**MCCF EDI TAS USEB-1**

System Design Document

IB\*2.0\*576



Department of Veterans Affairs

**November 2016**

Version 1.0

**User Story Number:** USEB-1

**User Story Name:** Payer Zip Code Field Adjustment

**Design/Resolution:**

The following fields need to be modified to satisfy the enhancement request described by the aforementioned User Story. Modifications highlighted in yellow:

* Zip Code field [#36, .116]
* Claims (INPT) Process Zip [#36, .126]
* Appeals Address Zip [#36, .146]
* Inquiry Address Zip Code [#36, .156]
* Claims (OPT) Process Zip [#36, .166]
* Claims (RX) Zip [#36, .186]

**Before:**

**36,.116** ZIP CODE .11;6 FREE TEXT

INPUT TRANSFORM: K:$L(X)>20!($L(X)<5) X I $D(X) D ZIPIN^VAFADDR

OUTPUT TRANSFORM: D ZIPOUT^VAFADDR

LAST EDITED: FEB 26, 1993

HELP-PROMPT: Answer with either 5 digit or 9 digit zip code.

DESCRIPTION: Enter the zip code of the mailing address for this insurance carrier. Answer with either the 5 digit zip code (format 12345) or with the 9 digit zip code (in format 123456789 or 12345-6789).

DELETE TEST: 1,0)= I $D(DGINS)

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

**After:**

**36,.116** ZIP CODE .11;6 FREE TEXT

INPUT TRANSFORM: K:($L(X)>10!($L(X)<9)) X I $D(X) K:'$$ZIPCHK9^IBCNSU(X) X I $D(X) D ZIPIN^VAFADDR

OUTPUT TRANSFORM: NOT EXECUTABLE!! -- SPECIFIER NEEDS AN "O"!

LAST EDITED: MAR 08, 2017

HELP-PROMPT: Answer must be nine (999999999) or ten characters (99999-9999) in length. The last 4 cannot be '0000' or '9999'.

DESCRIPTION: This is the ZIP code for the payer's main mailing address. Enter a 9 or 10 character ZIP code.

DELETE TEST: 1,0)= I $D(DGINS)

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

**Before:**

**36,.126** CLAIMS (INPT) PROCESS ZIP .12;6 FREE TEXT

INPUT TRANSFORM: K:$L(X)>20!($L(X)<5) X I $D(X) D ZIPIN^VAFADDR

OUTPUT TRANSFORM: D ZIPOUT^VAFADDR

LAST EDITED: JUL 15, 1993

HELP-PROMPT: If the inpatient claims process address of this company is different from its main address, enter zip code of the inpatient claims process address. Answer with either 5 digit or 9 digit zip code.

DESCRIPTION: Answer with either the 5 digit zip code (format 12345) or with the 9 digit zip code (in format 123456789 or 12345-6789).

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

**After:**

**36,.126** CLAIMS (INPT) PROCESS ZIP .12;6 FREE TEXT

INPUT TRANSFORM: K:($L(X)>10!($L(X)<9)) X I $D(X) K:'$$ZIPCHK9^IBCNSU(X) X I $D(X) D ZIPIN^VAFADDR

OUTPUT TRANSFORM: NOT EXECUTABLE!! -- SPECIFIER NEEDS AN "O"!

LAST EDITED: MAR 08, 2017

HELP-PROMPT: Answer must be nine (999999999) or ten characters (99999-9999) in length. The last 4 cannot be '0000' or '9999'.

DESCRIPTION: This is the ZIP code for the address of the inpatient claims processing location when it differs from the payer's main mailing address. Enter a 9 or 10 character ZIP code.

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

**Before:**

**36,.146** APPEALS ADDRESS ZIP .14;6 FREE TEXT

INPUT TRANSFORM: K:$L(X)>20!($L(X)<5) X I $D(X) D ZIPIN^VAFADDR

OUTPUT TRANSFORM: D ZIPOUT^VAFADDR

LAST EDITED: FEB 26, 1993

HELP-PROMPT: If the appeals address of this company is different from its main address, enter zip code of the appeals address. Answer with either 5 digit or 9 digit zip code.

DESCRIPTION: Answer with either the 5 digit zip code (format 12345) or with the 9 digit zip code (in format 123456789 or 12345-6789).

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

**After:**

**36,.146** APPEALS ADDRESS ZIP .14;6 FREE TEXT

INPUT TRANSFORM: K:($L(X)>10!($L(X)<9)) X I $D(X) K:'$$ZIPCHK9^IBCNSU(X) X I $D(X) D ZIPIN^VAFADDR

OUTPUT TRANSFORM: NOT EXECUTABLE!! -- SPECIFIER NEEDS AN "O"!

LAST EDITED: MAR 08, 2017

HELP-PROMPT: Answer must be nine (999999999) or ten characters (99999-9999) in length. The last 4 cannot be '0000' or '9999'.

DESCRIPTION: This is the ZIP code for the address of the appeals processing location when it differs from the payer's main mailing address. Enter a 9 or 10 character ZIP code.

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

**Before:**

**36,.156** INQUIRY ADDRESS ZIP CODE .15;6 FREE TEXT

INPUT TRANSFORM: K:$L(X)>20!($L(X)<5) X I $D(X) D ZIPIN^VAFADDR

OUTPUT TRANSFORM: D ZIPOUT^VAFADDR

LAST EDITED: FEB 26, 1993

HELP-PROMPT: If the inquiry address of this company is different from its main address, enter zip code of the inquiry address. Answer with either the 5 digit or 9 digit zip code.

DESCRIPTION: Answer with either the 5 digit zip code (format 12345) or with the 9 digit zip code (in format 123456789 or 12345-6789).

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

**After:**

**36,.156** INQUIRY ADDRESS ZIP CODE .15;6 FREE TEXT

INPUT TRANSFORM: K:($L(X)>10!($L(X)<9)) X I $D(X) K:'$$ZIPCHK9^IBCNSU(X) X I $D(X) D ZIPIN^VAFADDR

OUTPUT TRANSFORM: NOT EXECUTABLE!! -- SPECIFIER NEEDS AN "O"!

LAST EDITED: MAR 08, 2017

HELP-PROMPT: Answer must be nine (999999999) or ten characters (99999-9999) in length. The last 4 cannot be '0000' or '9999'.

DESCRIPTION: This is the ZIP code for the address of the inquiry processing location when it differs from the payer's main mailing address. Enter a 9 or 10 character ZIP code.

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

**Before:**

**36,.166** CLAIMS (OPT) PROCESS ZIP .16;6 FREE TEXT

INPUT TRANSFORM: K:$L(X)>20!($L(X)<5) X I $D(X) D ZIPIN^VAFADDR

OUTPUT TRANSFORM: D ZIPOUT^VAFADDR

LAST EDITED: AUG 16, 1993

HELP-PROMPT: If the outpatient claims process address of this company is different from its main address, enter zip code of the outpatient claims process address. Answer with either 5 digit or 9 digit zip code.

DESCRIPTION: Answer with either the 5 digit zip code (format 12345) or with the 9 digit zip code (in format 123456789 or 12345-6789).

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

**After:**

**36,.166** CLAIMS (OPT) PROCESS ZIP .16;6 FREE TEXT

INPUT TRANSFORM: K:($L(X)>10!($L(X)<9)) X I $D(X) K:'$$ZIPCHK9^IBCNSU(X) X I $D(X) D ZIPIN^VAFADDR

OUTPUT TRANSFORM: NOT EXECUTABLE!! -- SPECIFIER NEEDS AN "O"!

LAST EDITED: MAR 08, 2017

HELP-PROMPT: Answer must be nine (999999999) or ten characters (99999-9999) in length. The last 4 cannot be '0000' or '9999'.

DESCRIPTION: This is the ZIP code for the address of the outpatient claims processing location when it differs from the payer's main mailing address. Enter a 9 or 10 character ZIP code.

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

**Before:**

**36,.186** CLAIMS (RX) ZIP .18;6 FREE TEXT

INPUT TRANSFORM: K:$L(X)>20!($L(X)<5) X I $D(X) D ZIPIN^VAFADDR

OUTPUT TRANSFORM: D ZIPOUT^VAFADDR

LAST EDITED: JAN 04, 1994

HELP-PROMPT: If the prescription claims address of this company is different from its main address, enter zip code of the prescription claims address. Answer with either 5 digit or 9 digit zip code.

DESCRIPTION: Answer with either the 5 digit zip code (format 12345) or with the 9 digit zip code (in format 123456789 or 12345-6789).

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

**After:**

**36,.186** CLAIMS (RX) ZIP .18;6 FREE TEXT

INPUT TRANSFORM: K:($L(X)>10!($L(X)<9)) X I $D(X) K:'$$ZIPCHK9^IBCNSU(X) X I $D(X) D ZIPIN^VAFADDR

OUTPUT TRANSFORM: NOT EXECUTABLE!! -- SPECIFIER NEEDS AN "O"!

LAST EDITED: MAR 08, 2017

HELP-PROMPT: Answer must be nine (999999999) or ten characters (99999-9999) in length. The last 4 cannot be '0000' or '9999'.

DESCRIPTION: This is the ZIP code for the address of the prescription claims processing location when it differs from the payer's main mailing address. Enter a 9 or 10 character ZIP code.

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

* For each of the INPUT TRANSFORM elements in each of the above fields, the highlighted section of code will be added to make a call to a new separate “multi-useable” common subroutine (called ZIPCHK9) in the routine ^IBCNSU. The routine that is currently being referenced by the INPUT TRANSFORM elements for proper Zip Code validation is ^VAFADDR, for both acceptance and output display. The new module of code (ZIPCK9^IBCNSU) will properly validate the ZIP CODE for 9 digits prior to the INPUT TRANSFORM call to the module ZIPIN^VAFADDR. The new module of code (ZIPCK9^IBCNSU) will also validate to make sure the 9 digits do not end in either 4 Zeros (“0000”) or 4 Nines (“9999”).
* The HELP-PROMPT and DESCRIPTION elements in each of the above fields, are to be reworded/updated to reflect the 9-digit zip code requirement and also to inform of the last 4 digits not being allowed to be “0000” or “9999”.

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCNSU | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** |  | | | | | | | | |
| **Related Options** | None | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  |  | | | | |  | | | |
| **Data Dictionary (DD) References** | None | | | | | | | | |
| **Related Protocols** | None | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Current Logic** | | | | | | | | | |
| IBCNSU ;ALB/AAS - INSURANCE UTILITY ROUTINE ;19-MAY-93  ;;2.0;INTEGRATED BILLING;\*\*28,103,371\*\*; 21-MAR-94;Build 57  ;;Per VHA Directive 2004-038, this routine should not be modified.  ;  AB(IBCPOL,IBYR,IBASK) ; -- Return entry in Annual Benefits file  ; Input: IBCPOL = pointer to health insurance policy file  ; IBYR = fileman internal date, Default = dt  ; IBASK = 1 if want to ask okay to add new entry  ;  ; Output: IBCAB = pointer to Annual Benefits file if added, else null  ;  N DIR,IBCAB  S IBCAB=""  I $G(IBCPOL)="" G ABQ  I $G(IBYR)="" S IBYR=DT  ;S IBYR=$E(IBYR,1,3)\_"0000"  ;  ; -- try to find entry for policy for year  S IBCAB=$O(^IBA(355.4,"APY",IBCPOL,-IBYR,0))  ;  ; -- if no match add new entry  I 'IBCAB D  .I $G(IBASK) S DIR(0)="Y",DIR("A")="Are you adding a new Annual Benefits YEAR",DIR("B")="YES" D ^DIR I $D(DIRUT)!(Y<1) S VALMQUIT="" Q  .S IBCAB=$$ADDB(IBCPOL,IBYR)  .Q  ABQ Q IBCAB  ;  ADDB(IBCPOL,IBYR) ; -- add entries to Annual Benefits file  ; Input: IBCPOL = pointer to health insurance policy file  ; IBYR = fileman internal date, Default = dt  ;  ; Output: IBCAB = pointer to Annual Benefits file if added, else null  ;  N %DT,IBN1,IBCAB,DIC,DIE,DR,DA,DLAYGO,DO,DD  S IBCAB=""  I $G(IBCPOL)="" G ADDBQ  I $G(IBYR)="" S IBYR=DT  K DD,DO,DIC,DR S DIC="^IBA(355.4,",DIC(0)="L",DLAYGO=355.4  ;  ;S X=$E(IBYR,1,3)\_"0000"  S X=IBYR D FILE^DICN I +Y<0 G ADDBQ  S (IBCAB,DA)=+Y,DIE="^IBA(355.4,",DR=".02////"\_IBCPOL  D ^DIE K DIC,DIE,DA,DR  ADDBQ Q IBCAB  ;  CHIP(IBCDFND) ; -- convert node with no hip pointer to one with hip pointer  ; Input: IBCDFND = zeroth node of insurance type multiple  ; = ^dpt(dfn,.312,ibcdfn,0)  ;  ; Output: IBCPOL = pointer to policy file  ;  N IBCNS,IBGRP,IBGRNA,IBGRNU  S IBCNS=+IBCDFND,IBGRNA=$P(IBCDFND,"^",15),IBGRNU=$P(IBCDFND,"^",3),IBGRP=0  I IBGRNA'=""!(IBGRNU'="") S IBGRP=1  S IBCPOL=$$HIP(IBCNS,IBGRP,IBGRNA,IBGRNU)  CHIPQ Q IBCPOL  ;  HIP(IBCNS,IBGRP,IBGRNA,IBGRNU) ; -- find internal entry number in policy file  ; Input: IBCNS = pointer to ins co file  ; IBGRP = 1 if group policy, 0 if not  ; IBGRNA = group name  ; IBGRNU = group number  ;  ; Output: IBCPOL = pointer to policy file  ;  N %DT  S IBCPOL=""  I $G(^DIC(36,+$G(IBCNS),0))="" G HIPQ  S IBGRP=+$G(IBGRP) ; if undefine, is not a group policy  I 'IBGRP S IBCPOL=$$ADDH(IBCNS,IBGRP) G HIPQ  ;  S:$G(IBGRNU)="" IBGRNU="IB ZZZZZ"  I IBGRNU'="IB ZZZZZ" S IBCPOL=$O(^IBA(355.3,"AGNU",IBCNS,IBGRNU,0))  I IBCPOL,$P($G(^IBA(355.3,+IBCPOL,0)),"^",3)=IBGRNA G HIPQ ; match both  ;  S:$G(IBGRNA)="" IBGRNA="IB ZZZZZ"  S IBCPOL=$O(^IBA(355.3,"AGNA",IBCNS,IBGRNA,0))  I IBCPOL,$P($G(^IBA(355.3,+IBCPOL,0)),"^",4)=IBGRNU G HIPQ ; match both  ;  I 'IBCPOL S IBCPOL=$$ADDH(IBCNS,IBGRP) D  .I IBGRNA="",IBGRNU="" Q  .S:IBGRNA="IB ZZZZZ" IBGRNA="" S:IBGRNU="IB ZZZZZ" IBGRNU=""  .S DA=IBCPOL,DIE="^IBA(355.3,",DR=".03////"\_$$STRIP(IBGRNA,";")\_";.04////"\_$$STRIP(IBGRNU,";")  .D ^DIE K DA,DR,DIC,DIE  HIPQ Q IBCPOL  ;  ADDH(IBCNS,IBGRP,IBGNA,IBGNU) ; -- add entries to health insurance policy file (355.3)  ; Input: IBCNS = pointer to ins co file  ; IBGRP = 1 if group policy, 0 if no  ;  ; Output: IBCPOL = pointer to policy file, if added else null  ;  N %DT,IBN1,IBCAB,DIC,DIE,DR,DA,DLAYGO,DO,DD  S IBCPOL=""  I $G(IBCNS)="" G ADDHQ  K DD,DO,DIC,DR S DIC="^IBA(355.3,",DIC(0)="L",DLAYGO=355.3  ;  S X=IBCNS D FILE^DICN I +Y<0 G ADDHQ  S (DA,IBCPOL)=+Y,DIE="^IBA(355.3,",DR=".02////"\_+$G(IBGRP)  I IBGRP=0,$G(DFN) S DR=DR\_";.1////"\_DFN  I $D(IBGNU) S DR=DR\_";.04///^S X=IBGNU"  I $D(IBGNA) S DR=DR\_";.03///^S X=IBGNA"  D ^DIE K DA,DR,DIE,DIC  I $G(IBCNTP)'="" S IBCNTP=IBCNTP+1  ADDHQ Q IBCPOL  ;  ODELP(DFN,INS) ; -- can an insurance policy be deleted  ; -- called by ^dd(2.312,0,"del",.01) and by ibcnsm  ; -- input dfn: ien of patient in file 2.  ; ins: ien of ins. co in file 36  ;  ; -- output 1 if no deletion allowed  ; 0 if deletion allowed  N I,X,Y S X=0  ;  ; -- do not delete if any uncancelled bills  S J=0 F S J=$O(^DGCR(399,"AE",DFN,INS,J)) Q:'J I $P(^DGCR(399,J,"S"),"^",17)="" S X=1 Q  ODELPQ Q X  ;  STRIP(X,X1) ; -- strip characters from string  ; input: x = string  ; x1 = character to strip (default is ";"  N I,X2  S X2="" S:$G(X1)="" X1=";"  S X1=$E(X1)  F I=1:1 S X2=X2\_$P(X,X1,I) Q:($P(X,X1,I+1,999)'[X1)  Q X2  ;  ;  DELP(DFN,INS,IBC) ; -- can an insurance policy be deleted  ; -- called by ^dd(2.312,0,"del",.01) and by ibcnsm  ; -- input dfn: ien of patient in file 2.  ; ins: ien of ins. co in file 36  ; ibc: ien of policy in file 2.312 to do a match  ;  ; -- output 1 if no deletion allowed  ; 0 if deletion allowed  ;  N ARR,J,ONEPOL,X  ;  ; - check input  I '$G(DFN)!'$G(INS) S X=1 G DELPQ  ;  ; - see if vet has more than one policy with carrier; set flag  ; - also, if no policy is passed, assume the patient has one policy  I $G(IBC) D  .S J=0 F S J=$O(^DPT("AB",IBC,DFN,J)) Q:'J S ARR(J)=$G(^DPT(DFN,.312,J,0))  .S (J,ONEPOL)=0 S J=$O(ARR(J)) I J,'$O(ARR(J)) S ONEPOL=1  E S ONEPOL=1  ;  ;  ; -- do not delete if any uncancelled bills  S (J,X)=0 F S J=$O(^DGCR(399,"AE",DFN,INS,J)) Q:'J D Q:X  .;  .N ARRP,POL,K,L,M,MP,S,Z  .S Z=$G(^DGCR(399,J,0)),M=$G(^("M")),MP=$G(^("MP")),S=$G(^("S"))  .;  .; - skip cancelled bills  .I $P(S,"^",17)'="" Q  .;  .; - set flag if the patient has just one policy with the company  .I ONEPOL S X=1 Q  .;  .; - if there are no policy pointers in the claim,  .I '$P(M,"^",12),'$P(M,"^",13),'$P(M,"^",14),'$P(MP,"^",2) D Q  ..;  ..; - find all policies effective on the event date  ..S K=0 F S K=$O(ARR(K)) Q:'K S POL=ARR(K) D  ...I $P(POL,"^",8) Q:$P(Z,"^",3)<$P(POL,"^",8)  ...I $P(POL,"^",4) Q:$P(Z,"^",3)>$P(POL,"^",4)  ...S ARRP(K)=""  ..;  ..; - if there are two such policies, trust user judgement and assume  ..; - policy is not related to this claim.  ..S L=$O(ARRP(0)) I L,$O(ARR(L)) Q  ..;  ..; - if there is just one policy, and it is the same as the one  ..; - passed in, do not allow deletion.  ..I L=IBC S X=1  .;  .; - if one of the claim policy pointers is the same as the policy  .; - passed in, do not allow deletion.  .I $P(MP,"^",2)=IBC S X=1 Q  .I $P(M,"^",12)=IBC!($P(M,"^",13)=IBC)!($P(M,"^",14)=IBC) S X=1  ;  ;  DELPQ Q X  ;  DUPADDRL(DATA,IBCNS,FLD1,FLD2) ; Insurance address lines can not be duplicated  ; DATA - Value being compared  ; FLD1 - First field to check against  ; FLD2 - Second field to check against (OPTIONAL)  ;  ; Returns 1 if this field is a duplicate of another field.  ;  N Z1,Z2  Q:$G(DATA)="" 0 ; should not happen because this is invoked as an input transform  Q:'$G(IBCNS) 1 ; stop from editing through fileman  S DATA=$$UP^XLFSTR($G(DATA)),DATA=$$TRIM^XLFSTR(DATA)  S Z1=$$GET1^DIQ(36,+$G(IBCNS),+$G(FLD1),"I")  S Z1=$$UP^XLFSTR(Z1),Z1=$$TRIM^XLFSTR(Z1)  S Z2=$$GET1^DIQ(36,+$G(IBCNS),+$G(FLD2),"I")  S Z2=$$UP^XLFSTR(Z2),Z2=$$TRIM^XLFSTR(Z2)  I DATA=Z1 D CLEAN^DILF Q 1  I DATA=Z2 D CLEAN^DILF Q 1  D CLEAN^DILF  Q 0  ; | | | | | | | | | |
| **Modified Logic (Changes are highlighted in yellow)** | | | | | | | | | |
| IBCNSU ;ALB/AAS - INSURANCE UTILITY ROUTINE ;19-MAY-93  ;;2.0;INTEGRATED BILLING;\*\*28,103,371,576\*\*; 21-MAR-94;Build 57  ;;Per VHA Directive 2004-038, this routine should not be modified.  ;  ;AITC/JRA - IB\*2.0\*576 Added ZIPCHK9 function.  ;  AB(IBCPOL,IBYR,IBASK) ; -- Return entry in Annual Benefits file  ; Input: IBCPOL = pointer to health insurance policy file  ; IBYR = fileman internal date, Default = dt  ; IBASK = 1 if want to ask okay to add new entry  ;  ; Output: IBCAB = pointer to Annual Benefits file if added, else null  ;  N DIR,IBCAB  S IBCAB=""  I $G(IBCPOL)="" G ABQ  I $G(IBYR)="" S IBYR=DT  ;S IBYR=$E(IBYR,1,3)\_"0000"  ;  ; -- try to find entry for policy for year  S IBCAB=$O(^IBA(355.4,"APY",IBCPOL,-IBYR,0))  ;  ; -- if no match add new entry  I 'IBCAB D  .I $G(IBASK) S DIR(0)="Y",DIR("A")="Are you adding a new Annual Benefits YEAR",DIR("B")="YES" D ^DIR I $D(DIRUT)!(Y<1) S VALMQUIT="" Q  .S IBCAB=$$ADDB(IBCPOL,IBYR)  .Q  ABQ Q IBCAB  ;  ADDB(IBCPOL,IBYR) ; -- add entries to Annual Benefits file  ; Input: IBCPOL = pointer to health insurance policy file  ; IBYR = fileman internal date, Default = dt  ;  ; Output: IBCAB = pointer to Annual Benefits file if added, else null  ;  N %DT,IBN1,IBCAB,DIC,DIE,DR,DA,DLAYGO,DO,DD  S IBCAB=""  I $G(IBCPOL)="" G ADDBQ  I $G(IBYR)="" S IBYR=DT  K DD,DO,DIC,DR S DIC="^IBA(355.4,",DIC(0)="L",DLAYGO=355.4  ;  ;S X=$E(IBYR,1,3)\_"0000"  S X=IBYR D FILE^DICN I +Y<0 G ADDBQ  S (IBCAB,DA)=+Y,DIE="^IBA(355.4,",DR=".02////"\_IBCPOL  D ^DIE K DIC,DIE,DA,DR  ADDBQ Q IBCAB  ;  CHIP(IBCDFND) ; -- convert node with no hip pointer to one with hip pointer  ; Input: IBCDFND = zeroth node of insurance type multiple  ; = ^dpt(dfn,.312,ibcdfn,0)  ;  ; Output: IBCPOL = pointer to policy file  ;  N IBCNS,IBGRP,IBGRNA,IBGRNU  S IBCNS=+IBCDFND,IBGRNA=$P(IBCDFND,"^",15),IBGRNU=$P(IBCDFND,"^",3),IBGRP=0  I IBGRNA'=""!(IBGRNU'="") S IBGRP=1  S IBCPOL=$$HIP(IBCNS,IBGRP,IBGRNA,IBGRNU)  CHIPQ Q IBCPOL  ;  HIP(IBCNS,IBGRP,IBGRNA,IBGRNU) ; -- find internal entry number in policy file  ; Input: IBCNS = pointer to ins co file  ; IBGRP = 1 if group policy, 0 if not  ; IBGRNA = group name  ; IBGRNU = group number  ;  ; Output: IBCPOL = pointer to policy file  ;  N %DT  S IBCPOL=""  I $G(^DIC(36,+$G(IBCNS),0))="" G HIPQ  S IBGRP=+$G(IBGRP) ; if undefine, is not a group policy  I 'IBGRP S IBCPOL=$$ADDH(IBCNS,IBGRP) G HIPQ  ;  S:$G(IBGRNU)="" IBGRNU="IB ZZZZZ"  I IBGRNU'="IB ZZZZZ" S IBCPOL=$O(^IBA(355.3,"AGNU",IBCNS,IBGRNU,0))  I IBCPOL,$P($G(^IBA(355.3,+IBCPOL,0)),"^",3)=IBGRNA G HIPQ ; match both  ;  S:$G(IBGRNA)="" IBGRNA="IB ZZZZZ"  S IBCPOL=$O(^IBA(355.3,"AGNA",IBCNS,IBGRNA,0))  I IBCPOL,$P($G(^IBA(355.3,+IBCPOL,0)),"^",4)=IBGRNU G HIPQ ; match both  ;  I 'IBCPOL S IBCPOL=$$ADDH(IBCNS,IBGRP) D  .I IBGRNA="",IBGRNU="" Q  .S:IBGRNA="IB ZZZZZ" IBGRNA="" S:IBGRNU="IB ZZZZZ" IBGRNU=""  .S DA=IBCPOL,DIE="^IBA(355.3,",DR=".03////"\_$$STRIP(IBGRNA,";")\_";.04////"\_$$STRIP(IBGRNU,";")  .D ^DIE K DA,DR,DIC,DIE  HIPQ Q IBCPOL  ;  ADDH(IBCNS,IBGRP,IBGNA,IBGNU) ; -- add entries to health insurance policy file (355.3)  ; Input: IBCNS = pointer to ins co file  ; IBGRP = 1 if group policy, 0 if no  ;  ; Output: IBCPOL = pointer to policy file, if added else null  ;  N %DT,IBN1,IBCAB,DIC,DIE,DR,DA,DLAYGO,DO,DD  S IBCPOL=""  I $G(IBCNS)="" G ADDHQ  K DD,DO,DIC,DR S DIC="^IBA(355.3,",DIC(0)="L",DLAYGO=355.3  ;  S X=IBCNS D FILE^DICN I +Y<0 G ADDHQ  S (DA,IBCPOL)=+Y,DIE="^IBA(355.3,",DR=".02////"\_+$G(IBGRP)  I IBGRP=0,$G(DFN) S DR=DR\_";.1////"\_DFN  I $D(IBGNU) S DR=DR\_";.04///^S X=IBGNU"  I $D(IBGNA) S DR=DR\_";.03///^S X=IBGNA"  D ^DIE K DA,DR,DIE,DIC  I $G(IBCNTP)'="" S IBCNTP=IBCNTP+1  ADDHQ Q IBCPOL  ;  ODELP(DFN,INS) ; -- can an insurance policy be deleted  ; -- called by ^dd(2.312,0,"del",.01) and by ibcnsm  ; -- input dfn: ien of patient in file 2.  ; ins: ien of ins. co in file 36  ;  ; -- output 1 if no deletion allowed  ; 0 if deletion allowed  N I,X,Y S X=0  ;  ; -- do not delete if any uncancelled bills  S J=0 F S J=$O(^DGCR(399,"AE",DFN,INS,J)) Q:'J I $P(^DGCR(399,J,"S"),"^",17)="" S X=1 Q  ODELPQ Q X  ;  STRIP(X,X1) ; -- strip characters from string  ; input: x = string  ; x1 = character to strip (default is ";"  N I,X2  S X2="" S:$G(X1)="" X1=";"  S X1=$E(X1)  F I=1:1 S X2=X2\_$P(X,X1,I) Q:($P(X,X1,I+1,999)'[X1)  Q X2  ;  ;  DELP(DFN,INS,IBC) ; -- can an insurance policy be deleted  ; -- called by ^dd(2.312,0,"del",.01) and by ibcnsm  ; -- input dfn: ien of patient in file 2.  ; ins: ien of ins. co in file 36  ; ibc: ien of policy in file 2.312 to do a match  ;  ; -- output 1 if no deletion allowed  ; 0 if deletion allowed  ;  N ARR,J,ONEPOL,X  ;  ; - check input  I '$G(DFN)!'$G(INS) S X=1 G DELPQ  ;  ; - see if vet has more than one policy with carrier; set flag  ; - also, if no policy is passed, assume the patient has one policy  I $G(IBC) D  .S J=0 F S J=$O(^DPT("AB",IBC,DFN,J)) Q:'J S ARR(J)=$G(^DPT(DFN,.312,J,0))  .S (J,ONEPOL)=0 S J=$O(ARR(J)) I J,'$O(ARR(J)) S ONEPOL=1  E S ONEPOL=1  ;  ;  ; -- do not delete if any uncancelled bills  S (J,X)=0 F S J=$O(^DGCR(399,"AE",DFN,INS,J)) Q:'J D Q:X  .;  .N ARRP,POL,K,L,M,MP,S,Z  .S Z=$G(^DGCR(399,J,0)),M=$G(^("M")),MP=$G(^("MP")),S=$G(^("S"))  .;  .; - skip cancelled bills  .I $P(S,"^",17)'="" Q  .;  .; - set flag if the patient has just one policy with the company  .I ONEPOL S X=1 Q  .;  .; - if there are no policy pointers in the claim,  .I '$P(M,"^",12),'$P(M,"^",13),'$P(M,"^",14),'$P(MP,"^",2) D Q  ..;  ..; - find all policies effective on the event date  ..S K=0 F S K=$O(ARR(K)) Q:'K S POL=ARR(K) D  ...I $P(POL,"^",8) Q:$P(Z,"^",3)<$P(POL,"^",8)  ...I $P(POL,"^",4) Q:$P(Z,"^",3)>$P(POL,"^",4)  ...S ARRP(K)=""  ..;  ..; - if there are two such policies, trust user judgement and assume  ..; - policy is not related to this claim.  ..S L=$O(ARRP(0)) I L,$O(ARR(L)) Q  ..;  ..; - if there is just one policy, and it is the same as the one  ..; - passed in, do not allow deletion.  ..I L=IBC S X=1  .;  .; - if one of the claim policy pointers is the same as the policy  .; - passed in, do not allow deletion.  .I $P(MP,"^",2)=IBC S X=1 Q  .I $P(M,"^",12)=IBC!($P(M,"^",13)=IBC)!($P(M,"^",14)=IBC) S X=1  ;  ;  DELPQ Q X  ;  DUPADDRL(DATA,IBCNS,FLD1,FLD2) ; Insurance address lines can not be duplicated  ; DATA - Value being compared  ; FLD1 - First field to check against  ; FLD2 - Second field to check against (OPTIONAL)  ;  ; Returns 1 if this field is a duplicate of another field.  ;  N Z1,Z2  Q:$G(DATA)="" 0 ; should not happen because this is invoked as an input transform  Q:'$G(IBCNS) 1 ; stop from editing through fileman  S DATA=$$UP^XLFSTR($G(DATA)),DATA=$$TRIM^XLFSTR(DATA)  S Z1=$$GET1^DIQ(36,+$G(IBCNS),+$G(FLD1),"I")  S Z1=$$UP^XLFSTR(Z1),Z1=$$TRIM^XLFSTR(Z1)  S Z2=$$GET1^DIQ(36,+$G(IBCNS),+$G(FLD2),"I")  S Z2=$$UP^XLFSTR(Z2),Z2=$$TRIM^XLFSTR(Z2)  I DATA=Z1 D CLEAN^DILF Q 1  I DATA=Z2 D CLEAN^DILF Q 1  D CLEAN^DILF  Q 0  ;  ZIPCHK9(ZIP) ;AITC/JRA - IB\*2.0\*576 Check if ZIP is in proper 9-digit format  ;Zip must be in the form '123456789' or '12345-6789' and the last 4 digits can't be  ; '0000' or '9999'.  N ZIP4  I ZIP'?9N,(ZIP'?5N1"-"4N) Q 0  S ZIP4=$S(ZIP["-":$P(ZIP,"-",2),1:$E(ZIP,6,9)) I ZIP4="0000"!(ZIP4="9999") Q 0  Q ZIP  ; | | | | | | | | | |