**MCCF EDI TAS US22**

System Design Document



Department of Veterans Affairs

**April 2017**

Version 1.00

**User Story Number:** USRX-186

**User Story Name:**

**Product Backlog ID:** 186

**Backlog Priority:** High

**Initial Sizing Estimate:**

**Rational ID:** tbd

**Rally ID:** US22

# Design – Summary

Overview of the CMOP process as it pertains to this story:

When the nightly CMOP batch process runs, the CMOP system (PSX) loops through all the prescriptions in the queue. The batch calls Outpatient Pharmacy (specifically ECMESND^PSOBPSU1) to send a claim for each billable prescription. After it has gone through the entire queue, the CMOP process waits for 15 seconds for each claim sent, waiting a maximum of two hours. After waiting, if the status of a prescription’s claim is still “IN PROGRESS” (meaning a response has not yet been received for that claim), then the CMOP process puts that prescription back in the suspense queue. The CMOP process then sends to the CMOP facility a batch of prescriptions to be filled.

Background of the problem this story is seeking to remedy:

During the CMOP process, TRICARE/CHAMPVA prescriptions are sometimes being left in the suspense queue due to the claims having a status of “IN PROGRESS”. The claim submission process currently does the following:

* An email is sent to users indicating that a specific Rx/Fill, “could not be filled because of a delay in processing the third-party claim.” (See BULL1^BPSNCPD1.)
* An entry is added to the prescription’s Activity Log sub-file stating, “Rx placed on Suspense due to ECME IN PROGRESS status”. (See TACT^PSOREJU3.)

The CMOP process currently does the following:

* An email is sent to the users listing all prescriptions that were not filled but put back on suspense. (See MAIL^PSXBPSMS.)
* The prescription is put back on suspense and is not sent to the CMOP facility.

These things happen at different points in the process. In most instances, the problem occurs because something is being checked prematurely. If a claim is still “IN PROGRESS”, the determination of whether to leave the prescription in the suspense queue should not be made until after the CMOP process waits 15 seconds for each claim sent. There are three checks happening prior to that point.

This design calls for:

* Modifying a check in the claim submission process to include the BWHERE/FROM value of “PC”. This will prevent the claim submission process from sending the email to the users.
* Quitting out of a procedure early if the value of BWHERE/FROM is “PC” and the ECME status is “IN PROGRESS”. This will prevent the claim submission process from adding an entry to the prescription’s Activity Log.
* Modifying a check in the CMOP process to perform additional checks on a TRICARE/CHAMPVA prescription only if the ECME status is not “IN PROGRESS”. Modifying this check will result in the CMOP process *not* putting a prescription back on suspense inappropriately.
* Modifying the CMOP process to add an entry in the Activity Log of a prescription if it is being left on suspense due to a status of “IN PROGRESS” (since the creation of this Activity Log entry is being removed from the claim submission process).

List of Components:

* Routine: BPSNCPDP
* Routine: PSOREJU3
* Routine: PSXBPSMS
* Routine: PSXRPPL1
* Routine: PSXRPPL2

# Design – Detail

In END^BPSNCPDP, after the claim has been created and sent, and after it has waited the length of time specified by the ECME TIMEOUT value, if the claim status is still “IN PROGRESS”, then the system conditionally sends an email bulletin to communicate that the Rx/Fill could not be filled and has been place on suspense due to a delay in processing the third party claim. At this point in the process, it does not put the Rx/Fill on suspense; it simply sends the email if certain conditions are met.

The system currently does *not* send that email bulletin if the BWHERE/FROM variable is “CRLB”, “CRLR”, “CRLX”, “CRRL”, or “RRL”. This enhancement calls for adding to that list the value “PC” (Pull CMOPS). This is the value of BWHERE/FROM when the claim submission process in ECME/BPS is called from CMOP/PSX, including the nightly CMOP transmission run and the user selecting the CMOP pull from suspense.

If the value of “PC” is added to the list of possible BWHERE/FROM values described above, then the system will not call BULL^BPSNCPD1 when CMOP processing is performed, and it will not send the email bulletin. The CMOP process itself will continue to check for any claims that are still “IN PROGRESS”, as it does now, but that check will not happen until all the prescriptions have been sent and the process has waited an additional 15 seconds for every claim sent, waiting up to two hours. (The sending of claims is in SBTEMCE^PSXRPPL1, and the waiting is in SDT^PSXRPPL.)

| Subroutine Name | **EN^BPSNCPDP** | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Enhancement Category | New | Modify | | | Delete | | No Change | |
| Story | US22 | | | | | | | |
| Related Menu Options or ListMan Actions | Pull Early from Suspense [PSO PNDRX]  Print from Suspense File [PSO PNDLBL] | | | | | | | |
| Related Routines | This Subroutine is Called By | | | This Subroutine Calls | | | | |
| REVERSE^BPSBCKJ, REQST99^BPSOSRX5, SUBMCLM^BPSPRRX2, DOSELCTD^BPSRES, RESUBMIT^BPSSCRRS, REVERSE^BPSSCRRV, SUBMIT^IBNCPDPU, ECMESND^PSOBPSU1, REVERSE^PSOBPSU1 | | | CLOSE2^BPSBUTL, BULL^BPSNCPD1, $$STTM^BPSNCPD4, DISPL^BPSNCPD4, $$SCHREQ^BPSNCPD5, BB^BPSNCPD6, NEWCLAIM^BPSNCPD6, RSPAID^BPSNCPD6, RVNEW^BPSNCPD6, RVNPAID^BPSNCPD6, RVPAID^BPSNCPD6, RVRSNPD^BPSNCPD6, RVRSPAID^BPSNCPD6, LOG^BPSOSL, $$IEN59^BPSOSRX, $$LOCKRF^BPSOSRX, $$STATUS^BPSOSRX, UNLCKRF^BPSOSRX, $$ACTTYPE^BPSOSRX5, $$PAYABLE^BPSOSRX5, $$CHKREQST^BPSOSRX7, $$CHCKPAR^BPSOSRX8, $$GETSITE^BPSOSRX8, $$PRINTSCR^BPSOSRX8, $$FINDECLM^BPSPRRX5, $$SECDATA^BPSPRRX6, $$DOSDATE^BPSSCRRS, $$RXAPI1^BPSUTIL1, $$CLMINFO^BPSUTIL2, $$GET1^DIQ, $$GETNDC^PSONDCUT, $$DT^XLFDT, $$PROD^XUPROD | | | | |
| Related Integration Control Registrations (ICRs) | Reference to $$PROD^XUPROD supported by DBIA 4440  Reference to $$GETNDC^PSONDCUT supported by DBIA 4705  Reference to Patient file (#2) supported by DBIA 10035 | | | | | | | |
| Data Passing | Parameter Input | | Parameter Output | | | Function Return Value | | Global Modified |
| Input Attribute Name and Definition | Name: BRXIEN  Definition: Prescription IEN  New  Modify  Delete  No Change  Name: BFILL  Definition: Fill number  New  Modify  Delete  No Change  Name: DOS  Definition: Date of service  New  Modify  Delete  No Change  Name: BWHERE  Definition: User or system action initiating this claim process  New  Modify  Delete  No Change  Name: BILLNDC  Definition: Valid NDC  New  Modify  Delete  No Change  Name: REVREAS  Definition: Reversal reason  New  Modify  Delete  No Change  Name: DURREC  Definition: String of up to three sets of DUR info  Reason for Service Code ^ Prof. Srvc. Code ^ Result of Srvc. Code  Each set is delimited by “~”  New  Modify  Delete  No Change  Name: BPOVRIEN  Definition: Point to BPS NCPDP OVERRIDE file, if overrides entered  New  Modify  Delete  No Change  Name: BPSCLARF  Definition: Submission clarification code entered by pharmacist  New  Modify  Delete  No Change  Name: BPSAUTH  Definition: Prior authorization code  New  Modify  Delete  No Change  Name: BPCOBIND  Definition: COB indicator  New  Modify  Delete  No Change  Name: BPJOBFLG  Definition: “F” if in foreground, “B” if in background  New  Modify  Delete  No Change  Name: BPREQIEN  Definition: BPS Request  New  Modify  Delete  No Change  Name: BPSCLOSE  Definition: Array populated only if claim should be closed after reversal  New  Modify  Delete  No Change  Name: BPSPLAN  Definition: Pointer to GROUP INSURANCE PLAN file  New  Modify  Delete  No Change  Name: BPSPRDAT  Definition: Array of primary claim data needed for submitting a secondary claim  New  Modify  Delete  No Change  Name: BPSRTYPE  Definition: Rate type  New  Modify  Delete  No Change  Name: BPSDELAY  Definition: Delay reason code  New  Modify  Delete  No Change | | | | | | | |
| Output Attribute Name and Definition | Name: $$EN  Definition: RESPONSE ^ MESSAGE ^ ELIGIBILITY ^ CLAIMSTATUS ^ COB ^ RXCOB ^ INSURANCE  RESPONSE  0 Submitted through ECME  1 No submission through ECME  2 IB not billable  3 Claim was closed, not submitted (RTS/Deletes)  4 Unable to queue claim  5 Incorrect information supplied to ECME  6 Inactive ECME - Primarily used for TRICARE/CHAMPVA to say ok to process rx  10 Reversal but no resubmit  MESSAGE = Message associated with the response (error/submitted)  ELIGIBILITY = V - Veteran, T - TRICARE, C - CHAMPVA  CLAIMSTATUS = claim status (null or IN PROGRESS/E PAYABLE/etc...)  COB = Coordination Of Benefit indicator of the insurance as it is stored in the PATIENT file: 1-primary, 2-secondary, 3-tertiary  RXCOB = the payer sequence indicator of the claim which was sent to the payer as a result of this call: 1-primary, 2-secondary)  INSURANCE = Name of the insurance company that was billed as a result of this call  New  Modify  Delete  No Change | | | | | | | |
| Current Logic | | | | | | | | |
| BPSNCPDP ;BHAM ISC/LJE/SS - API to submit a claim to ECME ;11/7/07 16:58  ;;1.0;E CLAIMS MGMT ENGINE;\*\*1,3,4,2,5,6,7,8,10,11,19,20\*\*;JUN 2004;Build 27  ;;Per VA Directive 6402, this routine should not be modified.  ;  ; Reference to $$PROD^XUPROD supported by DBIA 4440  ; Reference to $$GETNDC^PSSNDCUT supported by DBIA 4707  ...  EN(BRXIEN,BFILL,DOS,BWHERE,BILLNDC,REVREAS,DURREC,BPOVRIEN,BPSCLARF,BPSAUTH,BPCOBIND,BPJOBFLG,  BPREQIEN,BPSCLOSE,BPSPLAN,BPSPRDAT,BPSRTYPE,BPSDELAY) ;  ...  ;  ; Initialize log  D LOG^BPSOSL(IEN59,$T(+0)\_"-Start of claim","DT")  D LOG^BPSOSL(IEN59,$T(+0)\_"-Job flag = "\_BPJOBFLG\_$S(BPJOBFLG="B":" BPS REQUEST ien = "\_$G(BPREQIEN),1:""))  ;  ...  END ;  ; BPSELIG and other variables are established by inference in BPSNCPD6.  I BPJOBFLG="F",BPLCK D UNLCKRF^BPSOSRX(BRXIEN,BFILL,$G(IEN59),$T(+0)) S BPLCK=0  ; Get Site in case we send a Bulletin  S SITE=$$GETSITE^BPSOSRX8(BRXIEN,BFILL)  ;if foreground AND we can't schedule request for any reason AND this is not OP - send bulletin  I BPJOBFLG="F",RESPONSE=4,",AREV,BB,ERES,ERWV,ERNB,EREV,P2,P2S,"'[(","\_BWHERE\_",") D BULL^BPSNCPD1(BRXIEN,BFILL,$G(SITE),$G(DFN),$G(PNAME),"",$G(CLMSTAT),$G(RESPONSE),$G(BPSCOB))  I $G(BPSELIG)="" S BPSELIG=""  ; Send Bulletin if TRICARE or CHAMPVA is IN PROGRESS and this is not a release process  S BPSSTAT=$S($G(BRXIEN):$P($$STATUS^BPSOSRX(BRXIEN,BFILL,,,BPSCOB),U),1:"")  I BPSELIG="T"!(BPSELIG="C"),BPSSTAT="IN PROGRESS",$G(REVREAS)'="RX RELEASE-NDC CHANGE",",CRLB,CRLR,CRLX,CRRL,RRL,"'[(","\_BWHERE\_",") D BULL^BPSNCPD1(BRXIEN,BFILL,SITE,$G(DFN),$G(PNAME),BPSELIG,"","",$G(BPSCOB))  ;  S:'$D(RESPONSE) RESPONSE=1  K MOREDATA  I $G(IEN59) D  . N MSG  . S MSG="Foreground Process Complete-RESPONSE="\_$G(RESPONSE)  . I $G(RESPONSE)'=0 S MSG=MSG\_", CLMSTAT="\_$G(CLMSTAT)  . D LOG^BPSOSL(IEN59,$T(+0)\_"-"\_MSG)  Q RESPONSE\_U\_$G(CLMSTAT)\_U\_BPSELIG\_U\_BPSSTAT\_U\_$$CLMINFO^BPSUTIL2(+$G(IEN59)) | | | | | | | | |
| Modified Logic (Changes are highlighted) | | | | | | | | |
| BPSNCPDP ;BHAM ISC/LJE/SS - API to submit a claim to ECME ;11/7/07 16:58  ;;1.0;E CLAIMS MGMT ENGINE;\*\*1,3,4,2,5,6,7,8,10,11,19,20\*\*;JUN 2004;Build 27  ;;Per VA Directive 6402, this routine should not be modified.  ;  ; Reference to $$PROD^XUPROD supported by DBIA 4440  ; Reference to $$GETNDC^PSONDCUT supported by DBIA 4705  ...  EN(BRXIEN,BFILL,DOS,BWHERE,BILLNDC,REVREAS,DURREC,BPOVRIEN,BPSCLARF,BPSAUTH,BPCOBIND,BPJOBFLG,  BPREQIEN,BPSCLOSE,BPSPLAN,BPSPRDAT,BPSRTYPE,BPSDELAY) ;  ...  ;  ; Initialize log  D LOG^BPSOSL(IEN59,$T(+0)\_"-Start of claim","DT")  D LOG^BPSOSL(IEN59,$T(+0)\_"-BWHERE = "\_BWHERE)  D LOG^BPSOSL(IEN59,$T(+0)\_"-Job flag = "\_BPJOBFLG\_$S(BPJOBFLG="B":" BPS REQUEST ien = "\_$G(BPREQIEN),1:""))  ;  ...  END ;  ; BPSELIG and other variables are established by inference in BPSNCPD6.  I BPJOBFLG="F",BPLCK D UNLCKRF^BPSOSRX(BRXIEN,BFILL,$G(IEN59),$T(+0)) S BPLCK=0  ; Get Site in case we send a Bulletin  S SITE=$$GETSITE^BPSOSRX8(BRXIEN,BFILL)  ;if foreground AND we can't schedule request for any reason AND this is not OP - send bulletin  I BPJOBFLG="F",RESPONSE=4,",AREV,BB,ERES,ERWV,ERNB,EREV,P2,P2S,"'[(","\_BWHERE\_",") D BULL^BPSNCPD1(BRXIEN,BFILL,$G(SITE),$G(DFN),$G(PNAME),"",$G(CLMSTAT),$G(RESPONSE),$G(BPSCOB))  I $G(BPSELIG)="" S BPSELIG=""  ; Send Bulletin if TRICARE or CHAMPVA is IN PROGRESS and this is not a release process  S BPSSTAT=$S($G(BRXIEN):$P($$STATUS^BPSOSRX(BRXIEN,BFILL,,,BPSCOB),U),1:"")  ;  ; \*\*\* Remove the below after BPS\*1.0\*22 changes have been confirmed to work \*\*\*  ;  I BPSELIG="T"!(BPSELIG="C"),BPSSTAT="IN PROGRESS",$G(REVREAS)'="RX RELEASE-NDC CHANGE",",BWHERE="PC" D LOG^BPSOSL(IEN59,$T(+0)\_"-BPS\*1.0\*22 A”)  ;  ; \*\*\* Remove the above after BPS\*1.0\*22 changes have been confirmed to work \*\*\*  ;  I BPSELIG="T"!(BPSELIG="C"),BPSSTAT="IN PROGRESS",$G(REVREAS)'="RX RELEASE-NDC CHANGE",",CRLB,CRLR,CRLX,CRRL,PC,RRL,"'[(","\_BWHERE\_",") D BULL^BPSNCPD1(BRXIEN,BFILL,SITE,$G(DFN),$G(PNAME),BPSELIG,"","",$G(BPSCOB))  ;  S:'$D(RESPONSE) RESPONSE=1  K MOREDATA  I $G(IEN59) D  . N MSG  . S MSG="Foreground Process Complete-RESPONSE="\_$G(RESPONSE)  . I $G(RESPONSE)'=0 S MSG=MSG\_", CLMSTAT="\_$G(CLMSTAT)  . D LOG^BPSOSL(IEN59,$T(+0)\_"-"\_MSG)  Q RESPONSE\_U\_$G(CLMSTAT)\_U\_BPSELIG\_U\_BPSSTAT\_U\_$$CLMINFO^BPSUTIL2(+$G(IEN59)) | | | | | | | | |

Near the end of the claim submission process, it calls TRICCHK^PSOREJU3 for TRICARE/CHAMPVA claims. Currently, if the status is still “IN PROGRESS”, that procedure and code below it eventually adds a comment to the ECME Activity Log stating that the Rx is being placed back on suspense. At this point in the process, it does not put the Rx/Fill on suspense.

This enhancement calls for modifying TRICCHK^PSOREJU3 so that it Quits out if the status is still “IN PROGRESS” and the value of BWHERE/FROM is “PC” (Pull CMOPS). That is the value of BWHERE/FROM when the claim submission process in ECME/BPS is called from CMOP/PSX, which includes when the nightly CMOP transmission is run and when the user selects the CMOP pull from suspense.

The CMOP process itself will continue to check for any claims that are still “IN PROGRESS”, as it does now, but that check will not happen until all the prescriptions have been sent and the process has waited an additional 15 seconds for every claim sent, waiting up to two hours. (The sending of claims is in SBTEMCE^PSXRPPL1, and the waiting is in SDT^PSXRPPL.)

| Subroutine Name | **TRICCHK^PSOREJU3** | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Enhancement Category | New | Modify | | | Delete | | No Change | |
| Story | US22 | | | | | | | |
| Related Menu Options or ListMan Actions | Pull Early from Suspense [PSO PNDRX]  Print from Suspense File [PSO PNDLBL] | | | | | | | |
| Related Routines | This Subroutine is Called By | | | This Subroutine Calls | | | | |
| ECMESND^PSOBPSU1, HDLG^PSOREJU1, STATUS^PSOREJU3 | | | $$GET1^DIQ, RXACT^PSOBPSU2, $$ELIGDISP^PSOREJP1, LOG^BPSOSL | | | | |
| Related Integration Control Registrations (ICRs) | N/A | | | | | | | |
| Data Passing | Parameter Input | | Parameter Output | | | Function Return Value | | Global Modified |
| Input Attribute Name and Definition | Name: RX  Definition: Prescription  New  Modify  Delete  No Change  Name: RFL  Definition: Fill number  New  Modify  Delete  No Change  Name: RESP  Definition: Response from $$EN^BPSNCPDP  New  Modify  Delete  No Change  Name: FROM  Definition: BWHERE value  New  Modify  Delete  No Change  Name: RVTX  Definition: Reversal reason  New  Modify  Delete  No Change | | | | | | | |
| Output Attribute Name and Definition | Name: N/A  Definition:  New  Modify  Delete  No Change | | | | | | | |
| Current Logic | | | | | | | | |
| TRICCHK(RX,RFL,RESP,FROM,RVTX) ;check to see if Rx is non-billable or in an "In Progress" state on ECME  ; Input: (r) RX - Rx IEN (#52)  ; (r) RFL - REFILL  ; (o) RESP - Response from $$EN^BPSNCPDP api  ; TRICCHK assumes that the calling routine has validated that the fill is TRICARE or CHAMPVA.  ;  ; - \Need to be mindful of foreground and background processing.  ;  N ETOUT,ESTAT,PSOBEI  S:'$D(FROM) FROM="" S ESTAT="",ESTAT=$P(RESP,"^",4),NFROM=0 I FROM="PL"!(FROM="PC") S NFROM=1  Q:ESTAT["PAYABLE"!(ESTAT["REJECTED")  S PSOBEI=$$ELIGDISP^PSOREJP1(RX,RFL)  I ESTAT["IN PROGRESS",FROM="RRL"!($G(RVTX)="RX RELEASE-NDC CHANGE") D Q  . I 'NFROM D  . . W !!,PSOBEI\_" Prescription "\_$$GET1^DIQ(52,RX,".01")\_" cannot be released until ECME 'IN PROGRESS'"  . . W !,"status is resolved payable.",!!  ;  I $D(RESP) D Q  . I +RESP=6 W:'NFROM&('$G(CMOP)) !!,"Inactive ECME "\_PSOBEI,!! D Q  . . S ACT="Inactive ECME "\_PSOBEI D RXACT^PSOBPSU2(RX,RFL,ACT,"M",DUZ)  . I +RESP=2!(+RESP=3) N PSONBILL S PSONBILL=1 D TRIC2 Q  . I +RESP=4!(ESTAT["IN PROGRESS") N PSONPROG S PSONPROG=1 D TRIC2 Q  Q | | | | | | | | |
| Modified Logic (Changes are highlighted) | | | | | | | | |
| TRICCHK(RX,RFL,RESP,FROM,RVTX) ;check to see if Rx is non-billable or in an "In Progress" state on ECME  ; Input: (r) RX - Rx IEN (#52)  ; (r) RFL - REFILL  ; (o) RESP - Response from $$EN^BPSNCPDP api  ; TRICCHK assumes that the calling routine has validated that the fill is TRICARE or CHAMPVA.  ;  ; - \Need to be mindful of foreground and background processing.  ;  N ETOUT,ESTAT,PSOBEI  S:'$D(FROM) FROM="" S ESTAT="",ESTAT=$P(RESP,"^",4),NFROM=0 I FROM="PL"!(FROM="PC") S NFROM=1  Q:ESTAT["PAYABLE"!(ESTAT["REJECTED")  S PSOBEI=$$ELIGDISP^PSOREJP1(RX,RFL)  ;  ; \*\*\* Remove the below after BPS\*1.0\*22 changes have been confirmed to work \*\*\*  ;  I ESTAT["IN PROGRESS",FROM="PC" D LOG^BPSOSL(IEN59,$T(+0)\_"-BPS\*1.0\*22 B”)  ;  ; \*\*\* Remove the above after BPS\*1.0\*22 changes have been confirmed to work \*\*\*  ;  I ESTAT["IN PROGRESS",FROM="PC" Q  I ESTAT["IN PROGRESS",FROM="RRL"!($G(RVTX)="RX RELEASE-NDC CHANGE") D Q  . I 'NFROM D  . . W !!,PSOBEI\_" Prescription "\_$$GET1^DIQ(52,RX,".01")\_" cannot be released until ECME 'IN PROGRESS'"  . . W !,"status is resolved payable.",!!  ;  I $D(RESP) D Q  . I +RESP=6 W:'NFROM&('$G(CMOP)) !!,"Inactive ECME "\_PSOBEI,!! D Q  . . S ACT="Inactive ECME "\_PSOBEI D RXACT^PSOBPSU2(RX,RFL,ACT,"M",DUZ)  . I +RESP=2!(+RESP=3) N PSONBILL S PSONBILL=1 D TRIC2 Q  . I +RESP=4!(ESTAT["IN PROGRESS") N PSONPROG S PSONPROG=1 D TRIC2 Q  Q | | | | | | | | |

Immediately after calling ECMESND^PSOBPSU1 to send a claim for a given Rx/Fill, the system is currently performing the check $$TRISTA^PSOREJU3, and if that function returns a “1” then the Rx/Fill is put into ^TMP(“PSXEPHNB”). Later in the process, any Rx/Fills in that temp global are not sent to CMOP and are put back on suspense. The system sends mail message listing all Rx/Fills left on suspense at the end of the CMOP process. The call to $$TRISTA^PSOREJU3 should remain there, but it should not be checked if the ECME status is currently “IN PROGRESS”.

| Subroutine Name | **SBTECME^PSXRPPL1** | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Enhancement Category | New | Modify | | | Delete | | No Change | |
| Story | US22 | | | | | | | |
| Related Menu Options or ListMan Actions | Pull Early from Suspense [PSO PNDRX]  Print from Suspense File [PSO PNDLBL] | | | | | | | |
| Related Routines | This Subroutine is Called By | | | This Subroutine Calls | | | | |
| SDT^PSXRPPL | | | $$CMOPON^BPSUTIL, $$ECMEON^BPSUTIL, $$GET1^DIQ, $$LSTRFL^PSOBPSU1, ECMESND^PSOBPSU1, $$RETRX^PSOBPSUT, $$STATUS^PSOBPSUT, $$TRISTA^PSOREJU3, $$FIND^PSOREJUT, PSOUL^PSSLOCK, $$XMIT^PSXBPSUT, $$DSH^PSXRPPL2, $$DUR^PSXRPPL2, $$ECMESTAT^PSXRPPL2, $$PATCH^XPDUTL, LOG^BPSOSL, $$IEN59^BPSOSRX | | | | |
| Related Integration Control Registrations (ICRs) | ;Reference to ^PS(52.5, supported by DBIA #1978  ;Reference to ^BPSUTIL supported by DBIA #4410  ;Reference to ^PSSLOCK supported by DBIA #2789  ;Reference to ^PSOBPSUT supported by DBIA #4701  ;Reference to ^PSOBPSU1 supported by DBIA #4702  ;Reference to ^PSOREJUT supported by DBIA #4706  ;Reference to ^PSOREJU3 supported by DBIA #5186 | | | | | | | |
| Data Passing | Parameter Input | | Parameter Output | | | Function Return Value | | Global Modified |
| Input Attribute Name and Definition | Name: PSXTP  Definition: Type of prescriptions (“C” = controlled substances, “N” = non-controlled substances)  New  Modify  Delete  No Change  Name: PSXDV  Definition: Pointer to DIVISION file# 59  New  Modify  Delete  No Change  Name: PSXDT  Definition: T+N when scheduling the THROUGH DATE to run CMOP transmission  New  Modify  Delete  No Change  Name: PULLDT  Definition: T+N+PULL DAYS parameter in the DIVISION file# 59  New  Modify  Delete  No Change | | | | | | | |
| Output Attribute Name and Definition | Name: $$SBTECME  Definition:  New  Modify  Delete  No Change | | | | | | | |
| Current Logic | | | | | | | | |
| SBTECME(PSXTP,PSXDV,THRDT,PULLDT) ; - Sumitting prescriptions to EMCE (3rd Party Billing)  ;Input: PSXTP - Type of prescriptions "C" - Controlled Subs / "N" Non-Controlled Subs  ; PSXDV - Pointer to DIVSION file (#59)  ; THRDT - T+N when scheduling the THROUGH DATE to run CMOP Transmission  ; PULLDT - T+N+PULL DAYS parameter in the DIVISION file (#59)  ;Output:SBTECME- Number of prescriptions submitted to ECME  N RX,RFL,SBTECME,PSOLRX,RESP,SDT,XDFN,REC,PSOLRX  I '$$ECMEON^BPSUTIL(PSXDV)!'$$CMOPON^BPSUTIL(PSXDV) Q  S (SDT,SBTECME)=0 K ^TMP("PSXEPHDFN",$J)  F S SDT=$O(^PS(52.5,"CMP","Q",PSXTP,PSXDV,SDT)) S XDFN=0 Q:(SDT>PULLDT)!(SDT'>0) D  . F S XDFN=$O(^PS(52.5,"CMP","Q",PSXTP,PSXDV,SDT,XDFN)) S REC=0 Q:(XDFN'>0)!(XDFN="") D  . . F S REC=$O(^PS(52.5,"CMP","Q",PSXTP,PSXDV,SDT,XDFN,REC)) Q:(REC'>0)!(REC="") D  . . . S (PSOLRX,RX)=+$$GET1^DIQ(52.5,REC,.01,"I") I 'RX Q  . . . S RFL=$$GET1^DIQ(52.5,REC,9,"I") I RFL="" S RFL=$$LSTRFL^PSOBPSU1(RX)  . . . I $$XMIT^PSXBPSUT(REC) D  . . . . I SDT>THRDT,'$D(^TMP("PSXEPHDFN",$J,XDFN)) Q  . . . . I $$PATCH^XPDUTL("PSO\*7.0\*148") D  . . . . . I $$RETRX^PSOBPSUT(RX,RFL),SDT>DT Q  . . . . . I $$DOUBLE(RX,RFL) Q  . . . . . I $$FIND^PSOREJUT(RX,RFL,,"79,88",,1) Q  . . . . . I '$$RETRX^PSOBPSUT(RX,RFL),'$$ECMESTAT^PSXRPPL2(RX,RFL) Q  . . . . . I $$PATCH^XPDUTL("PSO\*7.0\*289") Q:'$$DUR^PSXRPPL2(RX,RFL) ;ePharm Host error hold  . . . . . I $$PATCH^XPDUTL("PSO\*7.0\*289"),RFL>0,$$STATUS^PSOBPSUT(RX,RFL-1)'="" Q:'$$DSH^PSXRPPL2(REC) ;ePharm 3/4 days su  pply (refill)  . . . . . I $$PATCH^XPDUTL("PSO\*7.0\*289"),RFL=0 Q:'$$DSH^PSXRPPL2(REC) ;ePharm 3/4 days supply (original fill)  . . . . . D ECMESND^PSOBPSU1(RX,RFL,"","PC",,1,,,,.RESP)  . . . . . I $$PATCH^XPDUTL("PSO\*7.0\*287"),$$TRISTA^PSOREJU3(RX,RFL,.RESP,"PC") S ^TMP("PSXEPHNB",$J,RX,RFL)=$G(RESP)  . . . . . I $D(RESP),'RESP S SBTECME=SBTECME+1  . . . . . S ^TMP("PSXEPHDFN",$J,XDFN)=""  . . . D PSOUL^PSSLOCK(PSOLRX)  K ^TMP("PSXEPHDFN",$J)  Q SBTECME | | | | | | | | |
| Modified Logic (Changes are highlighted) | | | | | | | | |
| SBTECME(PSXTP,PSXDV,THRDT,PULLDT) ; - Sumitting prescriptions to EMCE (3rd Party Billing)  ;Input: PSXTP - Type of prescriptions "C" - Controlled Subs / "N" Non-Controlled Subs  ; PSXDV - Pointer to DIVSION file (#59)  ; THRDT - T+N when scheduling the THROUGH DATE to run CMOP Transmission  ; PULLDT - T+N+PULL DAYS parameter in the DIVISION file (#59)  ;Output:SBTECME- Number of prescriptions submitted to ECME  N RX,RFL,SBTECME,PSOLRX,RESP,SDT,XDFN,REC,PSOLRX  I '$$ECMEON^BPSUTIL(PSXDV)!'$$CMOPON^BPSUTIL(PSXDV) Q  S (SDT,SBTECME)=0 K ^TMP("PSXEPHDFN",$J)  F S SDT=$O(^PS(52.5,"CMP","Q",PSXTP,PSXDV,SDT)) S XDFN=0 Q:(SDT>PULLDT)!(SDT'>0) D  . F S XDFN=$O(^PS(52.5,"CMP","Q",PSXTP,PSXDV,SDT,XDFN)) S REC=0 Q:(XDFN'>0)!(XDFN="") D  . . F S REC=$O(^PS(52.5,"CMP","Q",PSXTP,PSXDV,SDT,XDFN,REC)) Q:(REC'>0)!(REC="") D  . . . S (PSOLRX,RX)=+$$GET1^DIQ(52.5,REC,.01,"I") I 'RX Q  . . . S RFL=$$GET1^DIQ(52.5,REC,9,"I") I RFL="" S RFL=$$LSTRFL^PSOBPSU1(RX)  . . . I $$XMIT^PSXBPSUT(REC) D  . . . . I SDT>THRDT,'$D(^TMP("PSXEPHDFN",$J,XDFN)) Q  . . . . I $$PATCH^XPDUTL("PSO\*7.0\*148") D  . . . . . I $$RETRX^PSOBPSUT(RX,RFL),SDT>DT Q  . . . . . I $$DOUBLE(RX,RFL) Q  . . . . . I $$FIND^PSOREJUT(RX,RFL,,"79,88",,1) Q  . . . . . I '$$RETRX^PSOBPSUT(RX,RFL),'$$ECMESTAT^PSXRPPL2(RX,RFL) Q  . . . . . I $$PATCH^XPDUTL("PSO\*7.0\*289") Q:'$$DUR^PSXRPPL2(RX,RFL) ;ePharm Host error hold  . . . . . I $$PATCH^XPDUTL("PSO\*7.0\*289"),RFL>0,$$STATUS^PSOBPSUT(RX,RFL-1)'="" Q:'$$DSH^PSXRPPL2(REC) ;ePharm 3/4 days su  pply (refill)  . . . . . I $$PATCH^XPDUTL("PSO\*7.0\*289"),RFL=0 Q:'$$DSH^PSXRPPL2(REC) ;ePharm 3/4 days supply (original fill)  . . . . . D ECMESND^PSOBPSU1(RX,RFL,"","PC",,1,,,,.RESP)  . . . . . ;  . . . . . ; \*\*\* Remove the below after BPS\*1.0\*22 changes have been confirmed to work \*\*\*  . . . . . ;  . . . . . I $P(RESP,"^",4)[“IN PROGRESS” D LOG^BPSOSL($$IEN59^BPSOSRX(RX,RFL),$T(+0)\_"-BPS\*1.0\*22 C”)  . . . . . ;  . . . . . ; \*\*\* Remove the above after BPS\*1.0\*22 changes have been confirmed to work \*\*\*  . . . . . ;  . . . . . I $P(RESP,"^",4)‘[“IN PROGRESS”,$$PATCH^XPDUTL("PSO\*7.0\*287"),  $$TRISTA^PSOREJU3(RX,RFL,.RESP,"PC") S ^TMP("PSXEPHNB",$J,RX,RFL)=$G(RESP)  . . . . . I $D(RESP),'RESP S SBTECME=SBTECME+1  . . . . . S ^TMP("PSXEPHDFN",$J,XDFN)=""  . . . D PSOUL^PSSLOCK(PSOLRX)  K ^TMP("PSXEPHDFN",$J)  Q SBTECME | | | | | | | | |

Since the claims submission process will no longer make an entry on the prescription’s Activity Log if a prescription will be left on suspense due to an “IN PROGRESS” status, this design calls for updating the CMOP process to do that. The most appropriate time in the process for this to happen is when the system creates the email listing all the prescriptions which are being left on suspense and not sent to CMOP.

| Subroutine Name | **EN^PSXBPSMS** | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Enhancement Category | New | Modify | | | Delete | | No Change | |
| Story | US22 | | | | | | | |
| Related Menu Options or ListMan Actions | Pull Early from Suspense [PSO PNDRX]  Print from Suspense File [PSO PNDLBL] | | | | | | | |
| Related Routines | This Subroutine is Called By | | | This Subroutine Calls | | | | |
| SDT^PSXRPPL, | | | DEM^VADPT, $$FMADD^XLFDT, LOG^BPSOSL, $$IEN59^BPSOSRX, $$STATUS^PSOBPSUT, $$ELIGDISP^PSOREJP1, RXACT^PSOBPSU2 | | | | |
| Related Integration Control Registrations (ICRs) | Reference to ^PSOBPSUT supported by DBIA #4701  Reference to ^PSOBPSU2 supported by DBIA #4970  A new ICR will have to be created to allow this PSX routine to call the API ELIGDISP^PSOREJP1 | | | | | | | |
| Data Passing | Parameter Input | | Parameter Output | | | Function Return Value | | Global Modified |
| Input Attribute Name and Definition | Name: N/A  Definition:  New  Modify  Delete  No Change | | | | | | | |
| Output Attribute Name and Definition | Name: N/A  Definition:  New  Modify  Delete  No Change | | | | | | | |
| Current Logic | | | | | | | | |
| EN ;Main entry point.  N EMCNT,DFN,ORCNT,PATCNT,DIV,RX,RFL,DFN,SSN,PATNM,PTLST,VADM  K ^TMP("PSXEPHOUT",$J)  S ^XTMP("PSXBPSMS",0)=$$FMADD^XLFDT(DT,35)\_"^"\_DT  S DIV="",(EMCNT,ORCNT,PATCNT)=0  F S DIV=$O(^TMP("PSXEPHIN",$J,DIV)) Q:DIV="" D  .D HEADER(DIV)  .S RX="" F S RX=$O(^TMP("PSXEPHIN",$J,DIV,RX)) Q:RX="" D  ..S RFL=+$G(^TMP("PSXEPHIN",$J,DIV,RX))  ..S ^XTMP("PSXBPSMS",1,RX,RFL,DT)=""  ..S DFN=+$P(^PSRX(RX,0),"^",2) D DEM^VADPT  ..S SSN=$E($P(VADM(2),U),6,9),PATNM=(VADM(1))  ..S ORCNT=$G(ORCNT)+1 D PATCNT(PATNM\_SSN)  ..D FORMAT  .D FOOTER(DIV)  D MAIL,CLEAN  Q | | | | | | | | |
| Modified Logic (Changes are highlighted) | | | | | | | | |
| EN ;Main entry point.  N EMCNT,DFN,ORCNT,PATCNT,DIV,RX,RFL,DFN,SSN,PATNM,PTLST,VADM,PSXACTIVITY  K ^TMP("PSXEPHOUT",$J)  S ^XTMP("PSXBPSMS",0)=$$FMADD^XLFDT(DT,35)\_"^"\_DT  S DIV="",(EMCNT,ORCNT,PATCNT)=0  F S DIV=$O(^TMP("PSXEPHIN",$J,DIV)) Q:DIV="" D  . D HEADER(DIV)  . S RX="" F S RX=$O(^TMP("PSXEPHIN",$J,DIV,RX)) Q:RX="" D  . . S RFL=+$G(^TMP("PSXEPHIN",$J,DIV,RX))  . . S ^XTMP("PSXBPSMS",1,RX,RFL,DT)=""  . . ;  . . ; \*\*\* Remove the below after BPS\*1.0\*22 changes have been confirmed to work \*\*\*  . . ;  . . D LOG^BPSOSL($$IEN59^BPSOSRX(RX,RFL),$T(+0)\_"-BPS\*1.0\*22 D”)  . . ;  . . ; \*\*\* Remove the above after BPS\*1.0\*22 changes have been confirmed to work \*\*\*  . . ;  . . I $$STATUS^PSOBPSUT(RX,RFL)="IN PROGRESS" D ; ICR #4701  . . . S PSXACTIVITY=$$ELIGDISP^PSOREJP1(RX,PSXFILL)\_“-Rx placed on Suspense due to ECME IN PROGRESS status” ; ICR #tbd  . . . D RXACT^PSOBPSU2(RX,RFL,PSXACTIVITY,“M”,DUZ) ; ICR # 4970  . . . Q  . . S DFN=+$P(^PSRX(RX,0),"^",2) D DEM^VADPT  . . S SSN=$E($P(VADM(2),U),6,9),PATNM=(VADM(1))  . . S ORCNT=$G(ORCNT)+1 D PATCNT(PATNM\_SSN)  . . D FORMAT  . D FOOTER(DIV)  D MAIL,CLEAN  Q | | | | | | | | |

The existing ICR #4970 will need to be updated into include the entry point $$RXACT^PSOBPSU2.

# Back Out/Rollback Procedure

This story does not include adding or changing data fields in any VistA files, so no rollback of data would be necessary if this enhancement needed to be backed out.

To back out this enhancement, the routines BPSNCPDP, PSOREJU3, PSXBPSMS, PSXRPPL1, and PSXRPPL2 should be backed up prior to installing this patch. Reverting to the previous version of these routines would back out this enhancement. Alternatively, the previous version of these routines could be distributed via an emergency patch.