**MCCF EDI TAS US591**

System Design Document



Department of Veterans Affairs

**April 2017**

Version 1.00

**User Story Number:** USRX-10

**User Story Name:** Enhance the ECME User Screen CV Change View action.

**Product Backlog ID:** 10

**Backlog Priority:**

**Initial Sizing Estimate:**

**Rational ID:** tbd

**Rally ID:** US591

# Design – Summary

Edit specified prompts of the Change View (CV) action of the ECME User Screen. Currently, the prompts allow all values or a single value. The updated prompts will allow the user to select multiple entries. The following prompts will be updated:

* Select Certain Eligibility Type or (A)ll: A//
* Display One ECME (U)ser or (A)LL: A//
* Display One (P)atient or (A)LL: A//
* Display One (R)x or (A)LL: A//
* Display (R)ejects or (P)ayables or (U)nstranded or (A)LL: A//
* Display (C)MOP or (M)ail or (W)indow or (A)LL: A//
* Display (R)ealTime, (B)ackbills, (P)RO Option, Re(S)ubmission or (A)LL: A//
* Display Specific (R)eject Code or (A)LL: A//

Of the prompts being updated, the following prompts will require confirmation from the user if a previously selected entry is to be deleted:

* Display One ECME (U)ser or (A)LL: A//
* Display One (P)atient or (A)LL: A//
* Display One (R)x or (A)LL: A//

Of the prompts being updated, the following prompts will not require confirmation if a previously selected entry is to be deleted:

* Select Certain Eligibility Type or (A)ll: A//
* Display (R)ejects or (P)ayables or (U)nstranded or (A)LL: A//
* Display (C)MOP or (M)ail or (W)indow or (A)LL: A//
* Display (R)ealTime, (B)ackbills, (P)RO Option, Re(S)ubmission or (A)LL: A//
* Display Specific (R)eject Code or (A)LL: A//

Division prompt will also be updated to require at least one division if user selects D. If user selects D and does not enter a division, the prompt will be issued again.

The header of the ECME Screen will be updated to remove the details pertaining to Selected Users. Lines 2 and 3 of the header will be changed to left justify, aligning each line by the colon.

List of Components:

* Routine: BPSSCRCV
* Routine: BPSSCR01
* Routine: BPSSCR03
* Routine: BPSSCR05

# Design – Detail

Change View Action user prompts.

| Subroutine Name | **EDITPROF^BPSSCRCV** | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Enhancement Category | New | Modify | | Delete | | No Change | |
| Story | US591 | | | | | | |
| Related Menu Options or ListMan Actions | ECME User Screen action CV | | | | | | |
| Related Routines | This Subroutine is Called By | | | This Subroutine Calls | | | |
| CV^BPSSCRCV | | | $$DS^BPSSCRDS, $$EDITRX^BPSSCRPR, $$INSURSEL^BPSSCRCU, $$PROMPTRX^BPSUTIL1, DIQ^PSODI, ^DIR | | | |
| Related Integration Control Registrations (ICRs) | 4858 | | | | | | |
| Data Passing | Parameter Input | | Parameter Output | | Function Return Value | | Global Modified |
| Input Attribute Name and Definition | Name: BPARR  Definition: User Profile Array – Parameter Definition “BPS USRSCR”  New  Modify  Delete  No Change  Name: BPDUZ7  Definition: DUZ  New  Modify  Delete  No Change | | | | | | |
| Output Attribute Name and Definition | Name: BPARR  Definition: User Profile Array – Parameter Definition “BPS USRSCR”  New  Modify  Delete  No Change  Name: BPDUZ7  Definition: DUZ  New  Modify  Delete  No Change | | | | | | |
| Current Logic | | | | | | | |
| EDITPROF(BPARR,BPDUZ7) ;  I +$G(DUZ)=0 D ERRMSG("Unknown User") Q  N BP1,BPTF,BPQ,BPINP  N BPRET  N DIR,DR,DIE,DA  ;get ONE/ALL USERS?  ;EDITFLD(FILENO,FLDNO,RECIEN,CODESET,PRMTMSG,DFLTCODE) ;  S BPRET=$$DS^BPSSCRDS(.BPARR,+BPDUZ7) ;get divisions  Q:BPRET=-2 ;quit due to timeout or ^  Q:$$EDITFLD(2.01,+BPDUZ7,"S^V:VETERAN;T:TRICARE;C:CHAMPVA;A:ALL","Select Certain Eligibility Type or (A)ll","V",.BPARR)=-1  S BPQ=0 F  D  Q:BPQ'=0  . S BPINP=$$EDITFLD(1.01,+BPDUZ7,"S^U:ONE USER;A:ALL","Display One ECME (U)ser or (A)LL","ALL",.BPARR)  . S:BPINP=-1 BPQ=-1 S:$P(BPINP,U,2)="A" BPQ=1 I BPQ'=0 Q  . S BPINP=$$EDITFLD(1.16,+BPDUZ7,"P^VA(200,","Select User","",.BPARR)  . S:BPINP=-1 BPQ=-1 S:$P(BPINP,U,2)'="" BPQ=1 I BPQ'=0 Q  Q:BPQ=-1 ;quit due to timeout or ^  S BPQ=0 F  D  Q:BPQ'=0  . S BPINP=$$EDITFLD(1.02,+BPDUZ7,"S^P:ONE PATIENT;A:ALL","Display One (P)atient or (A)LL","ALL",.BPARR)  . S:BPINP=-1 BPQ=-1 S:$P(BPINP,U,2)="A" BPQ=1 I BPQ'=0 Q  . S BPINP=$$EDITFLD(1.17,+BPDUZ7,"P^DPT(","Select Patient","",.BPARR)  . S:BPINP=-1 BPQ=-1 S:$P(BPINP,U,2)'="" BPQ=1 I BPQ'=0 Q  Q:BPQ=-1 ;quit due to timeout or ^  S BPQ=0 F  D  Q:BPQ'=0  . S BPINP=$$EDITFLD(1.03,+BPDUZ7,"S^R:ONE RX;A:ALL","Display One (R)x or (A)LL","ALL",.BPARR)  . S:BPINP=-1 BPQ=-1 S:$P(BPINP,U,2)="A" BPQ=1 I BPQ'=0 Q  . S BPINP=$$EDITRX^BPSSCRPR(1.18,+BPDUZ7,"Select RX","",.BPARR)  . S:BPINP=-1 BPQ=-1 S:$P(BPINP,U,2)'="" BPQ=1 I BPQ'=0 Q  Q:BPQ=-1 ;quit due to timeout or ^  S BPINP=$$EDITFLD(1.04,+BPDUZ7,"S^D:DAYS;H:HOURS","Activity Timeframe (H)ours or (D)ays","DAYS",.BPARR)  I BPINP=-1 Q  ;quit due to timeout or ^  S BPTF=$P(BPINP,U,2)  Q:$$EDITFLD(1.05,+BPDUZ7,"N^1:999:0","Activity Timeframe Value",$S(BPTF="H":24,1:7),.BPARR)=-1  Q:$$EDITFLD(2.02,+BPDUZ7,"S^O:OPEN CLAIMS;C:CLOSED CLAIMS;A:ALL","Select Open/Closed or All Claims","O",.BPARR)=-1  Q:$$EDITFLD(1.19,+BPDUZ7,"S^O:Open Non-Billable Entries;C:Closed Non-Billable Entries;A:ALL","Display (O)pen or (C)losed or (A)ll Non-Billable Entries","A",.BPARR)=-1  Q:$$EDITFLD(2.03,+BPDUZ7,"S^B:BILLING REQUESTS;R:REVERSALS;A:ALL","Select Submission Type","A",.BPARR)=-1  Q:$$EDITFLD(1.06,+BPDUZ7,"S^R:REJECTS;P:PAYABLES;U:UNSTRANDED;A:ALL","Display (R)ejects or (P)ayables or (U)nstranded or (A)LL","REJECTS",.BPARR)=-1  Q:$$EDITFLD(1.07,+BPDUZ7,"S^R:RELEASED;N:NON-RELEASED;A:ALL","Display (R)eleased Rxs or (N)on-Released Rxs or (A)LL","RELEASED",.BPARR)=-1  Q:$$EDITFLD(1.08,+BPDUZ7,"S^C:CMOP;M:MAIL;W:WINDOW;A:ALL","Display (C)MOP or (M)ail or (W)indow or (A)LL","ALL",.BPARR)=-1  Q:$$EDITFLD(1.09,+BPDUZ7,"S^R:REALTIME;B:BACKBILLS;P:PRO OPTION;S:RESUBMISSION;A:ALL","Display (R)ealTime, (B)ackbills, (P)RO Option, Re(S)ubmission or (A)LL","ALL",.BPARR)=-1  S BPQ=0 F  D  Q:BPQ'=0  . S BPINP=$$EDITFLD(1.1,+BPDUZ7,"S^R:REJECT CODE;A:ALL","Display Specific (R)eject Code or (A)LL","ALL",.BPARR)  . S:BPINP=-1 BPQ=-1 S:$P(BPINP,U,2)="A" BPQ=1 I BPQ'=0 Q  . S BPINP=$$EDITFLD(1.15,+BPDUZ7,"P^BPSF(9002313.93,","Select Reject Code","",.BPARR)  . S:BPINP=-1 BPQ=-1 S:$P(BPINP,U,2)'="" BPQ=1 I BPQ'=0 Q  Q:BPQ=-1 ;quit due to timeout or ^  Q:$$INSURSEL^BPSSCRCU(.BPARR,+BPDUZ7)=-1  Q | | | | | | | |
| Modified Logic (Changes are highlighted) | | | | | | | |
| EDITPROF(BPARR,BPDUZ7) ;  I +$G(DUZ)=0 D ERRMSG("Unknown User") Q  N BP1,BPTF,BPQ,BPINP  N BPRET  N DIR,DR,DIE,DA  N BPS106,BPS106STR,BPS108,BPS108STR,BPS109,BPS109STR,BPS115,BPS115AR,BPS115NM,BPS115X  N BPS116,BPS116AR,BPS116NM,BPS116X,BPS117,BPS117AR,BPS117NM,BPS117X,BPS118,BPS118AR,BPS118X  N BPS201,BPS201STR,BPSCNT,BPSDRUG,BPSERR,BPSRXD,BPSRXN,BPSSEL,BPSX  ;get ONE/ALL USERS?  ;EDITFLD(FILENO,FLDNO,RECIEN,CODESET,PRMTMSG,DFLTCODE) ;  ; BPS113 ; FIELD 1.13 - Division  S BPRET=$$DS^BPSSCRDS(.BPARR,+BPDUZ7) ;get divisions  Q:BPRET=-2 ;quit due to timeout or ^  ; If (D)ivision option was selected but no Division entered, set BPARR(1.13)to "A"  ; and re-prompt current question.  I ($G(BPARR(1.13))="D")&(($G(BPARR("DIVS"))="")!($G(BPARR("DIVS"))=";")) S BPARR(1.13)="A" G BPS113  ;  ; \*\*\*\*\*\*\*\*\*\* Start of BPS201 Logic \*\*\*\*\*\*\*\*\*\*  ;  ; BPS201STR = string of valid codes  ;   ; Upon completion of Eligibility codes, values will be placed into a string delimited  ; by commas. e.g. C,T  ;   ; If user includes (A)ll as an Eligibility code, only A will be stored in BPARR  ; array. e.g. Entry of C,T,A will save as BPARR(1.13)="A"  ;   ; User input values are temporarily stored in array BPS201 to eliminate duplicate   ; entries. e.g. Entry of T,C,T will save as BPARR(1.13)="C,T"  ;  BPS201 ; FIELD 2.01 - Eligibility Type  ;  S BPS201STR=",V,T,C,A,"  S DIR(0)="FO^0:7"  S DIR("A",1)=""  S DIR("A",2)=" Select one of the following:"  S DIR("A",3)=""  S DIR("A",4)=" V VETERAN"  S DIR("A",5)=" T TRICARE"  S DIR("A",6)=" C CHAMPVA"  S DIR("A",7)=" A ALL"  S DIR("A",8)=""  S DIR("A")="Select One or Many Eligibility Types or (A)ll"  S DIR("B")="V"   I $G(BPARR(2.01))'="" S DIR("B")=BPARR(2.01)  S DIR("?",1)="Enter a single response or multiple responses separated by commas."  S DIR("?",2)=" Example:"  S DIR("?",3)=" T"  S DIR("?")=" T,C"  D ^DIR K DIR  ;  I $D(DIRUT) Q  ;  ; Loop through user input (returned in variable X).  ; Display warning message if any user input selection is not included  ; in the string of acceptable codes (BPS201STR) and re-prompt question.  ; Assign valid selections to BPS201 array. This array will prevent  ; duplicate entries from being saved to the user's profile.  ;  K BPS201  S BPSERR=0  F BPSX=1:1:$L(X,",") D  . S BPSSEL=$P(X,",",BPSX)  . I BPS201STR'[(","\_BPSSEL\_",") W !," ",BPSSEL," is not a valid entry." S BPSERR=1 Q  . S BPS201(BPSSEL)=""  ;   I $G(BPSERR)=1 G BPS201  ;   ; If user included (A)ll as a selection, set profile setting to A.  ;  I $D(BPS201("A")) S BPARR(2.01)="A"  E  D  ; User did not enter "A".  . ;  . ; At this point user selections are valid and do not include "A".  . ; Loop through valid user selections. Set selections into a  . ; comma delimited string before assigning to BPARR array.  . ;  . S BPSSEL=""  . F  S BPSSEL=$O(BPS201(BPSSEL)) Q:BPSSEL=""  D  . . S BPS201=$G(BPS201)\_BPSSEL\_","  . S BPS201=$E(BPS201,1,($L(BPS201)-1))  . S BPARR(2.01)=BPS201  ;  ; ^^^^^^^^^^ End of BPS201 Logic ^^^^^^^^^^  ;  ; \*\*\*\*\*\*\*\*\*\* Start of BPS101 / BPS116 Logic \*\*\*\*\*\*\*\*\*\*  ;  ; User input will be temporarily stored in BPS116AR for display to user   ; of selected USERS while in CV Action.  ;   ; If U is selected, at least one USER must be selected. If not, selection   ; will default to (A)ll and the current question will be re-prompted.  ;   ; Upon completion of USERs entry, values will be placed into a string   ; delimited by semicolons. e.g. BPARR(1.16)=";12345;56789;"  ;   ; BPS116AR = array containing USER information  ; BPS116AR(BPS116)="" BPS116 = IEN from New Person File #200  ; BPS116AR("B",BPS116NM) BPS116NM = User Name - Index used to display   ; selected USERs in alphabetical order while in CV Action.  ; BPS101 ; Fields 1.01 and 1.16 - User(s)  ;  S BPINP=$$EDITFLD(1.01,+BPDUZ7,"S^U:ONE USER;A:ALL","Display One or Many ECME (U)sers or (A)LL","ALL",.BPARR)  I BPINP=-1 Q  ;  S BPSERR=0  ;  ; If user selection is U, assign existing entry(s) into BPS116AR array.  ;  I $P(BPINP,U,2)="U" D  . S BPS116=$G(BPARR(1.16))  . I BPS116'="" D  . . S BPSCNT=$L(BPS116,";")  . . I BPSCNT=1 D  . . . S BPS116NM=$$GET1^DIQ(200,BPS116,.01)  . . . I $G(BPS116NM)'="" S BPS116AR(BPS116)="",BPS116AR("B",BPS116NM)=""  . . I +BPSCNT>2 D  . . . F I=2:1:BPSCNT-1 D  . . . . S BPS116X=$P(BPS116,";",I),BPS116NM=$$GET1^DIQ(200,BPS116X,.01)  . . . . I $G(BPS116NM)'="" S BPS116AR(BPS116X)="",BPS116AR("B",BPS116NM)=""  . ;  . ; Display existing entry(s) to user.  . ;  . I $D(BPS116AR) D  . . W !,?2,"Selected:"  . . S BPS116NM="" F  S BPS116NM=$O(BPS116AR("B",BPS116NM)) Q:BPS116NM=""  W !,?10,BPS116NM  . ;  . W !!,"Enter a user to select."  . W !,"Once all users are selected, hit enter without making a selection.",!  . ;  . S BPS116="" F  D  Q:BPS116=-1  . . S DIR0="P^VA(200,"  . . S PRMTMSG="Select User"  . . S DFLTVAL=""  . . S BPS116=$$PROMPT(DIR0,PRMTMSG,DFLTVAL)  . . ;  . . ; Exit 'Select User' loop if user entered nil.  . . ;   . . I BPS116=-1 Q  . . ;   . . S BPS116NM=$$GET1^DIQ(200,BPS116,.01)  . . ;  . . ; If entry exists in BPS116AR array, prompt user to delete from list.  . . ;   . . I $D(BPS116AR(BPS116)) D  . . . S DIR(0)="S^Y:YES;N:NO"  . . . S DIR("A")="Delete "\_BPS116NM\_" from your list?"  . . . S DIR("B")="NO"   . . . D ^DIR  . . . I Y="Y" K BPS116AR(BPS116),BPS116AR("B",BPS116NM)  . . . ;   . . E  D  . . . ;   . . . ; Set new entry into BPS116AR array.  . . . ;  . . . I $G(BPS116NM)'="" S BPS116AR(BPS116)="",BPS116AR("B",BPS116NM)=""  . . ;  . . ; Display existing entry(s) to user.   . . ;  . . I $D(BPS116AR) D  . . . W !,?2,"Selected:"  . . . S BPS116NM="" F  S BPS116NM=$O(BPS116AR("B",BPS116NM)) Q:BPS116NM=""  W !,?10,BPS116NM  . ;  . ; If user selected (U)ser but has not selected any users,  . ; set profile setting to ALL and set BPSERR flag to re-prompt question.  . ;   . I '$D(BPS116AR) S BPARR(1.01)="A",BPARR(1.16)="" S BPSERR=1 Q  . ;  . ; Loop through selected users, setting selected users into   . ; BPARR array - delimited by semi-colon.  . ;   . S BPARR(1.16)=";"  . S BPS116=0 F  S BPS116=$O(BPS116AR(BPS116)) Q:+BPS116=0 S BPARR(1.16)=BPARR(1.16)\_BPS116\_";"  ;  ; If BPSERR flag is 1, re-prompt question.  ;   I $G(BPSERR)=1 G BPS101  ;  ; ^^^^^^^^^^ End of BPS101 / BPS116 Logic ^^^^^^^^^^  ;  ; \*\*\*\*\*\*\*\*\*\* Start of BPS102 / BPS117 Logic \*\*\*\*\*\*\*\*\*\*  ;  ; User input will be temporarily stored in BPS117AR for display to user   ; of selected PATIENTS while in CV Action.  ;   ; If P is selected, at least one PATIENT must be selected. If not, selection   ; will default to (A)ll and the current question will be re-prompted.  ;   ; Upon completion of PATIENTs entry, values will be placed into a string   ; delimited by semicolons. e.g. BPARR(1.17)=";12345;56789;"  ;   ; BPS117AR = array containing PATIENT information  ; BPS117AR(BPS117)="" BPS117 = IEN from Patient File ^DPT  ; BPS117AR("B",BPS117M) BPS117NM = Patient Name - Index used to display   ; selected PATIENTs in alphabetical order while in CV Action.   ; BPS102 ; Fields 1.02 and 1.17 - Patient(s)  ;  S BPINP=$$EDITFLD(1.02,+BPDUZ7,"S^P:ONE PATIENT;A:ALL","Display One or Many (P)atients or (A)LL","ALL",.BPARR)  I BPINP=-1 Q  ;  S BPSERR=0  ;  ; If user selection is P, assign existing entry(s) into BPS117AR array.  ;  I $P(BPINP,U,2)="P" D  . S BPS117=$G(BPARR(1.17))  . I BPS117'="" D  . . S BPSCNT=$L(BPS117,";")  . . I BPSCNT=1 D  . . . S BPS117NM=$$GET1^DIQ(2,BPS117,.01)  . . . I $G(BPS117NM)'="" S BPS117AR(BPS117)="",BPS117AR("B",BPS117NM)=""  . . I +BPSCNT>2 D  . . . F I=2:1:BPSCNT-1 D  . . . . S BPS117X=$P(BPS117,";",I),BPS117NM=$$GET1^DIQ(2,BPS117X,.01)  . . . . I $G(BPS117NM)'="" S BPS117AR(BPS117X)="",BPS117AR("B",BPS117NM)=""  . ;  . ; Display existing entry(s) to user.  . ;   . I $D(BPS117AR) D  . . W !,?2,"Selected:"  . . S BPS117NM="" F  S BPS117NM=$O(BPS117AR("B",BPS117NM)) Q:BPS117NM=""  W !,?10,BPS117NM  . ;  . W !!,"Enter a patient to select."  . W !,"Once all patients are selected, hit enter without making a selection.",!  . ;   . S BPS117="" F  D  Q:BPS117=-1  . . S DIR0="P^DPT("  . . S PRMTMSG="Select Patient"  . . S DFLTVAL=""  . . S BPS117=$$PROMPT(DIR0,PRMTMSG,DFLTVAL)  . . ;  . . ; Exit 'Select Patient' loop if user entered nil.  . . ;   . . I BPS117=-1 Q  . . ;   . . S BPS117NM=$$GET1^DIQ(2,BPS117,.01)  . . ;  . . ; If entry exists in BPS117AR array, prompt user to delete from list.  . . ;   . . I $D(BPS117AR(BPS117)) D  . . . S DIR(0)="S^Y:YES;N:NO"  . . . S DIR("A")="Delete "\_BPS117NM\_" from your list?"  . . . S DIR("B")="NO"  . . . D ^DIR  . . . I Y="Y" K BPS117AR(BPS117),BPS117AR("B",BPS117NM)  . . . ;   . . E  D  . . . ;   . . . ; Set new entry into BPS117AR array.  . . . ;   . . . I $G(BPS117NM)'="" S BPS117AR(BPS117)="",BPS117AR("B",BPS117NM)=""  . . ;  . . ; Display existing entry(s) to user.  . . ;   . . I $D(BPS117AR) D  . . . W !,?2,"Selected:"  . . . S BPS117NM="" F  S BPS117NM=$O(BPS117AR("B",BPS117NM)) Q:BPS117NM=""  W !,?10,BPS117NM  . ;  . ; If user selected (P)atient but has not selected any patients,  . ; set profile setting to ALL and re-prompt question.  . ;   . I '$D(BPS117AR) S BPARR(1.02)="A",BPARR(1.17)="" S BPSERR=1 Q  . ;  . ; Loop through selected patients, setting selected patients into  . ; BPARR array - delimited by semi-colon.  . ;  . S BPARR(1.17)=";"  . S BPS117=0 F  S BPS117=$O(BPS117AR(BPS117)) Q:+BPS117=0 S BPARR(1.17)=BPARR(1.17)\_BPS117\_";"  I $G(BPSERR)=1 G BPS102  ;  ; ^^^^^^^^^^ End of BPS102 / BPS117 Logic ^^^^^^^^^^  ;  ; \*\*\*\*\*\*\*\*\*\* Start of BPS103 / BPS118 Logic \*\*\*\*\*\*\*\*\*\*  ;  ; User input will be temporarily stored in BPS118AR for display to user   ; of selected RXs while in CV Action.  ;   ; If R is selected, at least one RX must be selected. If not, selection   ; will default to (A)ll and the current question will be re-prompted.  ;   ; Upon completion of RXs entry, values will be placed into a string   ; delimited by semicolons. e.g. BPARR(1.18)=";12345;56789;"  ;   ; BPS118AR = array containing RX information  ; BPS118AR(BPS118)="" BPS118 = IEN from Prescription File #52  ; BPS118AR("B",BPSRXN) BPSRXN = RX Number - Index used to display   ; selected RXs in numerical order while in CV Action.  ;  BPS103 ; Fields 1.03 and 1.18 - Rx(s)  ;  S BPINP=$$EDITFLD(1.03,+BPDUZ7,"S^R:ONE RX;A:ALL","Display One or Many (R)x or (A)LL","ALL",.BPARR)  I BPINP=-1 Q  ;  S BPSERR=0  ;   ; If user selection is R, assign existing entry(s) into BPS118AR array.  ;  I $P(BPINP,U,2)="R" D  . S BPS118=$G(BPARR(1.18))  . I BPS118'="" D  . . S BPSCNT=$L(BPS118,";")  . . I BPSCNT=1 D  . . . S DIC=52  . . . S DR=".01;6"  . . . S DA=BPS118  . . . S DIQ="BPSRXD"  . . . S DIQ(0)="E"  . . . D DIQ^PSODI(52,DIC,DR,DA,.DIQ) ; ICR 4858  . . . ;   . . . S BPSRXN=$G(BPSRXD(52,DA,.01,"E"))  . . . S BPSDRUG=$G(BPSRXD(52,DA,6,"E"))  . . . ;   . . . S BPS118AR(BPS118)="",BPS118AR("B",BPSRXN)=BPSDRUG  . . . ;   . . I +BPSCNT>2 D  . . . F I=2:1:BPSCNT-1 D  . . . . S BPS118X=$P(BPS118,";",I)  . . . . S DIC=52  . . . . S DR=".01;6"  . . . . S DA=BPS118X  . . . . S DIQ="BPSRXD"  . . . . S DIQ(0)="E"  . . . . D DIQ^PSODI(52,DIC,DR,DA,.DIQ) ; ICR 4858  . . . . ;  . . . . S BPSRXN=$G(BPSRXD(52,DA,.01,"E"))  . . . . S BPSDRUG=$G(BPSRXD(52,DA,6,"E"))  . . . . ;   . . . . S BPS118AR(BPS118X)="",BPS118AR("B",BPSRXN)=BPSDRUG  . ;  . ; Display existing entry(s) to user.  . ;   . I $D(BPS118AR) D  . . W !,?2,"Selected:"  . . S BPSRXN="" F  S BPSRXN=$O(BPS118AR("B",BPSRXN)) Q:BPSRXN=""  W !,?10,BPSRXN,?30,BPS118AR("B",BPSRXN)  . ;  . W !!,"Enter a prescription to select."  . W !,"Once all prescriptions are selected, hit enter without making a selection.",!  . ;   . S BPS118="" F  D  Q:BPS118=-1  . . S PRMTMSG="Select RX"  . . S DFLTVAL=""  . . S BPS118=$$PROMPTRX^BPSUTIL1(PRMTMSG,DFLTVAL)  . . ;   . . ; Exit 'Select RX' loop if user entered nil.  . . ;  . . I BPS118=-1 Q  . . ;  . . S DIC=52  . . S DR=".01;6"  . . S DA=BPS118  . . S DIQ="BPSRXD"  . . S DIQ(0)="E"  . . D DIQ^PSODI(52,DIC,DR,DA,.DIQ) ; ICR 4858  . . ;   . . S BPSRXN=$G(BPSRXD(52,DA,.01,"E"))  . . S BPSDRUG=$G(BPSRXD(52,DA,6,"E"))  . .  . . ;   . . ; If entry exists in BPS118AR array, prompt user to delete from list.  . . ;   . . I $D(BPS118AR(BPS118)) D  . . . S DIR(0)="S^Y:YES;N:NO"  . . . S DIR("A")="Delete "\_BPSRXN\_" from your list?"  . . . S DIR("B")="NO"  . . . D ^DIR  . . . I Y="Y" K BPS118AR(BPS118),BPS118AR("B",BPSRXN)  . . . ;   . . E  D  . . . ; Set new entry into BPS118AR array.  . . . ;   . . . S BPS118AR(BPS118)="",BPS118AR("B",BPSRXN)=BPSDRUG  . . ;  . . ; Display existing entry(s) to user.  . . ;  . . I $D(BPS118AR) D  . . . W !,?2,"Selected:"  . . . S BPSRXN="" F  S BPSRXN=$O(BPS118AR("B",BPSRXN)) Q:BPSRXN=""  W !,?10,BPSRXN,?30,BPS118AR("B",BPSRXN)  . ;  . ; If user selected (R)x but has not selected any RXs,  . ; set profile setting to ALL and set BPSERR flag to re-prompt question.  . ;   . I '$D(BPS118AR) S BPARR(1.03)="A",BPARR(1.18)="" S BPSERR=1 Q  . ;  . ; Loop through selected RXs, setting selected RXs into  . ; BPARR array - delimited by semi-colon.  . ;   . S BPARR(1.18)=";"  . S BPS118=0 F  S BPS118=$O(BPS118AR(BPS118)) Q:+BPS118=0 S BPARR(1.18)=BPARR(1.18)\_BPS118\_";"  ;  ; If BPSERR flag is 1, re-prompt question.  ;  I $G(BPSERR)=1 G BPS103  ;  ; ^^^^^^^^^^ End of BPS103 / BPS118 Logic ^^^^^^^^^^  ;  S BPINP=$$EDITFLD(1.04,+BPDUZ7,"S^D:DAYS;H:HOURS","Activity Timeframe (H)ours or (D)ays","DAYS",.BPARR)  I BPINP=-1 Q  ;quit due to timeout or ^  S BPTF=$P(BPINP,U,2)  Q:$$EDITFLD(1.05,+BPDUZ7,"N^1:999:0","Activity Timeframe Value",$S(BPTF="H":24,1:7),.BPARR)=-1  Q:$$EDITFLD(2.02,+BPDUZ7,"S^O:OPEN CLAIMS;C:CLOSED CLAIMS;A:ALL","Select Open/Closed or All Claims","O",.BPARR)=-1  Q:$$EDITFLD(1.19,+BPDUZ7,"S^O:Open Non-Billable Entries;C:Closed Non-Billable Entries;A:ALL","Display (O)pen or (C)losed or (A)ll Non-Billable Entries","A",.BPARR)=-1  Q:$$EDITFLD(2.03,+BPDUZ7,"S^B:BILLING REQUESTS;R:REVERSALS;A:ALL","Select Submission Type","A",.BPARR)=-1  ;  ; \*\*\*\*\*\*\*\*\*\* Start of BPS106 Logic \*\*\*\*\*\*\*\*\*\*  ;  ; BPS106STR = string of valid codes  ;  ; Upon completion of prompt, values will be placed into a string delimited  ; by commas. e.g. P,R  ;  ; If user includes (A)ll as a code, only A will be stored in BPARR  ; array. e.g. Entry of R,P,A will save as BPARR(1.06)="A"  ;  ; User input values are temporary stored in array BPS106 to eliminate duplicate   ; entries. e.g. Entry of R,P,R will save as BPARR(1.06)="P,R"  ;  BPS106 ; FIELD 1.06 - Rejects / Payables / Unstranded / All  ;   S BPS106STR=",R,P,U,A,"  S DIR(0)="FO^0:7"  S DIR("A",1)=""  S DIR("A",2)=" Select one of the following:"  S DIR("A",3)=""  S DIR("A",4)=" R REJECTS"  S DIR("A",5)=" P PAYABLES"  S DIR("A",6)=" U UNSTRANDED"  S DIR("A",7)=" A ALL"  S DIR("A",8)=""  S DIR("A")="Display (R)ejects or (P)ayables or (U)nstranded or (A)ll"  S DIR("B")="A"   I $G(BPARR(1.06))'="" S DIR("B")=BPARR(1.06)  S DIR("?",1)="Enter a single response or multiple responses separated by commas."  S DIR("?",2)=" Example:"  S DIR("?",3)=" P"  S DIR("?")=" P,R"  D ^DIR K DIR  ;  I $D(DIRUT) Q  ;  ; Loop through user input (returned in variable X).  ; Display warning message if any user input selection is not included  ; in the string of acceptable codes (BPS106STR) and re-prompt question.  ; Assign valid selections to BPS106 array. This array will prevent  ; duplicate entries from being saved to the user's profile.  ;  K BPS106  S BPSERR=0  F BPSX=1:1:$L(X,",") D  . S BPSSEL=$P(X,",",BPSX)  . I BPS106STR'[(","\_BPSSEL\_",") W !," ",BPSSEL," is not a valid entry." S BPSERR=1 Q  . S BPS106(BPSSEL)=""  ;  I $G(BPSERR)=1 G BPS106  ;  ; If user included (A)ll as a selection, set profile setting to A.  ;  I $D(BPS106("A")) S BPARR(1.06)="A"  E  D  ; User did not enter "A"  . ;   . ; At this point user selections are valid and do not include "A".  . ; Loop through valid user selections. Set selections into a  . ; comma delimited string before assigning to BPARR array.  . ;   . S BPSSEL=""  . F  S BPSSEL=$O(BPS106(BPSSEL)) Q:BPSSEL=""  D  . . S BPS106=$G(BPS106)\_BPSSEL\_","  . S BPS106=$E(BPS106,1,($L(BPS106)-1))  . S BPARR(1.06)=BPS106  ;  ; ^^^^^^^^^^ End of BPS106 Logic ^^^^^^^^^^  ;  Q:$$EDITFLD(1.07,+BPDUZ7,"S^R:RELEASED;N:NON-RELEASED;A:ALL","Display (R)eleased Rxs or (N)on-Released Rxs or (A)LL","RELEASED",.BPARR)=-1  ;  ; \*\*\*\*\*\*\*\*\*\* Start of BPS108 Logic \*\*\*\*\*\*\*\*\*\*  ;  ; BPS108STR = string of valid codes  ;   ; Upon completion of prompt, values will be placed into a string delimited  ; by commas. e.g. C,M  ;   ; If user includes (A)ll as a code, only A will be stored in BPARR  ; array. e.g. Entry of C,M,A will save as BPARR(1.08)="A"  ;   ; User input values are temporary stored in array BPS108 to eliminate duplicate   ; entries. e.g. Entry of C,M,C will save as BPARR(1.08)="C,M"  ;  BPS108 ; FIELD 1.08 - CMOP / Mail / Window / All  ;   S BPS108STR=",C,M,W,A,"  S DIR(0)="FO^0:7" S DIR("A",1)=""  S DIR("A",2)=" Select one of the following:"  S DIR("A",3)=""  S DIR("A",4)=" C CMOP"  S DIR("A",5)=" M MAIL"  S DIR("A",6)=" W WINDOW"  S DIR("A",7)=" A ALL"  S DIR("A",8)=""  S DIR("A")="Display (C)MOP or (M)ail or (W)indow or (A)ll"  S DIR("B")="A" S:$G(BPARR(1.08))'="" DIR("B")=BPARR(1.08)  S DIR("?",1)="Enter a single response or multiple responses separated by commas."  S DIR("?",2)=" Example:"  S DIR("?",3)=" C"  S DIR("?")=" C,M"  D ^DIR K DIR  ;  I $D(DIRUT) Q  ;  ; Loop through user input (returned in variable X).  ; Display warning message if any user input selection is not included  ; in the string of acceptable codes (BPS108STR) and re-prompt question.  ; Assign valid selections to BPS108 array. This array will prevent  ; duplicate entries from being saved to the user's profile.  ;  K BPS108  S BPSERR=""  F BPSX=1:1:$L(X,",") D  . S BPSSEL=$P(X,",",BPSX)  . I BPS108STR'[(","\_BPSSEL\_",") W !," ",BPSSEL," is not a valid entry." S BPSERR=1 Q  . S BPS108(BPSSEL)=""  ;  I $G(BPSERR)=1 G BPS108  ;  ; If user included (A)ll as a seleection, set profile setting to A.  ;  I $D(BPS108("A")) S BPARR(1.08)="A"  E  D  ; User did not enter "A".  . ;  . ; At this point user selections are valid and do not include "A".  . ; Loop through valid user selections. Set selections into a  . ; comma delimited string before assigning to BPARR array.  . ;  . S BPSSEL=""  . F  S BPSSEL=$O(BPS108(BPSSEL)) Q:BPSSEL=""  D  . . S BPS108=$G(BPS108)\_BPSSEL\_","  . S BPS108=$E(BPS108,1,($L(BPS108)-1))  . S BPARR(1.08)=BPS108   ;  ; ^^^^^^^^^^ End of BPS108 Logic ^^^^^^^^^^  ;  ; \*\*\*\*\*\*\*\*\*\* Start of BPS109 Logic \*\*\*\*\*\*\*\*\*\*  ;  ; BPS109STR = string of valid codes  ;  ; Upon completion of prompt, values will be placed into a string delimited  ; by commas. e.g. P,R  ;  ; If user includes (A)ll as a code, only A will be stored in BPARR  ; array. e.g. Entry of R,P,A will save as BPARR(1.09)="A"'  ;  ; User input values are temporary stored in array BPS106 to eliminate duplicate   ; entries. e.g. Entry of R,P,R will save as BPARR(1.09)="P,R"  ;  BPS109 ; FIELD 1.09 - Realtime / Backbills / Pro Option / Resubmission / All  ;  S BPS109STR=",R,B,P,S,A,"  S DIR(0)="FO^0:7"  S DIR("A",1)=""  S DIR("A",2)=" Select one of the following:"  S DIR("A",3)=""  S DIR("A",4)=" R REALTIME"  S DIR("A",5)=" B BACKBILLS"  S DIR("A",6)=" P PRO OPTION"  S DIR("A",7)=" S RESUBMISSION"  S DIR("A",8)=" A ALL"  S DIR("A",9)=""  S DIR("A")="Display (R)ealTime, (B)ackbills, (P)RO Option, Re(S)ubmission or (A)ll"  S DIR("B")="A"   I $G(BPARR(1.09))'="" S DIR("B")=BPARR(1.09)  S DIR("?",1)="Enter a single response or multiple responses separated by commas."  S DIR("?",2)=" Example:"  S DIR("?",3)=" B"  S DIR("?")=" B,P"   D ^DIR K DIR  ;  I $D(DIRUT) Q  ;  ; Loop through user input (returned in variable X).  ; Display warning message if any user input selection is not included  ; in the string of acceptable codes (BPS109STR) and re-prompt question.  ; Assign valid selections to BPS109 array. This array will prevent  ; duplicate entries from being saved to the user's profile.  ;  K BPS109  S BPSERR=""  F BPSX=1:1:$L(X,",") D  . S BPSSEL=$P(X,",",BPSX)  . I BPS109STR'[(","\_BPSSEL\_",") W !," ",BPSSEL," is not a valid entry." S BPSERR=1 Q  . S BPS109(BPSSEL)=""  ;  I $G(BPSERR)=1 G BPS109  ;  ; If user included (A)ll as a selection, set profile setting to A.  ;  I $D(BPS109("A")) S BPARR(1.09)="A"  E  D  ; User did not enter "A".  . ;  . ; At this point user selections are valid and do not include "A".  . ; Loop through valid user selections. Set selections into a  . ; comma delimited string before assigning to BPARR array.  . ;  . S BPSSEL=""  . F  S BPSSEL=$O(BPS109(BPSSEL)) Q:BPSSEL=""  D  . . S BPS109=$G(BPS109)\_BPSSEL\_","  . S BPS109=$E(BPS109,1,($L(BPS109)-1))  . S BPARR(1.09)=BPS109   ;  ; ^^^^^^^^^^ End of BPS109 Logic ^^^^^^^^^^   ;  ; \*\*\*\*\*\*\*\*\*\* Start of BPS110 / BPS115 Logic \*\*\*\*\*\*\*\*\*\*  ;  ; User input will be temporarily stored in BPS115AR for display to user   ; of selected REJECT CODES while in CV Action.  ;   ; If R is selected, at least one REJECT CODE must be selected. If not, selection   ; will default to (A)ll and the current question will be re-prompted.  ;   ; Upon completion of REJECT CODEs entry, values will be placed into a string   ; delimited by semicolons. e.g. BPARR(1.15)=";50;60;"  ;   ; BPS115AR = array containing REJECT CODE information  ; BPS115AR(BPS115)=BPS115NM BPS115 = IEN from File #9002313.93  ; BPS115NM = Reject Code Explanation   ;  BPS110 ; Fields 1.10 and 1.15 - Reject Code(s)  ;  S BPINP=$$EDITFLD(1.1,+BPDUZ7,"S^R:REJECT CODE;A:ALL","Display Specific (R)eject Code or (A)LL","ALL",.BPARR)  I BPINP=-1 Q  ;  S BPSERR=0  ;   ; If user selection is R, assign existing entry(s) into BPS115AR array.  ;  I $P(BPINP,U,2)="R" D  . S BPS115=$G(BPARR(1.15))  . I BPS115'="" D  . . S BPSCNT=$L(BPS115,";")  . . I BPSCNT=1 D  . . . S BPS115NM=$$GET1^DIQ(9002313.93,BPS115,.02)  . . . S BPS115AR(BPS115)=BPS115NM  . . I +BPSCNT>2 D  . . . F I=2:1:BPSCNT-1 D  . . . . S BPS115X=$P(BPS115,";",I),BPS115NM=$$GET1^DIQ(9002313.93,BPS115X,.02)  . . . . S BPS115AR(BPS115X)=BPS115NM  . ;  . ; Display existing entry(s) to user.  . ;   . I $D(BPS115AR) D  . . W !,?2,"Selected:"  . . S BPS115="" F  S BPS115=$O(BPS115AR(BPS115)) Q:BPS115=""  W !,?10,BPS115,?20,BPS115AR(BPS115)  . ;  . S BPS115="" F  D  Q:BPS115=-1  . . S DIR0="P^BPSF(9002313.93,"  . . S PRMTMSG="Select Reject Code"  . . S DFLTVAL=""  . . S BPS115=$$PROMPT(DIR0,PRMTMSG,DFLTVAL)  . . ;   . . ; Exit 'Select Reject Code' loop if user entered nil.  . . ;   . . I BPS115=-1 Q  . . ;  . . S BPS115NM=$$GET1^DIQ(9002313.93,BPS115,.02)  . .   . . ;  . . ; If entry exists in BPS115AR array, prompt user to delete from list.  . . ;   . . I $D(BPS115AR(BPS115)) D  . . . S DIR(0)="S^Y:YES;N:NO"  . . . S DIR("A")="Delete "\_BPS115\_" from your list?"  . . . S DIR("B")="NO"  . . . D ^DIR  . . . I Y="Y" K BPS115AR(BPS115)  . . . ;   . . E  D  . . . ;   . . . ; Set new entry intp BPS115AR array.  . . . ;   . . . S BPS115AR(BPS115)=BPS115NM  . . ;  . . ; Display existing entry(s) to user.  . . ;   . . I $D(BPS115AR) D  . . . W !,?2,"Selected:"  . . . S BPS115="" F  S BPS115=$O(BPS115AR(BPS115)) Q:BPS115=""  W !,?10,BPS115,?20,BPS115AR(BPS115)  . ;  . ; If user selected (R)eject but has not selected any Reject Codes,  . ; set profile setting to ALL and set BPSERR flag to re-prompt question.  . ;  . I '$D(BPS115AR) S BPARR(1.1)="A",BPARR(1.15)="" S BPSERR=1 Q  . ;  . ; Loop through selected reject codes, setting selected reject codes into  . ; BPARR array - delimited by semi-colon.   . ;   . S BPARR(1.15)=";"  . S BPS115=0 F  S BPS115=$O(BPS115AR(BPS115)) Q:+BPS115=0 S BPARR(1.15)=BPARR(1.15)\_BPS115\_";"  ;   ; If BPSERR flag is 1, re-prompt question.  ;   I $G(BPSERR)=1 G BPS110  ;  ; ^^^^^^^^^^ End of BPS110 / BPS115 Logic ^^^^^^^^^^  ;  Q:$$INSURSEL^BPSSCRCU(.BPARR,+BPDUZ7)=-1  Q  ; | | | | | | | |

Update the header to no longer include information pertaining to Users. In addition to removing User information, right justify the 2nd and 3rd header lines (aligning the colons). HDR right justifies the header lines. GETVDETS removes the User information.

| Subroutine Name | **HDR^BPSSCR01** | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Enhancement Category | New | Modify | | | Delete | | No Change | |
| Story | US591 | | | | | | | |
| Related Menu Options or ListMan Actions | ECME User Screen | | | | | | | |
| Related Routines | This Subroutine is Called By | | | This Subroutine Calls | | | | |
| ^BPSRSM  ^BPSSCR | | | RESTVIEW^BPSSCR01  $$GETVDIVS^BPSSCR01  $$GETVDETS^BPSSCR01  $$SORTTYPE^BPSSCRSL | | | | |
| Related Integration Control Registrations (ICRs) | N/A | | | | | | | |
| Data Passing | Parameter Input | | Parameter Output | | | Function Return Value | | Global Modified |
| Input Attribute Name and Definition | Name: BPSLN  Definition: Header Line Number  New  Modify  Delete  No Change | | | | | | | |
| Output Attribute Name and Definition | Name: N/A  Definition:  New  Modify  Delete  No Change | | | | | | | |
| Current Logic | | | | | | | | |
| HDR(BPSLN) ; -- header code  N BPARR,BPX,BPXSL  Q:'$D(@VALMAR@("VIEWPARAMS"))  D RESTVIEW(.BPARR)  I BPSLN=1 Q "SELECTED DIVISION(S): "\_$$GETVDIVS(.BPARR,58)  I BPSLN=2 Q $$GETVDETS(.BPARR)  I BPSLN=3 D  Q $$LINE^BPSSCRU3(80-$L(BPX)," ")\_BPX  . S BPXSL=$$SORTTYPE^BPSSCRSL($G(BPARR(1.12)))  . I BPXSL="" S BPXSL="Transaction date by default"  . S BPX="Sorted by: "\_BPXSL  Q "" | | | | | | | | |
| Modified Logic (Changes are highlighted) | | | | | | | | |
| HDR(BPSLN) ; -- header code  N BPARR,BPX,BPXSL  Q:'$D(@VALMAR@("VIEWPARAMS"))  D RESTVIEW(.BPARR)  I BPSLN=1 Q "SELECTED DIVISION(S): "\_$$GETVDIVS(.BPARR,58)  I BPSLN=2 Q $$GETVDETS(.BPARR)  I BPSLN=3 D  Q BPX  . S BPXSL=$$SORTTYPE^BPSSCRSL($G(BPARR(1.12)))  . I BPXSL="" S BPXSL="Transaction date by default"  . S BPX=" Sorted by: "\_BPXSL  Q "" | | | | | | | | |

| Subroutine Name | **GETVDETS^BPSSCR01** | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Enhancement Category | New | Modify | | | Delete | | No Change | |
| Story | US591 | | | | | | | |
| Related Menu Options or ListMan Actions | ECME User Screen | | | | | | | |
| Related Routines | This Subroutine is Called By | | | This Subroutine Calls | | | | |
| HDR^BPSCMT  HDR^BPSSCR01 | | | $$LJ^BPPSCR02 | | | | |
| Related Integration Control Registrations (ICRs) | N/A | | | | | | | |
| Data Passing | Parameter Input | | Parameter Output | | | Function Return Value | | Global Modified |
| Input Attribute Name and Definition | Name: BPARR  Definition: User Profile Array – Parameter Definition “BPS USRSCR”  New  Modify  Delete  No Change | | | | | | | |
| Output Attribute Name and Definition | Name: N/A  Definition:  New  Modify  Delete  No Change | | | | | | | |
| Current Logic | | | | | | | | |
| GETVDETS(BPARR) ;\*/  N BPSTR  I $G(BPARR(1.01))="A" S BPSTR=$$LJ^BPSSCR02("Transmitted by ALL users",31)  E  S BPSTR=$$LJ^BPSSCR02("Transmitted by "\_$$GETUSRNM^BPSSCRU1($G(BPARR(1.16))),31)  S BPSTR=BPSTR\_$$LJ^BPSSCR02(" Activity Date Range: within the past "\_$G(BPARR(1.05))\_$S($G(BPARR(1.04))="H":" hour(s)",1:" day(s)"),49)  Q BPSTR | | | | | | | | |
| Modified Logic (Changes are highlighted) | | | | | | | | |
| GETVDETS(BPARR) ;\*/  N BPSTR ~~I $G(BPARR(1.01))="A" S BPSTR=$$LJ^BPSSCR02("Transmitted by ALL users",31)  E  S BPSTR=$$LJ^BPSSCR02("Transmitted by "\_$$GETUSRNM^BPSSCRU1($G(BPARR(1.16))),31)~~  S BPSTR=" Activity Date Range: within the past "\_$G(BPARR(1.05))\_$S($G(BPARR(1.04))="H":" hour(s)",1:" day(s)")~~,49)~~  Q BPSTR | | | | | | | | |

Filter the information that displays on the User Screen based on the options selected in **Change View.** Previously the filters were based on ALL values or a SINGLE value. Entries that included a single value, set the value into the BPARR array as follows: BPARR(1.16)=12345. The BPARR array has been updated to allow a string of values, separated by semi-colons. Eg. BPARR(1.16)=”;12345;56789;”.

The updated filter logic has been written to check for the old method of saving the information and the new method. This prevents a conversion routine from having to be written, updating existing entries to include semi-colons. The remaining Subroutine Tables are all specific to the filtering logic.

Field 2.01: Eligibility

| Subroutine Name | **$$FLTELIG^BPSSCR05** | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Enhancement Category | New | Modify | | | Delete | | No Change | |
| Story | US591 | | | | | | | |
| Related Menu Options or ListMan Actions | ECME User Screen | | | | | | | |
| Related Routines | This Subroutine is Called By | | | This Subroutine Calls | | | | |
| FILTER^BPSSCR03 | | | $$ELIGCODE^BPSSCR05 | | | | |
| Related Integration Control Registrations (ICRs) | N/A | | | | | | | |
| Data Passing | Parameter Input | | Parameter Output | | | Function Return Value | | Global Modified |
| Input Attribute Name and Definition | Name: BP59  Definition: Ptr to #59  New  Modify  Delete  No Change  Name: BPARR  Definition: Array with user’s preferences  New  Modify  Delete  No Change | | | | | | | |
| Output Attribute Name and Definition | Name: N/A  Definition:  New  Modify  Delete  No Change | | | | | | | |
| Current Logic | | | | | | | | |
| FLTELIG(BP59,BPARR) ;  Q:$G(BPARR(2.01))="A" 1  I $G(BPARR(2.01))="" S BPARR(2.01)="V"  Q $S($$ELIGCODE(BP59)=$G(BPARR(2.01)):1,1:0) | | | | | | | | |
| Modified Logic (Changes are highlighted) | | | | | | | | |
| FLTELIG(BP59,BPARR) ;  Q:$G(BPARR(2.01))="A" 1  I $G(BPARR(2.01))="" S BPARR(2.01)="V"  I $G(BPARR(2.01))[$$ELIGCODE(BP59) Q 1  Q 0 | | | | | | | | |

Field 1.01/1.16 - User

| Subroutine Name | **FLTUSR^BPSSCR03** | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Enhancement Category | New | Modify | | | Delete | | No Change | |
| Story | US591 | | | | | | | |
| Related Menu Options or ListMan Actions | ECME User Screen | | | | | | | |
| Related Routines | This Subroutine is Called By | | | This Subroutine Calls | | | | |
| FILTER^BPSSCR03 | | | N/A | | | | |
| Related Integration Control Registrations (ICRs) | N/A | | | | | | | |
| Data Passing | Parameter Input | | Parameter Output | | | Function Return Value | | Global Modified |
| Input Attribute Name and Definition | Name: BPST0  Definition: Zero node of #9002313.59  New  Modify  Delete  No Change  Name: BPARR  Definition: Array with user’s preferences  New  Modify  Delete  No Change | | | | | | | |
| Output Attribute Name and Definition | Name: N/A  Definition:  New  Modify  Delete  No Change | | | | | | | |
| Current Logic | | | | | | | | |
| FLTUSR(BPST0,BPARR) ;  I $L($G(BPARR(1.16)))=0 Q 0  I $P(BPST0,U,10)'=$G(BPARR(1.16)) Q 0  Q 1 | | | | | | | | |
| Modified Logic (Changes are highlighted) | | | | | | | | |
| FLTUSR(BPST0,BPARR) ;  I $L($G(BPARR(1.16)))=0 Q 0  I $P(BPST0,U,10)=$G(BPARR(1.16)) Q 1  I $G(BPARR(1.16))[(";"\_$P(BPST0,U,10)\_";") Q 1  Q 0 ; | | | | | | | | |

Field 1.02/1.17 - Patient

| Subroutine Name | **FLTPAT^BPSSCR03** | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Enhancement Category | New | Modify | | | Delete | | No Change | |
| Story | US591 | | | | | | | |
| Related Menu Options or ListMan Actions | ECME User Screen | | | | | | | |
| Related Routines | This Subroutine is Called By | | | This Subroutine Calls | | | | |
| FILTER^BPSSCR03 | | | N/A | | | | |
| Related Integration Control Registrations (ICRs) | N/A | | | | | | | |
| Data Passing | Parameter Input | | Parameter Output | | | Function Return Value | | Global Modified |
| Input Attribute Name and Definition | Name: BPST0  Definition: Zero node of #9002313.59  New  Modify  Delete  No Change  Name: BPARR  Definition: Array with user’s preferences  New  Modify  Delete  No Change | | | | | | | |
| Output Attribute Name and Definition | Name: N/A  Definition:  New  Modify  Delete  No Change | | | | | | | |
| Current Logic | | | | | | | | |
| FLTPAT(BPST0,BPARR) ;  I $L($G(BPARR(1.17)))=0 Q 0  I $P(BPST0,U,6)'=$G(BPARR(1.17)) Q 0  Q 1 | | | | | | | | |
| Modified Logic (Changes are highlighted) | | | | | | | | |
| FLTPAT(BPST0,BPARR) ;  I $L($G(BPARR(1.17)))=0 Q 0  I $P(BPST0,U,6)=$G(BPARR(1.17)) Q 1  I $G(BPARR(1.17))[(";"\_$P(BPST0,U,6)\_";") Q 1  Q 0 | | | | | | | | |

Field 1.03/1.18 - Rx

| Subroutine Name | **FLTRX^BPSSCR03** | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Enhancement Category | New | Modify | | | Delete | | No Change | |
| Story | US591 | | | | | | | |
| Related Menu Options or ListMan Actions | ECME User Screen | | | | | | | |
| Related Routines | This Subroutine is Called By | | | This Subroutine Calls | | | | |
| FILTER^BPSSCR03 | | | N/A | | | | |
| Related Integration Control Registrations (ICRs) | N/A | | | | | | | |
| Data Passing | Parameter Input | | Parameter Output | | | Function Return Value | | Global Modified |
| Input Attribute Name and Definition | Name: BPST1  Definition: 1st node of #9002313.59  New  Modify  Delete  No Change  Name: BPARR  Definition: Array with user’s preferences  New  Modify  Delete  No Change | | | | | | | |
| Output Attribute Name and Definition | Name: N/A  Definition:  New  Modify  Delete  No Change | | | | | | | |
| Current Logic | | | | | | | | |
| FLTRX(BPST1,BPARR) ;  I $L($G(BPARR(1.18)))=0 Q 0  I $P(BPST1,U,11)'=$G(BPARR(1.18)) Q 0  Q 1 | | | | | | | | |
| Modified Logic (Changes are highlighted) | | | | | | | | |
| FLTRX(BPST1,BPARR) ;  I $L($G(BPARR(1.18)))=0 Q 0   I $P(BPST1,U,11)=$G(BPARR(1.18)) Q 1  I $G(BPARR(1.18))[(";"\_$P(BPST1,U,11)\_";") Q 1  Q 0 | | | | | | | | |

Field 1.06 – Rejected / Payable / Unstranded

| Subroutine Name | **FILTER^BPSSCR03** | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Enhancement Category | New | Modify | | | Delete | | No Change | |
| Story | US591 | | | | | | | |
| Related Menu Options or ListMan Actions | ECME User Screen | | | | | | | |
| Related Routines | This Subroutine is Called By | | | This Subroutine Calls | | | | |
| FILTRALL^BPSSCR03 | | | $$REJECTED^BPSSCR02  $$PAYABLE^BPSSCR02  $$UNSTRAND^BPSSCR02 | | | | |
| Related Integration Control Registrations (ICRs) | N/A | | | | | | | |
| Data Passing | Parameter Input | | Parameter Output | | | Function Return Value | | Global Modified |
| Input Attribute Name and Definition | Name: BP59  Definition: Pointer to file #9002313.59  New  Modify  Delete  No Change  Name: BPARR  Definition: Array with user’s preferences  New  Modify  Delete  No Change | | | | | | | |
| Output Attribute Name and Definition | Name: N/A  Definition:  New  Modify  Delete  No Change | | | | | | | |
| Current Logic | | | | | | | | |
| FILTER(BP59,BPARR) ;  N BPST0,BPST1,BPRXREF,BPRX52,BPREFNUM,BPRTBB  N BPRET  …   ;only rejected  I '$$NB(BP59),$G(BPARR(1.06))="R",$$REJECTED^BPSSCR02(BP59)=0 Q 0 ; n/a for non-billables  ;only payable  I '$$NB(BP59),$G(BPARR(1.06))="P",$$PAYABLE^BPSSCR02(BP59)=0 Q 0 ; n/a for non-billables  ;only unstranded  I '$$NB(BP59),$G(BPARR(1.06))="U",$$UNSTRAND^BPSSCR02(BP59)=0 Q 0 ; n/a for non-billables  ;  … | | | | | | | | |
| Modified Logic (Changes are highlighted) | | | | | | | | |
| FILTER(BP59,BPARR) ;  N BPST0,BPST1,BPRXREF,BPRX52,BPREFNUM,BPRTBB  N BPRET,BPSRPU  …   S BPSRPU=1  I '$$NB(BP59) D    ; n/a for non-billables  . S BPSRPU=0  . I ($G(BPARR(1.06))["R")&($$REJECTED^BPSSCR02(BP59)=1) S BPSRPU=1  . I ($G(BPARR(1.06))["P")&($$PAYABLE^BPSSCR02(BP59)=1) S BPSRPU=1  . I ($G(BPARR(1.06))["U")&($$UNSTRAND^BPSSCR02(BP59)=1) S BPSRPU=1  . I $G(BPARR(1.06))["A" S BPSRPU=1  I BPSRPU=0 Q 0  … | | | | | | | | |

Field 1.08 – CMOP / Mail / Window

| Subroutine Name | **ISMWC^BPSSCR03** | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Enhancement Category | New | Modify | | | Delete | | No Change | |
| Story | US591 | | | | | | | |
| Related Menu Options or ListMan Actions | ECME User Screen | | | | | | | |
| Related Routines | This Subroutine is Called By | | | This Subroutine Calls | | | | |
| FILTER^BPSSCR03 | | | $$MWCNAME^BPSSCRU2  $$MWC^BPSSCRU2 | | | | |
| Related Integration Control Registrations (ICRs) | N/A | | | | | | | |
| Data Passing | Parameter Input | | Parameter Output | | | Function Return Value | | Global Modified |
| Input Attribute Name and Definition | Name: BPRX59  Definition: Pointer to file #9002313.59  New  Modify  Delete  No Change  Name: BPREFNUM  Definition: Refill #  New  Modify  Delete  No Change  Name: BPMWC  Definition: Value from BPARR(1.08)  New  Modify  Delete  No Change | | | | | | | |
| Output Attribute Name and Definition | Name: N/A  Definition:  New  Modify  Delete  No Change | | | | | | | |
| Current Logic | | | | | | | | |
| ISMWC(BPRX52,BPREFNUM,BPMWC) ;  I $$MWCNAME^BPSSCRU2($$MWC^BPSSCRU2(BPRX52,BPREFNUM))=BPMWC Q 1  Q 0 | | | | | | | | |
| Modified Logic (Changes are highlighted) | | | | | | | | |
| ISMWC(BPRX52,BPREFNUM,BPMWC) ;  I BPMWC[$$MWCNAME^BPSSCRU2($$MWC^BPSSCRU2(BPRX52,BPREFNUM)) Q 1  Q 0 | | | | | | | | |

Field 1.09 – RealTime, Backbills, PRO Option, Resubmission

| Subroutine Name | **FILTER^BPSSCR03** | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Enhancement Category | New | Modify | | | Delete | | No Change | |
| Story | US591 | | | | | | | |
| Related Menu Options or ListMan Actions | ECME User Screen | | | | | | | |
| Related Routines | This Subroutine is Called By | | | This Subroutine Calls | | | | |
| FILTRALL^BPSSCR03 | | | $$RTBB^BPSSCRU2 | | | | |
| Related Integration Control Registrations (ICRs) | N/A | | | | | | | |
| Data Passing | Parameter Input | | Parameter Output | | | Function Return Value | | Global Modified |
| Input Attribute Name and Definition | Name: BP59  Definition: Pointer to file #9002313.59  New  Modify  Delete  No Change  Name: BPARR  Definition: Array with user’s preferences  New  Modify  Delete  No Change | | | | | | | |
| Output Attribute Name and Definition | Name: N/A  Definition:  New  Modify  Delete  No Change | | | | | | | |
| Current Logic | | | | | | | | |
| FILTER(BP59,BPARR) ;  N BPST0,BPST1,BPRXREF,BPRX52,BPREFNUM,BPRTBB  N BPRET  …   ; filter checks for fill type  S BPRTBB=$$RTBB^BPSSCRU2(BP59) I BPRTBB="\*\*" S BPRTBB="RT"  I $G(BPARR(1.09))="B",BPRTBB'="BB" Q 0 ; filter for back billing  I $G(BPARR(1.09))="P",BPRTBB'="P2" Q 0 ; filter for PRO Option  I $G(BPARR(1.09))="S",BPRTBB'="RS" Q 0 ; filter for ECME user screen resubmits (BPS\*1\*20)  I $G(BPARR(1.09))="R",BPRTBB'="RT" Q 0 ; filter for real time  … | | | | | | | | |
| Modified Logic (Changes are highlighted) | | | | | | | | |
| FILTER(BP59,BPARR) ;  N BPST0,BPST1,BPRXREF,BPRX52,BPREFNUM,BPRTBB  N BPRET,BPSFT  …   ; filter checks for fill type  S BPRTBB=$$RTBB^BPSSCRU2(BP59) I BPRTBB="\*\*" S BPRTBB="RT"  S BPSFT=1  I $G(BPARR(1.09))'["A" D  . S BPSFT=0  . I $G(BPARR(1.09))["B",BPRTBB="BB" S BPSFT=1 ; filter for back billing  . I $G(BPARR(1.09))["P",BPRTBB="P2" S BPSFT=1 ; filter for PRO Option  . I $G(BPARR(1.09))["S",BPRTBB="RS" S BPSFT=1 ; filter for ECME user screen resubmits (BPS\*1\*20)  . I $G(BPARR(1.09))["R",BPRTBB="RT" S BPSFT=1 ; filter for real time  I BPSFT=0 Q 0  ... | | | | | | | | |

Reject Codes – If BPARR(1.06) contains R for Rejects & specific Reject Codes are selected (BPARR(1.1)=”R”), call code ($$FLTREJ) to filter based on selected reject codes (BPARR(1.15)).

| Subroutine Name | **FILTER^BPSSCR03** | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Enhancement Category | New | Modify | | | Delete | | No Change | |
| Story | US591 | | | | | | | |
| Related Menu Options or ListMan Actions | ECME User Screen | | | | | | | |
| Related Routines | This Subroutine is Called By | | | This Subroutine Calls | | | | |
| FILTRALL^BPSSCR03 | | | $$FLTREJ^BPSSCR03 | | | | |
| Related Integration Control Registrations (ICRs) | N/A | | | | | | | |
| Data Passing | Parameter Input | | Parameter Output | | | Function Return Value | | Global Modified |
| Input Attribute Name and Definition | Name: BP59  Definition: Pointer to file #9002313.59  New  Modify  Delete  No Change  Name: BPARR  Definition: Array with user’s preferences  New  Modify  Delete  No Change | | | | | | | |
| Output Attribute Name and Definition | Name: N/A  Definition:  New  Modify  Delete  No Change | | | | | | | |
| Current Logic | | | | | | | | |
| FILTER(BP59,BPARR) ; …   ;if only rejected and only specific rejected codes should be displayed  I $G(BPARR(1.06))="R",$G(BPARR(1.1))="R",$$FLTREJ(BP59,.BPARR)=0 Q 0  … | | | | | | | | |
| Modified Logic (Changes are highlighted) | | | | | | | | |
| FILTER(BP59,BPARR) ; …   ;if only rejected and only specific rejected codes should be displayed  I $G(BPARR(1.06))["R",$G(BPARR(1.1))="R",$$FLTREJ(BP59,.BPARR)=0 Q 0  … | | | | | | | | |

Reject Codes (continued)

| Subroutine Name | **FLTREJ^BPSSCR03** | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Enhancement Category | New | Modify | | | Delete | | No Change | |
| Story | US591 | | | | | | | |
| Related Menu Options or ListMan Actions | ECME User Screen | | | | | | | |
| Related Routines | This Subroutine is Called By | | | This Subroutine Calls | | | | |
| FILTER^BPSSCR03 | | | REJCODES^BPSSCRU3 | | | | |
| Related Integration Control Registrations (ICRs) | N/A | | | | | | | |
| Data Passing | Parameter Input | | Parameter Output | | | Function Return Value | | Global Modified |
| Input Attribute Name and Definition | Name: BP59  Definition: Pointer to file #9002313.59  New  Modify  Delete  No Change  Name: BPARR  Definition: Array with user’s preferences  New  Modify  Delete  No Change | | | | | | | |
| Output Attribute Name and Definition | Name: N/A  Definition:  New  Modify  Delete  No Change | | | | | | | |
| Current Logic | | | | | | | | |
| FLTREJ(BP59,BPARR) ;  N BPRCODES  N BPRJCD  S BPRJCD=$P($G(^BPSF(9002313.93,+$G(BPARR(1.15)),0)),U)  I $L(BPRJCD)=0 Q 0  D REJCODES^BPSSCRU3(BP59,.BPRCODES,1) ; bps\*1\*20 include possible non-billable pseudo-reject codes too  I $D(BPRCODES(BPRJCD)) Q 1  Q 0 | | | | | | | | |
| Modified Logic (Changes are highlighted) | | | | | | | | |
| FLTREJ(BP59,BPARR) ;  N BPRCODES  N BPSRJCD,BPSRJFLAG,BPSRJIEN  D REJCODES^BPSSCRU3(BP59,.BPRCODES,1) ; bps\*1\*20 include possible non-billable pseudo-reject codes too  ;  S BPSRJCD="",BPSRJFLAG=0  F S BPSRJCD=$O(BPRCODES(BPSRJCD)) Q:BPSRJCD="" Q:BPSRJFLAG=1 D  . S BPSRJIEN="",BPSRJIEN=$O(^BPSF(9002313.93,"B",BPSRJCD,BPSRJIEN))  . I BPARR(1.15)=BPSRJIEN S BPSRJFLAG=1  . I BPARR(1.15)[(";"\_BPSRJIEN\_";") S BPSRJFLAG=1  Q BPSRJFLAG | | | | | | | | |

# Back Out/Rollback Procedure

This story includes modifications to existing routines. To back out this enhancement, all the routines which are part of this patch should be backed up prior to installation. 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.