**MCCF EDI TAS EINSURANCE US3511**

System Design Document

IB\*2.0\*602



Department of Veterans Affairs

**December 2017**

Version 1.0

**User Story Number:** US3511

**User Story Name:** Electronic Insurance Coverage Discovery: Data Extract

|  |  |
| --- | --- |
| **Epic Taxonomy** | eBiz Compliance  Port  Update  Increase No Touch  TAS Apps |

# Story:

|  |  |  |
| --- | --- | --- |
| **As a...** | **I want to...** | **So that...** |
| Consumer of Electronic Insurance Coverage Discovery (EICD) functionality | Extract patient demographics from VistA, based on scheduled appointments, for patients who have no active insurance information[[1]](#footnote-1) saved in VistA. | The EICD functionality can build an HL7 message to be sent to Financial Services Center (FSC) where it will be converted to an X12 270 EICD Request and sent to TransUnion (TU). |

**Clean up needed**

**:: Routine: IBCNEDE6**

Remove tag INAC

Remove tag INACSET

Remove tag BLANKTQ

**:: Routine: IBCNEDE5**

Remove tag SIDCHK2 (note: this tag is called from IBCNEDE4; we’ll be doing major renovations in that extract below; part of those changes involve tag removal including “POP” which contains the SIDCHK2 call)

**Call the EICD Extract**

**:: Routine: IBCNEDE** - main driver of eIV Nightly Process

Add the three lines shown below, directly after line 52 “D EN^IBCNEDE2 ; Pre Reg Extract”

 ; Check to see if background process has been stopped, if so quit.

I $G(ZTSTOP) G ENX

D EN^IBCNEDE4 ; No Insurance Extract (EICD)

**EICD Extract**

**:: Routine**: IBCNEDE7

**In Tag**: SETTINGS

Update function to accept EXTNUM=4 ; this is for the insurance discovery aka no insurance extract

With regards to checking for missing data like lines 49 thru 51 (see below) for EXTNUM=4, if XDAYS =”” set the variable to zero when passing it back out of this routine. Make sure STALEDYS is populated for EXTNUM=4 otherwise set EACTIVE to 0 to stop the extract from running.

 I EXTNUM=2,(XDAYS="") S EACTIVE=0 ; missing required data  
 I EXTNUM=3 D  
 . I XDAYS=""!(STALEDYS="") S EACTIVE=0 ; missing required data

Make sure you pull (and return) selection criteria #1, selection criteria #2 and selection criteria #3; Check STALEDYS as it may already be pulling selection criteria #2.

**:: Routine: IBCNEDE4** (major rewrite)

Add the patch number to the second line of the routine. Don’t make the mistake of restarting over with patch numbers just because we are reusing the routine.

Update the directive on the third line

Fix “Note” and “Program description” – lines 5 & 10

* (most comments for other patches can be removed, e.g. “IB\*2\*416”)
* Make reference to “EICD”
* Explain that we are rewriting the entire logic for determining insurance for those who don’t have active policies with patch xyz

When all is done make sure variables being new’d on line 19-22 are accurate and there are no leftovers

Keep lines 24 - 34 (Set INCNCNT=0 thru getting extract parameters)

 S IBCNCNT=0  
 ; Initialize count for periodic TaskMan check  
 S IBCNETOT=0  
 ;  
 ; Get Extract parameters  
 S EACTIVE=$$SETTINGS^IBCNEDE7(4)  
 I 'EACTIVE Q                   ; quit if not active  
 S XDAYS=$P(EACTIVE,U,2)  
 S YDAYS=$P(EACTIVE,U,3)  
 S MAXCNT=$P(EACTIVE,U,4)  
 S:MAXCNT="" MAXCNT=9999999999

Pull start and end date ranges from site parameters within IBCENDE7

Using the EICD payer from the site parameters confirm using file (#365.12) that the payer is both nationally active and for the IIV application is locally active. If either field is inactive quit and stop processing for this extract. No need to provide any message – quiet quit.

Change line 37 – Calculate the start and end date range from the site parameters that you’ve already pulled

 S IBBDT=$$FMADD^XLFDT(DT,-XDAYS),IBEDT=DT

**Remove lines:**

* Lines that control the main routine: lines 41 – 64

**Remove tags:**

* INP
* OUTP
* REST
* PROCESS
* POP
* POPSET

**You should be left with nothing past line 37 which you corrected above**.

Now … using IBCNEDE2 as a template (utilize the copy pasted at the end of this SDD doc as a guide since it could switch with other changes during this build)

The rest of this code is to be placed into IBCNEDE4 after your last line which is line 37 as described above.

**Gather clinics**

* Copy lines 34 – 49 from IBCNEDE2 … change references to IBCNEDE2 to IBCNEDE4 & make the call to CLINICEX^IBCNEDE2

*From:* “K ^TMP($J,"SDAMA301"),^TMP("IBCNEDE2",$J) ; Clean TMP globals”  
 ;  
.

.

.

*Through:* “S NUM=$$SDAPI^SDAMA301(.IBSDA) I NUM<1 D:NUM<0 ERRMSG G ENQ”

Loop through clinics, identify patients and populate IIV Transmission Queue file #365.1

* Copy lines 52 – 83 from IBCNEDE2 ... make call to ELG^IBCNEDE2 & drop commented out line for deceased patients we are not excluding them (you can’t reference patch #549)
* Make use of the start and end dates that you derived from the data you pulled from the site parameters (see above) when looping through the appointment dates
* Set FRESHDT= extract date - selection criteria #2 (#350.9002,.04)
* CAREFUL – A patient can have more than one appointment in the date range you don’t want to reevaluate the same patient twice and you don’t want more than one EICD for the patient.
* In the middle of this logic you find the DFN of the patient … determine if we can skip this patient based on the “Date Last EICD Run” (#2.xxxx,.01) and the selection criteria #2 (#350.9002,.04) that you pulled from the site parameters (see above). If the Date Last EICD Run greater than FRESHDT (see above) skip the patient
* Once the patient is identified … determine if they have active insurance as defined in US3511 user story
  + Loop through policies
  + Use the INSURANCE TYPE (#2.312,.01) field to determine the insurance company name
  + Use the EFFECTIVE DATE OF POLICY (#2.312,8) field
  + Use the INSURANCE EXPIRATION DATE (#2.312,3) field
  + Use the GROUP PLAN (#2.312, .18) -> TYPE OF PLAN (#355.3,.09) get external value to check
  + If the ACTINS has no policies then continue processing patient
  + If any of the active policies does NOT meet the criteria of “don’t consider them active” then skip the patient
* Determine if there is already an entry in the TQ waiting to go out for this EICD payer
* Make a call to ADDTQ^IBCNEUT5 pass in DFN, payer ien of the EICD payer, Today’s date as the service date, and FRESHDAY as selection critieria #3 from the site parameters that you pulled from IBCNEDE7. See below for an example:

; Quit before filing if outstanding entries in TQ  
 I '$$ADDTQ^IBCNEUT5(DFN,PIEN,SRVICEDT,FRESHDAY) Q

* For every patient that we continue processing with …
* To match the same look, feel, and flow from the other eIV extracts : call a tag “SET” to define your variables and call SETTQ^IBCNEDE7

(see IBCNEDE1 and IBCNEDE2 for examples)

* + DFN = patient’s ien
  + QURYFLAG=”I” ; the letter “i" in Caps
  + FRESHDT= see calculation above
  + ; The hard coded '1' in the 3rd piece of DATA1 sets the Transmission status of file 365.1 to "Ready to Transmit"
  + DATA1=DFN^payer ien ^1^^^FRESHDT^
  + ; The hardcoded '4' in the 1st piece of DATA2 is the value to tell the file 365.1 that it is the No Insurance (EICD) extract.
  + DATA2=4^QURYFLAG^DT
  + ; source of information pulled from file #355.12
  + DATA5= ien for the source of info entry “CONTRACT SERVICES”

**Finished!**

**Test!**

IMPORTANT … USE THIS COPY OF IBCNEDE2 as a guide since the routine on the development or CIT account could be different by the time one is making these coding changes.

\*\*\* The super small font is on purpose so that nothing is wrapped as the programmer will be literally copying lines from this section of the document. \*\*\*

For old eyes, please use the view tab in word and zoom … don’t change the font.

IBCNEDE2 ;DAOU/DAC - eIV PRE REG EXTRACT (APPTS) ;23-SEP-2015  
 ;;2.0;INTEGRATED BILLING;\*\*184,271,249,345,416,438,506,549,593\*\*;21-MAR-94;Build 31  
 ;;Per VA Directive 6402, this routine should not be modified.  
 ;  
 ;\*\*Program Description\*\*  
 ; This program finds veterans who are scheduled to be seen within a  
 ; specified date range.  
 ; Periodically check for stop request for background task  
 ;  
 Q   ; can't be called directly  
 ;  
EN ; Loop through designated cross-references for updates  
 ; Pre reg extract (Appointment extract)  
 ; IB\*2.0\*593 - Added EXCLTOC,EXCLTOP now initialized at top. Removed YY.  
 ; IB\*2.0\*549 - Added YY,ZZ, Re-Arranged in alphabetical order  
 N ACTINS,APTDT,CLNC,CNT,DATA1,DATA2,DFN,DISYS,ELG,ENDDT,EXCLTOC,EXCLTOP,FOUND1,FOUND2,FRESHDAY  
 N FRESHDT,GIEN,IBCNETOT,IBDDI,IBINDT,IBINS,IBSDA,IBSDATA,IBOUTP,INREC,INS,INSIEN,INSNAME  
 N MAXCNT,MCAREFLG,NUM,OK,PATID,PAYER,PAYERID,PAYERSTR,PIEN  
 N SETSTR,SID,SIDACT,SIDARRAY,SIDCNT,SIDDATA,SLCCRIT1,SRVICEDT,SUPPBUFF,SYMBOL  
 N TODAYSDT,TQIEN,QURYFLAG,VAIN,VDATE,YY,ZZ  
 ;  
 S SETSTR=$$SETTINGS^IBCNEDE7(2) ; Get setting for pre reg. extract   
 I 'SETSTR Q                         ; Quit if extract is not active  
 S SLCCRIT1=$P(SETSTR,U,2) ; Selection Criteria #1  
 S MAXCNT=$P(SETSTR,U,4) ; Max # of TQ entries to create  
 S:MAXCNT="" MAXCNT=9999999999  
 S SUPPBUFF=$P(SETSTR,U,5) ; Suppress Buffer Flag  
 S FRESHDAY=$P($G(^IBE(350.9,1,51)),U,1) ; Freshness days span  
 S CNT=0 ; Init. TQ entry counter  
 S ENDDT=$$FMADD^XLFDT(DT,SLCCRIT1) ; End of appt. date selection range  
 S IBCNETOT=0 ; Initialize count for periodic TaskMan check  
 S EXCLTOC=$$GETELST(355.2) ; Initialize excluded TYPEs OF COVERAGE IB\*2.0\*593  
 S EXCLTOP=$$GETELST(355.1) ; Initialize excluded TYPEs OF PLAN IB\*2.0\*593  
 K ^TMP($J,"SDAMA301"),^TMP("IBCNEDE2",$J) ; Clean TMP globals  
 ;  
 S CLNC=0 ; Init. clinic  
 ; Loop through clinics   
 F  S CLNC=$O(^SC(CLNC)) Q:'CLNC!(CNT'<MAXCNT) D  Q:$G(ZTSTOP)  
 . ;  
 . D CLINICEX Q:'OK     ; Check for clinic exclusion  
 . ;  
 . S ^TMP("IBCNEDE2",$J,CLNC)=""  
 ;  
 ; Set up variables for scheduling call and call  
 S IBSDA("FLDS")=8  
 S IBSDA(1)=DT\_";"\_ENDDT  
 S IBSDA(2)="^TMP(""IBCNEDE2"",$J,"  
 S IBSDA(3)="R"  
 S NUM=$$SDAPI^SDAMA301(.IBSDA) I NUM<1 D:NUM<0 ERRMSG G ENQ  
 ;  
 ;  
 S CLNC=0 ; Init. clinic  
 ; Loop through clinics returned  
 F  S CLNC=$O(^TMP($J,"SDAMA301",CLNC)) Q:'CLNC  D  Q:$G(ZTSTOP)!(CNT'<MAXCNT)  
 . ;  
 . ; Loop through patients returned  
 . S DFN=0 F  S DFN=$O(^TMP($J,"SDAMA301",CLNC,DFN)) Q:'DFN!(CNT'<MAXCNT) D  Q:$G(ZTSTOP)  
 .. ;  
 .. S APTDT=DT           ; Check for appointment date  
 .. S MCAREFLG=0  
 .. ;  
 .. ; Loop through dates in range at clinic  
 .. F  S APTDT=$O(^TMP($J,"SDAMA301",CLNC,DFN,APTDT)) Q:('APTDT)!((APTDT\1)>ENDDT)!(CNT'<MAXCNT) D  Q:$G(ZTSTOP)  
 ... ;  
 ... S SRVICEDT=APTDT\1 ;Set service date equal to appointment date  
 ... S FRESHDT=$$FMADD^XLFDT(SRVICEDT,-FRESHDAY)  
 ... ;  
 ... ; Update count for periodic check  
 ... S IBCNETOT=IBCNETOT+1  
 ... ; Check for request to stop background job, periodically  
 ... I $D(ZTQUEUED),IBCNETOT#100=0,$$S^%ZTLOAD() S ZTSTOP=1 Q  
 ... ;  
 ... S IBSDATA=$G(^TMP($J,"SDAMA301",CLNC,DFN,APTDT))  
 ... S ELG=$P(IBSDATA,U,8)  
 ... S ELG=$S(ELG'="":ELG,1:$P($G(^DPT(DFN,.36)),U,1))  
 ... I $P($G(^DPT(DFN,0)),U,21) Q         ; Exclude if test patient  
 ... ; IB\*2.0\*549 removed the following line  
 ... ;I $P($G(^DPT(DFN,.35)),"^",1)'="" Q ; Exclude if patient is deceased  
 ... ;  
 ... D ELG Q:'OK     ; Check for eligibility exclusion  
 ... ;  
 ... K ACTINS  
 ... D ALL^IBCNS1(DFN,"ACTINS",2)  
 ... ;  
 ... I '$D(ACTINS(0)) Q  ; Patient has no active ins  
 ... ;  
 ... S INREC=0 ; Record IEN  
 ... F  S INREC=$O(ACTINS(INREC)) Q:('INREC)!(CNT'<MAXCNT) D  
 ... . S INSIEN=$P($G(ACTINS(INREC,0)),U,1) ; Insurance ien  
 ... . S INSNAME=$P($G(^DIC(36,INSIEN,0)),U)  
 ... . ;  
 ... . ; IB\*2.0\*549 Added next 3 lines to exclude certain Type of Coverages  
 ... . ; IB\*2.0\*593 Moved exclusion list initialization to top execution level.  
 ... . S ZZ=$$GET1^DIQ(36,INSIEN\_",",.13,"I") ; Type of Coverage  
 ... . ;S YY=$$GETELST(355.2) ; Type of Coverages to exclude  
 ... . ;Q:YY[("^"\_ZZ\_"^") ; Excluded Type of Coverage  
 ... . Q:EXCLTOC[("^"\_ZZ\_"^") ; Excluded Type of Coverage  
 ... . ;  
 ... . ; Exclude policies that have been verified within "freshness days"  
 ... . S VDATE=$P($G(ACTINS(INREC,1)),U,3)  
 ... . I VDATE'="",SRVICEDT'>$$FMADD^XLFDT(VDATE,FRESHDAY) Q  
 ... . ; Allow only one MEDICARE transmission per patient  
 ... . I INSNAME["MEDICARE",MCAREFLG Q  
 ... . ; Exclude pharmacy policies IB\*2.0\*549 - Commented out following line  
 ... . ;I $$GET1^DIQ(36,INSIEN\_",",.13)="PRESCRIPTION ONLY" Q  
 ... . S GIEN=+$P($G(ACTINS(INREC,0)),U,18)  
 ... . ;  
 ... . ; IB\*2.0\*549 Added next 3 lines to exclude certain Type of Plans  
 ... . ; IB\*2.0\*593/TAZ Moved exclusion list initialization to top execution level.  
 ... . S ZZ=$$GET1^DIQ(355.3,GIEN\_",",.09,"I") ; Type of Plan  
 ... . ;S YY=$$GETELST(355.1) ; Type of Plans to exclude  
 ... . ;Q:YY[("^"\_ZZ\_"^") ; Excluded Type of Plan  
 ... . Q:EXCLTOP[("^"\_ZZ\_"^") ; Excluded Type of Plan  
 ... . ;  
 ... . ;I GIEN,$$GET1^DIQ(355.3,GIEN\_",",.09)="PRESCRIPTION" Q ; IB\*2.0\*549 - Removed line  
 ... . ; check for ins. to exclude (i.e. Medicaid)  
 ... . I $$EXCLUDE^IBCNEUT4(INSNAME) Q  
 ... . ; check insurance policy expiration date  
 ... . I $$EXPIRED($P($G(ACTINS(INREC,0)),U,4)) Q  
 ... . ;  
 ... . ; set patient id field IB\*2\*416  
 ... . S PATID=$P($G(ACTINS(INREC,5)),U,1) ; 5.01 field  
 ... . ;  
 ... . S PAYERSTR=$$INSERROR^IBCNEUT3("I",INSIEN) ; Get payer info  
 ... . ;  
 ... . S SYMBOL=+PAYERSTR ; error symbol  
 ... . S PAYERID=$P(PAYERSTR,U,3) ; (National ID) payer id  
 ... . S PIEN=$P(PAYERSTR,U,2) ; Payer ien  
 ... . ;  
 ... . ; If Payer is Nationally Inactive create an Insurance Buffer record w/blank SYMBOL & quit. - IB\*2.0\*506  
 ... . I '$$PYRACTV^IBCNEDE7(PIEN) D  Q  
 ... .. S SYMBOL=""  
 ... .. I 'SUPPBUFF,'$$BFEXIST^IBCNEUT5(DFN,INSNAME) D PT^IBCNEBF(DFN,INREC,SYMBOL,"",1)  
 ... .. Q  
 ... . ;  
 ... . ; If error symbol exists, set record in insurance buffer & quit  
 ... . I SYMBOL D  Q  
 ... . . I 'SUPPBUFF,'$$BFEXIST^IBCNEUT5(DFN,INSNAME) D PT^IBCNEBF(DFN,INREC,SYMBOL,"",1)  
 ... . ;  
 ... . ; Update service date and freshness date based on payers allowed  
 ... . ; date range  
 ... . D UPDDTS^IBCNEDE6(PIEN,.SRVICEDT,.FRESHDT)  
 ... . ;  
 ... . ; Update service dates for inquiry to be transmitted  
 ... . D TQUPDSV^IBCNEUT5(DFN,PIEN,SRVICEDT)  
 ... . ;  
 ... . ; Quit before filing if outstanding entries in TQ  
 ... . I '$$ADDTQ^IBCNEUT5(DFN,PIEN,SRVICEDT,FRESHDAY) Q  
 ... . ;  
 ... . S QURYFLAG="V"  
 ... . K SIDARRAY  
 ... . S SIDDATA=$$SIDCHK^IBCNEDE5(PIEN,DFN,,.SIDARRAY,FRESHDT)  
 ... . S SIDACT=$P(SIDDATA,U),SIDCNT=$P(SIDDATA,U,2)  
 ... . I SIDACT=3,'SUPPBUFF,'$$BFEXIST^IBCNEUT5(DFN,INSNAME) D PT^IBCNEBF(DFN,INREC,18,"",1) Q  
 ... . I CNT+SIDCNT>MAXCNT S CNT=MAXCNT Q  ;exceeds MAXCNT  
 ... . ;  
 ... . S SID=""  
 ... . F  S SID=$O(SIDARRAY(SID)) Q:SID=""  D:$P(SID,"\_")'="" SET($P(SID,"\_"),$P(SID,"\_",2),PATID) S:INSNAME["MEDICARE" MCAREFLG=1  
 ... . I SIDACT=4 D  
 ... . . D SET("","",PATID)  
 ... . . S:INSNAME["MEDICARE" MCAREFLG=1  
 ... . Q  
 ... Q  
ENQ K ^TMP($J,"SDAMA301"),^TMP("IBCNEDE2",$J)  
 Q  
 ;  
GETELST(FILE) ; Returns a '^' delimited list of Type of Plans or Type of  
 ; coverages to be excluded with leading and trailing '^'s  
 ; IB\*2.0\*549 Added method  
 ; IB\*2.0\*593 Added NO-FAULT INSURANCE. Refactored.  
 ; Input: FILE - 355.1 - Return a list of Type of Plans to be excluded  
 ; 355.2 - Return a list of Type of Coverages to be excluded  
 ; Returns: '^' delimited list of Type of Plans or Type of Coverages  
 ; to be excluded  
 ;N EXCLIST,IEN,NM,XX  
 ;S EXCLIST="",NM("AUTOMOBILE")="",NM("MEDI-CAL")="",NM("TORT FEASOR")=""  
 ;S NM("WORKERS' COMPENSATION INSURANCE")="",NM("VA SPECIAL CLASS")=""  
 ;S NM("MEDICAID")=""  
 ;S XX=""  
 ;F D Q:XX=""  
 ;. S XX=$O(NM(XX))  
 ;. Q:XX=""  
 ;. S IEN=""  
 ;. F D Q:IEN=""  
 ;. . S IEN=$O(^IBE(FILE,"B",XX,IEN))  
 ;. . Q:IEN=""  
 ;. . S EXCLIST=$S(EXCLIST="":IEN,1:EXCLIST\_"^"\_IEN)  
 N EXCLIST,TYPE  
 S EXCLIST=""  
 F TYPE="AUTOMOBILE","MEDICAID","MEDI-CAL","NO-FAULT INSURANCE","TORT FEASOR","WORKERS' COMPENSATION INSURANCE","VA SPECIAL CLASS" D  
 . N IEN S IEN=$O(^IBE(FILE,"B",TYPE,""))  
 . Q:IEN=""  
 . S EXCLIST=$S(EXCLIST="":IEN,1:EXCLIST\_"^"\_IEN)  
 Q "^"\_EXCLIST\_"^"  
 ;  
CLINICEX ; Clinic exclusion  
 S OK=1  
 I $D(^DG(43,1,"DGPREC","B",CLNC)) S OK=0  
 Q  
 ;  
ELG ; Eligibility exclusion  
 I ELG="" S OK=0 Q  
 I $D(^DG(43,1,"DGPREE","B",ELG)) S OK=0 Q  
 S OK=1  
 Q  
 ;  
INP ; Inpatient status  
 D INP^VADPT  
 I $G(VAIN(1))'="" K VAIN S OK=0 Q  
 K VAIN  
 S OK=1  
 Q  
 ;  
SET(SID,INR,PATID) ; Set data in TQ  
 ;  
 ; The hard coded '1' in the 3rd piece of DATA1 sets the Transmission  
 ; status of file 365.1 to "Ready to Transmit"  
 S DATA1=DFN\_U\_PIEN\_U\_1\_U\_""\_U\_SID\_U\_FRESHDT ; SETTQ 1st parameter  
 S $P(DATA1,U,8)=PATID     ; IB\*2\*416  
 ;  
 ; The hardcoded '2' in the 1st piece of DATA2 is the value to tell  
 ; the file 365.1 that it is the appointment extract.  
 S DATA2=2\_U\_QURYFLAG\_U\_SRVICEDT\_U\_INR    ; SETTQ 2nd parameter  
 ;  
 S TQIEN=$$SETTQ^IBCNEDE7(DATA1,DATA2) ; Sets in TQ  
 I TQIEN'="" S CNT=CNT+1 ; If filed increment count  
 ;  
 Q  
 ;  
ERRMSG ; Send a message indicating an extract error has occurred  
 N MGRP,XMSUB,MSG,IBX,IBM  
 ;  
 ; Set to IB site parameter MAILGROUP  
 S MGRP=$$MGRP^IBCNEUT5()  
 ;  
 S XMSUB="eIV Problem: Appointment Extract"  
 S MSG(1)="On "\_$$FMTE^XLFDT(DT)\_" the Appointment Extract for eIV encountered one or more"  
 S MSG(2)="errors while attempting to get Appointment data from the scheduling"  
 S MSG(3)="package."  
 S MSG(4)=""  
 S MSG(5)="Error(s) encountered: "  
 S MSG(6)=""  
 S MSG(7)=" Error Code Error Message"  
 S MSG(8)=" ---------- -------------"  
 S IBM=8,IBX=0 F  S IBX=$O(^TMP($J,"SDAMA301",IBX)) Q:IBX=""  S IBM=IBM+1,MSG(IBM)=" "\_$$LJ^XLFSTR(IBX,13)\_$G(^TMP($J,"SDAMA301",IBX))  
 S IBM=IBM+1,MSG(IBM)=""  
 S IBM=IBM+1,MSG(IBM)="As a result of this error the extract was not done. The extract"  
 S IBM=IBM+1,MSG(IBM)="will be attempted again the next night automatically. If you"  
 S IBM=IBM+1,MSG(IBM)="continue to receive error messages you should contact your IRM"  
 S IBM=IBM+1,MSG(IBM)="and possibly call the Help Desk for assistance."  
 ;  
 D MSG^IBCNEUT5(MGRP,XMSUB,"MSG(")  
 ;  
 Q  
 ;  
EXPIRED(EXPDT) ; check if insurance policy has already expired  
 ; EXPDT - expiration date (2.312/3)  
 ; returns 1 if expiration date is in the past, 0 otherwise  
 N X1,X2  
 S X1=+$G(DT),X2=+$G(EXPDT)  
 I X1,X2 Q $S($$FMDIFF^XLFDT(DT,EXPDT,1)>0:1,1:0)  
 Q 0

1. Refer to the user story -- TAS eIns US3511 Insurance Coverage Electronic Discovery Data Extract.docx for a detailed description of “no active insurance” [↑](#footnote-ref-1)