**MCCF EDI TAS EINSURANCE US3513**

System Design Document

IB\*2.0\*602



Department of Veterans Affairs

**December 2017**

Version 1.0

**User Story Number:** US3513

**User Story Name:** Electronic Insurance Coverage Discovery: Request (Inbound Message)

|  |  |
| --- | --- |
| **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 | Receive and process (in VistA) the EICD Response sent by Financial Services Center (FSC) after FSC receives the EICD Response from TransUnion (TU) | Found insurance for a patient can be verified for eligibility benefits and persisted in the patient’s insurance file. |

# Assumptions:

1. The below design assumes that FSC (and ultimately Transunion) will send the following data (we are not capturing any other data) for the insurance found: the VA Payer ID, , group ID, subscriber ID, member ID, subscriber SSN, subscriber DOB
2. At this time, the eIV Response Report will not display the insurance information for the A1 responses with this current design. A new user story will be needed – for a future release – to update the eIV Response Report if that information is desired on the report.
3. The eIV purge file logic will need to updated to include the new file #365.16 – possible future story.

# Design:

Note : “TQ” is a reference to file #365.1 (IIV TRANSMISSION QUEUE)

* Edit existing file IIV TRANSMISSION QUEUE #365.1
  + Add new field .21 EIV INS DISCOVERY RESULTS IEN (Number corresponding to the #365.16 node “INS-A2” multiple)
* Edit existing file IIV RESPONSE #365
  + Add new field pointer or logical pointer to the EIV INSURANCE DISCOVERY RESULTS #365.16
* A new file will be created to store inbound message information for A1 responses and provide a ‘crosswalk’ of sorts
  + Name #365.16 EIV INSURANCE DISCOVERY RESULTS
  + Global ^IBCN(365.16

^IBCN(365.16,IEN,0)=A1 TQ Pointer ^ A1 DT Created ^ A1 EICD Payer ^ A1 EC Trace Number ^ A1 Response Pointer

^IBCN(365.16,IEN,”INS-A2”)= Insurance info discovered (MULTIPLE)

^IBCN(365.16,IEN,”INS-A2”,DA,0)= A2 TQ Pointer ^ A2 DT Created ^ PAYER (VA NATIONAL ID) ^ GROUP ID ^ SUBSCRIBER ID ^ MEMBER ID^ SUBSCRIBER SSN ^ SUBSCRIBER DOB

* Create new protocol as follows:
  + NAME: IBCNE eIV RQI IN  
      ITEM TEXT: eIV IDENTIFICATION CLIENT  
      TYPE: event driver  
      DESCRIPTION: This protocol is for Identification messages   
      SENDING APPLICATION: IIV VISTA  
      TRANSACTION MESSAGE TYPE: RQP   
      EVENT TYPE: I04  
      ACCEPT ACK CODE: AL  
      APPLICATION ACK TYPE: NE  
      VERSION ID: 2.4  
      RESPONSE PROCESSING ROUTINE: D ^IBCNEHLI  
      SUBSCRIBERS: IBCNE EIV ID RESPONSE
* Edit routine: IBCNEHLI
  + Add a new line after 38, “I EVENT=”RQP^IO4”  S TAG="EICD",IBPRTCL=“IBCNE eIV RQP IN”
  + Edit line tag RSP to now call EN^IBCNEHL1(2) instead of the current call.
  + Add line tag EICD ; A1 Response
  + Add line “D EN^IBCNEHL1(1) Q”
* Edit routine: IBCNEHL1
  + Add a “Q” before line tag EN
  + Edit Line tag EN, change it to EN(EVENTYP)
    - EVENTYP=1 > A1 Response (RPI^IO4)
    - EVENTYP=2 > Normal 270 (RPI^IO1)
  + For lines 63 & 66 where it calls tags GT1 & IN1 of IBNCEHL2
    - Only call these original tags if the EVENTYP=2 and the rest of the existing if statement is right
    - Call new tag EICDGT1 and EICDIN1 in IBCNEHL7 if EVENTPY=1 instead
  + Add line right above line 119 which says “S AUTO=$$AUTOUPD(RIEN)”
    - If EVENTYP=1 call new tag SAVEA1^IBCNEHL7
  + Edit Line tag FIL, to pass the variable EVENTYP to IBCNEHL6
* Edit routine: FIL^IBCNEHL6
  + Change line tag FIL to receive variable EVENTYP
  + After line #55, add line “I $G(EVENTYP)=1 D EN^IBCNEHL7 Q”

* Create a new routine: IBCNEHL7
  + Create new tags EICDGT1 & EICDIN1
    - Grab data from the corresponding segment (EICDGT1 processes GT1s etc.) and save the data to an array as we need to gather it all before saving to cross reference
    - If an IN1 has **no** corresponding GT1 then the SUBSCRIBER ID and MEMBER ID are the same and they come from the IN1
    - If an IN1 has a corresponding GT1 then the MEMBER ID is pulled from the IN1 and the SUBSCRIBER ID is pulled from the GT1
  + Create new tag SAVEA1
    - Loop through array gathered in EICDGT1 & EICDIN1 and save to EIV INSURANCE DISCOVERY RESULTS file #365.16
  + Within tag EN, process the incoming A1 that was saved
    - Using the IIV Transmission Queue ien variable, loop through the EIV INSURANCE DISCOVERY RESULTS file #365.16
      * **If payer is both locally and nationally active** …. Create A2 TQ entry and send it out the door, then get next item in the loop
        + Create a TQ entry #365.1 by calling a tag “SET” (( you are mimicking what all the other extracts and real time do))

If subscriber ID = “” in #365.16 set subscriber ID = MEMBER ID on file

Payer – get from #365.16

Set freshness date = today – the freshness date #350.9

S DATA1 = DFN of patient^payer ien^1^^subscriber ID^FRESHNESS DATE^^PATIENT ID

S DATA2=4^”V”^today date

S DATA5= the pointer to “contract services”^EIV INS DISCOVERY RESULTS IEN (#365.1, .21)

\*\*\* now call S TQIENA2=$$SETTQ^IBCNEDE7(DATA1,DATA2,,,DATA5)

Now you have the TQ ien for the A2 … use that to save to EIV INSURANCE DISCOVERY RESULTS file #365.16

* + - * + Load and Send HL7 Message

S RESPONSE=$$PROCSEND^IBCNERTQ(TQIENA2) ; watch your variables

Look at cleanup ENDTRIG^IBCNERTQ for what array may need to be cleaned up

* + - * **If payer is NOT both locally and nationally active** …. Create buffer entry and then skip to next item in the loop
        + ;write a buffer entry
        + ;Patient fields, name, dob and ssn will be populated automatically
        + K IBBUF
        + S IBBUF(.02)=DUZ ; Entered By ((\* NEW DUZ but then pass in the value for the eiv interface user”))
        + S IBBUF(.12)= “” ; need to determine the appropriate buffer symbol (buffer can’t handle payer name) – some analysis/testing needed during coding
        + S IBBUF(.18)=DT
        + S IBBUF(20.01)=payer ien
        + S IBBUF(60.01)=DFN ; Patient IEN
        + S IBBUF(60.02)=EFFECTIVE DATE
        + S IBBUF(60.06)= set to self if member id and subscriber id are same; otherwise set to <TBD>
        + S IBBUF(62.01)=MEMBER ID
        + S IBBUF(80.01)=30
        + S IBBUF(90.02)=GROUP NUMBER
        + S IBBUF(90.03)=subscriber ID
        + ; the following call in-turn, calls EDITSTF^IBCNBES which will make sure to file subscriber ID last, automatically
        + S IBFDA=$$ADDSTF^IBCNBES($$FIND1^DIC(355.12,,,"CONTRACT SERVICES","C"),DFN,.IBBUF)
  + Edit routine: IBCNEHLQ
    - Modify NTE – 3rd occurrence (line 176)

from ”$S($$MBICHK^IBCNEUT7(BUFF):"MBI",1:"ELI")”

to Where if MBI set to ‘MBI’, if EXT=4 and Query=”I” set to “OHI”, otherwise “ELI”

* + - Change line 90 (within GT1 tag) from “I EXT'=1 D” to “I EXT=2 D”
    - Within GT1 tag write new section for IF EXT=4 …. build the GT1 using the data from the EIV INSURANCE DISCOVERY RESULTS file #365.16. \* Note the field EIV INS DISCOVERY RESULTS IEN field (#365.1,.21) will indicate which subfile on #365.16 to use.
    - Change line 145 (within IN1 tag) from “I EXT'=1 D” to “I EXT=2 D”
    - Within IN1 tag write new section for IF EXT=4 …. build the IN1 using the data from the EIV INSURANCE DISCOVERY RESULTS file #365.16. \* Note the field EIV INS DISCOVERY RESULTS IEN field (#365.1,.21) will indicate which subfile on #365.16 to use.
  + Edit routine: PROC^IBCNEDEP
    - Add logic for NTE occurrences for 4 and 5 - make sure you address all scenarios
      * 4th occurrence – blank when Query = “I” ; expired insurances for the past year
      * 5th occurrence – blank unless it is EXT=4 & Query = “V” ; Trace # from the associated A1
  + Edit routine: SETTQ^IBCNEDE7
    - Make it so the 2nd piece of DATA5 is to be saved to EIV INS DISCOVERY RESULTS IEN (#365.1, .21)