**MCCF EDI TAS US578**

System Design Document



Department of Veterans Affairs

**December 2017**

Version 1.00

**User Story Number:** USRX-18

**User Story Name:** Improve existing ePharmacy reports – TRI/CVA Bypass Override Report QTY for partial fills

**Product Backlog ID:**

**Backlog Priority:** 34

**Initial Sizing Estimate:**

**Rational ID:**

**Rally ID:** US578

**Epic Taxonomy:** Update

# Design – Summary

The TRICARE-CHAMPVA Bypass Override Report [PSO TRI CVA OVERRIDE RPORT] displays incorrect quantity and cost for partially filled prescriptions.

List of Components:

* Routine: PSOTRI

# Design – Detail

The TRICARE CHAMPVA Bypass/Override Report pulls the data displayed from the PSO AUDIT LOG file #52.87. This design calls for modifying the routine PSOTRI so that the correct data is stored into file #52.87. The report will then display the correct quantity and cost for partial fills. The cost should display in $BILLED column even though a claim is not submitted to bill. The quantity is entered by the user when partially filling a prescription.

| Subroutine Name | **AUDIT^PSOTRI** | | | |
| --- | --- | --- | --- | --- |
| Enhancement Category | New | Modify | Delete | No Change |
| Current Logic | | | | |
| PSOTRI ;BIRM/BNT - OP TRICARE/CHAMPVA Audit Log Utilities ;07/21/2010  ;;7.0;OUTPATIENT PHARMACY;\*\*358,385,427\*\*;DEC 1997;Build 21 . . .  AUDIT(RX,RFL,RXCOB,JST,AUD,ELIG) ;  ; Main entry to create a new record in the PSO AUDIT LOG file #52.87  ; Note that AUDIT^PSOTRI is called by ECME (BPSECMP2) - ICR 6156  ; INPUT: RX (r) = Prescription IEN  ; RFL (o) = Prescription Fill # (Default is original zero fill)  ; RXCOB (o) = Coordination of Benefits  ; 1 = Primary (Default)  ; 2 = Secondary  ; JST (o) = Justification text  ; AUD (r) = Audit Type  ; R = NCPDP REJECT - Associated with an Override audit action  ; N = NON BILLABLE - Associated with an Override audit action  ; I = INPATIENT - Associated with a Bypass audit action  ; P = PARTIAL FILL  ; ELIG (r) = Eligibility Type  ; T = TRICARE  ; C = CHAMPVA  ; RETURN: Successful Audit entry will return the IEN of the new entry in file 52.87  ; Unsuccessful Audit entry will return "0^Error Description"  ;  N PSOTRIC,PSODIV,RXFLDS,RFLFLDS,RXECME,PSOFDA,FN,SFN,PSOIEN,PSOIENS,PSOUSER,PSOTC,PSOET  N I,PSOAIEN,PSOREJ,DFN,PSODOA,PSODOS,PSOERR,PSOX,PSOY,RXARR,RFLARR,PSOPHRM,PSOQTY  Q:'$D(^PSRX(RX,0)) "0^Prescription does not exist"  . . .     ; Date of Action is NOW  S PSODOA=$$NOW^XLFDT()  ; Date of Service  S PSODOS=$$DOS^PSOBPSU1(RX,RFL)  ; User (If null OR Audit Type is Inpatient OR bypass-type reject, set to POSTMASTER)  S PSOUSER=DUZ  I (PSOUSER="")!(AUD="I")!$$BYPASS^PSOBPSU1(ELIG,JST) S PSOUSER=.5  ; Quantity  S PSOQTY=$S(RFL>0:$G(RFLARR(52.1,RFL\_","\_RX\_",",1,"I")),1:$G(RXARR(52,RX\_",",7,"I")))  ;  ; Set up FDA array  S PSOIEN="+1,"  S PSOAIEN=$P($G(^PS(52.87,0)),U,3)+1  ; AUDIT ID field  S PSOFDA(FN,PSOIEN,.01)=PSOAIEN  ; PRESCRIPTION field  S PSOFDA(FN,PSOIEN,1)=RX  ; FILL field  S PSOFDA(FN,PSOIEN,2)=RFL  ; PATIENT field  S PSOFDA(FN,PSOIEN,3)=$G(RXARR(52,RX\_",",2,"I"))  ; DIVISOIN field  S PSOFDA(FN,PSOIEN,4)=PSODIV  ; PROVIDER field  S PSOFDA(FN,PSOIEN,5)=$S(RFL>0:$G(RFLARR(52.1,RFL\_","\_RX\_",",15,"I")),1:$G(RXARR(52,RX\_",",4,"I")))  ; NDC field  S PSOFDA(FN,PSOIEN,6)=$S(RFL>0:$G(RFLARR(52.1,RFL\_","\_RX\_",",11,"I")),1:$G(RXARR(52,RX\_",",27,"I")))  ; DRUG field  S PSOFDA(FN,PSOIEN,7)=$G(RXARR(52,RX\_",",6,"I"))  ; BILL COST field (This needs to be verified)  S PSOFDA(FN,PSOIEN,8)=$G(RXARR(52,RX\_",",17,"I"))\*PSOQTY+8  ; ECME NUMBER field  S PSOFDA(FN,PSOIEN,9)=RXECME  ; QTY field  S PSOFDA(FN,PSOIEN,10)=PSOQTY  ; PATIENT STATUS field  S PSOFDA(FN,PSOIEN,11)=$G(RXARR(52,RX\_",",3,"I"))  ; AUDIT TYPE field  S PSOFDA(FN,PSOIEN,12)=AUD  ; USER field  S PSOFDA(FN,PSOIEN,14)=PSOUSER  ; DATE OF ACTION field  S PSOFDA(FN,PSOIEN,15)=PSODOA  ; DATE OF SERVICE field  S PSOFDA(FN,PSOIEN,16)=PSODOS  ; TRICARE JUSTIFICATION field  S PSOFDA(FN,PSOIEN,17)=JST  ; Eligibility Code  S PSOFDA(FN,PSOIEN,18)=ELIG  D DUR1^BPSNCPD3(RX,RFL,.PSOREJ,.PSOERR,RXCOB)  S PSOET=$$PSOET^PSOREJP3(RX,RFL) ;check to see if eT or eC is the reject code as no ecme claim.  I PSOET S PSOTC=$S(PSOTRIC=1:"eT",PSOTRIC=2:"eC",1:"")  I PSOTC]"",'$D(PSOREJ(RXCOB,"REJ CODES")) S PSOREJ(RXCOB,"REJ CODES",1,PSOTC)="",PSOREJ(RXCOB,"REJ CODE LST")=PSOTC  I $G(PSOREJ(RXCOB,"REJ CODE LST"))]"" D  . S PSOX="",PSOY=1 F I=1:1 S PSOX=$O(PSOREJ(RXCOB,"REJ CODES",I,0)) Q:PSOX=""  D  . . S PSOY=PSOY+1,PSOIENS=PSOY\_","\_PSOIEN  . . S PSOFDA(SFN,"+"\_PSOIENS,.01)=PSOX  ;  D UPDATE^DIE("","PSOFDA","","PSOERR")  I $D(PSOERR("DIERR")) D BMES^XPDUTL(PSOERR("DIERR",1,"TEXT",1))  Q | | | | |
| Modified Logic (Changes are highlighted) | | | | |
| PSOTRI ;BIRM/BNT - OP TRICARE/CHAMPVA Audit Log Utilities ;07/21/2010  ;;7.0;OUTPATIENT PHARMACY;\*\*358,385,427,48#\*\*;DEC 1997;Build 21 . . .  AUDIT(RX,RFL,RXCOB,JST,AUD,ELIG) ;  ; Main entry to create a new record in the PSO AUDIT LOG file #52.87  ; Note that AUDIT^PSOTRI is called by ECME (BPSECMP2) - ICR 6156  ; INPUT: RX (r) = Prescription IEN  ; RFL (o) = Prescription Fill # (Default is original zero fill)  ; RXCOB (o) = Coordination of Benefits  ; 1 = Primary (Default)  ; 2 = Secondary  ; JST (o) = Justification text  ; AUD (r) = Audit Type  ; R = NCPDP REJECT - Associated with an Override audit action  ; N = NON BILLABLE - Associated with an Override audit action  ; I = INPATIENT - Associated with a Bypass audit action  ; P = PARTIAL FILL  ; ELIG (r) = Eligibility Type  ; T = TRICARE  ; C = CHAMPVA  ; RETURN: Successful Audit entry will return the IEN of the new entry in file 52.87  ; Unsuccessful Audit entry will return "0^Error Description"  ;   N PSOTRIC,PSODIV,RXFLDS,RFLFLDS,RXECME,PSOFDA,FN,SFN,PSOIEN,PSOIENS,PSOUSER,PSOTC,PSOET  N I,PSOAIEN,PSOREJ,DFN,PSODOA,PSODOS,PSOERR,PSOX,PSOY,RXARR,RFLARR,PSOPHRM,PSOQTY   N DFN,IBADT,IBBI,IBCHG,IBD,IBFEE,IBINS,IBRES,IBTYP,PFARR,PFFLDS,PFIEN,PSOUNITCOST  ;  Q:'$D(^PSRX(RX,0)) "0^Prescription does not exist" . . .   ; Date of Action is NOW  S PSODOA=$$NOW^XLFDT()  ; Date of Service  S PSODOS=$$DOS^PSOBPSU1(RX,RFL)  ; User (If null OR Audit Type is Inpatient OR bypass-type reject, set to POSTMASTER)  S PSOUSER=DUZ  I (PSOUSER="")!(AUD="I")!$$BYPASS^PSOBPSU1(ELIG,JST) S PSOUSER=.5   ; Set up FDA array  S PSOIEN="+1,"  S PSOAIEN=$P($G(^PS(52.87,0)),U,3)+1  ; Quantity, Unit Cost, Provider, NDC and Bill Cost fields  I AUD="P" D  . ; For Partial Fills pull the QTY, Unit Cost, PROVIDER, NDC and BILLED from  . ;the appropriate entry in the PARTIAL DATE sub-file.  . ;  . ; Attempt to identify a partial fill for today’s date.  . S PFIEN=0  . F  S PFIEN=$O(^PSRX(RX,"P",PFIEN)) Q:'PFIEN I $P($G(^PSRX(RX,"P",PFIEN,0)),U,1)=$P(PSODOA,".",1) Q   . I 'PFIEN Q ; partial fill entry for today not found  . ;  . ; QTY;CURRENT UNIT PRICE OF DRUG;PROVIDER;NDC  . S PFFLDS=".04;.042;6;1"   . D GETS^DIQ(52.2,PFIEN\_","\_RX,PFFLDS,"I","PFARR")  . ;  . S PSOQTY=$G(PFARR(52.2,PFIEN\_","\_RX\_",",.04,"I"))  . S PSOUNITCOST=$G(PFARR(52.2,PFIEN\_","\_RX\_",",.042,"I"))   . S PSOFDA(FN,PSOIEN,5)=$G(PFARR(52.2,PFIEN\_","\_RX\_",",6,"I"))  . S PSOFDA(FN,PSOIEN,6)=$G(PFARR(52.2,PFIEN\_","\_RX\_",",1,"I"))  . D GETCOST S PSOFDA(FN,PSOEIN,8)=+IBCHG  E  D  . S PSOQTY=$S(RFL>0:$G(RFLARR(52.1,RFL\_","\_RX\_",",1,"I")),1:$G(RXARR(52,RX\_",",7,"I")))  . S PSOFDA(FN,PSOIEN,5)=$S(RFL>0:$G(RFLARR(52.1,RFL\_","\_RX\_",",15,"I")),1:$G(RXARR(52,RX\_",",  4,"I")))  . S PSOFDA(FN,PSOIEN,6)=$S(RFL>0:$G(RFLARR(52.1,RFL\_","\_RX\_",",11,"I")),1:$G(RXARR(52,RX\_",",  27,"I")))  . S PSOFDA(FN,PSOIEN,8)=$G(RXARR(52,RX\_",",17,"I"))\*PSOQTY+8 ;This needs to be verified  ; AUDIT ID field  S PSOFDA(FN,PSOIEN,.01)=PSOAIEN  ; PRESCRIPTION field  S PSOFDA(FN,PSOIEN,1)=RX  ; FILL field  S PSOFDA(FN,PSOIEN,2)=RFL  ; PATIENT field  S PSOFDA(FN,PSOIEN,3)=$G(RXARR(52,RX\_",",2,"I"))  ; DIVISION field  S PSOFDA(FN,PSOIEN,4)=PSODIV  ; DRUG field  S PSOFDA(FN,PSOIEN,7)=$G(RXARR(52,RX\_",",6,"I"))  ; ECME NUMBER field  S PSOFDA(FN,PSOIEN,9)=RXECME  ; QTY field  S PSOFDA(FN,PSOIEN,10)=PSOQTY  ; PATIENT STATUS field  S PSOFDA(FN,PSOIEN,11)=$G(RXARR(52,RX\_",",3,"I"))  ; AUDIT TYPE field  S PSOFDA(FN,PSOIEN,12)=AUD  ; USER field  S PSOFDA(FN,PSOIEN,14)=PSOUSER  ; DATE OF ACTION field  S PSOFDA(FN,PSOIEN,15)=PSODOA  ; DATE OF SERVICE field  S PSOFDA(FN,PSOIEN,16)=PSODOS  ; TRICARE JUSTIFICATION field  S PSOFDA(FN,PSOIEN,17)=JST  ; Eligibility Code  S PSOFDA(FN,PSOIEN,18)=ELIG  D DUR1^BPSNCPD3(RX,RFL,.PSOREJ,.PSOERR,RXCOB)  S PSOET=$$PSOET^PSOREJP3(RX,RFL) ;check to see if eT or eC is the reject code as no ecme claim.  I PSOET S PSOTC=$S(PSOTRIC=1:"eT",PSOTRIC=2:"eC",1:"")  I PSOTC]"",'$D(PSOREJ(RXCOB,"REJ CODES")) S PSOREJ(RXCOB,"REJ CODES",1,PSOTC)="",PSOREJ(RXCOB,"REJ CODE LST")=PSOTC  I $G(PSOREJ(RXCOB,"REJ CODE LST"))]"" D  . S PSOX="",PSOY=1 F I=1:1 S PSOX=$O(PSOREJ(RXCOB,"REJ CODES",I,0)) Q:PSOX=""  D  . . S PSOY=PSOY+1,PSOIENS=PSOY\_","\_PSOIEN  . . S PSOFDA(SFN,"+"\_PSOIENS,.01)=PSOX  ;  D UPDATE^DIE("","PSOFDA","","PSOERR")  I $D(PSOERR("DIERR")) D BMES^XPDUTL(PSOERR("DIERR",1,"TEXT",1))  Q   ; GETCOST ; For Partial Fills the BILL COST needs to be calculated using the same calculation that   ; is used during claim submission (modeled after RX^IBNCPDP1)  ;  S DFN=$G(RXARR(52,RX\_",",2,"I"))  S IBADT=DT  D RXINS^IBNCPDPU(DFN,IBADT,.IBINS)  S IBRT=$$RT^IBNCPDPU(DFN,IBADT,.IBINS,.IBTYP)  D SETINSUR^IBNCPDP1(IBADT,IBRT,"T",.IBINS,.IBD,.IBRES)  S IBBI=$$EVNTITM^IBCRU3(+IBRT,3,"PRESCRIPTION FILL",IBADT,.IBRS)  S IBRS=+$O(IBRS($P(IBBI,";"),0))  S IBCHG=$$RATECHG^IBCRCC(+IBRS,$S($P(IBRT,U,2)'="C":1,1:PSOQTY\*PSOUNITCOST),IBADT,.IBFEE)  Q | | | | |

With the addition of the GETCOST subroutine, an existing ICR #5714 will need to be updated as described in the table below. PSO (Outpatient Pharmacy) will be added as a subscribing package, and the component $$RT will be added, along with input and output variables.

|  |  |
| --- | --- |
| Integration Control Registration # | **5714** |
| Name | IB PHARMACY INSURANCE |
| Enhancement Category | New  Modify  Expand |
| Custodial Package | Integrated Billing |
| Subscribing Package(s) | E Claims Mgmt Engine  Oupatient Pharmacy |
| Usage | Private |
| Type | Routine |
| Routine | IBNCPDPU |
| Component | $$RT |
| Input Variables | Name: IBINS  Description: insurance array (pass by reference) |
| Output Variables | Name: IBRT  Description: 3 piece string in the following format  [1] rate type ien,  [2] Rate Type (Tort or Awp or Cost)  [3] Eligibility Basis (V=VETERAN, T=TRICARE, C=CHAMPVA) |

Additionally a new ICR will be created to allow the PSO routine to call SETINSUR^IBNCPDP1.

|  |  |
| --- | --- |
| Integration Control Registration # | **TBD** |
| Name | SET INSURANCE |
| Enhancement Category | New  Modify  Expand |
| Custodial Package | Integrated Billing |
| Subscribing Package(s) | Outpatient Pharmacy |
| Usage | Private |
| Type | Routine |
| Routine | IBNCPDP1 |
| Component | SETINSUR |
| Input Variables | Name: IBADT  Description: date of service/identify insurance as of this date  Name: IBRT  Description: rate type variable – [1] rate type ien, [2] type (A/C/T), [3] eligibility (V/T/C)  Name: IBELIG  Description: eligibility request flag (1/0)  Name: IBINS  Description: insurance array as returned by RXINS^IBNCPDPU  Name: IBD  Description: input/output – array entries passed in and certain array entries returned |
| Output Variables | Name: IBRES  Description: only returned if insurance errors  Name: IBD  Description: input/output – array entries passed in and certain array entries returned |