHARACTER: NO PAD REQUIRED

FORMAT CODE: N Z K IBXDATA S Z=0 F S Z=$O(IBXSAVE("FRM",Z)) Q:'Z S IBXDATA(Z

)=$P(IBXSAVE("FRM",Z),U,4)

**CMN QUESTION RESPONSE % & DECIMAL**

FORM FIELD REFERENCE: IB 837 TRANSMISSION

SECURITY LEVEL: NATIONAL,NO EDIT

DATA ELEMENT: N-GET FROM PREVIOUS EXTRACT

FORMAT CODE: N Z K IBXDATA S Z=0 F S Z=$O(IBXSAVE("FRM",Z)) Q:'Z S IBXDATA(Z)=$P(IBXSAVE("FRM",Z),U,5)

**LQ Segment**

**CMN RECORD ID 'LQ '**

FORM FIELD REFERENCE: IB 837 TRANSMISSION

SECURITY LEVEL: NATIONAL,NO EDIT DATA ELEMENT: N-CMN RECORD ID 'LQ '

PAD CHARACTER: NO PAD REQUIRED REQUIRED: NO

EDIT STATUS: DISPLAY ONLY FORMAT CODE: K IBXDATA S IBXDATA="LQ "

**LQ DATA EXTRACT**

FORM FIELD REFERENCE: IB 837 TRANSMISSION

SECURITY LEVEL: NATIONAL,NO EDIT

DATA ELEMENT: N-GET FROM PREVIOUS EXTRACT

PAD CHARACTER: NO PAD REQUIRED

FORMAT CODE: D:$D(IBXSAVE("CMNDEX"))'>1 CMNDEX^IBCEF31(IBXIEN,.IBXSAVE)

**SERVICE LINE #**

FORM FIELD REFERENCE: IB 837 TRANSMISSION

SECURITY LEVEL: NATIONAL,NO EDIT

DATA ELEMENT: N-GET FROM PREVIOUS EXTRACT

PAD CHARACTER: NO PAD REQUIRED REQUIRED: NO

EDIT STATUS: DISPLAY ONLY

FORMAT CODE: N Z K IBXDATA S Z=0 F S Z=$O(IBXSAVE("CMNDEX",Z)) Q:'Z S IBXDATA(Z)=$P(IBXSAVE("CMNDEX",Z),U,2) D:Z>1 ID^IBCEF2(Z,"LQ")

**CMN FORM TYPE QUALIFIER**

FORM FIELD REFERENCE: IB 837 TRANSMISSION

SECURITY LEVEL: NATIONAL,NO EDIT

DATA ELEMENT: N-GET FROM PREVIOUS EXTRACT

PAD CHARACTER: NO PAD REQUIRED REQUIRED: NO

EDIT STATUS: DISPLAY ONLY

FORMAT CODE: N Z K IBXDATA S Z=0 F S Z=$O(IBXSAVE("CMNDEX",Z)) Q:'Z S IBXDATA(Z)="UT"

FORMAT CODE DESCRIPTION: The CMN FORM TYPE QUALIFIER is always 'UT'.

**CMN INDUSTRY CODE**

FORM FIELD REFERENCE: IB 837 TRANSMISSION

SECURITY LEVEL: NATIONAL,NO EDIT

DATA ELEMENT: N-GET FROM PREVIOUS EXTRACT

PAD CHARACTER: NO PAD REQUIRED REQUIRED: NO

EDIT STATUS: DISPLAY ONLY

FORMAT CODE: N Z K IBXDATA S Z=0 F S Z=$O(IBXSAVE("CMNDEX",Z)) Q:'Z S IBXDATA(Z)=$$CMNDATA^IBCEF31(IBXIEN,+IBXSAVE("CMNDEX",Z),"24:2")

**MEA Segment**

**CMN RECORD ID ‘MEA ‘**

FORM FIELD REFERENCE: IB 837 TRANSMISSION

SECURITY LEVEL: NATIONAL,NO EDIT

DATA ELEMENT: N-CMN RECORD ID 'MEA '

PAD CHARACTER: NO PAD REQUIRED

FORMAT CODE: K IBXDATA S IBXDATA="MEA "

**MEA DATA EXTRACT**

FORM FIELD REFERENCE: IB 837 TRANSMISSION

SECURITY LEVEL: NATIONAL,NO EDIT

DATA ELEMENT: N-GET FROM PREVIOUS EXTRACT

PAD CHARACTER: NO PAD REQUIRED

FORMAT CODE: D:$D(IBXSAVE("CMNDEX"))'>1 CMNDEX^IBCEF31(IBXIEN,.IBXSAVE)

**SERVICE LINE #**

FORM FIELD REFERENCE: IB 837 TRANSMISSION

SECURITY LEVEL: NATIONAL,NO EDIT

DATA ELEMENT: N-GET FROM PREVIOUS EXTRACT

PAD CHARACTER: NO PAD REQUIRED

FORMAT CODE: N Z K IBXDATA S Z=0 F S Z=$O(IBXSAVE("CMNDEX",Z)) Q:'Z S IBXDATA(Z)=$P(IBXSAVE("CMNDEX",Z),U,2) D:Z>1 ID^IBCEF2(Z,"MEA")

**CMN MEASUREMENT REFERENCE ID CODE**

FORM FIELD REFERENCE: IB 837 TRANSMISSION

SECURITY LEVEL: NATIONAL,NO EDIT

DATA ELEMENT: N-GET FROM PREVIOUS EXTRACT

PAD CHARACTER: NO PAD REQUIRED REQUIRED: NO

FORMAT CODE: N Z K IBXDATA S Z=0 F S Z=$O(IBXSAVE("CMNDEX",Z)) Q:'Z S IBXDATA(Z)="TR"

**CMN MEASUREMENT QUALIFIER**

FORM FIELD REFERENCE: IB 837 TRANSMISSION

SECURITY LEVEL: NATIONAL,NO EDIT

DATA ELEMENT: N-GET FROM PREVIOUS EXTRACT

PAD CHARACTER: NO PAD REQUIRED

FORMAT CODE: N Z K IBXDATA S Z=0 F S Z=$O(IBXSAVE("CMNDEX",Z)) Q:'Z S IBXDATA(Z)="" I $$CMNDATA^IBCEF31(IBXIEN,+IBXSAVE("CMNDEX",Z),24.02) S IBXDATA(Z)="HT"

**CMN TEST RESULTS**

FORM FIELD REFERENCE: IB 837 TRANSMISSION

SECURITY LEVEL: NATIONAL,NO EDIT

DATA ELEMENT: N-GET FROM PREVIOUS EXTRACT

PAD CHARACTER: NO PAD REQUIRED

FORMAT CODE: N Z K IBXDATA S Z=0 F S Z=$O(IBXSAVE("CMNDEX",Z)) Q:'Z S IBXDATA(Z)=$$CMNDATA^IBCEF31(IBXIEN,+IBXSAVE("CMNDEX",Z),24.02)

**PT1 Segment**

**CMN PATIENT WEIGHT (LBS)**

FORM FIELD REFERENCE: IB 837 TRANSMISSION

SECURITY LEVEL: NATIONAL,NO EDIT

DATA ELEMENT: N-GET FROM PREVIOUS EXTRACT

PAD CHARACTER: NO PAD REQUIRED REQUIRED: NO

EDIT STATUS: EDITABLE

FORMAT CODE: K IBXDATA S IBXDATA=$$PTWT^IBCEF31(IBXIEN)

**CMN PATIENT WEIGHT MODIFIER**

FORM FIELD REFERENCE: IB 837 TRANSMISSION

SECURITY LEVEL: NATIONAL,NO EDIT

DATA ELEMENT: N-GET FROM PREVIOUS EXTRACT

PAD CHARACTER: NO PAD REQUIRED REQUIRED: NO

EDIT STATUS: EDITABLE

FORMAT CODE: K IBXDATA S IBXDATA="" I $$PTWT^IBCEF31(IBXIEN) S IBXDATA="01"

FORMAT CODE DESCRIPTION: This is the PATIENT WEIGHT MODIFIER which is always '01'.

**VI) New entries to the IB ERROR file 350.8 for missing or incorrect CMN Data:**

**IB CMN NOT REQ BUT DATA**

NAME: **IB CMN NOT REQ BUT DATA**

ERROR MESSAGE: - "CMN Required?" set to NO, but CMN data exists.

ERROR CODE: **IB901**

PACKAGE REPORTING ERROR: INTEGRATED BILLING

ERROR ACTION: DISPLAY MESSAGE

**IB CMN FORM TYPE**

NAME: **IB CMN FORM TYPE**

ERROR MESSAGE: - "CMN Form type" missing.

ERROR CODE: **IB902**

PACKAGE REPORTING ERROR: INTEGRATED BILLING

ERROR ACTION: DISPLAY MESSAGE

**IB CMN NO DATA NODE**

NAME: **IB CMN NO DATA NODE**

ERROR MESSAGE: - CMN form-specific data missing for the Form Type chosen.

ERROR CODE: **IB903**

PACKAGE REPORTING ERROR: INTEGRATED BILLING

ERROR ACTION: DISPLAY MESSAGE

**IB CMN BAD DATA NODE**

NAME: **IB CMN BAD DATA NODE**

ERROR MESSAGE: - CMN data does not match the chosen Form Type.

ERROR CODE: **IB904**

PACKAGE REPORTING ERROR: INTEGRATED BILLING

ERROR ACTION: DISPLAY MESSAGE

**IB CMN CERT TYPE**

NAME: **IB CMN CERT TYPE**

ERROR MESSAGE: - "Certification Type" missing.

ERROR CODE: **IB905**

PACKAGE REPORTING ERROR: INTEGRATED BILLING

ERROR ACTION: DISPLAY MESSAGE

**IB CMN PEB**

NAME: **IB CMN PEB**

ERROR MESSAGE: - "Is this for Parenteral nutrition, Enteral nutrition, or Both?" missing.

ERROR CODE: **IB906**

PACKAGE REPORTING ERROR: INTEGRATED BILLING

ERROR ACTION: DISPLAY MESSAGE

**IB CMN THERAPY DT**

NAME: **IB CMN THERAPY DT**

ERROR MESSAGE: - "Date Therapy Started" missing.

ERROR CODE: **IB907**

PACKAGE REPORTING ERROR: INTEGRATED BILLING

ERROR ACTION: DISPLAY MESSAGE

**IB CMN LAST CERT DT**

NAME: **IB CMN LAST CERT DT**

ERROR MESSAGE: - "Last Certification Date" missing.

ERROR CODE: **IB908**

PACKAGE REPORTING ERROR: INTEGRATED BILLING

ERROR ACTION: DISPLAY MESSAGE

**IB CMN RECERT/REVISION DT**

NAME: **IB CMN RECERT/REVISION DT**

ERROR MESSAGE: - "Recertification/Revision Date" missing.

ERROR CODE: **IB909**

PACKAGE REPORTING ERROR: INTEGRATED BILLING

ERROR ACTION: DISPLAY MESSAGE

**IB CMN ABG SAT DT**

NAME: IB CMN ABG SAT DT

ERROR MESSAGE: - Date of last "ABG PO2" and/or "O2 Saturation" Test(s) missing.

ERROR CODE: **IB912**

PACKAGE REPORTING ERROR: INTEGRATED BILLING

ERROR ACTION: DISPLAY MESSAGE

**IB CMN 4 LPM DATE**

NAME: IB CMN 4 LPM DATE

ERROR MESSAGE: - "Date of Latest 4 LPM Test(s) missing.

ERROR CODE: **IB914**

PACKAGE REPORTING ERROR: INTEGRATED BILLING

ERROR ACTION: DISPLAY MESSAGE

**IB CMN ERRORS HEADER**

NAME: IB CMN ERRORS HEADER

ERROR MESSAGE: The following CMN field(s) missing or in error for at least 1 procedure:

ERROR CODE: **IB915**

PACKAGE REPORTING ERROR: INTEGRATED BILLING

ERROR ACTION: DISPLAY MESSAGE

**Routines Modified**

**V1) Check for missing CMN data after entering a bill and display appropriate error messages:**

**IBCBB1** – Calls routine IBCBB13 to perform the checks for missing CMN data

| **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** |  | | | | | | | | |
| **Related Protocols** | None | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Current Logic** | | | | | | | | | |
| IBCBB1 ;ALB/AAS - CONTINUATION OF EDIT CHECK ROUTINE ;2-NOV-89  ;;2.0;INTEGRATED BILLING;\*\*27,52,80,93,106,51,151,148,153,137,232,280,155,320,343,349,363,371,395,384,432,447,488,554,577,592\*\*;21-MAR-94;Build 25  ;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\_"IB365;" ;IB\*2\*577;JWS;11/20/17 FIX  ; 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),$$EDIACTV^IBCEF4(2) D MRA  Q  ;  MRA N IBEOB S IBEOB=0  ;  K PRCASV("MEDURE"),PRCASV("MEDCA")  ; Get EOB data  F S IBEOB=$O(^IBM(361.1,"B",IBIFN,IBEOB)) Q:'IBEOB D  . D MRACALC^IBCEMU2(IBEOB,IBIFN,1,.PRCASV)  Q ;MRA  ;  ;; PREGNANCY DX CODES: V22\*\*-V24\*\*, V27\*\*-V28\*\*, 630\*\*-677\*\*  ;; FLU SHOTS PROCEDURE CODES: 90724, G0008, 90732, G0009  ;  NONVAFLG(IBIFN) ; Check if Non-VA bill  ; Function returns 1 if Non-VA bill  ; IB\*2.0\*554/DRF 10/9/2015  N FLAG,PTF  S FLAG=0  I $P($G(^DGCR(399,IBIFN,"U2")),U,10)]"" S FLAG=1 ;Non-VA provider defined  S PTF=$P($G(^DGCR(399,IBIFN,0)),U,8)  I PTF,$P($G(^DGPT(PTF,0)),U,4)=1 S FLAG=1 ;PTF entry indicates Non-VA  Q FLAG | | | | | | | | | |
| **Modified Logic (Changes are highlighted in yellow)** | | | | | | | | | |
| IBCBB1 ;ALB/AAS - CONTINUATION OF EDIT CHECK ROUTINE ;2-NOV-89  ;;2.0;INTEGRATED BILLING;\*\*27,52,80,93,106,51,151,148,153,137,232,280,155,320,343,349,363,371,395,384,432,447,488,554,577,592,608\*\*;21-MAR-94;Build 21  ;Per VA Directive 6402, this routine should not be modified.  ;  ; \*\*\* Begin IB\*2.0\*488 VD (Issue 46 RBN)  N I  S I=""  S X=+$G(^DGCR(399,IBIFN,"MP"))  I 'X,$$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,$$GET1^DIQ(399,IBIFN\_",",158)="" S IBER=IBER\_"IB349;"  ;  I $$FT^IBCEF(IBIFN)=2 S IBER=IBER\_$$CMNCHK^IBCBB13(IBIFN) ;JRA;IB\*2.0\*608 Check for missing CMN info  ;  END ;Don't kill IBIFN, IBER, DFN  I $O(^TMP($J,"BILL-WARN",0)),$G(IBER)="" S IBER="WARN" ;Warnings only  K IBBNO,IBEVDT,IBLOC,IBCL,IBTF,IBAT,IBWHO,IBST,IBFDT,IBTDT,IBTC,IBFY,IBFY1,IBAU,IBRU,IBEU,IBARTP,IBFYC,IBMRA,IBTOB,IBTOB12,IBNDU2,IBNDUF3,IBNDUF31,IBNDTX  K IBNDS,IBND0,IBNDU,IBNDM,IBNDMP,IBNDU1,IBFFY,IBTFY,IBFT,IBRTCHV,IBPICHV,IBXDATA,IBOK  I $D(IBER),IBER="" W !,"No Errors found for National edits"  Q  ;  ARRAY ;Build PRCASV(array)  N IBCOBN,X  K PRCASV  Q:$$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 | | | | | | | | | |

**IBCBB13** – Perform the actual checks for missing CMN data (called by IBCBB1)

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | **IBCBB13** | | | | | | | | |
| **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** | | | | | | | | | |
| IBCBB13 ;ALB/BI - PROCEDURE AND LINE LEVEL PROVIDER EDITS ;5-OCT-2011  ;;2.0;INTEGRATED BILLING;\*\*447\*\*;21-MAR-94;Build 80  ;;Per VHA Directive 2004-038, this routine should not be modified.  Q  ;  IBLNTOT(IBIFN) ; Calculate Line total charges. IB\*2.0\*447 BI  N X,SUM S SUM=0  S X=0 F S X=$O(^DGCR(399,IBIFN,"RC",X)) Q:+X=0 D  . S SUM=SUM+$P($G(^DGCR(399,IBIFN,"RC",X,0)),"^",4)  Q SUM  ;  IBSYEI(IBIFN) ; Test for valid EIN/SY ID Values. IB\*2.0\*447 BI  N X12CODE,RESULT,IBPIEN,IBWIEN,IBLIEN  S RESULT=0  ; Check Claim Level Providers  S IBWIEN=IBIFN\_","  S X12CODE=$$GET1^DIQ(355.97,$$GET1^DIQ(399,IBWIEN,128,"I")\_",",.03)  I ((X12CODE="SY")!(X12CODE="EI")),$TR($$GET1^DIQ(399,IBWIEN,122),"-","")'?9N S RESULT=1 Q RESULT  S X12CODE=$$GET1^DIQ(355.97,$$GET1^DIQ(399,IBWIEN,129,"I")\_",",.03)  I ((X12CODE="SY")!(X12CODE="EI")),$TR($$GET1^DIQ(399,IBWIEN,123),"-","")'?9N S RESULT=1 Q RESULT  S X12CODE=$$GET1^DIQ(355.97,$$GET1^DIQ(399,IBWIEN,130,"I")\_",",.03)  I ((X12CODE="SY")!(X12CODE="EI")),$TR($$GET1^DIQ(399,IBWIEN,124),"-","")'?9N S RESULT=1 Q RESULT  ; Check Claim Level Providers  S IBPIEN=0 F S IBPIEN=$O(^DGCR(399,IBIFN,"PRV",IBPIEN)) Q:+IBPIEN=0 Q:RESULT=1 D  .S IBWIEN=IBPIEN\_","\_IBIFN\_","  .; Test for each provider listed.  .S X12CODE=$$GET1^DIQ(355.97,$$GET1^DIQ(399.0222,IBWIEN,.12,"I")\_",",.03)  .I ((X12CODE="SY")!(X12CODE="EI")),$TR($$GET1^DIQ(399.0222,IBWIEN,.05),"-","")'?9N S RESULT=1 Q  .S X12CODE=$$GET1^DIQ(355.97,$$GET1^DIQ(399.0222,IBWIEN,.13,"I")\_",",.03)  .I ((X12CODE="SY")!(X12CODE="EI")),$TR($$GET1^DIQ(399.0222,IBWIEN,.06),"-","")'?9N S RESULT=1 Q  .S X12CODE=$$GET1^DIQ(355.97,$$GET1^DIQ(399.0222,IBWIEN,.14,"I")\_",",.03)  .I ((X12CODE="SY")!(X12CODE="EI")),$TR($$GET1^DIQ(399.0222,IBWIEN,.07),"-","")'?9N S RESULT=1 Q  ; Check Line Level Providers  ; For each charge code / line.  S IBLIEN=0 F S IBLIEN=$O(^DGCR(399,IBIFN,"CP",IBLIEN)) Q:+IBLIEN=0 Q:RESULT=1 D  .; For each provider associated with the line.  .S IBPIEN=0 F S IBPIEN=$O(^DGCR(399,IBIFN,"CP",IBLIEN,"LNPRV",IBPIEN)) Q:+IBPIEN=0 Q:RESULT=1 D  ..S IBWIEN=IBPIEN\_","\_IBLIEN\_","\_IBIFN\_","  ..; Test for each provider listed.  ..S X12CODE=$$GET1^DIQ(355.97,$$GET1^DIQ(399.0404,IBWIEN,.12,"I")\_",",.03)  ..I ((X12CODE="SY")!(X12CODE="EI")),$TR($$GET1^DIQ(399.0404,IBWIEN,.05),"-","")'?9N S RESULT=1 Q  ..S X12CODE=$$GET1^DIQ(355.97,$$GET1^DIQ(399.0404,IBWIEN,.13,"I")\_",",.03)  ..I ((X12CODE="SY")!(X12CODE="EI")),$TR($$GET1^DIQ(399.0404,IBWIEN,.06),"-","")'?9N S RESULT=1 Q  ..S X12CODE=$$GET1^DIQ(355.97,$$GET1^DIQ(399.0404,IBWIEN,.14,"I")\_",",.03)  ..I ((X12CODE="SY")!(X12CODE="EI")),$TR($$GET1^DIQ(399.0404,IBWIEN,.07),"-","")'?9N S RESULT=1 Q  Q RESULT  ;  IBMICN(IBIFN) ; Test for a missing ICN. IB\*2.0\*447 BI  N IBTFOB ; TIMEFRAME OF BILL  N IBCBPS ; CURRENT BILL PAYER SEQUENCE, P-PRI, S-SEC, T-TER, A-PATIENT  S IBTFOB=$$GET1^DIQ(399,IBIFN\_",",.06,"I")  I '((IBTFOB=7)!(IBTFOB=8)) Q 0  S IBCBPS=$$GET1^DIQ(399,IBIFN\_",",.21,"I")  I IBCBPS="P",$$GET1^DIQ(399,IBIFN\_",",101)'="",$$GET1^DIQ(399,IBIFN\_",",453)="" Q 1  I IBCBPS="S",$$GET1^DIQ(399,IBIFN\_",",102)'="",$$GET1^DIQ(399,IBIFN\_",",454)="" Q 1  I IBCBPS="T",$$GET1^DIQ(399,IBIFN\_",",103)'="",$$GET1^DIQ(399,IBIFN\_",",455)="" Q 1  Q 0  ;  IBRCCHK(IBIFN) ; Test for a ZERO charge amounts. IB\*2.0\*447 BI  N IBN0  N IBRCCNT S IBRCCNT=0  N IBRCCHG S IBRCCHG=0  F S IBRCCNT=$O(^DGCR(399,IBIFN,"RC",IBRCCNT)) Q:+IBRCCNT=0 Q:IBRCCHG=1 D  .S IBN0=$G(^DGCR(399,IBIFN,"RC",IBRCCNT,0))  .I $P(IBN0,U,1)'="",+$P(IBN0,U,4)=0 S IBRCCHG=1  Q IBRCCHG  ;  IBPRV3(IBIFN) ; Test for missing "Patient reason for visit". IB\*2.0\*447 BI  I $$GET1^DIQ(399,IBIFN\_",",249)="",$$GET1^DIQ(399,IBIFN\_",",250)="",$$GET1^DIQ(399,IBIFN\_",",251)="" Q 1  Q 0  ;  IBMPID(IBIFN) ; Test for multiple payers. IB\*2.0\*447 BI  N IBPAY1 S IBPAY1=$$GET1^DIQ(399,IBIFN\_",",101,"I")  N IBPAY2 S IBPAY2=$$GET1^DIQ(399,IBIFN\_",",102,"I")  N IBPAY3 S IBPAY3=$$GET1^DIQ(399,IBIFN\_",",103,"I")  N IBCNT S IBCNT=0  S:IBPAY1 IBCNT=IBCNT+1 S:IBPAY2 IBCNT=IBCNT+1 S:IBPAY3 IBCNT=IBCNT+1 I IBCNT<2 Q 0  N IBINSTIT S IBINSTIT=$$INSPRF^IBCEF(IBIFN)  I IBPAY1,$S(IBINSTIT:$$GET1^DIQ(36,IBPAY1\_",",3.04),1:$$GET1^DIQ(36,IBPAY1\_",",3.02))="" Q 1  I IBPAY2,$S(IBINSTIT:$$GET1^DIQ(36,IBPAY2\_",",3.04),1:$$GET1^DIQ(36,IBPAY2\_",",3.02))="" Q 1  I IBPAY3,$S(IBINSTIT:$$GET1^DIQ(36,IBPAY3\_",",3.04),1:$$GET1^DIQ(36,IBPAY3\_",",3.02))="" Q 1  Q 0 | | | | | | | | | |
| **Modified Logic (Changes are highlighted in yellow)** | | | | | | | | | |
| IBCBB13 ;ALB/BI - PROCEDURE AND LINE LEVEL PROVIDER EDITS ;5-OCT-2011  ;;2.0;INTEGRATED BILLING;\*\*447,608\*\*;21-MAR-94;Build 40  ;;Per VHA Directive 2004-038, this routine should not be modified.  Q  ;  IBLNTOT(IBIFN) ; Calculate Line total charges. IB\*2.0\*447 BI  N X,SUM S SUM=0  S X=0 F S X=$O(^DGCR(399,IBIFN,"RC",X)) Q:+X=0 D  . S SUM=SUM+$P($G(^DGCR(399,IBIFN,"RC",X,0)),"^",4)  Q SUM  ;  IBSYEI(IBIFN) ; Test for valid EIN/SY ID Values. IB\*2.0\*447 BI  N X12CODE,RESULT,IBPIEN,IBWIEN,IBLIEN  S RESULT=0  ; Check Claim Level Providers  S IBWIEN=IBIFN\_","  S X12CODE=$$GET1^DIQ(355.97,$$GET1^DIQ(399,IBWIEN,128,"I")\_",",.03)  I ((X12CODE="SY")!(X12CODE="EI")),$TR($$GET1^DIQ(399,IBWIEN,122),"-","")'?9N S RESULT=1 Q RESULT  S X12CODE=$$GET1^DIQ(355.97,$$GET1^DIQ(399,IBWIEN,129,"I")\_",",.03)  I ((X12CODE="SY")!(X12CODE="EI")),$TR($$GET1^DIQ(399,IBWIEN,123),"-","")'?9N S RESULT=1 Q RESULT  S X12CODE=$$GET1^DIQ(355.97,$$GET1^DIQ(399,IBWIEN,130,"I")\_",",.03)  I ((X12CODE="SY")!(X12CODE="EI")),$TR($$GET1^DIQ(399,IBWIEN,124),"-","")'?9N S RESULT=1 Q RESULT  ; Check Claim Level Providers  S IBPIEN=0 F S IBPIEN=$O(^DGCR(399,IBIFN,"PRV",IBPIEN)) Q:+IBPIEN=0 Q:RESULT=1 D  .S IBWIEN=IBPIEN\_","\_IBIFN\_","  .; Test for each provider listed.  .S X12CODE=$$GET1^DIQ(355.97,$$GET1^DIQ(399.0222,IBWIEN,.12,"I")\_",",.03)  .I ((X12CODE="SY")!(X12CODE="EI")),$TR($$GET1^DIQ(399.0222,IBWIEN,.05),"-","")'?9N S RESULT=1 Q  .S X12CODE=$$GET1^DIQ(355.97,$$GET1^DIQ(399.0222,IBWIEN,.13,"I")\_",",.03)  .I ((X12CODE="SY")!(X12CODE="EI")),$TR($$GET1^DIQ(399.0222,IBWIEN,.06),"-","")'?9N S RESULT=1 Q  .S X12CODE=$$GET1^DIQ(355.97,$$GET1^DIQ(399.0222,IBWIEN,.14,"I")\_",",.03)  .I ((X12CODE="SY")!(X12CODE="EI")),$TR($$GET1^DIQ(399.0222,IBWIEN,.07),"-","")'?9N S RESULT=1 Q  ; Check Line Level Providers  ; For each charge code / line.  S IBLIEN=0 F S IBLIEN=$O(^DGCR(399,IBIFN,"CP",IBLIEN)) Q:+IBLIEN=0 Q:RESULT=1 D  .; For each provider associated with the line.  .S IBPIEN=0 F S IBPIEN=$O(^DGCR(399,IBIFN,"CP",IBLIEN,"LNPRV",IBPIEN)) Q:+IBPIEN=0 Q:RESULT=1 D  ..S IBWIEN=IBPIEN\_","\_IBLIEN\_","\_IBIFN\_","  ..; Test for each provider listed.  ..S X12CODE=$$GET1^DIQ(355.97,$$GET1^DIQ(399.0404,IBWIEN,.12,"I")\_",",.03)  ..I ((X12CODE="SY")!(X12CODE="EI")),$TR($$GET1^DIQ(399.0404,IBWIEN,.05),"-","")'?9N S RESULT=1 Q  ..S X12CODE=$$GET1^DIQ(355.97,$$GET1^DIQ(399.0404,IBWIEN,.13,"I")\_",",.03)  ..I ((X12CODE="SY")!(X12CODE="EI")),$TR($$GET1^DIQ(399.0404,IBWIEN,.06),"-","")'?9N S RESULT=1 Q  ..S X12CODE=$$GET1^DIQ(355.97,$$GET1^DIQ(399.0404,IBWIEN,.14,"I")\_",",.03)  ..I ((X12CODE="SY")!(X12CODE="EI")),$TR($$GET1^DIQ(399.0404,IBWIEN,.07),"-","")'?9N S RESULT=1 Q  Q RESULT  ;  IBMICN(IBIFN) ; Test for a missing ICN. IB\*2.0\*447 BI  N IBTFOB ; TIMEFRAME OF BILL  N IBCBPS ; CURRENT BILL PAYER SEQUENCE, P-PRI, S-SEC, T-TER, A-PATIENT  S IBTFOB=$$GET1^DIQ(399,IBIFN\_",",.06,"I")  I '((IBTFOB=7)!(IBTFOB=8)) Q 0  S IBCBPS=$$GET1^DIQ(399,IBIFN\_",",.21,"I")  I IBCBPS="P",$$GET1^DIQ(399,IBIFN\_",",101)'="",$$GET1^DIQ(399,IBIFN\_",",453)="" Q 1  I IBCBPS="S",$$GET1^DIQ(399,IBIFN\_",",102)'="",$$GET1^DIQ(399,IBIFN\_",",454)="" Q 1  I IBCBPS="T",$$GET1^DIQ(399,IBIFN\_",",103)'="",$$GET1^DIQ(399,IBIFN\_",",455)="" Q 1  Q 0  ;  IBRCCHK(IBIFN) ; Test for a ZERO charge amounts. IB\*2.0\*447 BI  N IBN0  N IBRCCNT S IBRCCNT=0  N IBRCCHG S IBRCCHG=0  F S IBRCCNT=$O(^DGCR(399,IBIFN,"RC",IBRCCNT)) Q:+IBRCCNT=0 Q:IBRCCHG=1 D  .S IBN0=$G(^DGCR(399,IBIFN,"RC",IBRCCNT,0))  .I $P(IBN0,U,1)'="",+$P(IBN0,U,4)=0 S IBRCCHG=1  Q IBRCCHG  ;  IBPRV3(IBIFN) ; Test for missing "Patient reason for visit". IB\*2.0\*447 BI  I $$GET1^DIQ(399,IBIFN\_",",249)="",$$GET1^DIQ(399,IBIFN\_",",250)="",$$GET1^DIQ(399,IBIFN\_",",251)="" Q 1  Q 0  ;  IBMPID(IBIFN) ; Test for multiple payers. IB\*2.0\*447 BI  N IBPAY1 S IBPAY1=$$GET1^DIQ(399,IBIFN\_",",101,"I")  N IBPAY2 S IBPAY2=$$GET1^DIQ(399,IBIFN\_",",102,"I")  N IBPAY3 S IBPAY3=$$GET1^DIQ(399,IBIFN\_",",103,"I")  N IBCNT S IBCNT=0  S:IBPAY1 IBCNT=IBCNT+1 S:IBPAY2 IBCNT=IBCNT+1 S:IBPAY3 IBCNT=IBCNT+1 I IBCNT<2 Q 0  N IBINSTIT S IBINSTIT=$$INSPRF^IBCEF(IBIFN)  I IBPAY1,$S(IBINSTIT:$$GET1^DIQ(36,IBPAY1\_",",3.04),1:$$GET1^DIQ(36,IBPAY1\_",",3.02))="" Q 1  I IBPAY2,$S(IBINSTIT:$$GET1^DIQ(36,IBPAY2\_",",3.04),1:$$GET1^DIQ(36,IBPAY2\_",",3.02))="" Q 1  I IBPAY3,$S(IBINSTIT:$$GET1^DIQ(36,IBPAY3\_",",3.04),1:$$GET1^DIQ(36,IBPAY3\_",",3.02))="" Q 1  Q 0  ;  CMNCHK(IBIFN) ;JRA;IB\*2.0\*608 Check for missing required Certificate of Medical Necessity (CMN) data  ; Input : IBIFN = IEN of the Bill/Claim (D399)  ; Output: IBER = NULL if no errors  ; = String of IB Error Message codes delimited by ';'  ; => Note that the return value is appended to the 'IBER' variable in routine ^IBCBB1  Q:IBIFN="" ""  N CERTYP,CMNNODE,CMNREQ,DA,DIE,ERR,FRMNAM,FRMIEN,FORM,FRMTYP,IBER,IBPROCP,PROCNUM  S IBER=""  ;Set up array of each existing Form Type (i.e. Form IENs) and associated ^DGCR data node.  S FRMNAM="" F S FRMNAM=$O(^IBE(399.6,"B",FRMNAM)) Q:FRMNAM="" S FRMIEN=+$O(^IBE(399.6,"B",FRMNAM,"")) I FRMIEN D  . S FORM(FRMIEN)=$P($G(^IBE(399.6,FRMIEN,0)),U,4)  ;Loop thru all procedures on the claim searching for missing CMN data  S IBPROCP=0 F S IBPROCP=$O(^DGCR(399,IBIFN,"CP",IBPROCP)) Q:'IBPROCP D Q:IBER]""  . ;If "CMN Required?" is NULL then QUIT w/out further checking  . S CMNREQ=$$CVALCHK(IBPROCP,23,,"I") Q:CMNREQ=""  . I 'CMNREQ,$D(FORM)>1 D Q ;"CMN Required?" flagged as "NO" so check if data node(s) exist anyway for at least 1 form  . . S ERR=0,FRMIEN="" F S FRMIEN=$O(FORM(FRMIEN)) Q:FRMIEN="" I FORM(FRMIEN)]"" D Q:ERR  . . . S CMNNODE="^DGCR(399,"\_IBIFN\_",""CP"","\_IBPROCP\_","""\_FORM(FRMIEN)\_""")" I $D(@CMNNODE) S ERR=1,IBER=IBER\_"IB901;"  . S FRMTYP=$$CVALCHK(IBPROCP,24,"IB902","I") Q:'FRMTYP ;Check for "CMN FORM TYPE" (Internal value)  . I $G(FORM(FRMTYP))]"" D Q:ERR  . . ;Check if any data exists at the node specific to the Form Type  . . S ERR=0,CMNNODE="^DGCR(399,"\_IBIFN\_",""CP"","\_IBPROCP\_","""\_FORM(FRMTYP)\_""")"  . . I '$D(@CMNNODE) S ERR=1,IBER=IBER\_"IB903;" Q  . . Q:FORM(FRMTYP)'[10126  . . N ND10126  . . S ND10126=@CMNNODE  . . I $P(ND10126,U,17)]"" S $P(ND10126,U,17)="" I $TR(ND10126,U)="" S ERR=1,IBER=IBER\_"IB903;"  . ;Check if any data exists for at least 1 node other than that associated with the Form Type  . S ERR=0,FRMIEN="" F S FRMIEN=$O(FORM(FRMIEN)) Q:FRMIEN="" I FRMIEN'=FRMTYP,FORM(FRMIEN)]"" D Q:ERR  . . S CMNNODE="^DGCR(399,"\_IBIFN\_",""CP"","\_IBPROCP\_","""\_FORM(FRMIEN)\_""")" I $D(@CMNNODE) S ERR=1,IBER=IBER\_"IB904;"  . ;Check for Required fields at the data node common to all forms (node 'CMN')  . S CERTYP=$$CVALCHK(IBPROCP,24.01,"IB905","I") Q:CERTYP="" ;Check for "CMN CERTIFICATION TYPE"  . D CVALCHK(IBPROCP,24.05,"IB907","I") ;Check for "CMN DATE THERAPY STARTED"  . D CVALCHK(IBPROCP,24.06,"IB908","I") ;Check for "CMN LAST CERTIFICATION DATE"  . ;IF Certificate Type is "RENEWAL" (R) or "REVISED" (S) then "CMN RECERTIFICATION/REVISN DT" is Required.  . I CERTYP="R"!(CERTYP="S") D CVALCHK(IBPROCP,24.07,"IB909","I")  . ;  . ;Check for required fields specific to the CMN-484 form  . I FORM(FRMTYP)[484 D ;Check for required fields/dates  . . I $$CVALCHK(IBPROCP,24.1,,"I")]""!($$CVALCHK(IBPROCP,24.102,,"I")]"") D CVALCHK(IBPROCP,24.103,"IB912","I")  . . I $$CVALCHK(IBPROCP,24.111,,"I")]""!($$CVALCHK(IBPROCP,24.113,,"I")]"") D CVALCHK(IBPROCP,24.114,"IB914","I")  . ;  . ;Check for required fields specific to the CMN-10126 form  . I FORM(FRMTYP)[10126 D  . . D CVALCHK(IBPROCP,24.217,"IB906","I")  . . N PROCMSG  . . S PROCMSG="CMN ""Procedure ",PROCMSG(1)=""" has no associated Calories."  . . I $$CVALCHK(IBPROCP,24.204,,"I")]"",'$$CVALCHK(IBPROCP,24.203,,"I") D WARN^IBCBB11(PROCMSG\_"A"\_PROCMSG(1))  . . I $$CVALCHK(IBPROCP,24.219,,"I")]"",'$$CVALCHK(IBPROCP,24.218,,"I") D WARN^IBCBB11(PROCMSG\_"B"\_PROCMSG(1))  ;  I IBER]"" S IBER="IB915;"\_IBER  Q IBER  ;  CVALCHK(IBPROCP,FLD,ERROR,FLG) ;JRA;IB\*2.0\*608 Check value of CMN field & append Error Code (if any) to list of errors  Q:($G(FLD)=""!('$G(IBPROCP)))  N VAL  S VAL=$$CMNDATA^IBCEF31(IBIFN,IBPROCP,FLD,$G(FLG))  I $G(ERROR)]"",VAL="" S IBER=IBER\_ERROR\_";"  Q VAL  ; | | | | | | | | | |

**VII) Use DR Prompting to collect CMN information:**

**IBCU7** – Calls new routine IBCU75 to prompt user for CMN info (due to large size of IBCU7)

| **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;\*\*62,52,106,125,51,137,210,245,228,260,348,371,432,447,488,461,516,522,577,592\*\*;21-MAR-94;Build 25  ;;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 daterestrict 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//"\_$$DEFDIV(IBIFN)\_";"  . ;JWS;IB\*2.0\*592 US1108 - Dental  . ;IA# 10018  . S DR=DR\_$S(IBFT=7:"8;9//;",IBFT=2:"8;9;17//NO;",1:"")\_3,DIE=DIC,(IBPROCP,DA)=+Y D ^DIE Q:'$D(DA)!($E($G(Y))=U)  . K DR ;WCJ;IB\*2.0\*432  . ;  . ; MRD;IB\*2.0\*516 - Allow user to add an NDC and Units. Ask only if  . ; coding system is not ICD and this is not a prescription claim. If  . ; an NDC is entered, prompt for Units.  . I $P($G(^DGCR(399,IBIFN,0)),U,9)'=9,'$$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  ; | | | | | | | | | |
| **Modified Logic (Changes are highlighted in yellow)** | | | | | | | | | |
| IBCU7 ;ALB/AAS - INTERCEPT SCREEN INPUT OF PROCEDURE CODES ;29-OCT-91  ;;2.0;INTEGRATED BILLING;\*\*62,52,106,125,51,137,210,245,228,260,348,371,432,447,488,461,516,522,577,604,592,608\*\*;21-MAR-94;Build 40  ;;Per VA Directive 6402, this routine should not be modified.  ;  ;MAP TO DGCRU7  ;  ; This routine is a copy of IBUC7 for testing purposes.  ;  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 daterestrict 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  . S IBCODE=X ;Get the code  . ; 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 $$MAINPRV^IBCEU(IBIFN) is for claim  . ; level provider defaults.  . ; 1. For new line level providers we don't need  . ; or want default claim level provider  . ; (requirement).  . ; 2. We don't want to default claim level to  . ; line level provider (requirement).  . ;  . K DIC("V") ; DEM;432 - KILL DIC("V") because this was for previous variable pointer use.  . ;  . N IBPROCSV ; DEM;432 - Variable IBPROCSV is variable to preserve value of 'Y', which is procedure code info returned by call to ^DIC.  . S IBPROCSV=Y ; DEM;432 - Preserve value of Y for after calls to FileMan (Y = procedure code info returned by call to ^DIC).  . K DR ;WCJ;IB\*2.0\*432  . ;  . I IBPROCSV["ICD0" S DR=".01",DIE=DIC,(IBPROCP,DA)=+Y D ^DIE Q:'$D(DA)!($D(Y)) K DR ; IB\*2.0\*461  . ;  . ;JRA;IB\*2.0\*608 Prompt user for Certificate of Medical Necessity (CMN) info  . I $$FT^IBCEF(IBIFN)=2,$$CMNPRMT^IBJPS8(IBIFN,IBPROCP,$P($P(IBPROCSV,U,2),";")) D CMN^IBCU75(IBIFN,IBPROCP)  . ;  . I IBPROCSV["ICPT" S DR=".01;16",DIE=DIC,(IBPROCP,DA)=+Y D ^DIE Q:'$D(DA)!($D(Y)) K DR ; IB\*2.0\*447 BI  . ;  . S DR=""  . ;  . ; MRD;IB\*2.0\*516 - Added line level PROCEDURE DESCRIPTION field,  . ; asked only if the procedure is an "NOC".  . I IBPROCSV["ICPT",$$NOCPROC(IBPROCSV,IBCODE,DGPROCDT) D ; added IBCODE,DGPROCDT in \*604  . . 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//"\_$$DEFDIV(IBIFN)\_";"  . ;JWS;IB\*2.0\*592 US1108 - Dental  . ;IA# 10018  . S DR=DR\_$S(IBFT=7:"8;9//;",IBFT=2:"8;9;17//NO;",1:"")\_3,DIE=DIC,(IBPROCP,DA)=+Y D ^DIE Q:'$D(DA)!($E($G(Y))=U)  . K DR ;WCJ;IB\*2.0\*432  . ;  . ; MRD;IB\*2.0\*516 - Allow user to add an NDC and Units. Ask only if  . ; coding system is not ICD and this is not a prescription claim. If  . ; an NDC is entered, prompt for Units.  . I $P($G(^DGCR(399,IBIFN,0)),U,9)'=9,'$$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="""";52R~//UN;54R~QUANTITY//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 IBFT=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//"\_$$OBSHOUR^IBCU74(DFN,$P(IBCPT,U,2))\_";" G SPCUNTQ ; hours  I +IBFT=2,$P($G(^IBE(353.2,+$P(IBCPT,U,10),0)),U,2)="ANESTHESIA" S IBDR="15;" ; minutes  SPCUNTQ Q IBDR  ;  ATTACH ; DEM;432 - Attachment control number.  ; Ask if user wants to enter Attachment Control Number.  N DIR,X,Y,DA,DIE,DR  S DIR("A")="Enter Attachment Control Number"  S DIR(0)="Y",DIR("B")="NO"  D ^DIR  Q:'Y  ; User chose to enter Attachment Control Number.  ; User enters Attachment Control fields.  S DA(1)=IBIFN,DA=IBPROCP  S DIE="^DGCR(399,"\_DA(1)\_",""CP"","  S DR="71Report Type;72Report Transmission Method;70Attachment Control Number"  D ^DIE  Q  ;  NOCPROC(IBPROCSV,IBCODE,IBDATE) ; MRD;IB\*2.0\*516 - Function to determine if procedure is an  ; "NOC". Returns '1' if "NOC" procedure, otherwise '0'.  ;  N IBNOC,IBPROCEX,IBPROCIN,IBPROCNM,IBX,IBLINES,IBSTR,IBEND,IBLN  S IBNOC=0  I $G(IBPROCSV)="" G NOCPROCQ  I $G(IBCODE)="" G NOCPROCQ  I $G(IBDATE)'?7N G NOCPROCQ  S IBPROCIN=$P($P(IBPROCSV,U,2),";") ;parsing out the IEN  I IBPROCIN="" G NOCPROCQ  ;  ; If procedure code ends in '99', quit with a '1'.  ;  I $E(IBCODE,$L(IBCODE)-1,$L(IBCODE))=99 S IBNOC=1 G NOCPROCQ ;Does code end with 99? If so NOC  ;  ; Pull procedure name, then check to see if it contains one of the  ; specified strings.  ;  S IBPROCNM=$$CPT^ICPTCOD(IBCODE,IBDATE)  S IBPROCNM=$P(IBPROCNM,U,3)  I IBPROCNM'="",($$NOC(IBPROCNM)) S IBNOC=1 G NOCPROCQ ; Does external match NOC strings? if so NOC  ;  ;Does array strings match any of the specified strings  S IBLINES=$$CPTD^ICPTCOD(IBCODE,"IBINFO",,IBDATE) ;get number of lines/array of lines  S IBEND=1 S:IBLINES>1 IBEND=IBLINES-1 ;set up counter for loop  F IBLN=1:1:IBEND D Q:IBNOC=1 ;loop through array so we can check if node values = NOC  . N IBSTR S IBSTR=$$TM($G(IBINFO(IBLN)))\_" "\_$$TM($G(IBINFO(IBLN+1)))\_" " ;Build strings for NOC comparison  . S IBNOC=$$NOC(IBSTR) ;is current combination of strings a NOC?  . Q  ;  NOCPROCQ ; Quit out.  K IBINFO ;killing the array made in CPTD^ICPTCOD  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))  Q 0  ;  TM(IBX,IBY) ; Trim Character Y - Default " "  S IBX=$G(IBX) Q:IBX="" IBX S IBY=$G(IBY) S:'$L(IBY) IBY=" "  F Q:$E(IBX,1)'=IBY S IBX=$E(IBX,2,$L(IBX))  F Q:$E(IBX,$L(IBX))'=IBY S IBX=$E(IBX,1,($L(IBX)-1))  Q IBX  ;  ORALCAV(FLD) ;EP;IB\*2.0\*592  ; 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 399 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 $P(NDE,"^",4)=Y S RTN=0 Q  Q RTN  ;  TOOTHS(FLD) ;EP;IB\*2.0\*592  ; Dictionary Screen function called from Dental Service Line Tooth fields:  ; 399,91,.02, 399,91,.03, 399,91,.04, 399,91,.05, 399,91,.06. Prevents the  ; same Tooth Surface from being selected more than once.  ; Input: FLD - Field # of the field being checked  ; DA - Tooth Surface multiple IEN  ; DA(1) - Service Line multiple IEN  ; DA(2) - IEN of the 399 entry being edited  ; Y - Internal Value of the user response  ; Returns: 1 - Data input by the user is valid, 0 otherwise  N NDE,RTN  S NDE=$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  ; | | | | | | | | | |

**IBCU75** – Use DR Prompting to prompt user for CMN information (called by IBCU7)

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | **IBCU75** | | | | | | | | |
| **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** | | | | | | | | | |
| IBCU75 ;ALB/JRA - INTERCEPT SCREEN INPUT OF PROCEDURE CODES (ENTER CMN INFO) ;23-Apr-18  ;;2.0;INTEGRATED BILLING;\*\*608\*\*;23-Apr-18  ;;Per VA Directive 6402, this routine should not be modified.  ;  Q  ;  CMN(IBXIEN,IBPROCP) ;JRA;IB\*2.0\*608 Prompt user for CMN info  ;Input: IBXIEN = Internal bill/claim number  ; IBPROCP = Procedure line subscript in ^DGCR  ;  Q:('$G(IBXIEN)!('$G(IBPROCP)))  N ABGMSG,ABGPO2,CERTDT,CERTYP,CHNGFRM,CMNNODE,CMNREQ,CMSG,DA,DIC,DIE,DIR,DGLB,DR,DRTAG,DTOLD,EDIT,EVNTDT,FIEN,FNAM,FORM,FRMTAG  N FRMTYP,HT,HTOLD,I,IBPEB,WTOLD,LKGLB,LPM4ABG,LPM4SAT,MSG,NODE0,O2SAT,OK,OLDVAL,PROCA,PROCB,QUIT,RRDT,TDY,THERPYDT,X,Y  S DGLB="^TMP(""CMN"",$J)" K @DGLB  S LKGLB="^DGCR(399,"\_IBXIEN\_")" L +@LKGLB:0 I '$T W !,$C(7),"Another user is editing this entry -- EXITING" H 2 Q  S EVNTDT=$$FMTE^DILIBF($G(IBDT),"5U") ;Get the Event Date - will be the default for several date fields.  S TDY=$$HTFM^DILIBF(+$H)  S ABGMSG="""ABG PO2"" and/or ""O2 Saturation"" Test(s) REQUIRED"  S DA=IBPROCP,DA(1)=IBXIEN,DIE="^DGCR(399,"\_IBXIEN\_",""CP"","  ;Set FORM array of CMN Data Nodes (D399.6 field 3) indexed by CMN Form Type ien  S FNAM="" F S FNAM=$O(^IBE(399.6,"B",FNAM)) Q:FNAM="" S FIEN=+$O(^IBE(399.6,"B",FNAM,"")) I FIEN D  . S FORM(FIEN)=$P($G(^IBE(399.6,FIEN,0)),U,4) K:$TR(FORM(FIEN)," ")="" FORM(FIEN)  I $D(FORM)'>1 S FORM(1)="CMN-484",FORM(2)="CMN-10126" ;Default nodes for CMN data  S DIE("NO^")="BACKOUTOK"  S CMNREQ("MSG")="If ""CMN Required?"" is changed to ""NO"", existing CMN data will be deleted!"  S FRMTYP("MSG")="Changing the Form Type will delete any data specific to the current Form Type!"  S CERTYP("MSG")="You are changing the Certification Type!"  S CERTYP("MSGI")="Changing Certification Type to ""I"" will delete ""Recertification/Revision Date!"""  D CMNREQ  S QUIT=0 F D Q:QUIT  . D ^DIE  . S CMNREQ=$G(CMNREQ),FRMTYP=$G(FRMTYP),CERTYP=$G(CERTYP)  . S CMNREQ=$$CMNDATA^IBCEF31(IBXIEN,IBPROCP,23,"I") I CMNREQ=0 S QUIT=1 Q  . S FRMTYP=$$CMNDATA^IBCEF31(IBXIEN,IBPROCP,24,"I")  . S CERTYP=$$CMNDATA^IBCEF31(IBXIEN,IBPROCP,24.01,"I")  . I FRMTYP,CERTYP'="" S QUIT=1 Q  . I CMNREQ="" W $C(7),!,?3,"""CMN Required?"" is a REQUIRED field!" D CMNREQ Q  . S MSG=""  . I FRMTYP="" S MSG="""Form Type"" and ""Certification Type"" are REQUIRED!",DRTAG="CMNREQ"  . E I CERTYP="" S MSG="""Certification Type"" is REQUIRED!",DRTAG="CMNREQ"  . I MSG]"" S DR="",MSG=MSG\_$C(13,10)\_" \*\* To exit, set ""CMN Required?"" to ""NO""" W $C(7),!,?3,MSG D @DRTAG Q  . S QUIT=1  ;  ;If CMN is not required, delete all CMN data that may be associated with this procedure & exit  I $G(CMNREQ)=0 D Q  . S FIEN="" F S FIEN=$O(FORM(FIEN)) Q:FIEN="" I FORM(FIEN)]"" D  . . S CMNNODE="^DGCR(399,"\_IBXIEN\_",""CP"","\_IBPROCP\_","""\_FORM(FIEN)\_""")" K @CMNNODE  . S CMNNODE="^DGCR(399,"\_IBXIEN\_",""CP"","\_IBPROCP\_",""CMN"")" K @CMNNODE S @CMNNODE=0  ;  ;If user selected Form Type we need to remove data that may exist for any other Form Type.  I $G(FRMTYP) S FIEN="" F S FIEN=$O(FORM(FIEN)) Q:FIEN="" I FIEN'=FRMTYP D  . S CMNNODE="^DGCR(399,"\_IBXIEN\_",""CP"","\_IBPROCP\_","""\_FORM(FIEN)\_""")" K @CMNNODE  ;  I $G(CERTYP)="I" D SETFLD(24.07,"@") ;If "Certification Type" is "INITIAL" delete "Recertification/Revision Date"  ;  I (($D(EDIT)&($G(EDIT)'="Y"))!(X=""!('$G(CMNREQ)!('$G(FRMTYP)!($G(CERTYP)=""))))) Q  ;  S FRMTAG="DR"\_$S($G(FORM(FRMTYP))[484:484,1:10126) ;Set tag to call to set DR with form-specific logic  D DRCOMM  ;  ;Prompt user for remaining questions & check for missing required fields  S (QUIT,UPCT)=0,DRTAG(1)="" F D Q:QUIT  . D ^DIE  . K MSG S MSG=0  . S DRTAG=""  . S CERTYP=$$CMNDATA^IBCEF31(IBXIEN,IBPROCP,24.01,"I")  . S HT=$$CMNDATA^IBCEF31(IBXIEN,IBPROCP,24.02,"I")  . S THERPYDT=$$CMNDATA^IBCEF31(IBXIEN,IBPROCP,24.05,"I")  . S CERTDT=$$CMNDATA^IBCEF31(IBXIEN,IBPROCP,24.06,"I")  . S RRDT=$$CMNDATA^IBCEF31(IBXIEN,IBPROCP,24.07,"I")  . I 'CERTDT S MSG=MSG+1,MSG(MSG)="""Last Certification Date""" S DRTAG="DRCOMM"  . I 'RRDT,CERTYP'="I" S MSG=MSG+1,MSG(MSG)="""Recertification/Revision Date""" S:DRTAG="" DRTAG="RRDT"  . I 'THERPYDT S MSG=MSG+1,MSG(MSG)="""Date Therapy Started""" S:DRTAG="" DRTAG="STRTDT"  . I FORM(FRMTYP)[10126 D  . . I $$CMNDATA^IBCEF31(IBXIEN,IBPROCP,24.217,"I")="" S MSG=MSG+1,MSG(MSG)="""Is this for Parenteral nutrition, Enteral nutrition, or Both?""" S:DRTAG="" DRTAG="DR10126"  . I +MSG D Q  . . S:X="" UPCT=UPCT+1 I UPCT>1,DRTAG=DRTAG(1) S QUIT=1 Q  . . S DR="" W $C(7) F I=1:1:MSG W !,?3,MSG(I)\_" is REQUIRED!"  . . W !,?3,"\*\* Exiting now will leave required fields unanswered."  . . W !,?3,"\*\* If you must exit, enter '^' again."  . . S DRTAG(1)=DRTAG D @DRTAG  . S QUIT=1  ;  ;Delete dates associated with result fields that were deleted  I $D(@DGLB)>1 D  . N FLD  . S FLD="" F S FLD=$O(@DGLB@(FLD)) Q:FLD="" D SETFLD(FLD,"@")  . K @DGLB  Q  ;  CMNREQ ; Set DR with logic for 1st 3 fields: "CMN Required?", "Form Type" and "Certification Type"  S DR="@23;S CMNREQ(""OLD"")=$$CMNDATA^IBCEF31(IBXIEN,IBPROCP,23,""I"");23R~T//NO;S CMNREQ=X I 'X,'CMNREQ(""OLD"") S Y=""@999"";"  S DR=DR\_"I CMNREQ=0,CMNREQ(""OLD"")=1 S FRM=$$CMNDATA^IBCEF31(IBXIEN,IBPROCP,24,""I"") S:'FRM OK=1 S:FRM OK=$$USEROK^IBCU75(23,1,CMNREQ(""MSG""))"  S DR=DR\_" S:OK Y=""@999"" I 'OK S Y=""@23"";"  FRMTYP ;Entry point to set DR with logic for "Form Type" and "Certification Type" fields in preparation for re-prompting.  S DR=DR\_"@24;S DIC(0)=""N"" S FRMTYP(""OLD"")=$$CMNDATA^IBCEF31(IBXIEN,IBPROCP,24,""I"");24R~T;S FRMTYP=X I FRMTYP(""OLD"")]"""",FRMTYP]"""""  S DR=DR\_",FRMTYP'=FRMTYP(""OLD"") S OK=$$USEROK^IBCU75(24,FRMTYP(""OLD""),FRMTYP(""MSG"")) S:OK CHNGFRM=1 S:'OK Y=""@24"";"  S DR=DR\_"I $G(CHNGFRM)!($$CMNDATA^IBCEF31(IBXIEN,IBPROCP,24.01,""I"")="""") D COPYCMN^IBCU75(IBXIEN,IBPROCP,FRMTYP);"  S DR=DR\_"I $$CMNDATA^IBCEF31(IBXIEN,IBPROCP,24.01,""I"")]"""",'$G(CHNGFRM) R !,""Edit CMN Information for this Procedure? NO// "",EDIT S EDIT=$E($ZCONVERT(EDIT,""U"")) "  S DR=DR\_"W:(EDIT]""""&(EDIT'=""^"")) "" ""\_$S(EDIT=""Y"":""YES"",1:""NO"") I EDIT'=""Y"" S Y=""@999"";"  CERTYP ;Entry point to set DR with logic for "Certification Type" field in preparation for re-prompting.  S DR=DR\_"@01;S CERTYP(""OLD"")=$$CMNDATA^IBCEF31(IBXIEN,IBPROCP,24.01,""I"");24.01R~T//INITIAL"  S DR=DR\_";S CERTYP=X I CERTYP(""OLD"")]"""",CERTYP]"""",CERTYP'=CERTYP(""OLD"")"  S DR=DR\_" S CMSG=$S(CERTYP=""I"":CERTYP(""MSGI""),1:CERTYP(""MSG""))"  S DR=DR\_" S OK=$$USEROK^IBCU75(24.01,CERTYP(""OLD""),CMSG) S:'OK Y=""@01"";@999;"  Q  ;  DRCOMM ;Set DR with logic for the remaining fields common to all form types  S DR="@06;S DTOLD=$$CMNDATA^IBCEF31(IBXIEN,IBPROCP,24.06,""I"");24.06R~T//"\_EVNTDT\_";D DTCHK^IBCU75(X,TDY,""06"",$G(DTOLD));"  S DR=DR\_"I CERTYP=""I"" S @DGLB@(24.07)="""",Y=""@02"";"  RRDT ;Entry point to set DR with logic for "Recertification/Revision Date"... fields in preparation for re-prompting.  S DR=DR\_"@07;S DTOLD=$$CMNDATA^IBCEF31(IBXIEN,IBPROCP,24.07,""I"");24.07R~T//"\_EVNTDT\_";D DTCHK^IBCU75(X,TDY,""07"",$G(DTOLD));"  S DR=DR\_"@02;S HTOLD=$$CMNDATA^IBCEF31(IBXIEN,IBPROCP,24.02,""I"");24.02T;I X>96 S OK=$$USEROK^IBCU75(24.02,HTOLD,""Patient is over 8 feet tall!"")"  S DR=DR\_" I 'OK S Y=""@02"";@03;S WTOLD=$$CMNDATA^IBCEF31(IBXIEN,IBPROCP,24.03,""I"");24.03T;I X>500 S OK=$$USEROK^IBCU75(24.03,WTOLD,"  S DR=DR\_"""Patient is over 500 pounds!"") I 'OK S Y=""@03"";24.04T;"  STRTDT ;Entry point to set DR with logic for "Date Therapy Started"... fields in preparation for re-prompting.  S DR=DR\_"@05;S DTOLD=$$CMNDATA^IBCEF31(IBXIEN,IBPROCP,24.05,""I"");24.05R~T//"\_EVNTDT\_";D DTCHK^IBCU75(X,TDY,""05"",$G(DTOLD));@08;24.08T//N;"  D @FRMTAG  Q  ;  DR484 ;Set DR with logic specific for form CMN-484  S DR=DR\_"@100;24.1T;S ABGPO2=X;@102;24.102T;S O2SAT=X;I ABGPO2="""",O2SAT="""" S Y=""@104"";"  S DR=DR\_"@103;S DTOLD=$$CMNDATA^IBCEF31(IBXIEN,IBPROCP,24.103,""I"");24.103T;D DTCHK^IBCU75(X,TDY,103,$G(DTOLD));"  S DR=DR\_"@104;I (ABGPO2<56!(ABGPO2>59)),(O2SAT'=89) S @DGLB@(24.104)="""",@DGLB@(24.105)="""""  S DR=DR\_",@DGLB@(24.106)="""",Y=""@107"";24.104T//NO;24.105T//NO;24.106T//NO;@107;24.107T;24.108T;24.109T;24.11T;I X'>4 S @DGLB@(24.111)="""""  S DR=DR\_",@DGLB@(24.113)="""",@DGLB@(24.114)="""",Y=""@115"";24.111T;S ABG4LPM=X;"  S DR=DR\_"@113;24.113T;I 'ABG4LPM,'X S Y=""@115"",@DGLB@(24.114)="""";"  S DR=DR\_"@114;S DTOLD=$$CMNDATA^IBCEF31(IBXIEN,IBPROCP,24.114,""I"");24.114T;D DTCHK^IBCU75(X,TDY,114,$G(DTOLD));@115;24.115T;@999;"  Q  ;  DR10126 ;Set DR with logic specific to the CMN-10126  S DR=DR\_"@217;S IBPEB(""OLD"")=$$CMNDATA^IBCEF31(IBXIEN,IBPROCP,24.217,""I"");24.217R~T//P;S IBPEB=X I IBPEB(""OLD"")]"""",IBPEB]"""",IBPEB(""OLD"")'=IBPEB "  S DR=DR\_"S OK=$$USEROK^IBCU75(24.217,IBPEB(""OLD""),""You are changing the nutrition type!"") S:'OK Y=""@217"";I $G(IBPEB)=""P"" S Y=""@206"" "  S DR=DR\_"N I F I=24.201:.001:24.205,24.218,24.219 S @DGLB@(I)="""";24.201T;24.202T;"  S DR=DR\_"24.204T;I '+X S Y=""@205"",@DGLB@(24.203)="""" I $$CMNDATA^IBCEF31(IBXIEN,IBPROCP,24.219)]"""" S Y=""@219"";"  S DR=DR\_"24.203T;I '+X S Y=""@205"" I $$CMNDATA^IBCEF31(IBXIEN,IBPROCP,24.219)]"""" S Y=""@219"";"  S DR=DR\_"@219;24.219T;I '+X S Y=""@205"",@DGLB@(24.218)="""";"  S DR=DR\_"24.218T;@205;24.205T;@206;24.206T;I $G(IBPEB)=""E"" S Y=""@999"" "  S DR=DR\_"N I F I=24.207:.001:24.216 S @DGLB@(I)="""";"  S DR=DR\_"24.207T;24.208T;24.209T;24.21T;24.211T;24.212T;24.213T;24.215T;24.216T;@214;24.214T;@999;"  Q  ;  COPYCMN(IBXIEN,IBPROCP,FRMTYP) ;Copy CMN information from last procedure entered that has it to current procedure  ;Input: IBXIEN = Internal bill/claim number  ; IBPROCP = Procedure line subscript  ; FRMTYP = CMN Form Type ien  ;  N DONE  S DONE=0  Q:('$G(IBXIEN)!('$G(IBPROCP)!('$G(FRMTYP))))  N FRMND,FRMNDI,IBPROC,IBXSAVE,Z  S FRMNDI=FORM(FRMTYP)  D CMNDEX^IBCEF31(IBXIEN,.IBXSAVE)  S Z="" F S Z=$O(IBXSAVE("CMNDEX",Z),-1) Q:'Z S IBPROC=+IBXSAVE("CMNDEX",Z) I IBPROCP,IBPROC'=IBPROCP D Q:DONE  . Q:('$D(^DGCR(399,IBXIEN,"CP",IBPROC,"CMN"))!('$D(^DGCR(399,IBXIEN,"CP",IBPROC,FRMNDI))))  . S FRMND=$O(^DGCR(399,IBXIEN,"CP",IBPROC,"CMN")) Q:(FRMND=""!(FRMND'=FRMNDI))  . S ^DGCR(399,IBXIEN,"CP",IBPROCP,"CMN")=^DGCR(399,IBXIEN,"CP",IBPROC,"CMN")  . S ^DGCR(399,IBXIEN,"CP",IBPROCP,FRMND)=^DGCR(399,IBXIEN,"CP",IBPROC,FRMND)  . S DONE=1  Q  ;  USEROK(FLD,OLDVAL,MSG) ;JRA;IB\*2.0\*608 Prompt user if OK to change field value  ;Input: FLD = Field for which we are asking the user to confirm the change  ; OLDVAL = Value of the field before user changed  ; MSG = Warning message to display to user regarding the implications of the change  ;  Q:'$G(FLD) 0  N DIC,DIR,X,Y  S OLDVAL=$G(OLDVAL)  W $C(7) I $TR($G(MSG)," ")]"" W !,MSG  S DIR(0)="Y",DIR("A")="OK to continue",DIR("B")="NO" D ^DIR  I Y'=1 D SETFLD(FLD,OLDVAL) ;Set field back to old value if user doesn't want to continue  I Y=1 S X="^"  Q Y  ;  SETFLD(FLD,VAL) ;JRA;IB\*2.0\*608 Set/Delete field data w/out user prompting  ;Input: FLD = Field to set/delete  ; VAL = Value to set FLD to (Note: '@' will delete field value)  ;  Q:('$G(FLD)!($G(VAL)=""))  N DIE,DI,DL,DP,DQ,DR,X,Y  S DIE="^DGCR(399,"\_IBXIEN\_",""CP"","  S DR=FLD\_"////"\_VAL  D ^DIE  Q  ;  DTCHK(X,TDY,TAG,DTOLD) ;JRA;IB\*2.0\*608 Check if future date entered by user  ;Input: X = User entry for date field (internal FileMan date format)  ; TDY = Today's internal FileMan date  ; TAG = Field tag to jump to if user enters a future date (usually re-prompt same date)  ; DTOLD = The value of the date field prior to user edit  ;  Q:('$G(X))!('$G(TAG))  N FLD  S:$G(DTOLD)="" DTOLD="@"  S:'$G(TDY) TDY=$$HTFM^DILIBF(+$H) Q:X'>TDY  ;User entered future date so display error and change date back to previous value.  W $C(7),!,?3,"Future dates not allowed??"  S Y="@"\_TAG  D SETFLD("24."\_TAG,DTOLD) ;set back to prior date  Q  ; | | | | | | | | | |

**VIII) Create extract code for to pull the values for the new CMN fields:**

**IBCEF31**

| **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;\*\*155,296,349,400,432,488,516,592\*\*;21-MAR-94;Build 25  ;;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  ; | | | | | | | | | |
| **Modified Logic (Changes are highlighted in yellow)** | | | | | | | | | |
| IBCEF31 ;ALB/ESG - FORMATTER SPECIFIC BILL FLD FUNCTIONS - CONT ;14-NOV-03  ;;2.0;INTEGRATED BILLING;\*\*155,296,349,400,432,488,516,592,608\*\*;21-MAR-94;Build 40  ;;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  ;  CMNDATA(IBXIEN,IBPROC,FLD,INT) ;JRA;IB\*2.0\*608 Return data for specified Certificate of Medical Necessity (CMN) field.  ;Created to return data for a specific CMN field, which is a subfield of file 399, field 304 (Procedure). Returns data  ; in External format by default.  ;  ;Input: IBXIEN = Internal bill/claim number  ; IBPROC = Procedure # (subscript in ^DGCR)  ; FLD = Field number of desired field  ; INT = Flag set to 'I' if the subfield's Internal value is to be returned (optional)  ;  ;Output: VAL = External (or optionally Internal) value of the CMN subfield specified by FLD  ;  Q:('$G(IBXIEN)!('$G(FLD)!('$G(IBPROC)))) ""  S INT=$G(INT)  N ND,VAL,X  S ND=IBPROC\_","\_IBXIEN  S VAL=$$GET1^DIQ(399.0304,ND,FLD,INT)  Q VAL  ;  CMNDEX(IBXIEN,IBXSAVE) ;JRA;IB\*2.0\*608 Data Extract for LQ, CMN and MEA segments  Q:'$G(IBXIEN)  ;  N CMNREQ,ND,X,IBXDATA  ;Get Procedure Links for all Procedures on the claim.  D OUTPT^IBCEF11(IBXIEN,0) Q:'$D(IBXDATA)  N LP,Z,CNT  S LP=0 F S LP=$O(IBXDATA(LP)) Q:'+LP D  . S CNT=$G(CNT)+1  . Q:'$D(IBXDATA(LP,"CPLNK"))  . S ND=IBXDATA(LP,"CPLNK")  . S ND=ND\_","\_IBXIEN\_","  . S CMNREQ=$$GET1^DIQ(399.0304,ND,23,"I")  . S:CMNREQ="" CMNREQ=0  . Q:'+CMNREQ  . S Z=$G(Z)+1  . S IBXSAVE("CMNDEX",Z)=IBXDATA(LP,"CPLNK")\_U\_CNT  Q  ;  FRM(IBXIEN,IBXSAVE) ;JRA;IB\*2.0\*608 Data Extract for FRM segment  Q:'$G(IBXIEN)  ;  N CMNREQ,CNT,DEL,IBXDATA,LP,ND,QUIT,X,Z,Z1  ;Get Procedure Data for all Procedures on the claim.  D OUTPT^IBCEF11(IBXIEN,0) Q:'$D(IBXDATA)  S LP=0 F S LP=$O(IBXDATA(LP)) Q:'+LP D  . Q:'$D(IBXDATA(LP,"CPLNK"))  . S CNT=$G(CNT)+1  . S ND=IBXDATA(LP,"CPLNK")  . S ND=ND\_","\_IBXIEN\_","  . S CMNREQ=$$GET1^DIQ(399.0304,ND,23,"I")  . S:CMNREQ="" CMNREQ=0  . Q:'+CMNREQ  . S Z=$G(Z)+1  . ;WHAT FORM  . N DATA,FORM,FLD,FLDS,INTEXT,QUES,QUESNUM,X  . S FORM=$TR($$GET1^DIQ(399.0304,ND,"24:3","I"),"-") ; get the form number to figure what fields go with it  . Q:FORM="" ; quit if no form number  . ;  . S FLDS=$P($T(@FORM),";;",2,9999) ; get all the associated data fields from below  . ;  . N PAIREDQA  . ;Parse FLDS to get DD field, question number, type of response (2=Y/N, 3=text/code, 4=date, 5=percent/decimal), and the response data.  . F X=1:1 S QUES=$P(FLDS,"~",X) Q:QUES="" D  .. S FLD=$P(QUES,U)  .. S QUESNUM=$P(QUES,U,2)  .. S RESPTYP=$P(QUES,U,3)  .. I RESPTYP=4 S INTEXT="I"  .. E S INTEXT=$P(QUES,U,4) S:INTEXT="" INTEXT="E"  .. S DATA=$$GET1^DIQ(399.0304,ND,FLD,INTEXT)  .. ;  .. ; KLUDGE; On form CMN10126 If 4A or 3A is blank, don't send ther other (which means get rid of the previous Q/A)  .. ; same for 4B/3B  .. I FORM="CMN10126",".3A.3B.4A.4B."[QUESNUM S PAIRQ=0 D Q:PAIRQ  ... I QUESNUM="3A"!(QUESNUM="3B") S PAIREDQA(QUESNUM)=DATA Q  ... I QUESNUM="4A",$G(PAIREDQA("3A"))="" S PAIRQ=1 Q  ... I QUESNUM="4B",$G(PAIREDQA("3B"))="" S PAIRQ=1 Q  ..;  .. Q:DATA="" ;Do not include FRM rec for unanswered questions  .. ;  .. S:RESPTYP=2 DATA=$E(DATA) ; only want Y or N  .. S:RESPTYP=4 DATA=$$DT^IBCEFG1(DATA,"","D8") ;YYYYMMDD date format  .. ;Procedure# has a 1 to many ratio with Question# but can't have 2 subscripts so combine into 1, ordering IBXSAVE by Question#.  .. S IBXSAVE("FRM",(Z\_"\_"\_(X/10)))=QUESNUM\_U  .. S $P(IBXSAVE("FRM",(Z\_"\_"\_(X/10))),U,RESPTYP)=DATA  .. S $P(IBXSAVE("FRM",(Z\_"\_"\_(X/10))),U,6)=CNT  ;  ;Re-subscript IBXSAVE with sequential integers as current subscript format will not work with Output Formatter.  S (Z,Z1)=0 F S Z=$O(IBXSAVE("FRM",Z)) Q:'Z S Z1=Z1+1,IBXSAVE("FRM",Z1)=IBXSAVE("FRM",Z),DEL(Z)=""  S Z=0 F S Z=$O(DEL(Z)) Q:'Z K IBXSAVE("FRM",Z)  Q  ;  PTWT(IBXIEN) ;JRA;IB\*2.0\*608 Return CMN Patient Weight from 1st Service Line # that has it (or NULL if none)  Q:'$G(IBXIEN)  N FOUND,IBPROC,IBXSAVE,PTWT  D CMNDEX(IBXIEN,.IBXSAVE)  S (FOUND,Z)=0,PTWT="" F S Z=$O(IBXSAVE("CMNDEX",Z)) Q:Z="" D Q:FOUND  . S IBPROC=+IBXSAVE("CMNDEX",Z) Q:'IBPROC  . S PTWT=$$CMNDATA(IBXIEN,IBPROC,24.03) S:PTWT FOUND=1  Q PTWT  ;  ;JRA;IB\*2.0\*608 Tags CMN484 & CMN10126 added  ; FIELD#^QUESTION#^RESPONSE\_TYPE^INT/EXT  CMN484 ;;24.1^1A^3~24.102^1B^5~24.103^1C^4~24.107^2^3^I~24.108^3^3^I~24.109^4^3^I~24.11^5^3~24.111^6A^3~24.113^6B^5~24.114^6C^4~24.104^7^2~24.105^8^2~24.106^9^2~24.115^C^3  ;  CMN10126 ;;24.201^1^2~24.202^2^2~24.204^3A^3~24.219^3B^3~24.203^4A^3~24.218^4B^3~24.205^5^3^I~24.206^6^3~24.207^7^2~24.208^8A^3~24.209^8B^5~24.21^8C^3~24.211^8D^3~24.212^8E^5~24.213^8F^3~24.215^8G^3~24.216^8H^5~24.214^9^3^I  ; | | | | | | | | | |

**IX) Add the CMN nodes to comparison code:**

**IBCF23A**

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | **IBCF23A** | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** |  | | | | | | | | |
| **Related Options** | None | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  |  | | | | |  | | | |
| **Data Dictionary (DD) References** |  | | | | | | | | |
| **Related Protocols** | None | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Current Logic** | | | | | | | | | |
| IBCF23A ;ALB/ARH - HCFA 1500 19-90 DATA - Split from IBCF23 ;12-JUN-93  ;;2.0;INTEGRATED BILLING;\*\*51,432,516,547,577,592\*\*;21-MAR-94;Build 25  ;;Per VA Directive 6402, this routine should not be modified.  ;  ; $$INSTALDT^XPDUTL(IBPATCH,.IBARY) - ICR 10141  ;  B24 ; set individual entries in print array, external format  ; IBAUX = additional data for EDI output  ; IBRXF = array of RX procedures  ;JWS;IB\*2.0\*592;US131  ; IBDEN = Dental data for EDI output  ; IBDEN1 = array of Dental data for EDI outputdded NDC and Units to line level of claim.  ;I IBCPLINK'="" S $P(IBFLD(24,IBI),U,14,15)=$TR($P($G(^DGCR(399,IBIFN,"CP",IBCPLINK,1)),U,7,8),"-")  ; vd/Beginning of IB\*2\*577 - Added Unit/Basis of Measurment to line level of claim.  I IBCPLINK'="" S $P(IBFLD(24,IBI),U,14,16)=$TR($P($G(^DGCR(399,IBIFN,"CP",IBCPLINK,1)),U,7,8),"-")\_U\_$P($G(^DGCR(399,IBIFN,"CP",IBCPLINK,2)),U)  ; vd/End of IB\*2\*577  ;JWS;IB\*2.0\*592;US131  I $G(IBDEN)'="" S IBFLD(24,IBI,"DEN")=$G(IBDEN)  I $D(IBDEN1) M IBFLD(24,IBI,"DEN1")=IBDEN1  I $D(IBDEND) S IBFLD(24,IBI,"DEND")=$G(IBDEND)  ;end ;JWS;IB\*2.0\*592;US131  Q  ;  AUXOK(IBSS,IBSS1) ; Check all other flds are the same to combine procs  ; IBSS = subscript of IBCP to check for dups to combine - pass by ref  ; IBSS(IBSS,"AUX-X",n) = all the previously extracted line items for the  ; same set of basic data, but having different "AUX" data  ; IBSS1 = the "AUX" data of the current IBCP entry  ;  ; Returns entry # in IBSS array if match found, or 0 if no match  ; Set the IBSS "AUX-X" node for no match  N Z,Z0,Z1,XIEN  S Z=0 F S Z=$O(IBSS(IBSS,"AUX-X",Z)) Q:'Z I IBSS1=IBSS(IBSS,"AUX-X",Z) Q  ;JWS;IB\*2.0\*592;Dental fields to check for roll-up  S XIEN=$G(IBSS(IBSS,1))  I $D(IBCP(IBPO,"DEN"))!($D(IBCP(IBPO,"DEN1")))!($D(IBCP(IBPO,"DEND")))!($D(IBCP(XIEN,"DEN")))!($D(IBCP(XIEN,"DEN1")))!($D(IBCP(XIEN,"DEND"))) D  . I $G(IBCP(IBPO,"DEN"))'=$G(IBCP(XIEN,"DEN")) S Z=0 Q  . I $G(IBCP(IBPO,"DEND"))'=$G(IBCP(XIEN,"DEND")) S Z=0 Q  . S Z1=0 F S Z1=$O(IBCP(IBPO,"DEN1",Z1)) Q:'Z1 I $G(IBCP(IBPO,"DEN1",Z1,0))'=$G(IBCP(XIEN,"DEN1",Z1,0)) S Z=0 Q  I 'Z S Z0=+$O(IBSS(IBSS,"AUX-X",""),-1)+1,IBSS(IBSS,"AUX-X",Z0)=IBSS1  Q +Z  ;  PRC ; Extract procedure data for HCFA 1500  ; IBRC(IBSS) = #rev codes with same billing criteria (IBSS)  ; IBLINK('CP' ien,'RC' ien) = IBSS including modifiers,rx seq in pc 7,8  ; IBLINK1(IBSS, 'RC' ien) = auto (1)^ 'CP' ien (soft link)  ;  ; proc array w/chrg  ;JWS;IB\*2.0\*592;US131; added IBLN1, IBDENLN  ;IA# 3820  N IBPR,IBP,IBDENLN,IBLN1  S IBI=0 F S IBI=$O(^DGCR(399,IBIFN,"CP",IBI)) Q:'IBI S IBLN=^(IBI,0),IBLN1=$G(^(1)),IBAUXLN=$G(^("AUX")),IBDENLN=$G(^("DEN")) D  . I $O(^DGCR(399,IBIFN,"CP",IBI,"DEN1",0)) M IBDENLN("DEN1")=^DGCR(399,IBIFN,"CP",IBI,"DEN1")  . ;end ;JWS;IB\*2.0\*592;US131  . N Z,Z0,Z1,Q1  . S IBPDT=$P(IBLN,U,2)  . S IBSS=$$IBSS(IBI,.IBDXI,IBLN)  . S IBPO=$S($P(IBLN,U,4):+$P(IBLN,U,4),1:IBI+1000) ;Set print order  . S IBCP(IBPO)=IBPDT\_"^"\_IBSS,IBCP(IBPO,"AUX")=IBAUXLN  . S IBCP(IBPO,"LNK")=IBI  . ;JWS;IB\*2.0\*592;US131  . I $G(IBLN1)'="" S IBCP(IBPO,"DEND")=IBLN1  . I $G(IBDENLN)'="" S IBCP(IBPO,"DEN")=IBDENLN  . I $O(IBDENLN("DEN1",0)) M IBCP(IBPO,"DEN1")=IBDENLN("DEN1")  . ;end ;JWS;IB\*2.0\*592;US131  . ; Rx  . N IBZ,IBITEM  . S IBZ=$S($P(IBSS,U):$P(IBSS,U),1:"")  . I IBZ'="",$D(IBLINKRX(IBZ,IBI)) D Q:IBCHARG'=""  .. S IBPO1=IBPO  .. S IBITEM=+$O(IBLINKRX(IBZ,IBI,0)),IBRV=$G(IBLINKRX(IBZ,IBI,IBITEM))  .. Q:$S(IBRV="":1,1:'$G(IBRC(IBRV)))  .. S IBCHARG=$P(IBRV,U,6),IBRC(IBRV)=IBRC(IBRV)-1  .. S $P(IBCP(IBPO1),U,9)=IBCHARG,IBCP(IBPO1,"RX")=IBITEM K IBLINKRX(IBZ,IBI,IBITEM)  . ; find chrgs directly linked to proc  . S IBK=0 F S IBK=$O(IBLINK(IBI,IBK)) Q:'IBK S IBRV1=IBLINK(IBI,IBK),IBRV=$P(IBRV1,U,1,6) I +IBRC(IBRV1) D  .. S IBCHARG=$P(IBRV,U,6),IBRC(IBRV1)=IBRC(IBRV1)-1  .. I IBCHARG'="" S $P(IBSS,U,8)=IBCHARG,IBCP(IBPO)=IBPDT\_"^"\_IBSS,IBPO=IBPO+.1  ;  ; add chrgs associated with a proc (not a direct link)  ; find chrg associated with proc, if any (match proc,div,+/-basc)  K IBP(0)  F IBP=3,2 Q:$D(IBP(0)) S IBPO="" F S IBPO=$O(IBCP(IBPO)) Q:'IBPO I $P(IBCP(IBPO),U,9)="" D  . S IBSS=$P(IBCP(IBPO),U,2,9)  . S IBCHARG="",(IBRV,IBSS)=$P(IBSS,U,1,IBP) F S IBRV=$O(IBRC(IBRV)) Q:$P(IBRV,U,1,IBP)'=IBSS S IBP(0)=0 I +IBRC(IBRV) D Q  .. S IBCHARG=$P(IBRV,U,6),IBRC(IBRV)=IBRC(IBRV)-1  .. I IBRC(IBRV) S Z=0 F S Z=$O(IBCP(IBPO,Z)) Q:'Z S IBRC(IBRV)=IBRC(IBRV)-1  . S $P(IBCP(IBPO),U,9)=IBCHARG  . I IBCHARG'="" S Z=$O(IBLINK1(IBRV,0)) I Z S IBCP(IBPO,"L",Z)=IBLINK1(IBRV,Z) K IBLINK1(IBRV,Z)  ;  ; add chrgs not associated with a proc to first proc with no chrg  ; Aggggh!!! TP  S IBPO="" F S IBPO=$O(IBCP(IBPO)) Q:'IBPO I $P(IBCP(IBPO),U,9)="" D  . S IBCHARG="",IBRV="^" F S IBRV=$O(IBRC(IBRV)) Q:IBRV=""!+IBRV I +IBRC(IBRV) D Q  .. S IBCHARG=$P(IBRV,U,6),IBRC(IBRV)=IBRC(IBRV)-1  .. S Z=$O(IBLINK1(IBRV,0)) I Z S IBCP(IBPO,"L",Z)=IBLINK1(IBRV,Z) K IBLINK1(IBRV,Z)  . S $P(IBCP(IBPO),U,9)=IBCHARG  ;  Q  IBSS(IBI,IBDXI,IBLN) ; Creates index sequence for procedure  N IBPC,IBJ,IBSS,IBLPI,IBX,IBLPAR  S (IBPC,IBLPI)=0  F IBJ=1,6,5,0,9,10 S IBPC=IBPC+1 S:IBJ $P(IBSS,U,IBPC,IBPC+1)=($P(IBLN,U,IBJ)\_U)  S $P(IBSS,U,7)=($$GETMOD^IBEFUNC(IBIFN,IBI)\_U) ;Modifiers  ;IB\*547/TAZ - IBDXI not defined, use internal DX pointer  I '$G(IBNWPTCH) F IBJ=11:1:14 I $P(IBLN,U,IBJ) S $P(IBSS,U,4)=$P(IBSS,U,4)\_$S(IBJ>11:",",1:"")\_$G(IBDXI(+$P(IBLN,U,IBJ))) ; dx  I $G(IBNWPTCH) F IBJ=11:1:14 S IBX=$P(IBLN,U,IBJ) I IBX S $P(IBSS,U,4)=$P(IBSS,U,4)\_$S(IBJ>11:",",1:"")\_$G(IBDXI(IBX),IBX) ; dx  S $P(IBSS,U,10)=$P(IBLN,U,16),$P(IBSS,U,9)=$P(IBLN,U,19),$P(IBSS,U,11)=+$P(IBLN,U,17)  G:'$G(IBNWPTCH) IBSSX  ;IB\*547/TAZ - Add additional fields for roll-up compare  S $P(IBSS,U,21)=$$GET1^DIQ(399.0304,IBI\_","\_IBIFN\_",","ASSOCIATED CLINIC","I")  S $P(IBSS,U,22)=$$GET1^DIQ(399.0304,IBI\_","\_IBIFN\_",","TYPE OF SERVICE","I")  S $P(IBSS,U,23)=$$GET1^DIQ(399.0304,IBI\_","\_IBIFN\_",","ATTACHMENT CONTROL NUMBER","I")  S $P(IBSS,U,24)=$$GET1^DIQ(399.0304,IBI\_","\_IBIFN\_",","NDC","I")  S $P(IBSS,U,25)=$$GET1^DIQ(399.0304,IBI\_","\_IBIFN\_",","PROCEDURE DESCRIPTION","I")  S $P(IBSS,U,26)=$$GET1^DIQ(399.0304,IBI\_","\_IBIFN\_",","ADDITIONAL OB MINUTES","I")  ;Add Provider info in pieces 41-49  M IBLPAR=^DGCR(399,IBIFN,"CP",IBI,"LNPRV")  F S IBLPI=$O(IBLPAR(IBLPI)) Q:'IBLPI S IBX=IBLPAR(IBLPI,0),$P(IBSS,U,40+IBX)=$TR(IBX,"^","~")  K IBLPAR  IBSSX ;  Q IBSS  ;  IBNWPTCH(IBIFN,IBPATCH) ;  ;Checks the date the primary claim was 1st transmitted and returns 1 if the transmitted date is after the patch  ;referenced in variable IBPATCH was released. This allows the MRA/EOBs returning to roll up procedures the same  ;way as they went out. Otherwise the order changes and the MRA/EOB won't match up.  ;  N IBARY,IBIDT,IBPFN,IBEFN,IBBN,IBX,IBBDT  S IBX=0  I $$INSTALDT^XPDUTL(IBPATCH,.IBARY) D ;ICR 10141  . S IBX=1  . S IBIDT=$O(IBARY(""))  . ; Get Primary Bill Number. This will insure COB data is consistent across all bills.  . S IBPFN=$$GET1^DIQ(399,IBIFN\_",","PRIMARY BILL #","I") I 'IBPFN S IBPFN=IBIFN  . ; Find 1st Accepted Entry (A1, A2, or Z) of Primary Bill in EDI TRANSMIT BILL FILE (364) to determine Batch Number  . S (IBEFN,IBBN)=0 F S IBEFN=$O(^IBA(364,"B",IBPFN,IBEFN)) Q:'IBEFN D I IBBN Q  .. I ",A1,A2,Z,"'[(","\_$$GET1^DIQ(364,IBEFN\_",","TRANSMISSION STATUS","I")\_",") Q  .. S IBBN=$$GET1^DIQ(364,IBEFN\_",","BATCH NUMBER","I")  . ;Retrieve the date the batch was 1st sent. If IBBN="" IBBDT will be null  . S IBBDT=$$GET1^DIQ(364.1,$$GET1^DIQ(364,IBBN\_",","BATCH NUMBER","I")\_",","DATE FIRST SENT","I")  . I IBBDT,(IBBDT<IBIDT) S IBX=0  Q IBX | | | | | | | | | |
| **Modified Logic (Changes are highlighted in yellow)** | | | | | | | | | |
| IBCF23A ;ALB/ARH - HCFA 1500 19-90 DATA - Split from IBCF23 ;12-JUN-93  ;;2.0;INTEGRATED BILLING;\*\*51,432,516,547,577,592,608\*\*;21-MAR-94;Build 3  ;;Per VA Directive 6402, this routine should not be modified.  ;  ; $$INSTALDT^XPDUTL(IBPATCH,.IBARY) - ICR 10141  ;  B24 ; set individual entries in print array, external format  ; IBAUX = additional data for EDI output  ; IBRXF = array of RX procedures  ;JWS;IB\*2.0\*592;US131  ; IBDEN = Dental data for EDI output  ; IBDEN1 = array of Dental data for EDI outputdded NDC and Units to line level of claim.  ;I IBCPLINK'="" S $P(IBFLD(24,IBI),U,14,15)=$TR($P($G(^DGCR(399,IBIFN,"CP",IBCPLINK,1)),U,7,8),"-")  ; vd/Beginning of IB\*2\*577 - Added Unit/Basis of Measurment to line level of claim.  I IBCPLINK'="" S $P(IBFLD(24,IBI),U,14,16)=$TR($P($G(^DGCR(399,IBIFN,"CP",IBCPLINK,1)),U,7,8),"-")\_U\_$P($G(^DGCR(399,IBIFN,"CP",IBCPLINK,2)),U)  ; vd/End of IB\*2\*577  ;JWS;IB\*2.0\*592;US131  I $G(IBDEN)'="" S IBFLD(24,IBI,"DEN")=$G(IBDEN)  I $D(IBDEN1) M IBFLD(24,IBI,"DEN1")=IBDEN1  I $D(IBDEND) S IBFLD(24,IBI,"DEND")=$G(IBDEND)  ;end ;JWS;IB\*2.0\*592;US131  Q  ;  AUXOK(IBSS,IBSS1) ; Check all other flds are the same to combine procs  ; IBSS = subscript of IBCP to check for dups to combine - pass by ref  ; IBSS(IBSS,"AUX-X",n) = all the previously extracted line items for the  ; same set of basic data, but having different "AUX" data  ; IBSS1 = the "AUX" data of the current IBCP entry  ;  ; Returns entry # in IBSS array if match found, or 0 if no match  ; Set the IBSS "AUX-X" node for no match  N Z,Z0,Z1,XIEN  S Z=0 F S Z=$O(IBSS(IBSS,"AUX-X",Z)) Q:'Z I IBSS1=IBSS(IBSS,"AUX-X",Z) Q  ;JWS;IB\*2.0\*592;Dental fields to check for roll-up  S XIEN=$G(IBSS(IBSS,1))  I $D(IBCP(IBPO,"DEN"))!($D(IBCP(IBPO,"DEN1")))!($D(IBCP(IBPO,"DEND")))!($D(IBCP(XIEN,"DEN")))!($D(IBCP(XIEN,"DEN1")))!($D(IBCP(XIEN,"DEND"))) D  . I $G(IBCP(IBPO,"DEN"))'=$G(IBCP(XIEN,"DEN")) S Z=0 Q  . I $G(IBCP(IBPO,"DEND"))'=$G(IBCP(XIEN,"DEND")) S Z=0 Q  . S Z1=0 F S Z1=$O(IBCP(IBPO,"DEN1",Z1)) Q:'Z1 I $G(IBCP(IBPO,"DEN1",Z1,0))'=$G(IBCP(XIEN,"DEN1",Z1,0)) S Z=0 Q  I 'Z S Z0=+$O(IBSS(IBSS,"AUX-X",""),-1)+1,IBSS(IBSS,"AUX-X",Z0)=IBSS1  Q +Z  ;  PRC ; Extract procedure data for HCFA 1500  ; IBRC(IBSS) = #rev codes with same billing criteria (IBSS)  ; IBLINK('CP' ien,'RC' ien) = IBSS including modifiers,rx seq in pc 7,8  ; IBLINK1(IBSS, 'RC' ien) = auto (1)^ 'CP' ien (soft link)  ;  ; proc array w/chrg  ;JWS;IB\*2.0\*592;US131; added IBLN1, IBDENLN  N IBPR,IBP,IBDENLN,IBLN1  S IBI=0 F S IBI=$O(^DGCR(399,IBIFN,"CP",IBI)) Q:'IBI S IBLN=^(IBI,0),IBLN1=$G(^(1)),IBAUXLN=$G(^("AUX")),IBDENLN=$G(^("DEN")) D  . I $O(^DGCR(399,IBIFN,"CP",IBI,"DEN1",0)) M IBDENLN("DEN1")=^DGCR(399,IBIFN,"CP",IBI,"DEN1")  . ;end ;JWS;IB\*2.0\*592;US131  . N Z,Z0,Z1,Q1  . S IBPDT=$P(IBLN,U,2)  . S IBSS=$$IBSS(IBI,.IBDXI,IBLN)  . S IBPO=$S($P(IBLN,U,4):+$P(IBLN,U,4),1:IBI+1000) ;Set print order  . S IBCP(IBPO)=IBPDT\_"^"\_IBSS,IBCP(IBPO,"AUX")=IBAUXLN  . S IBCP(IBPO,"LNK")=IBI  . ;JWS;IB\*2.0\*592;US131  . I $G(IBLN1)'="" S IBCP(IBPO,"DEND")=IBLN1  . I $G(IBDENLN)'="" S IBCP(IBPO,"DEN")=IBDENLN  . I $O(IBDENLN("DEN1",0)) M IBCP(IBPO,"DEN1")=IBDENLN("DEN1")  . ;end ;JWS;IB\*2.0\*592;US131  . ; Rx  . N IBZ,IBITEM  . S IBZ=$S($P(IBSS,U):$P(IBSS,U),1:"")  . I IBZ'="",$D(IBLINKRX(IBZ,IBI)) D Q:IBCHARG'=""  .. S IBPO1=IBPO  .. S IBITEM=+$O(IBLINKRX(IBZ,IBI,0)),IBRV=$G(IBLINKRX(IBZ,IBI,IBITEM))  .. Q:$S(IBRV="":1,1:'$G(IBRC(IBRV)))  .. S IBCHARG=$P(IBRV,U,6),IBRC(IBRV)=IBRC(IBRV)-1  .. S $P(IBCP(IBPO1),U,9)=IBCHARG,IBCP(IBPO1,"RX")=IBITEM K IBLINKRX(IBZ,IBI,IBITEM)  . ; find chrgs directly linked to proc  . S IBK=0 F S IBK=$O(IBLINK(IBI,IBK)) Q:'IBK S IBRV1=IBLINK(IBI,IBK),IBRV=$P(IBRV1,U,1,6) I +IBRC(IBRV1) D  .. S IBCHARG=$P(IBRV,U,6),IBRC(IBRV1)=IBRC(IBRV1)-1  .. I IBCHARG'="" S $P(IBSS,U,8)=IBCHARG,IBCP(IBPO)=IBPDT\_"^"\_IBSS,IBPO=IBPO+.1  ;  ; add chrgs associated with a proc (not a direct link)  ; find chrg associated with proc, if any (match proc,div,+/-basc)  K IBP(0)  F IBP=3,2 Q:$D(IBP(0)) S IBPO="" F S IBPO=$O(IBCP(IBPO)) Q:'IBPO I $P(IBCP(IBPO),U,9)="" D  . S IBSS=$P(IBCP(IBPO),U,2,9)  . S IBCHARG="",(IBRV,IBSS)=$P(IBSS,U,1,IBP) F S IBRV=$O(IBRC(IBRV)) Q:$P(IBRV,U,1,IBP)'=IBSS S IBP(0)=0 I +IBRC(IBRV) D Q  .. S IBCHARG=$P(IBRV,U,6),IBRC(IBRV)=IBRC(IBRV)-1  .. I IBRC(IBRV) S Z=0 F S Z=$O(IBCP(IBPO,Z)) Q:'Z S IBRC(IBRV)=IBRC(IBRV)-1  . S $P(IBCP(IBPO),U,9)=IBCHARG  . I IBCHARG'="" S Z=$O(IBLINK1(IBRV,0)) I Z S IBCP(IBPO,"L",Z)=IBLINK1(IBRV,Z) K IBLINK1(IBRV,Z)  ;  ; add chrgs not associated with a proc to first proc with no chrg  ; Aggggh!!! TP  S IBPO="" F S IBPO=$O(IBCP(IBPO)) Q:'IBPO I $P(IBCP(IBPO),U,9)="" D  . S IBCHARG="",IBRV="^" F S IBRV=$O(IBRC(IBRV)) Q:IBRV=""!+IBRV I +IBRC(IBRV) D Q  .. S IBCHARG=$P(IBRV,U,6),IBRC(IBRV)=IBRC(IBRV)-1  .. S Z=$O(IBLINK1(IBRV,0)) I Z S IBCP(IBPO,"L",Z)=IBLINK1(IBRV,Z) K IBLINK1(IBRV,Z)  . S $P(IBCP(IBPO),U,9)=IBCHARG  ;  Q  IBSS(IBI,IBDXI,IBLN) ; Creates index sequence for procedure  N IBPC,IBJ,IBSS,IBLPI,IBX,IBLPAR  S (IBPC,IBLPI)=0  F IBJ=1,6,5,0,9,10 S IBPC=IBPC+1 S:IBJ $P(IBSS,U,IBPC,IBPC+1)=($P(IBLN,U,IBJ)\_U)  S $P(IBSS,U,7)=($$GETMOD^IBEFUNC(IBIFN,IBI)\_U) ;Modifiers  ;IB\*547/TAZ - IBDXI not defined, use internal DX pointer  I '$G(IBNWPTCH) F IBJ=11:1:14 I $P(IBLN,U,IBJ) S $P(IBSS,U,4)=$P(IBSS,U,4)\_$S(IBJ>11:",",1:"")\_$G(IBDXI(+$P(IBLN,U,IBJ))) ; dx  I $G(IBNWPTCH) F IBJ=11:1:14 S IBX=$P(IBLN,U,IBJ) I IBX S $P(IBSS,U,4)=$P(IBSS,U,4)\_$S(IBJ>11:",",1:"")\_$G(IBDXI(IBX),IBX) ; dx  S $P(IBSS,U,10)=$P(IBLN,U,16),$P(IBSS,U,9)=$P(IBLN,U,19),$P(IBSS,U,11)=+$P(IBLN,U,17)  G:'$G(IBNWPTCH) IBSSX  ;IB\*547/TAZ - Add additional fields for roll-up compare  S $P(IBSS,U,21)=$$GET1^DIQ(399.0304,IBI\_","\_IBIFN\_",","ASSOCIATED CLINIC","I")  S $P(IBSS,U,22)=$$GET1^DIQ(399.0304,IBI\_","\_IBIFN\_",","TYPE OF SERVICE","I")  S $P(IBSS,U,23)=$$GET1^DIQ(399.0304,IBI\_","\_IBIFN\_",","ATTACHMENT CONTROL NUMBER","I")  S $P(IBSS,U,24)=$$GET1^DIQ(399.0304,IBI\_","\_IBIFN\_",","NDC","I")  S $P(IBSS,U,25)=$$GET1^DIQ(399.0304,IBI\_","\_IBIFN\_",","PROCEDURE DESCRIPTION","I")  S $P(IBSS,U,26)=$$GET1^DIQ(399.0304,IBI\_","\_IBIFN\_",","ADDITIONAL OB MINUTES","I")  ;JRA;IB\*2.0\*608 Put Certificate of Medical Necessity (CMN) info in pieces 30,31,32  M IBLPAR=^DGCR(399,IBIFN,"CP",IBI)  S $P(IBSS,U,30)=$TR($G(IBLPAR("CMN")),U,"~")  S $P(IBSS,U,31)=$TR($G(IBLPAR("CMN-10126")),U,"~")  S $P(IBSS,U,32)=$TR($G(IBLPAR("CMN-484")),U,"~")  K IBLPAR  ;Add Provider info in pieces 41-49  M IBLPAR=^DGCR(399,IBIFN,"CP",IBI,"LNPRV")  F S IBLPI=$O(IBLPAR(IBLPI)) Q:'IBLPI S IBX=IBLPAR(IBLPI,0),$P(IBSS,U,40+IBX)=$TR(IBX,"^","~")  K IBLPAR  IBSSX ;  Q IBSS  ;  IBNWPTCH(IBIFN,IBPATCH) ;  ;Checks the date the primary claim was 1st transmitted and returns 1 if the transmitted date is after the patch  ;referenced in variable IBPATCH was released. This allows the MRA/EOBs returning to roll up procedures the same  ;way as they went out. Otherwise the order changes and the MRA/EOB won't match up.  ;  N IBARY,IBIDT,IBPFN,IBEFN,IBBN,IBX,IBBDT  S IBX=0  I $$INSTALDT^XPDUTL(IBPATCH,.IBARY) D ;ICR 10141  . S IBX=1  . S IBIDT=$O(IBARY(""))  . ; Get Primary Bill Number. This will insure COB data is consistent across all bills.  . S IBPFN=$$GET1^DIQ(399,IBIFN\_",","PRIMARY BILL #","I") I 'IBPFN S IBPFN=IBIFN  . ; Find 1st Accepted Entry (A1, A2, or Z) of Primary Bill in EDI TRANSMIT BILL FILE (364) to determine Batch Number  . S (IBEFN,IBBN)=0 F S IBEFN=$O(^IBA(364,"B",IBPFN,IBEFN)) Q:'IBEFN D I IBBN Q  .. I ",A1,A2,Z,"'[(","\_$$GET1^DIQ(364,IBEFN\_",","TRANSMISSION STATUS","I")\_",") Q  .. S IBBN=$$GET1^DIQ(364,IBEFN\_",","BATCH NUMBER","I")  . ;Retrieve the date the batch was 1st sent. If IBBN="" IBBDT will be null  . S IBBDT=$$GET1^DIQ(364.1,$$GET1^DIQ(364,IBBN\_",","BATCH NUMBER","I")\_",","DATE FIRST SENT","I")  . I IBBDT,(IBBDT<IBIDT) S IBX=0  Q IBX | | | | | | | | | |

**X) Modify cloning of a claim:**

**IBCCC2** – Copy CMN nodes to new claim when doing a cancel/copy

| **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;\*\*80,106,124,138,51,151,137,161,182,211,245,155,296,320,348,349,371,400,433,432,447,516,577,592\*\*;21-MAR-94;Build 25  ;;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; IA# 3820  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 flagon't copy value codes from inpatient inst to inpatient prof billsadd 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)  . ; WCJ;IB\*2.0\*577 - Added piece 1 (UNITS/BASIS OF MEASUREMENT) to 2-node.  . F K=1:1:1 S $P(^DGCR(399,IBIFN,I,J,2),"^",K)=$P(IBND("CP2"),"^",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($$MOD^ICPTMOD(+$P($G(^DGCR(399,IBIFN1,I,J,"MOD",K,0)),U,2),"I"),U,2)="TC" Q ; Don't copy TC modifier from inst to prof bill  ... S ^DGCR(399,IBIFN,I,J,"MOD",K,0)=^DGCR(399,IBIFN1,I,J,"MOD",K,0)  . ;JWS;IB\*2.0\*592;add new Dental claim form fieldsopy providers for cloned claim  N Z,Z0,CNT  S Z=$P($G(^DGCR(399,IBIFN,0)),U,19),Z0=$P($G(^DGCR(399,IBIFN1,0)),U,19),CNT=0  S IBDD=399.0222 F J=0:0 S J=$O(^DGCR(399,IBIFN1,I,J)) Q:'J I $D(^(J,0)) D  . I $$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 $P(^DGCR(399,IBIFN,"RC",IBCNT,0),U,16)=IBMAED(IBCNT)  Q | | | | | | | | | |
| **Modified Logic (Changes are highlighted in yellow)** | | | | | | | | | |
| IBCCC2 ;ALB/AAS - CANCEL AND CLONE A BILL - CONTINUED ;6/6/03 9:56am  ;;2.0;INTEGRATED BILLING;\*\*80,106,124,138,51,151,137,161,182,211,245,155,296,320,348,349,371,400,433,432,447,516,592,608\*\*;21-MAR-94;Build 40  ;;Per VA Directive 6402, this routine should not be modified.  ;  ;MAP TO DGCRCC2  ;  ;STEP 5 - get remainder of data to move and store in MCCR then x-ref  ;STEP 6 - go to screens, come out to IBB1 or something like that  ;  STEP5 S IBIFN1=$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 flagon't copy value codes from inpatient inst to inpatient prof billsadd 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($$MOD^ICPTMOD(+$P($G(^DGCR(399,IBIFN1,I,J,"MOD",K,0)),U,2),"I"),U,2)="TC" Q ; Don't copy TC modifier from inst to prof bill  ... S ^DGCR(399,IBIFN,I,J,"MOD",K,0)=^DGCR(399,IBIFN1,I,J,"MOD",K,0)  . ;JWS;IB\*2.0\*592;add new Dental claim form fields  . I $D(^DGCR(399,IBIFN1,I,J,"DEN")) S ^DGCR(399,IBIFN,I,J,"DEN")=^DGCR(399,IBIFN1,I,J,"DEN")  . I $D(^DGCR(399,IBIFN1,I,J,"DEN1",0)) S ^DGCR(399,IBIFN,I,J,"DEN1",0)=^DGCR(399,IBIFN1,I,J,"DEN1",0) D  .. S K=0 F S K=$O(^DGCR(399,IBIFN1,I,J,"DEN1",K)) Q:'K D  ... S ^DGCR(399,IBIFN,I,J,"DEN1",K,0)=^DGCR(399,IBIFN1,I,J,"DEN1",K,0)  . ;JRA;IB\*2.0\*608 Add CMN info - Node 'CMN-10126' contains data specific to only the CMS-10126 form, node 'CMN-484' contains data specific to  . ; only the CMN-484 form, and node 'CMN' contains data common to both formsopy providers for cloned claim  N Z,Z0,CNT  S Z=$P($G(^DGCR(399,IBIFN,0)),U,19),Z0=$P($G(^DGCR(399,IBIFN1,0)),U,19),CNT=0  S IBDD=399.0222 F J=0:0 S J=$O(^DGCR(399,IBIFN1,I,J)) Q:'J I $D(^(J,0)) D  . I $$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 | | | | | | | | | |

**IBJPS** – Main entry for IB Site Parameters which calls IBJPS8 to set up CMN CPT Inclusion

| **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;\*\*39,52,70,115,143,51,137,161,155,320,348,349,377,384,400,432,494,461,516,547,592\*\*;21-MAR-94;Build 40  ;;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  ; | | | | | | | | | |
| **Modified Logic (Changes are highlighted in yellow)** | | | | | | | | | |
| IBJPS ;ALB/MAF,ARH - IBSP IB SITE PARAMETER SCREEN ;22-DEC-1995  ;;2.0;INTEGRATED BILLING;\*\*39,52,70,115,143,51,137,161,155,320,348,349,377,384,400,432,494,461,516,547,592,608\*\*;21-MAR-94;Build 40  ;;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  . I IBSET=21 D EN^IBJPS8 Q ; WCJ;IB\*2.0\*608;US3;  . 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  ; | | | | | | | | | |

**IBJPS8** – Set up CMN CPT Inclusions in IB System Parameters & check before CMN prompt

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | **IBJPS8** | | | | | | | | |
| **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** | | | | | | | | | |
| IBJPS8 ;AITC/WCJ - IB Site Parameters, CMN CPT Inclusions CPT Codes ;02-Feb-2018  ;;2.0;INTEGRATED BILLING;\*\*608\*\*;21-MAR-94;Build 40  ;;Per VA Directive 6402, this routine should not be modified.  ;  EN ; -- main entry point for IBJP IB CMN CPTS  D EN^VALM("IBJPS CMN CPTS")  Q  ;  HDR ; -- header code  S VALMSG=""  Q  ;  INIT ; -- init variables and list array  N ERROR,IBCNT,IBLN,IBSTR  N CPTDATA,CIENS,CPTIEN,RTYDSC  ;  S (VALMCNT,IBCNT,IBLN)=0  I $D(^IBE(350.9,1,16,"B")) D  . S CPTIEN=0 F S CPTIEN=$O(^IBE(350.9,1,16,"B",CPTIEN)) Q:'CPTIEN D  . . ;  . . S CIENS=CPTIEN\_","  . . D GETS^DIQ(81,CIENS,".001;.01;2","I","CPTDATA","ERROR")  . . S IBCNT=IBCNT+1  . . S IBSTR=$$SETSTR^VALM1($J(IBCNT,4)\_".","",2,6)  . . S IBSTR=$$SETSTR^VALM1($G(CPTDATA(81,CIENS,.01,"I")),IBSTR,10,10)  . . S IBSTR=$$SETSTR^VALM1($G(CPTDATA(81,CIENS,2,"I")),IBSTR,25,30)  . . S IBLN=$$SET(IBLN,IBSTR)  . . ;S @VALMAR@("ZIDX",IBCNT,$G(CPTDATA(81,CIENS,.001,"I")))=""  . . S @VALMAR@("ZIDX",IBCNT,+CIENS)=""  . . Q  ;  I 'IBLN S IBLN=$$SET(IBLN,$$SETSTR^VALM1("No CMN CPTs defined.","",13,40))  ;  S VALMCNT=IBLN,VALMBG=1  Q  ;  HELP ; -- help code  S X="?" D DISP^XQORM1 W !!  Q  ;  EXIT ; -- exit code  D CLEAR^VALM1,CLEAN^VALM10  Q  ;  EXPND ; -- expand code  Q  ;  RTADD(IBTCFLAG) ; -- Add a new CPT Codes  N X,Y,DIE,DIR,DIRUT,DR,DTOUT,DUOUT,ERRMSG,FDA,RETIEN  ;  S VALMBCK="R"  D FULL^VALM1  D RTADD1  D INIT  Q  ;  RTADD1 ; Looping tag for Adding CPT Codes  K DA,DIE,DIR,DIRUT,DR,DTOUT,DUOUT,ERRMSG,FDA,RETIEN,X,Y  ;  S DIR(0)="350.916,.01"  S DIR("A")="CPT Code"  D ^DIR  Q:'+Y  ;  I $D(^IBE(350.9,1,16,"B",+Y)) D G RTADD1  . D FULL^VALM1  . W @IOF  . W !,"This CPT Code already exists on the Inclusion list."  . W !,"Please enter another CPT Code."  . Q  ;  S FDA(350.916,"+1,1,",.01)=+Y  D UPDATE^DIE("","FDA","RETIEN","ERRMSG")  G RTADD1  ;  RTDEL ; -- Delete a CPT Coode  N DR  D RTDEL1  S VALMBCK="R"  Q  ;  RTDEL1 ; Looping tag for deleting CPT Codes  N Z,VALMY  D FULL^VALM1  D EN^VALM2($G(XQORNOD(0)))  S Z=0  F S Z=$O(VALMY(Z)) Q:'Z D  . N DIK,IEN,RIEN  . S IEN=$O(@VALMAR@("ZIDX",Z,""))  . Q:IEN=""  . S RIEN=$O(^IBE(350.9,1,16,"B",IEN,""))  . I +RIEN S DIK="^IBE(350.9,1,16,",DA(1)=1,DA=RIEN D ^DIK  K @VALMAR  D INIT  Q  ;  SET(IBLN,IBSTR) ; -- Add a line to display list  ; returns line number added  S IBLN=IBLN+1 D SET^VALM10(IBLN,IBSTR,IBLN)  Q IBLN  ;  CMNPRMT(IBXIEN,IBPROCP,CPTIEN) ;JRA Determine if procedure requires prompting for CMN Info  ;Basically checks if CPTIEN is in the "CMN CPT Code Inclusion" list  ; Input: IBXIEN = Internal bill/claim number  ; IBPROCP = Procedure line subscript  ; CPTIEN = CPT code ien  ;  ; Output: 1 = Prompt user for CMN info  ; 0 = Don't prompt user for CMN info  ;  I '$G(IBXIEN)!('$G(IBPROCP)!('$G(CPTIEN))) Q 0  ;Prompt if the CPT is in IB Site Parameters "CMN CPT Code Inclusion" list -OR- if "CMN Required?" already set to "YES"  I $D(^IBE(350.9,1,16,"B",CPTIEN))>1!($$CMNDATA^IBCEF31(IBXIEN,IBPROCP,23,"I")) Q 1  Q 0  ; | | | | | | | | | |

**IBY608PR** - The new entries for files 364.5, 364.6, 364.7 and 350.8 are added

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | **IBY608PR** | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** |  | | | | | | | | |
| **Related Options** | None | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  |  | | | | |  | | | |
| **Data Dictionary (DD) References** |  | | | | | | | | |
| **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** | | | | | | | | | |
| IBY608PR ;EDE/JRA - Pre-Installation for IB patch 608 ; 10/12/17 2:12 pm  ;;2.0;INTEGRATED BILLING;\*\*608\*\*;21-MAR-94;Build 40  ;;Per VA Directive 6402, this routine should not be modified.  ;  ; delete all output formatter (O.F.) data elements included in build  D DELOF  Q  ;  INCLUDE(FILE,Y) ; function to determine if O.F. entry should be included in the build  ; FILE=5,6,7 indicating file 364.x or FILE=8 indicating file 350.8 (IB ERROR)  ; Y=ien to file  NEW OK,LN,TAG,DATA  S OK=0  F LN=2:1 S TAG="ENT"\_FILE\_"+"\_LN,DATA=$P($T(@TAG),";;",2) Q:DATA="" I $F(DATA,U\_Y\_U) S OK=1 Q  Q OK  ;  ;Delete edited entries to insure clean install of new entries  ;Delete obsolete entries.  DELOF ; Delete included OF entries  NEW FILE,DIK,LN,TAG,TAGLN,DATA,PCE,DA,Y  F FILE=5:1:8 S DIK=$S(FILE=8:"^IBE(350.",1:"^IBA(364.")\_FILE\_"," D  . F TAG="ENT"\_FILE,"DEL"\_FILE D  .. F LN=2:1 S TAGLN=TAG\_"+"\_LN,DATA=$P($T(@TAGLN),";;",2) Q:DATA="" D  ... F PCE=2:1 S DA=$P(DATA,U,PCE) Q:'DA D  .... I FILE=8,$D(^IBE(350.8,DA,0)) D ^DIK  .... Q:FILE=8  .... I $D(^IBA("364."\_FILE,DA,0)) D ^DIK  Q  ;  ; Example for ENT5, ENT6, ENT7, ENT8, DEL5, DEL6, and DEL7:  ;;^195^254^259^269^324^325^  ; Note: Must have beginning and ending up-carat  ;  ;-----------------------------------------------------------------------  ; 364.5 O.F. entries added:  ;  ; 225 N-COB CLAIM LEVEL AMOUNTS 'COB1-1.9' (US2486)  ; 226 N-MEDICARE INPT CLAIM COB AMTS 'MIA1-1.9' (US2486)  ; 227 N-MEDICARE OUTPT CLAIM COB AMT 'MOA1-1.9' (US2486)  ; 228 N-COB CLAIM LEVEL ADJUSTMENTS 'CCAS-1.9' (US2486)  ; 396 N-CMN RECORD ID 'LQ '  ; 438 N-CMN RECORD ID 'FRM '  ; 440 N-CMN RECORD ID 'CMN '  ; 442 N-CMN RECORD ID 'MEA '  ;  ENT5 ;O.F. entries in file 364.5 to be added  ;  ;;^225^226^227^228^396^438^440^442^  ;  ;-----------------------------------------------------------------------  ; 364.6 O.F. entries added:  ;  ; 2383 CMN RECORD ID 'LQ '  ; 2384 CMN FORM TYPE QUALIFIER  ; 2385 SERVICE LINE #  ; 2387 CMN INDUSTRY CODE  ; 2388 CMN CERTIFICATION TYPE  ; 2390 CMN CERTIFICATION TYPE QUAL  ; 2392 CMN MEASUREMENT REFERENCE ID CODE  ; 2393 CMN PATIENT WEIGHT (LBS)  ; 2394 CMN PATIENT WEIGHT MODIFIER  ; 2395 CMN MONTHS DME EQUIPMENT NEEDED  ; 2396 CMN DATE THERAPY STARTED  ; 2397 CMN DATE THERAPY STARTED QUALIFIER  ; 2398 CMN LAST CERTIFICATION DATE  ; 2399 CMN LAST CERTIFICATION DATE QUALIFIER  ; 2400 CMN RECERTIFICATION/REVISION DATE  ; 2401 CMN REPLACEMENT ITEM?  ; 2433 LQ DATA EXTRACT  ; 2436 FRM DATA EXTRACT  ; 2438 CMN RECORD ID 'FRM '  ; 2439 SERVICE LINE #  ; 2442 CMN QUESTION NUMBER/LETTER  ; 2443 CMN QUESTION RESPONSE Y/N  ; 2444 CMN QUESTION RESPONSE REF ID  ; 2445 CMN QUESTION RESPONSE DATE  ; 2446 CMN QUESTION RESPONSE % & DECIMAL  ; 2447 SERVICE LINE #  ; 2448 CMN DATA EXTRACT  ; 2449 CMN RECORD ID 'CMN '  ; 2451 CMN UNIT OR BASIS FOR MEASUREMENT CODE  ; 2452 CMN CERTIFICATION CONDITION INDICATOR  ; 2453 CMN CONDITION INDICATOR  ; 2454 CMN ATTACHMENT REPORT TYPE CODE  ; 2455 CMN ATTACHMENT TRANSMISSION CODE  ; 2456 CMN CODE CATEGORY  ; 2457 CMN RECORD ID 'MEA '  ; 2458 MEA DATA EXTRACT  ; 2461 SERVICE LINE #  ; 2462 CMN MEASUREMENT QUALIFIER  ; 2463 CMN TEST RESULTS  ;  ENT6 ;O.F. entries in file 364.6 to be added  ;  ;;^2383^2384^2385^2387^2388^2390^2392^2393^2394^2395^2396^2397^2398^  ;;^2399^2400^2401^2433^2436^2438^2439^2442^2443^2444^2445^2446^2447^  ;;^2448^2449^2451^2452^2453^2454^2455^2456^2457^2458^2461^2462^2463^  ;  ;-----------------------------------------------------------------------  ; 364.7 O.F. entries added:  ;  ; 105 VC1 VALUE CODE (837 Transaction) (PC 2) (US9)  ; 176 INS SERVICE LINE COUNTER (PC 2) (US9)  ; 178 INS SERVICE UNIT COUNT (PC 5) (US9)  ; 179 INS BLANK (PC 6) (US9)  ; 180 INS SERVICE LINE CHARGE AMT (PC 9) (US9)  ; 181 INS PROCEDURE MODIFIER (1) (PC 7) (US9)  ; 189 INS PROCEDURE CODE (PC 4) (US9)  ; 478 BGN N-RECORD ID (PC1) (US9)  ; 482 INS SERVICE LINE NON-COVERED CHARGE AMT (PC 12) (US9)  ; 805 INS UNITS/BASIS FOR MEASUREMENT CODE (PC 13) (US9)  ; 985 N-GET FROM PREVIOUS EXTRACT 'LCOB-1.9' - US2486  ; 1015 GEN-7  ; 1751 LDATE SERVICE LINE COUNTER (PC 2)  ; 1752 LDAT DATA EXTRACT (837 Transaction) (PC 1.9) (US9)  ; 1765 LDAT CLEANUP (837 Transaction) (PC 99.9) (US9)  ; 1969 CMN RECORD ID 'LQ '  ; 1970 SERVICE LINE #  ; 1971 CMN FORM TYPE QUALIFIER  ; 1973 CMN INDUSTRY CODE  ; 1974 CMN CERTIFICATION TYPE  ; 1975 CMN CERTIFICATION TYPE QUAL  ; 1977 CMN MEASUREMENT REFERENCE ID CODE  ; 1978 CMN PATIENT WEIGHT (LBS)  ; 1979 CMN PATIENT WEIGHT MODIFIER  ; 1980 CMN MONTHS DME EQUIPMENT NEEDED  ; 1981 CMN DATE THERAPY STARTED  ; 1982 CMN DATE THERAPY STARTED QUALIFIER  ; 1983 CMN LAST CERTIFICATION DATE  ; 1984 CMN LAST CERTIFICATION DATE QUALIFIER  ; 1985 CMN RECERTIFICATION/REVISION DATE  ; 1986 CMN REPLACEMENT ITEM?  ; 2018 LQ DATA EXTRACT  ; 2019 FRM DATA EXTRACT  ; 2020 CMN RECORD ID 'FRM '  ; 2021 CMN QUESTION NUMBER/LETTER  ; 2022 CMN QUESTION RESPONSE Y/N  ; 2023 CMN QUESTION RESPONSE REF ID  ; 2024 CMN QUESTION RESPONSE DATE  ; 2025 CMN QUESTION RESPONSE % & DECIMAL  ; 2026 SERVICE LINE #  ; 2027 SERVICE LINE #  ; 2028 CMN DATA EXTRACT  ; 2029 CMN RECORD ID 'CMN '  ; 2030 CMN UNIT OR BASIS FOR MEASUREMENT CODE  ; 2031 CMN CERTIFICATION CONDITION INDICATOR  ; 2032 CMN ATTACHMENT REPORT TYPE CODE  ; 2033 CMN ATTACHMENT TRANSMISSION CODE  ; 2034 CMN CODE CATEGORY  ; 2035 CMN CONDITION INDICATOR  ; 2038 CMN RECORD ID 'MEA '  ; 2039 MEA DATA EXTRACT  ; 2040 SERVICE LINE #  ; 2041 CMN MEASUREMENT QUALIFIER  ; 2042 CMN TEST RESULTS  ;  ENT7 ; O.F. entries in file 364.7 to be added  ;  ;;^105^176^178^179^180^181^189^478^482^805^985^1015^1751^1752^1765^  ;;^1969^1970^1971^1973^1974^1975^1977^1978^1979^1980^1981^  ;;^1982^1983^1984^1985^1986^2018^2019^2020^2021^2022^2023^  ;;^2024^2025^2026^2027^2028^2029^2030^2031^2032^2033^2034^  ;;^2035^2038^2039^2040^2041^2042^  ;  ;-----------------------------------------------------------------------  ; 350.8 O.F. entries added:  ;  ; 239 IB CMN NOT REQ BUT DATA (IB901)  ; 240 IB CMN FORM TYPE (IB902)  ; 241 IB CMN NO DATA NODE (IB903)  ; 243 IB CMN BAD DATA NODE (IB904)  ; 244 IB CMN CERT TYPE (IB905)  ; 246 IB CMN THERAPY DT (IB907)  ; 247 IB CMN LAST CERT DT (IB908)  ; 248 IB CMN RECERT/REVISION DT (IB909)  ; 259 IB CMN ABG SAT DT (IB912)  ; 271 IB CMN 4 LPM DATE (IB914)  ; 272 IB CMN ERRORS HEADER (IB915)  ; 273 IB CMN PEB (IB906)  ;  ENT8 ;O.F. entries in file 350.8 to be added  ;  ;;^239^240^241^243^244^246^247^248^259^271^272^273^  ;  ;-----------------------------------------------------------------------  ; 364.5 entries deleted:  ;  DEL5 ; remove O.F. entries in file 364.5 (not re-added)  ;  ;;  ;  ;-----------------------------------------------------------------------  ; 364.6 entries deleted:  ;  DEL6 ; remove O.F. entries in file 364.6 (not re-added)  ;  ;;  ;  ;-----------------------------------------------------------------------  ; 364.7 entries deleted:  ;  ;  DEL7 ; remove O.F. entries in file 364.7 (not re-added)  ;  ;;  ;  ;-----------------------------------------------------------------------  ; 350.8 Entries deleted:  ;  ; 238 IB CMN REQ  ;  DEL8 ; remove entries from 350.8 (IB ERROR)  ;  ;;  ; | | | | | | | | | |

**IBY608PO –** Add 50 CPT codes to the CMN CPT Code Inclusion list in IB System Parameters – if one of these codes in entered on a claim, the user will be prompted for CMN info.

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | **IBY608PO** | | | | | | | | |
| **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** | | | | | | | | | |
| IBY608PO ;ALB/KDM - POST-INSTALL FOR IB\*2.0\*608 ;13-DEC-2017  ;;2.0;INTEGRATED BILLING;\*\*608\*\*;21-MAR-94;Build 40  ;;Per VA Directive 6402, this routine should not be modified.  ;  ;KDM 12/2017 US1909  ; run report of all insurance companies that have the current setting for Transmit Electronically set to zero- which is NO  ; send email of report to eBiz rapid response group  N IBA,RNAME  S RNAME="IBY608PO"  K ^TMP(RNAME)  S IBA(2)="IB\*2\*608 Post-Install...",(IBA(1),IBA(3))=" " D MES^XPDUTL(.IBA) K IBA  D MES^XPDUTL(">> Running Insurance Company EDI Parameter Report...please stand by....")  D RPT  D MES^XPDUTL(">> Report Completed.")  D CMNCPT  D:$$PROD^XUPROD(1) EMAIL ;LIVE  D EMAIL ;TESTING  S IBA(2)="IB\*2\*608 Post-Install Complete.",(IBA(1),IBA(3))=" " D MES^XPDUTL(.IBA) K IBA  Q  ;  RPT ; Get all Insurance companies that have the 3.01- transmit electronically field blank or set to No.  ;N IBADDRESS,IBCITY,IBNAME,IBPIEN,IBSTATE,STATE,TRANSCD,TRANSMIT  N IBADDRESS,IBCITY,IBNAME,IBPIEN,IBSTATE,INACTFLG,STATE,TRANSMIT  S IBNAME=""  F S IBNAME=$O(^DIC(36,"B",IBNAME)) Q:IBNAME="" D  . S IBPIEN=0  . F S IBPIEN=$O(^DIC(36,"B",IBNAME,IBPIEN)) Q:'+IBPIEN D  . . S TRANSMIT=$$GET1^DIQ(36,IBPIEN,3.01,"I")  . . Q:+TRANSMIT ;Only want to report the insurance companies that have a setting of 0 or NULL  . . S (IBADDRESS,IBCITY,IBSTATE,INACTFLG,STATE)=""  . . S IBADDRESS=$$GET1^DIQ(36,IBPIEN,.111)  . . S IBCITY=$$GET1^DIQ(36,IBPIEN,.114)  . . S IBSTATE=$$GET1^DIQ(36,IBPIEN,.115,"I")  . . I +IBSTATE S STATE=$$GET1^DIQ(5,+IBSTATE,1)  . . S INACTFLG=$$GET1^DIQ(36,IBPIEN,.05)  . . I INACTFLG="" S INACTFLG=""  . . S ^TMP(RNAME,$J,IBNAME,IBPIEN)=IBADDRESS\_U\_IBCITY\_U\_STATE\_U\_INACTFLG\_U\_$S(TRANSMIT="":"",1:"NO")  Q  ;  EMAIL ; Send an email message to eBiz Rapid Response group with the report.  N ADDRESS,CITY,DATA,FULLADD,IBNAME,IBNAMEX,IBPIEN,INACTFLG,LN,MSG  N SPACES,SITE,SITENAME,SITENO,STATE,STATION,SUBJ,TOTAL,TRANS,TRANSCD,XMINSTR,XMTO  D BMES^XPDUTL(">> Sending Email...")  D MES^XPDUTL("-------------")  D MES^XPDUTL("Sending email notification to eBiz Rapid response group ... ")  ;S SPACES=$J(" ",100)  S $P(SPACES,"\_",100)="\_"  S SITE=$$SITE^VASITE,SITENAME=$P(SITE,U,2),SITENO=$P(SITE,U,1),STATION=$P(SITE,U,3)  S SUBJ="PATCH IB\*2.0\*608 - Insurance Company EDI Report"\_" for Station# "\_$P(SITE,U,3)\_" - "\_$P(SITE,U,2)  S SUBJ=$E(SUBJ,1,65)  S MSG(1)="PATCH IB\*2.0\*608 - Insurance Company EDI Parameter Report"  S MSG(2)=""  S MSG(3)="Site: "\_SITENO\_" "\_SITENAME\_" - Station "\_STATION  S MSG(4)="Domain: "\_$G(^XMB("NETNAME"))  S MSG(5)="Date/Time: "\_$$FMTE^XLFDT($$NOW^XLFDT)  S MSG(6)=""  S MSG(7)="INSURANCE COMPANY\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ADDRESS\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_INACTIVE\_\_\_\_EDI-TRANSMIT"  S MSG(8)="============================================================================================================================"  S MSG(9)=""  S LN=10,IBNAME="",TOTAL=0  F S IBNAME=$O(^TMP(RNAME,$J,IBNAME)) Q:IBNAME="" D  . S IBPIEN=""  . F S IBPIEN=$O(^TMP(RNAME,$J,IBNAME,IBPIEN)) Q:IBPIEN="" D  . . S DATA=^TMP(RNAME,$J,IBNAME,IBPIEN)  . . S IBNAMEX=$$UNSPACE($E(IBNAME,1,30))  . . S ADDRESS=$$UNSPACE($E($P(DATA,U,1),1,30)),CITY=$$UNSPACE($E($P(DATA,U,2),1,25)),STATE=$$UNSPACE($P(DATA,U,3))  . . S FULLADD=ADDRESS\_", "\_CITY\_", "\_STATE  . . I '$L(ADDRESS),'$L(CITY),'$L(STATE) S FULLADD=""  . . S INACTFLG=$P(DATA,U,4)  . . S TRANS=$P(DATA,U,5)  . . S LN=LN+1,MSG(LN)=IBNAMEX\_$E(SPACES,1,35-$L(IBNAMEX))\_FULLADD\_$E(SPACES,1,68-$L(FULLADD))  . . S MSG(LN)=MSG(LN)\_INACTFLG\_$E(SPACES,1,15-$L(INACTFLG))\_TRANS  . . S TOTAL=TOTAL+1  S LN=LN+1,MSG(LN)=""  S LN=LN+1,MSG(LN)="Total: "\_+TOTAL  S LN=LN+1,MSG(LN)=""  S LN=LN+1,MSG(LN)="End of Report"  ;  ; \*\*\*testing email to vito,anne,cj,jane vs live\*\*\* must change back to live before putting in build \*\*\*  ;S XMTO("vito.d'amico@va.gov")=""  ;S XMTO("anne.debacker@va.gov")=""  ;S XMTO("cherie.minch@va.gov")=""  ;S XMTO("jane.balchunas@va.gov")=""  ;S XMTO("william.jutzi@va.gov")=""  S XMTO("VHAeBillingRR@va.gov")=""  ;  S XMINSTR("FROM")="VistA-eBilling"  D SENDMSG^XMXAPI(DUZ,SUBJ,"MSG",.XMTO,.XMINSTR)  ;  EMAILX ;  D MES^XPDUTL(" Done.")  D CLEAN^DILF  Q  ;  UNSPACE(FLDX) ; Eliminate spaces at the end of the field.  N I  F S I=$L(FLDX) Q:($E(FLDX,I)'=" ") I $E(FLDX,I)=" " S FLDX=$E(FLDX,1,I-1)  Q FLDX  ;  CMNCPT ;Set CMN CPT CODES in IB System Parameters  D MES^XPDUTL("Setting CMN CPT Codes in IB SITE PARAMETER file.....")  N CODES,CPTCD,CPTIEN,CPTS,DA,DIC,DIE,DR,ERRMSG,FDA,I,RETIEN  S CODES=""  F I=1:1 S CPTS=$P($T(CPTCD+I),";;",2) Q:CPTS="" S CODES=$S(CODES="":CPTS,1:CODES\_U\_CPTS)  F I=1:1 S CPTCD=$P(CODES,U,I) Q:CPTCD="" D  . S CPTIEN=$$FIND1^DIC(81,,"X",CPTCD) Q:'CPTIEN  . I $D(^IBE(350.9,1,16,"B",CPTIEN)) Q  . K FDA,ERRMSG,RETIEN  . S FDA(350.916,"+1,1,",.01)=CPTIEN  . D UPDATE^DIE("","FDA","RETIEN","ERRMSG")  D MES^XPDUTL(".....CMN CPT Codes set. ")  Q  ;  CPTCD ;  ;;B4102^B4103^B4104^B4149^B4150^B4152^B4153^B4154^B4155^B4157^B4158^B4159^B4160^B4161^B4162^B4164^B4168  ;;B4172^B4176^B4178^B4180^B4185^B4189^B4193^B4197^B4199^B4216^B5000^B5100^B5200^B9002^B9004^B9006^E0424  ;;E0431^E0433^E0434^E0439^E0441^E0442^E0443^E0444^E0776^E0791^E1390^E1391^E1392^E1405^E1406^K0738  ; | | | | | | | | | |