**MCCF EDI TAS EINSURANCE US3691**

System Design Document

IB\*2.0\*602



Department of Veterans Affairs

**August 2018**

Version 1.1

**User Story ID:** US3691

**User Story Name:** Purge Old and Abnormal Records from IIV Response File and the

IIV Transmission Queue

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

# Story:

|  |  |  |
| --- | --- | --- |
| **As the...** | **I need...** | **So that...** |
| eInsurance Team member | VistA to automatically remove as part of the existing automatic monthly purge:   * Any records in *the IIV Response File* (except *Do Not Purge*)that point to record locations in the IIV Transmission Queue where no actual records exist   VistA to remove as part of a new pre/post-install routine:   * Any records older than six months in the *IIV Transmission Queue* (except those with associated *Do Not Purge* Response records) | * No *orphan* records remain in the IIV Response File (unless marked *Do Not Purge*) after the monthly purge routine runs   and   * No records older than six months (which cannot be transmitted to FSC and meet the criteria for purging) remain in the IIV Transmission Queue after the pre/post-install routine runs. |

# Design:

Note: Both US3691 and US3754 modify the IBCNEKIT routines for IB\*2.0\*602

*To make the code more robust over time, we want to avoid using hardcoded IEN values directly. When preparing the “STATLIST” variable, use a FILEMAN call to get the proper IEN values.*

*While calculating the default start purge date to display to the user, the first TRANSMISSION QUEUE (#365.1) record that has a purge-able TRANSMISSION STATUS (#365.1, .04) field is utilized.*

*In addition, we’ll need to make sure that the corresponding IIV RESPONSE (#365) field DO NOT PURGE (#365, .11) is not set. If we need to continue searching past the first record for the earliest purge-able date, we’ll want to display a message informing the user that it may take some time.*

| **Routines** | **Activities** | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | **IBCNEKIT** | | | | | | |
| **Enhancement Category** | New | Modify | Delete | | No Change | | |
| **RTM** |  | | | | | | |
| **Related Options** | “Purge eIV Transactions” [IBCNE PURGE IIV DATA] | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | **Routines “Called”** | | | |
|  | IBCNEKI2 | | | ^DIR  ^DIK  ^%ZTLOAD  $$GET1^DIQ  $$MGRP^IBCNEUT5  MSG^IBCNEUT5  $$SITE^VASITE  $$FMADD^XLFDT  $$FMTE^XLFDT | | | |
| **Routines** | **Activities** | | | | | | |
| **Data Dictionary (DD) References** | ^IBCN(365  ^IBCN(365.1 | | | | | | |
| **Related Protocols** |  | | | | | | |
| **Related Integration Control Registrations (ICRs)** |  | | | | | | |
| **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** | | | | | | | |
| IBCNEKIT ;DAOU/ESG - PURGE eIV DATA FILES ;11-JUL-2002  ;;2.0;INTEGRATED BILLING;\*\*184,271,316,416,549,595\*\*;21-MAR-94;Build 22  ;;Per VA Directive 6402, this routine should not be modified.  ; | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | |
| IBCNEKIT ;DAOU/ESG - PURGE eIV DATA FILES ;11-JUL-2002  ;;2.0;INTEGRATED BILLING;\*\*184,271,316,416,549,595**,602**\*\*;21-MAR-94;Build 22  ;;Per VA Directive 6402, this routine should not be modified.  ; | | | | | | | |
| **Current Logic** | | | | | | | |
| INIT ; This procedure calculates the default beginning and ending dates  ; and displays screen messages about this option to the user.  ;  NEW DATE,FOUND,TQIEN,TQS,RPIEN,RPS  NEW DIR,X,Y,DTOUT,DUOUT,DIRUT,DIROUT  ;  S STOP=0  ;  ; This is the list of statuses that are OK to purge  ; 3=Response Received  ; 5=Communication Failure  ; 7=Cancelled  S STATLIST=",3,5,7,"  ;  ; Try to find a beginning date in the eIV Transmission Queue file  S DATE="",FOUND=0,BEGDT=DT  F  S DATE=$O(^IBCN(365.1,"AE",DATE)) Q:'DATE!FOUND  S TQIEN=0 F  S TQIEN=$O(^IBCN(365.1,"AE",DATE,TQIEN)) Q:'TQIEN  D  Q:FOUND  . S TQS=$P($G(^IBCN(365.1,TQIEN,0)),U,4) ; status  . I '$F(STATLIST,","\_TQS\_",") Q  . S FOUND=1  . S BEGDT=$P(DATE,".",1)  . Q  ;  ; If not successful, try to find a beginning date in the eIV Response file.  I 'FOUND D  . S DATE=""  . F  S DATE=$O(^IBCN(365,"AE",DATE)) Q:'DATE!FOUND  S RPIEN=0 F  S RPIEN=$O(^IBCN(365,"AE",DATE,RPIEN)) Q:'RPIEN  D  Q:FOUND  .. S RPS=$P($G(^IBCN(365,RPIEN,0)),U,6) ; status  .. I '$F(STATLIST,","\_RPS\_",") Q  .. S FOUND=1  .. S BEGDT=$P(DATE,".",1)  .. Q  . Q  ;  …  INITX ;  Q  ; | | | | | | | |
| Modified Logic (Changes are in bold) | | | | | | | |
| INIT ; This procedure calculates the default beginning and ending dates  ; and displays screen messages about this option to the user.  ;  NEW DATE,FOUND,TQIEN,TQS,RPIEN,RPS**,IBHL7,IBDNP**  NEW DIR,X,Y,DTOUT,DUOUT,DIRUT,DIROUT  ;  S STOP=0  ;  ; This is the list of statuses that are OK to purge  ; 3=Response Received  ; 5=Communication Failure  ; 7=Cancelled **; IB\*2.0\*602/DM make sure we get the proper IENs for STATLIST  S STATLIST=","\_$$FIND1^DIC(365.14,,"B","Response Received")  S STATLIST=STATLIST\_","\_$$FIND1^DIC(365.14,,"B","Communication Failure")  S STATLIST=STATLIST\_","\_$$FIND1^DIC(365.14,,"B","Cancelled")\_","**  ;  ; Try to find a beginning date in the eIV Transmission Queue file  S DATE="",FOUND=0,BEGDT=DT  F  S DATE=$O(^IBCN(365.1,"AE",DATE)) Q:'DATE!FOUND  S TQIEN=0 F  S TQIEN=$O(^IBCN(365.1,"AE",DATE,TQIEN)) Q:'TQIEN  D  Q:FOUND  . S TQS=$P($G(^IBCN(365.1,TQIEN,0)),U,4) ; status  . I '$F(STATLIST,","\_TQS\_",") Q **. ;IB\*2.0\*602/DM make sure the default earliest date is not a DO NOT PURGE entry   . ;check the HL7 messages multiple to see if DO NOT PURGE is set on any response  . S (IBDNP,IBHL7)=0  . F  S IBHL7=$O(^IBCN(365.1,TQIEN,2,IBHL7)) Q:'IBHL7!IBDNP  D  .. S RPIEN=$P($G(^IBCN(365.1,TQIEN,2,IBHL7,0)),U,3) Q:'RPIEN  .. I +$$GET1^DIQ(365,RPIEN\_",","DO NOT PURGE","I") S IBDNP=1  .. Q  . ;  . I IBDNP,IBVER=2 Q   . I IBDNP W !,"Please wait, checking for the earliest purge date ...",! Q  . ;**  . S FOUND=1  . S BEGDT=$P(DATE,".",1)  . Q  ;  ; If not successful, try to find a beginning date in the eIV Response file.  I 'FOUND D  . S DATE=""  . F  S DATE=$O(^IBCN(365,"AE",DATE)) Q:'DATE!FOUND  S RPIEN=0 F  S RPIEN=$O(^IBCN(365,"AE",DATE,RPIEN)) Q:'RPIEN  D  Q:FOUND  .. S RPS=$P($G(^IBCN(365,RPIEN,0)),U,6) ; status  .. I '$F(STATLIST,","\_RPS\_",") Q **.. ;IB\*2.0\*602/DM do not choose a DO NOT PURGE response   .. I +$$GET1^DIQ(365,RPIEN\_",","DO NOT PURGE","I") Q**  .. S FOUND=1  .. S BEGDT=$P(DATE,".",1)  .. Q  . Q  ;  … INITX ;  Q  ; | | | | | | | |

| Current Logic |
| --- |
| PURGE ; This procedure is queued to run in the background and does the  ; actual purging. Variables available from the TaskMan call are:  ;  ; STATLIST = list of statuses that are OK to purge  ; BEGDT = beginning date for purging  ; ENDDT = ending date for purging  ;  ; First loop through the eIV Transmission Queue file and delete all  ; records in the date range whose status is in the list  ;  N CNT,DA,DATE,DIK,HLIEN,PFLAG,TQIEN,TQS    ;IB\*2.0\*549 added PFLAG  S DATE=$O(^IBCN(365.1,"AE",BEGDT),-1),CNT=0  F  S DATE=$O(^IBCN(365.1,"AE",DATE)) Q:'DATE!($P(DATE,".",1)>ENDDT)!$G(ZTSTOP) S TQIEN=0 F  S TQIEN=$O(^IBCN(365.1,"AE",DATE,TQIEN)) Q:'TQIEN  D  Q:$G(ZTSTOP)  . S CNT=CNT+1  . I $D(ZTQUEUED),CNT#100=0,$$S^%ZTLOAD() S ZTSTOP=1 Q  . S TQS=$P($G(^IBCN(365.1,TQIEN,0)),U,4) ; trans queue status  . I '$F(STATLIST,","\_TQS\_",") Q             ; must be in the list  . S PFLAG=$$GET1^DIQ(365,TQIEN\_",",.11,"I") ; Do Not Purge Flag IB\*2.0\*549 added line  . Q:+PFLAG                                  ; IB\*2.0\*549 added line  . ;  . ; loop through the HL7 messages multiple and kill any response  . ; records that are found for this transmission queue entry  . S HLIEN=0,DIK="^IBCN(365,"  . F  S HLIEN=$O(^IBCN(365.1,TQIEN,2,HLIEN)) Q:'HLIEN  D  .. S DA=$P($G(^IBCN(365.1,TQIEN,2,HLIEN,0)),U,3) I DA D ^DIK  .. Q  . ;  . ; now we can kill the transmission queue entry itself  . S DA=TQIEN,DIK="^IBCN(365.1," D ^DIK  . Q  ;  ; Check for a stop request  I $G(ZTSTOP) G PURGEX  ;  ; Now we must loop through the eIV Response file itself to purge any  ; response records that do not have a corresponding transmission  ; queue entry. These are the unsolicited responses. The status of  ; these responses is always 'response received' so we don't need to  ; check the status. For this loop, start from the very beginning of  ; the file.  ;  S DATE="",DIK="^IBCN(365,",CNT=0  F  S DATE=$O(^IBCN(365,"AE",DATE)) Q:'DATE!($P(DATE,".",1)>ENDDT)!$G(ZTSTOP) S DA=0 F  S DA=$O(^IBCN(365,"AE",DATE,DA)) Q:'DA  D  Q:$G(ZTSTOP)  . S CNT=CNT+1  . I $D(ZTQUEUED),CNT#100=0,$$S^%ZTLOAD() S ZTSTOP=1 Q  . ;  . ; If there is a pointer to the transmission queue file, then we  . ; should get out of this loop because the purpose of this section  . ; is to purge those responses with no link to the transmission  . ; queue file.  . ;  . I $P($G(^IBCN(365,DA,0)),U,5) Q  . D ^DIK  . Q  ; PURGEX ;  ; Tell TaskManager to delete the task's record  I $D(ZTQUEUED) S ZTREQ="@"  Q  ; |
| Modified Logic (Changes are in bold) |
| PURGE ; This procedure is queued to run in the background and does the  ; actual purging. Variables available from the TaskMan call are:  ;  ; STATLIST = list of statuses that are OK to purge  ; BEGDT = beginning date for purging  ; ENDDT = ending date for purging  ;  ; First loop through the eIV Transmission Queue file and delete all  ; records in the date range whose status is in the list  ;  N CNT,DA,DATE,DIK,HLIEN,PFLAG,TQIEN,TQS  S DATE=$O(^IBCN(365.1,"AE",BEGDT),-1),CNT=0  F  S DATE=$O(^IBCN(365.1,"AE",DATE)) Q:'DATE!($P(DATE,".",1)>ENDDT)!$G(ZTSTOP) S TQIEN=0 F  S TQIEN=$O(^IBCN(365.1,"AE",DATE,TQIEN)) Q:'TQIEN  D  Q:$G(ZTSTOP)  . S CNT=CNT+1  . I $D(ZTQUEUED),CNT#100=0,$$S^%ZTLOAD() S ZTSTOP=1 Q  . S TQS=$P($G(^IBCN(365.1,TQIEN,0)),U,4) ; trans queue status  . I '$F(STATLIST,","\_TQS\_",") Q             ; must be in the list  . ; loop through the HL7 messages multiple and kill any response  . ; records that are found for this transmission queue entry  . ; IB\*2.0\*602/DM Preserve any TQ and response that has DO NOT PURGE set to 1 (YES)  . S PFLAG=0,HLIEN=0,DIK="^IBCN(365,"  . F  S HLIEN=$O(^IBCN(365.1,TQIEN,2,HLIEN)) Q:'HLIEN  D  .. S DA=$P($G(^IBCN(365.1,TQIEN,2,HLIEN,0)),U,3) Q:'DA  .. I +$$GET1^DIQ(365,DA\_",","DO NOT PURGE","I") S PFLAG=1 Q  .. D ^DIK  .. Q  . ;  . ; now we can kill the transmission queue entry itself  . ; IB\*2.0\*602/DM as long as there was no DO NOT PURGE responses  . I 'PFLAG S DA=TQIEN,DIK="^IBCN(365.1," D ^DIK  . Q  ;  ; Check for a stop request  I $G(ZTSTOP) G PURGEX  ;  ; Now we must loop through the eIV Response file itself to purge any  ; response records that do not have a corresponding transmission  ; queue entry. These are the unsolicited responses. The status of  ; these responses is always 'response received' so we don't need to  ; check the status. For this loop, start from the very beginning of  ; the file.  ;  S DATE="",DIK="^IBCN(365,",CNT=0  F  S DATE=$O(^IBCN(365,"AE",DATE)) Q:'DATE!($P(DATE,".",1)>ENDDT)!$G(ZTSTOP) S DA=0 F  S DA=$O(^IBCN(365,"AE",DATE,DA)) Q:'DA  D  Q:$G(ZTSTOP)  . S CNT=CNT+1  . I $D(ZTQUEUED),CNT#100=0,$$S^%ZTLOAD() S ZTSTOP=1 Q  . ; **. ; IB\*2.0\*602/DM never drop a DO NOT PURGE response  . Q:+$$GET1^DIQ(365,DA\_",","DO NOT PURGE","I")**  . ; If there is a pointer to the transmission queue file, **. ; make sure the transmission queue record actually exists.  . ; If the TQ exists, quit this loop, if not, remove this response.**  . ; **. S TQIEN=+$$GET1^DIQ(365,DA\_",","TRANSMISSION QUEUE","I")  . I TQIEN,$D(^IBCN(365.1,TQIEN,0)) Q**   . D ^DIK  . Q  ; PURGEX ;  ; Tell TaskManager to delete the task's record  I $D(ZTQUEUED) S ZTREQ="@"  Q  ; |

Post Install:

*As part of the post install, we’ll queue a task to run the* ***FIXTQ*** *subroutine since it may take some time to complete.*

*The FIXTQ’s mission is to locate any abnormal IIV TRANSMISSION QUEUE (#365.1) records (TQ). For any found that are older than T-182 , the abnormal TQ record along with any corresponding IIV RESPONSE (#365) record will be removed.*

*For any abnormal TQ records found within T-182 through T-14, we’ll email the eIns insurance team (PII) a listing detailing the first 10 entries discovered along with a total count. The email must contain the site number (station). Additionally, if no abnormal records are discovered, we’ll still email the eIns team to let them know none were found. The purpose of the email is to provide the eInsurance team a sample of records (if any was found) that could point to a potential problem with the existing eInsurance software.*

*This is what constitutes an abnormal TQ record*

* *Any TQ record without a DATE/TIME CREATED (#365.1, .06)*
* *Any TQ record without a QUERY FLAG (#365.1,.11)*
* *Any TQ record with a QUERY FLAG that is set to “I” (Identification) as this aspect of eIV was terminated long time ago.*
* *Any TQ record that has a TRANSMISSION STATUS (#365.1, .04) that is NOT equal to “Response Received” and not equal to “Communication Failure” and not equal to “Cancelled”. There is no reason for any TQ entry more than 6 months ago to have any status other than those statuses listed above. If they are, they can’t be transmitted to FSC as a 270 message for one reason or another.*

| Routines | Activities | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | **IBY602PO** | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** |  | | | | | | | | |
| **Related Options** |  | | | | | | | | |
| Related Routines | Routines “Called By” | | | | | Routines “Called” | | | |
|  | Patch Install Post | | | | | ^%DT  ^%ZTLOAD  ^DIC, ^DIK, ^DIQ  ^IBCNEUT5  ^VASITE  ^XLFDT  ^XPDUTL  ^XUPROD | | | |
| Routines | Activities | | | | | | | | |
| **Data Dictionary (DD) References** | ^IBCN(365.1 | | | | | | | | |
| **Related Protocols** |  | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** |  | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| Current Logic | | | | | | | | | |
| N/A | | | | | | | | | |
| Modified Logic (Changes are in bold) | | | | | | | | | |
| **IBY602PO ;EDE/DM - Post-Installation for IB\*2.8\*602 ; 23-MAR-2018  ;;2.0;INTEGRATED BILLING;\*\*602\*\*;09-AUG-2018  ;;Per VA Directive 6402, this routine should not be modified.  ;** | | | | | | | | | |
| Modified Logic (Changes are in bold) | | | | | | | | | |
| **;  ; This routine handles the purging of the eIV data stored in the  ; eIV Transmission Queue file (#365.1), the eIV Response file (#365) and  ; the EIV EICD TRACKING file (#365.18) IB\*2.0\*621/DM  ; User can pick a date range for the purge. Data created within 6 months  ; cannot be purged. The actual global kills are done by a background  ; task after hours (8:00pm).  ; EN ;  NEW STOP,BEGDT,ENDDT,STATLIST,IBVER  S IBVER=1  D INIT I STOP G EXIT       ; initialize/calculate default dates  D DEFLT I STOP G EXIT      ; allow user to change default end date if test system ;IB\*2.0\*621  D BEGDT I STOP G EXIT      ; user interface for beginning date  D ENDDT I STOP G EXIT      ; user interface for ending date  D CONFIRM I STOP G EXIT    ; confirmation message/final check  D QUEUE                    ; queuing process EXIT ;  Q  ; EN1 ; Automated Monthly Purge \*IB\*2\*595  NEW STOP,BEGDT,ENDDT,STATLIST,IBVER  S IBVER=2  D INIT I STOP G EXIT1       ; initialize/calculate default dates  D QUEUE                    ; queuing process EXIT1 ;  Q PURGE ; This procedure is queued to run in the background and does the  ; actual purging. Variables available from the TaskMan call are:  ;  ; STATLIST = list of statuses that are OK to purge  ; BEGDT = beginning date for purging  ; ENDDT = ending date for purging  ;  ; First loop through the eIV Transmission Queue file and delete all  ; records in the date range whose status is in the list  ;  N CNT,DA,DATE,DIK,HLIEN,PFLAG,TQIEN,TQS   ;IB\*2.0\*549 added PFLAG  N IBWEXT,IBIORV  S DATE=$O(^IBCN(365.1,"AE",BEGDT),-1),CNT=0  F  S DATE=$O(^IBCN(365.1,"AE",DATE)) Q:'DATE!($P(DATE,".",1)>ENDDT)!$G(ZTSTOP) S TQIEN=0 F  S TQIEN=$O(^IBCN(365.1,"AE",DATE,TQIEN)) Q:'TQIEN  D  Q:$G(ZTSTOP)  . S CNT=CNT+1  . I $D(ZTQUEUED),CNT#100=0,$$S^%ZTLOAD() S ZTSTOP=1 Q  . S TQS=$P($G(^IBCN(365.1,TQIEN,0)),U,4) ; trans queue status  . S IBWEXT=$P($G(^IBCN(365.1,TQIEN,0)),U,10) ; IB\*2.0\*621/DM WHICH EXTRACT  . S IBIORV=$P($G(^IBCN(365.1,TQIEN,0)),U,11) ; IB\*2.0\*621/DM QUERY FLAG  . I IBWEXT=4,IBIORV="V" Q                    ; skip EICD Verification entries as they   . ; will be addressed with EICD Identifications  . I '$F(STATLIST,","\_TQS\_",") Q              ; must be in the list  . I IBWEXT=4,IBIORV="I" D CHKTRK(TQIEN) Q    ; check EIV EICD TRACKING for purge  . ; loop through the HL7 messages multiple and kill any response  . ; records that are found for this transmission queue entry  . ; IB\*2.0\*621/DM Preserve any TQ and response that has DO NOT PURGE set to 1 (YES)  . S PFLAG=0,HLIEN=0,DIK="^IBCN(365,"  . F  S HLIEN=$O(^IBCN(365.1,TQIEN,2,HLIEN)) Q:'HLIEN  D  .. S DA=$P($G(^IBCN(365.1,TQIEN,2,HLIEN,0)),U,3) Q:'DA  .. I +$$GET1^DIQ(365,DA\_",",.11,"I") S PFLAG=1 Q  ;"DO NOT PURGE"  .. D ^DIK  .. Q  . ;  . ; now we can kill the transmission queue entry itself  . ; as long as there was no DO NOT PURGE responses IB\*2.0\*621/DM   . I 'PFLAG S DA=TQIEN,DIK="^IBCN(365.1," D ^DIK K DA,DIK  . Q  ;  ; Check for a stop request  I $G(ZTSTOP) G PURGEX  ;  ; Now we must loop through the eIV Response file itself to purge any  ; response records that do not have a corresponding transmission  ; queue entry. These are the unsolicited responses. The status of  ; these responses is always 'response received' so we don't need to  ; check the status. For this loop, start from the very beginning of  ; the file.  ;  S DATE="",DIK="^IBCN(365,",CNT=0  F  S DATE=$O(^IBCN(365,"AE",DATE)) Q:'DATE!($P(DATE,".",1)>ENDDT)!$G(ZTSTOP) S DA=0 F  S DA=$O(^IBCN(365,"AE",DATE,DA)) Q:'DA  D  Q:$G(ZTSTOP)  . S CNT=CNT+1  . I $D(ZTQUEUED),CNT#100=0,$$S^%ZTLOAD() S ZTSTOP=1 Q  . ;  . ; If there is a pointer to the transmission queue file, then we  . ; should get out of this loop because the purpose of this section  . ; is to purge those responses with no link to the transmission  . ; queue file.  . ;  . I $P($G(^IBCN(365,DA,0)),U,5) Q  . D ^DIK  . Q  ;  K DA,DIK PURGEX ;  ; Tell TaskManager to delete the task's record  I $D(ZTQUEUED) S ZTREQ="@"  Q  ; INIT ; This procedure calculates the default beginning and ending dates  ; and displays screen messages about this option to the user.  ;  NEW DATE,FOUND,TQIEN,TQS,RPIEN,RPS,IBHL7,IBDNP  NEW DIR,X,Y,DTOUT,DUOUT,DIRUT,DIROUT  ;  S STOP=0  ;  ; This is the list of statuses that are OK to purge  ; 3=Response Received  ; 5=Communication Failure  ; 7=Cancelled  S STATLIST=","\_$$FIND1^DIC(365.14,,"B","Response Received")  S STATLIST=STATLIST\_","\_$$FIND1^DIC(365.14,,"B","Communication Failure")  S STATLIST=STATLIST\_","\_$$FIND1^DIC(365.14,,"B","Cancelled")\_","  ;  ; Try to find a beginning date in the eIV Transmission Queue file  S DATE="",FOUND=0,BEGDT=DT  F  S DATE=$O(^IBCN(365.1,"AE",DATE)) Q:'DATE!FOUND  S TQIEN=0 F  S TQIEN=$O(^IBCN(365.1,"AE",DATE,TQIEN)) Q:'TQIEN  D  Q:FOUND  . S TQS=$P($G(^IBCN(365.1,TQIEN,0)),U,4) ; status  . I '$F(STATLIST,","\_TQS\_",") Q  . ;IB\*2.0\*602/DM make sure the default earliest date is not a DO NOT PURGE entry   . ;check the HL7 messages multiple to see if DO NOT PURGE is set on any response  . S (IBDNP,IBHL7)=0  . F  S IBHL7=$O(^IBCN(365.1,TQIEN,2,IBHL7)) Q:'IBHL7!IBDNP  D  .. S RPIEN=$P($G(^IBCN(365.1,TQIEN,2,IBHL7,0)),U,3) Q:'RPIEN  .. I +$$GET1^DIQ(365,RPIEN\_",","DO NOT PURGE","I") S IBDNP=1  .. Q  . ;  . I IBDNP,IBVER=2 Q   . I IBDNP W !,"Please wait, checking for the earliest purge date ...",! Q  . ;  . S FOUND=1  . S BEGDT=$P(DATE,".",1)  . Q  ;  ; If not successful, try to find a beginning date in the eIV Response file.  I 'FOUND D  . S DATE=""  . F  S DATE=$O(^IBCN(365,"AE",DATE)) Q:'DATE!FOUND  S RPIEN=0 F  S RPIEN=$O(^IBCN(365,"AE",DATE,RPIEN)) Q:'RPIEN  D  Q:FOUND  .. S RPS=$P($G(^IBCN(365,RPIEN,0)),U,6) ; status  .. I '$F(STATLIST,","\_RPS\_",") Q  .. ;IB\*2.0\*602/DM do not choose a DO NOT PURGE response   .. I +$$GET1^DIQ(365,RPIEN\_",","DO NOT PURGE","I") Q  .. S FOUND=1  .. S BEGDT=$P(DATE,".",1)  .. Q  . Q  ;  ; default end date, Today minus 182 days (approx 6 months)  S ENDDT=$$FMADD^XLFDT(DT,-182)  ;  ;I IBVER=1,'FOUND!(BEGDT>ENDDT) D S STOP=1 G INITX ; IB\*2.0\*621  I IBVER=1,'FOUND,'$$PROD^XUPROD(1)!(BEGDT>ENDDT) D  S STOP=1 G INITX  . W !!?5,"Purging of eIV data is not possible at this time."  . I 'FOUND W !?5,"There are no entries in the file that are eligible to be",!?5,"purged or there is no data in the file."  . E  W !?5,"The oldest date in the file is ",$$FMTE^XLFDT(BEGDT,"5Z"),".",!?5,"Data cannot be purged unless it is at least 6 months old."  . W ! S DIR(0)="E" D ^DIR K DIR  . Q  I IBVER=2,'FOUND!(BEGDT>ENDDT) D  S STOP=1 G INITX  .; Send a MailMan message with Eligible Purge counts ; IB\*2.0\*621 - Updated Message  .N MGRP,MSG,IBXMY  .S MSG(1)="Purge Electronic Insurance Verification (eIV) Data Files did not find records"  .S MSG(2)="for station "\_+$$SITE^VASITE()\_"."  .S MSG(3)=""  .S MSG(4)="The option runs automatically on a monthly basis and purges data from the"  .S MSG(5)="IIV RESPONSE file (#365), the IIV TRANSMISSION QUEUE file (#365.1), and the"  .S MSG(6)="EIV EICD TRACKING file (#365.18). The data must be at least six months old"  .S MSG(7)="before it can be purged. Only insurance transactions that have a transmission"  .S MSG(8)="status of ""Response Received"", ""Communication Failure"", or ""Cancelled"""  .S MSG(9)="may be purged."  .; Set to IB site parameter MAILGROUP - IBCNE EIV MESSAGE  .S MGRP=$$MGRP^IBCNEUT5()  .S IBXMY("PII")=""  .D MSG^IBCNEUT5(MGRP,"eIV Purge No Data Found for Station "\_+$$SITE^VASITE(),"MSG(",,.IBXMY)  .; Duplicate message to Outlook group  .; S MGRP="PII"  .; D MSG^IBCNEUT5(MGRP,"eIV Data Background Purge","MSG(")  .Q  ;  ; At this point, we know that there are some entries eligible for  ; purging. Display a message to the user about this option.  I IBVER=2 G INITX  W @IOF  W !?8,"Purge Electronic Insurance Verification (eIV) Data Files"  W !!!," This option will allow you to purge data from the eIV Response File (#365)"  W !," and the eIV Transmission Queue File (#365.1). The data must be at least six"  W !," months old before it can be purged. Only insurance transactions that have a"  W !," transmission status of ""Response Received"", ""Communication Failure"", or"  W !," ""Cancelled"" may be purged. You will be allowed to select a date range for"  W !," this purging. The default beginning date will be the date of the oldest"  W !," eligible record in the system. The default ending date will be six months"  W !," ago from today's date. You may modify this default date range. However, you"  W !," may not select an ending date that is more recent than six months ago."  W !! INITX ;  Q  ; DEFLT ; IB\*621/DW Added to assist with testing  I IBVER=1,('$$PROD^XUPROD(1)) D  . W ?5,"\*\*\* For Test Purposes Only:"  . W !!?5,"In test systems one may override the DEFAULT end date."  . W !!?5,"Current default end date is TODAY - 182 DAYS: "\_$$FMTE^XLFDT(ENDDT,"5Z"),!!  . NEW DIR,X,Y,DTOUT,DUOUT,DIRUT,DIROUT  . S DIR(0)="DOA^"\_BEGDT\_":"\_DT\_":AEX"  . S DIR("A")="Enter the purge default date: "  . S DIR("B")=$$FMTE^XLFDT(ENDDT,"5Z")  . S DIR("?")="This response must be a date between "\_$$FMTE^XLFDT(BEGDT,"5Z")\_" and "\_$$FMTE^XLFDT(DT,"5Z")\_"."  . D ^DIR K DIR  . I $D(DIRUT)!'Y S STOP=1 G DEFLTX  . S ENDDT=Y  W !!! DEFLTX ;  Q  ; BEGDT ; This procedure captures the beginning date from the user.  NEW DIR,X,Y,DTOUT,DUOUT,DIRUT,DIROUT  S DIR(0)="DOA^"\_BEGDT\_":"\_ENDDT\_":AEX"  S DIR("A")="Enter the purge begin date: "  S DIR("B")=$$FMTE^XLFDT(BEGDT,"5Z")  S DIR("?")="This response must be a date between "\_$$FMTE^XLFDT(BEGDT,"5Z")\_" and "\_$$FMTE^XLFDT(ENDDT,"5Z")\_"."  D ^DIR K DIR  I $D(DIRUT)!'Y S STOP=1 G BEGDTX  S BEGDT=Y BEGDTX ;  Q  ; ENDDT ; This procedure captures the ending date from the user.  NEW DIR,X,Y,DTOUT,DUOUT,DIRUT,DIROUT  W !  S DIR(0)="DOA^"\_BEGDT\_":"\_ENDDT\_":AEX"  S DIR("A")=" Enter the purge end date: "  S DIR("B")=$$FMTE^XLFDT(ENDDT,"5Z")  S DIR("?")="This response must be a date between "\_$$FMTE^XLFDT(BEGDT,"5Z")\_" and "\_$$FMTE^XLFDT(ENDDT,"5Z")\_"."  D ^DIR K DIR  I $D(DIRUT)!'Y S STOP=1 G ENDDTX  S ENDDT=Y ENDDTX ;  Q  ; CONFIRM ; This procedure displays a confirmation message to the user and  ; asks if it is OK to proceed with the purge.  NEW DIR,X,Y,DTOUT,DUOUT,DIRUT,DIROUT  W !!!," You want to purge all eIV data created between "  W $$FMTE^XLFDT(BEGDT,"5Z")," and ",$$FMTE^XLFDT(ENDDT,"5Z"),"."  W !  S DIR(0)="YO",DIR("A")=" OK to continue"  S DIR("B")="NO"  D ^DIR K DIR  I 'Y S STOP=1 CONFX ;  Q  ; QUEUE ; This procedure queues the purge process for later at night.  ; The concept for queuing the purge came from the insurance buffer  ; purge routine, IBCNBPG. That purge process is also hard-coded to  ; be run at 8:00 PM just like this one is.  ;  NEW ZTRTN,ZTDESC,ZTDTH,ZTIO,ZTUCI,ZTCPU,ZTPRI,ZTSAVE,ZTKIL,ZTSYNC,ZTSK  NEW DIR,X,Y,DTOUT,DUOUT,DIRUT,DIROUT  ;  ; IB\*621/DW Added loop below to assist with testing  I IBVER=1,('$$PROD^XUPROD(1)) D  I Y D PURGE^IBCNEKIT G QUEUEX  . W !!!!,"\*\*\* TEST System only - you may run this immediately",!  . S DIR("A")="Do you want to run this now instead of tasking it for 8:00pm"  . S DIR(0)="Y",DIR("B")="YES"  . D ^DIR  . I Y="^" S STOP=1  ;  I STOP G QUEUEX              ; IB\*2.0\*621  S ZTRTN="PURGE^IBCNEKIT"     ; TaskMan task entry point  S ZTDESC="Purge eIV Data"    ; Task description  S ZTDTH=DT\_".20"             ; start it at 8:00 PM tonight  S ZTIO=""  S ZTSAVE("BEGDT")=""  S ZTSAVE("ENDDT")=""  S ZTSAVE("STATLIST")=""  D ^%ZTLOAD  I IBVER=2 G QUEUEX  I $G(ZTSK) W !!," Task# ",ZTSK," has been scheduled to purge the eIV data tonight at 8:00 PM."  E  W !!," TaskManager could not schedule this task.",!," Contact IRM for technical assistance."  W ! S DIR(0)="E" D ^DIR K DIR QUEUEX ;  Q  ; CHKTRK(IBTQ1) ; IB\*621, Evaluate associated records for one EICD transaction  ; IBTQ1 = EICD Identification TQ IEN  ;  N FILE,HLIEN,IBTQIEN1,IBTQIEN2,IBFIELDS,IBPURGE,IBSKIP,IBTQIEN,IBTQS  N IBTRKIEN,PFLAG  ;  S (IBSKIP,PFLAG)=0  K IBPURGE  S IBTQIEN1=+$$FIND1^DIC(365.18,,"QX",IBTQ1,"B")  Q:'IBTQIEN1  ; the passed TQ IEN is not in the tracking file  S IBPURGE("EICD",365.1,IBTQ1)=""               ;EICD TQ for identifications  S IBTQIEN=+$$GET1^DIQ(365.18,IBTQIEN1,.06,"I") ;EICD RESPONSE for identifications  I IBTQIEN S IBPURGE("EICD",365,IBTQIEN)=""  ;   ; loop through the EICD verification entries looking for exclusions   S IBTRKIEN=0 F  S IBTRKIEN=$O(^IBCN(365.18,IBTQIEN1,"INS-FND",IBTRKIEN)) Q:'IBTRKIEN  D  Q:IBSKIP  . ;  . ; check the 1 node data for associated TQs & their responses  . S IBTQIEN2=IBTRKIEN\_","\_IBTQIEN1\_","  . K IBFIELDS D GETS^DIQ(365.185,IBTQIEN2,"1.01:1.04","I","IBFIELDS")  . ;  . I IBFIELDS(365.185,IBTQIEN2,1.02,"I")="" Q                ; No TQ was created  . I IBFIELDS(365.185,IBTQIEN2,1.02,"I")>ENDDT S IBSKIP=1 Q  ; TQ not old enough   . S IBTQIEN=+IBFIELDS(365.185,IBTQIEN2,1.01,"I") ; EICD VER INQ TQ  . S IBTQS=+$$GET1^DIQ(365.1,IBTQIEN\_",",.04,"I") ; TQ Transmission Status   . I IBTQS,('$F(STATLIST,","\_IBTQS\_",")) S IBSKIP=1 Q        ; must be in the list  . ;  . ; Loop thru all EICD Verifications if any are DO NOT PURGE then kill  . ; nothing associated with it  . S HLIEN=0  . F  S HLIEN=$O(^IBCN(365.1,IBTQIEN,2,HLIEN)) Q:'HLIEN!PFLAG  D  .. S DA=$P($G(^IBCN(365.1,IBTQIEN,2,HLIEN,0)),U,3) Q:'DA  .. I +$$GET1^DIQ(365,DA\_",",.11,"I") S PFLAG=1 Q  ;"DO NOT PURGE"  .. S IBPURGE("EICD",365,DA)=""  ; array of Verifications to purge (responses)  . I PFLAG Q  . S IBPURGE("EICD",365.1,IBTQIEN)="" ; array of Verifications to purge (inquiries)  ;  I PFLAG!IBSKIP K IBPURGE  ; DO NOT PURGE is set or Not all records are old enough  ;  I '$D(IBPURGE) Q  ; No records associated with this entry to purge  S IBPURGE("EICD",365.18,IBTQ1)=""  S FILE="" F  S FILE=$O(IBPURGE("EICD",FILE)) Q:'FILE  D  . S DIK="^IBCN("\_FILE\_","  . S DA="" F  S DA=$O(IBPURGE("EICD",FILE,DA)) Q:'DA  D  .. D ^DIK  K IBPURGE,DA,DIK  Q  ;** | | | | | | | | | |

# Revision History

| Date | Version | Description | Author |
| --- | --- | --- | --- |
| 11/29/2018 | 1.2 | After IOC Testing | Henry Normand |
| 03/27/2018 | 1.1 | Formal SDD, pre-IOC | Daniel Moran |
| 02/27/2018 | 1.0 | Initial draft | Daniel Moran |