**MCCF EDI TAS US399**

System Design Document



Department of Veterans Affairs

**July 2017**

Version 1.01

**User Story Number:** USRX-45

**User Story Name:** Modify the RED Resubmit w/Edits option to allow a user to submit fields not on the payer sheet

**Product Backlog ID:**

**Backlog Priority:**

**Initial Sizing Estimate:**

**Rational ID:**

**Rally ID:** US399

# Design – Summary

A claim request sent from VistA to a payer currently includes only those fields indicated by the payer sheet provided by the payer. This story calls for giving the user the ability to add to the claim additional fields not on the payer sheet.

When the user performs the RED / Resubmit with Edits action, the user can enter values for several fields, the last being Delay Reason Code. Following that prompt, the system will now allow the user to select one or more fields to be added to the claim which are not already on the payer sheet and which are not “stubs” (fields that have no logic for pulling values). The system will display the values to be sent on the claim, and the user will be able to confirm whether or not to include that field. The list of fields added to the claim will be stored in a new sub-file on the existing file BPS NCPDP OVERRIDE. The values to be sent will not be stored at that time.

During the claim submission process, the creation of a claim entry in BPS CLAIMS is done first and then the claim request to be sent to the payer is created.

When the claim entry in BPS CLAIMS is built, the system loops through each field on each segment on the payer sheet. At the conclusion of looping through all the fields on a given segment, the system will check if there are any additional fields added by the user for that segment (via a cross-reference of fields by segment in the BPS NCPDP OVERRIDE file). If there are additional fields to be added, those fields will be added to that entry in BPS CLAIMS. The data for those fields will be pulled at that time.

When the claim request is built, again the system loops through each field on each segment on the payer sheet. The data for each field is pulled from BPS CLAIMS. At the conclusion of looping through all the fields on a given segment, the system will check if there are any additional fields added by the user for that segment. If there are additional fields to be added, those fields will be added to the claim request.

List of Components:

* Routine: BPSECX0
* Routine: BPSOSCF
* Routine: BPSOSH2
* Routine: BPSRES
* Routine: BPSRES1
* Routine: PSOBPSU1
* Routine: PSOBPSU2
* File: BPS NCPDP FIELD DEFS
* File: BPS NCPDP OVERRIDE
* File: BPS NCPDP SEGMENTS

# Design – Detail

When the user performs the RED action, a message is added to the Activity Log. That message will be enhanced to a) indicate specifically that the user performed the RED action, and b) indicate the Date of Service used on the resubmitted claim if the user opted to use a date different than the Release Date.

| Subroutine Name | **DOSELCTD^BPSRES** | | | |
| --- | --- | --- | --- | --- |
| Enhancement Category | New | Modify | Delete | No Change |
| Current Logic | | | | |
| DOSELCTD(BPRXI) ;  N BP02,BP59,BPBILL,BPCLTOT,BPDFN,BPDOSDT,BPOVRIEN,BPQ,BPRXIEN,BPRXR,BPSTATUS,BPUPDFLG  N BPCOB,BPSURE,BPPTRES,BPPHSRV,BPDLYRS,COBDATA,BPPRIOPN,BPSPCLS  S (BPQ)=""  S (BPCLTOT,BPUPDFLG)=0  ;  ...  ;Display selected claim and ask to submit  S BPDFN=+$P($G(^BPST(BP59,0)),U,6)  W !,"You've chosen to RESUBMIT the following prescription for "\_$E($$PATNAME^BPSSCRU2(BPDFN),1,13)  W !,@VALMAR@(+$P(BPRXI,U,5),0)  S BPQ=$$YESNO^BPSSCRRS("Are you sure?(Y/N)")  I BPQ'=1 S BPQ="^" G XRES  ;  ...  ;Prompt for EDIT Information  S BPOVRIEN=$$PROMPTS(BP59,BP02,BPRXIEN,BPRXR,BPCOB,.BPDOSDT,.COBDATA) I BPOVRIEN=-1 G XRES  ;  ; Submit the claim  S BPBILL=$$EN^BPSNCPDP(BPRXIEN,BPRXR,BPDOSDT,"ERES","","ECME RESUBMIT","",BPOVRIEN,"","",BPCOB,"F","","",$G(COBDATA("PLAN")  ),.COBDATA,$G(COBDATA("RTYPE")))  ;  ;Print Return Value Message  W !!  W:+BPBILL>0 $S(+BPBILL=10:"Reversal but no Resubmit:",1:"Not Processed:"),!," "  ;  ;Change Return Message for SC/EI  S:$P(BPBILL,U,2)="NEEDS SC DETERMINATION" $P(BPBILL,U,2)="NEEDS SC/EI DETERMINATION"  W $P(BPBILL,U,2)  ;  ;0 Prescription/Fill successfully submitted to ECME  ;1 ECME did not submit prescription/fill  ;2 IB says prescription/fill is not ECME billable or the data returned from IB is not valid  ;3 ECME closed the claim but did not submit it to the payer  ;4 Unable to queue the ECME claim  ;5 Invalid input  ;10 Reversal Processed But Claim Was Not Resubmitted  ;  I +BPBILL=0 D  . D ECMEACT^PSOBPSU1(+BPRXIEN,+BPRXR,"Claim resubmitted to 3rd party payer: ECME USER's SCREEN-"\_$S(BPCOB=1:"p",BPCOB=2:"s"  ,1:"")\_$$INSNAME^BPSSCRU6(BP59))  . S BPUPDFLG=1,BPCLTOT=1  XRES I BPCLTOT W !,BPCLTOT," claim",$S(BPCLTOT'=1:"s have",1:" has")," been resubmitted.",!  D PAUSE^VALM1  Q BPUPDFLG  ;  XRES2 I BPCLTOT W !,BPCLTOT," claim",$S(BPCLTOT'=1:"s have",1:" has")," been resubmitted.",!  Q BPUPDFLG | | | | |
| Modified Logic (Changes are highlighted) | | | | |
| DOSELCTD(BPRXI) ;  N BP02,BP59,BPADDLTXT,BPBILL,BPCLTOT,BPDFN,BPDOSDT,BPOVRIEN,BPQ,BPRXIEN,BPRXR,BPSTATUS,BPUPDFLG  N BPCOB,BPSURE,BPPTRES,BPPHSRV,BPDLYRS,COBDATA,BPPRIOPN,BPSPCLS,BPMSG  S BPQ=""  S BPADDLTXT=""  S (BPCLTOT,BPUPDFLG)=0  ;  ...  ;Display selected claim and ask to submit  S BPDFN=+$P($G(^BPST(BP59,0)),U,6)  W !,"You've chosen to RESUBMIT the following prescription for "\_$E($$PATNAME^BPSSCRU2(BPDFN),1,13)  W !,@VALMAR@(+$P(BPRXI,U,5),0)  S BPQ=$$YESNO^BPSSCRRS("Are you sure~~?~~(Y/N)")  I BPQ'=1 S BPQ="^" G XRES  ;  ...  ;Prompt for EDIT Information  S BPOVRIEN=$$PROMPTS(BP59,BP02,BPRXIEN,BPRXR,BPCOB,.BPDOSDT,.COBDATA,.BPADDLTXT) I BPOVRIEN=-1 G XRES  ;  ; Submit the claim  S BPBILL=$$EN^BPSNCPDP(BPRXIEN,BPRXR,BPDOSDT,"ERES","","ECME RESUBMIT","",BPOVRIEN,"","",BPCOB,"F","","",$G(COBDATA("PLAN")  ),.COBDATA,$G(COBDATA("RTYPE")))  ;  ;Print Return Value Message  W !!  W:+BPBILL>0 $S(+BPBILL=10:"Reversal but no Resubmit:",1:"Not Processed:"),!," "  ;  ;Change Return Message for SC/EI  S:$P(BPBILL,U,2)="NEEDS SC DETERMINATION" $P(BPBILL,U,2)="NEEDS SC/EI DETERMINATION"  W $P(BPBILL,U,2)  ;  ;0 Prescription/Fill successfully submitted to ECME  ;1 ECME did not submit prescription/fill  ;2 IB says prescription/fill is not ECME billable or the data returned from IB is not valid  ;3 ECME closed the claim but did not submit it to the payer  ;4 Unable to queue the ECME claim  ;5 Invalid input  ;10 Reversal Processed But Claim Was Not Resubmitted  ;  I +BPBILL=0 D  . S BPMSG="ECME RED Resubmit Claim w/Edits"  . I BPADDLTXT'="" S BPMSG=BPMSG\_": "\_BPADDLTXT  . S BPMSG=BPMSG\_"-"\_$S(BPCOB=1:"p",BPCOB=2:"s",1:"")\_$$INSNAME^BPSSCRU6(BP59)  . S BPMSG=$E(BPMSG,1,100)  . D ECMEACT^PSOBPSU1(+BPRXIEN,+BPRXR,BPMSG)  . S BPUPDFLG=1,BPCLTOT=1  ;  XRES ;  I BPCLTOT W !,BPCLTOT," claim",$S(BPCLTOT'=1:"s have",1:" has")," been resubmitted.",!  D PAUSE^VALM1  Q BPUPDFLG  ;  XRES2 ;  I BPCLTOT W !,BPCLTOT," claim",$S(BPCLTOT'=1:"s have",1:" has")," been resubmitted.",!  Q BPUPDFLG | | | | |

The subroutine PROMPTS^BPSRES is part of the RED action. It prompts the user for the current override fields. This subroutine will be modified to call a new subroutine, ADDLFLDS^BPSRES, to allow the user to add fields to the claim. The existing logic for saving the override fields will be moved to a new subroutine in BPSRES due to space considerations, and the logic for saving the additional fields will be added to that new subroutine.

| Subroutine Name | **PROMPTS^BPSRES** | | | |
| --- | --- | --- | --- | --- |
| Enhancement Category | New | Modify | Delete | No Change |
| Current Logic | | | | |
| BPSRES ;BHAM ISC/BEE - ECME SCREEN RESUBMIT W/EDITS ;3/12/08 14:01  ;;1.0;E CLAIMS MGMT ENGINE;\*\*3,5,7,8,10,11,20,21\*\*;JUN 2004;Build 27  ...  PROMPTS(BP59,BP02,BPRXIEN,BPRXR,BPCOB,BPSDOSDT,BPSECOND) ;  N %,BP300,BP35401,BPCLCD1,BPCLCD2,BPCLCD3,BPFDA,BPFLD,BPOVRIEN,BPMED,BPMSG,BPPSNCD  N BPPREAUT,BPPRETYP,BPQ,BPRELCD,DIC,DIR,DIROUT,DTOUT,DUOUT,X,Y,DIRUT,DUP  N BPCLCDN,BPCLCDX,BPSX  ...  P1 ;  ;  I $$RELDATE^BPSBCKJ(+BPRXIEN,+BPRXR)]"" S BPDOSDT=$$EDITDT(1,BPRXIEN,BPRXR,BP02) I BPDOSDT="^" S BPQ=-1 G XPROMPTS  ...  ;Delay Reason Code  N X,DIC,Y  S DIC("B")=BPDLYRS  S DIC(0)="QEAM",DIC=9002313.29,DIC("A")="Delay Reason Code: "  D ^DIC  ;Check for "^" or timeout  I ($D(DUOUT))!($D(DTOUT)) S BPQ=-1 K X,DIC,Y G XPROMPTS  S BPDLYRS=$P(Y,U,2)  K X,DIC,Y  ;  ; If secondary claim, setup secondary data and allow user to edit  ; Get data from the primary claim, if it exists  I BPCOB=2 D I BPQ=-1 G XPROMPTS  . N BPSPL59,BPRTTP59  . S BPRET=$$PRIMDATA^BPSPRRX6(BPRXIEN,BPRXR,.BPSECOND)  . ; If the primary claim data is missing, get data from the most recent secondary claim  . I 'BPRET,$$SECDATA^BPSPRRX6(BPRXIEN,BPRXR,.BPSPL59,.BPSECOND,.BPRTTP59)  . ; The PRIMARY BILL element is set by $$SECDATA. If SECDATA is not called, this element will be  . ; missing and we will need to create it  . I '$D(BPSECOND("PRIMARY BILL")) D  .. N BPBILL  .. S BPBILL=$$PAYBLPRI^BPSUTIL2(BP59)  .. I BPBILL>0 S BPSECOND("PRIMARY BILL")=BPBILL  . ; Set flag telling BPSNCPDP not to recompile the data from the BPS Transaction and the secondary claim  . S BPSECOND("NEW COB DATA")=1  . ; $$PROMPTS displays the data and allows the user edit the data.  . S BPQ=$$PROMPTS^BPSPRRX3(BPRXIEN,BPRXR,BPDOSDT,.BPSECOND)  ;  ;Ask to proceed  W ! S BPQ=$$YESNO^BPSSCRRS("Are you sure?(Y/N)") I BPQ'=1 S BPQ=-1 G XPROMPTS  S BPQ=1  ;  ;Save into BPS NCPDP OVERRIDES (#9002313.511)  S BPFDA(9002313.511,"+1,",.01)=BP59  D NOW^%DTC  S BPFDA(9002313.511,"+1,",.02)=%  S BPFLD=$O(^BPSF(9002313.91,"B",303,"")) I BPFLD]"" S BPFDA(9002313.5111,"+2,+1,",.01)=BPFLD,BPFDA(9002313.5111,"+2,+1,",.02)=BPPSNCD  S BPFLD=$O(^BPSF(9002313.91,"B",306,"")) I BPFLD]"" S BPFDA(9002313.5111,"+3,+1,",.01)=BPFLD,BPFDA(9002313.5111,"+3,+1,",.02)=BPRELCD  S BPFLD=$O(^BPSF(9002313.91,"B",462,"")) I BPFLD]"" S BPFDA(9002313.5111,"+4,+1,",.01)=BPFLD,BPFDA(9002313.5111,"+4,+1,",.02)=BPPREAUT  S BPFLD=$O(^BPSF(9002313.91,"B",461,"")) I BPFLD]"" S BPFDA(9002313.5111,"+5,+1,",.01)=BPFLD,BPFDA(9002313.5111,"+5,+1,",.02)=BPPRETYP  S BPFLD=$O(^BPSF(9002313.91,"B",420,"")) I BPFLD]"" S BPFDA(9002313.5111,"+6,+1,",.01)=BPFLD,BPFDA(9002313.5111,"+6,+1,",.02)=BPCLCD1\_"~"\_$G(BPCLCD2)\_"~"\_$G(BPCLCD3)  S BPFLD=$O(^BPSF(9002313.91,"B",384,"")) I BPFLD]"" S BPFDA(9002313.5111,"+7,+1,",.01)=BPFLD,BPFDA(9002313.5111,"+7,+1,",.02)=BPPTRES  S BPFLD=$O(^BPSF(9002313.91,"B",147,"")) I BPFLD]"" S BPFDA(9002313.5111,"+8,+1,",.01)=BPFLD,BPFDA(9002313.5111,"+8,+1,",.02)=BPPHSRV  S BPFLD=$O(^BPSF(9002313.91,"B",357,"")) I BPFLD]"" S BPFDA(9002313.5111,"+9,+1,",.01)=BPFLD,BPFDA(9002313.5111,"+9,+1,",.02)=BPDLYRS  D UPDATE^DIE("","BPFDA","BPOVRIEN","BPMSG")  ;  I $D(BPMSG("DIERR")) W !!,"Could Not Save Override information into BPS NCPDP OVERRIDES FILES",! S BPQ=-1 G XPROMPTS  ;  XPROMPTS ;  S BPOVRIEN=$S(BPQ=-1:BPQ,$G(BPOVRIEN(1))]"":BPOVRIEN(1),1:-1)  Q BPOVRIEN | | | | |
| Modified Logic (Changes are highlighted) | | | | |
| BPSRES ;BHAM ISC/BEE - ECME SCREEN RESUBMIT W/EDITS ;3/12/08 14:01  ;;1.0;E CLAIMS MGMT ENGINE;\*\*3,5,7,8,10,11,20,21,23\*\*;JUN 2004;Build 27  ...  PROMPTS(BP59,BP02,BPRXIEN,BPRXR,BPCOB,BPDOSDT,BPSECOND,BPADDLTXT) ;  N %,BP300,BP35401,BPCLCD1,BPCLCD2,BPCLCD3,BPFDA,BPFLD,BPOVRIEN,BPMED~~,BPMSG~~,BPPSNCD  N BPPREAUT,BPPRETYP,BPQ,BPRELCD,BPRELEASEDT,DIC,DIR,DIROUT,DTOUT,DUOUT,X,Y,DIRUT,DUP  N BPCLCDN,BPCLCDX,BPSX,BPSADDLFLDS  ...  P1 ;  ;  ; If the user opts to use the Date of Service instead of the  ; Release Date, then set BPADDLTXT, which will be used when creating  ; an entry in the Activity Log.  ;  S BPADDLTXT=""  S BPRELEASEDT=$$RELDATE^BPSBCKJ(+BPRXIEN,+BPRXR)  I BPRELEASEDT]"" D I BPQ=-1 G XPROMPTS  . S BPDOSDT=$$EDITDT(1,BPRXIEN,BPRXR,BP02)  . I BPDOSDT="^" S BPQ=-1 Q  . I BPDOSDT'=(BPRELEASEDT\1) S BPADDLTXT="Date of Service ("\_$$FMTE^XLFDT(BPDOSDT,5)\_")"  . Q  ...  ;Delay Reason Code  N X,DIC,Y  S DIC("B")=BPDLYRS  S DIC(0)="QEAM",DIC=9002313.29,DIC("A")="Delay Reason Code: "  D ^DIC  ;Check for "^" or timeout  I ($D(DUOUT))!($D(DTOUT)) S BPQ=-1 K X,DIC,Y G XPROMPTS  S BPDLYRS=$P(Y,U,2)  K X,DIC,Y  ;  ; If secondary claim, setup secondary data and allow user to edit  ; Get data from the primary claim, if it exists  I BPCOB=2 D I BPQ=-1 G XPROMPTS  . N BPSPL59,BPRTTP59  . S BPRET=$$PRIMDATA^BPSPRRX6(BPRXIEN,BPRXR,.BPSECOND)  . ; If the primary claim data is missing, get data from the most recent secondary claim  . I 'BPRET,$$SECDATA^BPSPRRX6(BPRXIEN,BPRXR,.BPSPL59,.BPSECOND,.BPRTTP59)  . ; The PRIMARY BILL element is set by $$SECDATA. If SECDATA is not called, this element will be  . ; missing and we will need to create it  . I '$D(BPSECOND("PRIMARY BILL")) D  .. N BPBILL  .. S BPBILL=$$PAYBLPRI^BPSUTIL2(BP59)  .. I BPBILL>0 S BPSECOND("PRIMARY BILL")=BPBILL  . ; Set flag telling BPSNCPDP not to recompile the data from the BPS Transaction and the secondary claim  . S BPSECOND("NEW COB DATA")=1  . ; $$PROMPTS displays the data and allows the user edit the data.  . S BPQ=$$PROMPTS^BPSPRRX3(BPRXIEN,BPRXR,BPDOSDT,.BPSECOND)  ;  ; Allow user to add to the claim additional fields which are  ; not on the payer sheet. $$ADDLFLDS will return 0 if no  ; additional fields were selected or -1 if the user exited out.  ;  S BPQ=$$ADDLFLDS^BPSRES1(BP02,BP59,.BPSADDLFLDS)  I BPQ=-1 G XPROMPTS  ;  ;Ask to proceed  W ! S BPQ=$$YESNO^BPSSCRRS("Are you sure~~?~~(Y/N)")  I BPQ'=1 S BPQ=-1 G XPROMPTS  S BPQ=1  ;  ; Save the override values and the list of additional fields  ; in file# 9002313.511, BPS NCPDP OVERRIDES.  ;  I '$$SAVE^BPSRES1("RED",BP59,.BPSADDLFLDS) S BPQ=-1  ;  XPROMPTS ;  S BPOVRIEN=$S(BPQ=-1:BPQ,$G(BPOVRIEN(1))]"":BPOVRIEN(1),1:-1)  Q BPOVRIEN | | | | |

The new subroutine ADDLFLDS^BPSRES1 will be created to allow the user to enter one or more additional fields to be added to the claim request. The subroutine ARRAYS will also be added, as described below.

| Subroutine Name | **ADDLFLDS^BPSRES1 and ARRAYS^BPSRES1** | | | |
| --- | --- | --- | --- | --- |
| Enhancement Category | New | Modify | Delete | No Change |
| Current Logic | | | | |
| n/a | | | | |
| Modified Logic (Changes are highlighted) | | | | |
| BPSRES1 ;AITC MRD - ECME SCREEN RESUBMIT W/EDITS ;10/23/17  ;;1.0;E CLAIMS MGMT ENGINE;\*\*23\*\*;JUN 2004;  ;;Per VA Directive 6402, this routine should not be modified.  ;  ADDLFLDS(BPS02,BPS59,BPSADDLFLDS) ; Add fields to a claim.  ; This function allows the user to add to claim fields not on payer  ; sheet. It is used by the RED/Resubmit with Edits Action on the  ; ECME User Screen and by the PRO Option (Process Secondary/TRICARE  ; Rx to ECME).  ; Input: BPS02 = Pointer to BPS CLAIMS  ; BPS59 = Pointer to BSP TRANSACTION  ; This function will return:  ; 1 - If user entered additional fields.  ; 0 - If user added no fields.  ; -1 - If user exited out via "^".  ; This function will also set up the array BPSADDFLDS if the user  ; chooses to add any fields to the claim.  ; BPSADDLFLDS(Field IEN) = Value to Send  ;  N BPS,BPSFIELD,BPSGETCODE,BPSPAYER,BPSQ,BPSSEGMENT  N DIC,MEDN,TRANLIST,X,Y  ;  ; Prompt user whether to enter additional fields. Quit with -1 if  ; user entered "^". Quit with 0 if user entered "No".  ;  S BPSQ=$$YESNO^BPSSCRRS("Submit NCPDP Field Not on Payer Sheet (Y/N)","N")  I BPSQ'=1 Q BPSQ  ;  ; Kill array that will contain list of fields to be added.  ;  K BPSADDLFLDS  ;  ; Build an array listing the fields already on the payer sheet and an  ; array listing all segments on the payer sheet. Include all segments,  ; though some may be excluded later.  ;  D ARRAYS(BPS02,.BPSPAYER,.BPSSEGMENT)  I 'BPSPAYER Q 0  ;  ; Build BPS array. While each field in the file BPS NCPDP FIELD DEFS  ; has Get Code (executable M code) for pulling the data value, the way  ; this has been implemented is that first the subroutine $$BPS^BPSOSCB  ; pulls many fields of data, building the BPS array, and then the  ; Get Code for each field puts into BPS("X") a value from the BPS  ; array. Because the system needs to display to the user the value  ; that would be sent with a field being added to the claim, we need to  ; build the entire BPS array, which will be used by the Get Code for  ; any fields selected by the user. TRANLIST is an array listing all  ; BPS Transactions in this batch of claims. However, the VA does not  ; ever batch claims, so there is always only one transaction in that  ; list. BPS(9002313.0201) must be set to 1. It should never return a  ; value other than 0, but if it does, Quit.  ;  S TRANLIST(BPS59)=""  S BPS(9002313.0201)=1  S X=$$BPS^BPSOSCB  I X W !,$P(X,U,2),".",!,"Fields may not be added at this time." Q 0  ;  ; Display help text.  ;  W !!,"Enter a valid NCPDP Field name or number. Enter '??' for"  W !,"a list of possible choices. Fields already on the payer sheet"  W !,"are excluded from the list of possible choices. Also excluded"  W !,"are any fields that do not have logic to pull data from VistA"  W !,"(i.e. fields that will always be <blank>)."  ;  A1 ; Prompt user for an NCPDP field to add to the claim.  ;  K DIC  ;  S DIC=9002313.91  S DIC(0)="AEMQ"  S DIC("A")="NCPDP Field Name or Number: "  S DIC("S")="I $$CHECK^BPSRES1(Y,.BPSPAYER,.BPSSEGMENT)"  S DIC("T")=""  ;  W !  D ^DIC  ;  ; When the user just hits <return>, skip down to A2.  ;  I X="" G A2  ;  I Y=-1 Q -1  S BPSFIELD=+Y  ;  ; Disallow a field already added.  ;  I $D(BPSADDLFLDS(BPSFIELD)) W !,?4,"This field has already been added to the claim.",\*7 G A1  ;  ; Display to the user the value to be sent with this field.  ;  S BPS("X")=""  S MEDN=1 ; Required for some GET codes.  S BPSGETCODE=0  F S BPSGETCODE=$O(^BPSF(9002313.91,BPSFIELD,10,BPSGETCODE)) Q:'BPSGETCODE X $G(^BPSF(9002313.91,BPSFIELD,10,BPSGETCODE,0))  W !,?4,"Value to transmit: ",BPS("X")  S BPSQ=$$YESNO^BPSSCRRS("Transmit with claim (Y/N)","Y")  I BPSQ=0 G A1  I BPSQ=-1 K BPSADDLFLDS Q -1  ;  ; Add selected field to array (Y=internal field #).  ;  S BPSADDLFLDS(BPSFIELD)=BPS("X")  ;  G A1  ;  A2 ; User is done selecting fields to add.  ;  ; If user added no fields, Quit with 0.  ;  I '$D(BPSADDLFLDS) Q 0  ;  ; Display to the user the list of fields being added to the claim.  ;  W !!,"Fields entered to transmit:"  S BPSFIELD=""  F S BPSFIELD=$O(BPSADDLFLDS(BPSFIELD)) Q:'BPSFIELD D  . W !,?4,$$GET1^DIQ(9002313.91,BPSFIELD,.01),"-"  . W $$GET1^DIQ(9002313.91,BPSFIELD,.06)," "  . W $$GET1^DIQ(9002313.91,BPSFIELD,.03),": "  . W BPSADDLFLDS(BPSFIELD)  . Q  ;  ; Do not ask "Are you sure?" here -- that will be asked in  ; PROMPTS^BPSRES after Quitting out of this subroutine.  ;  Q 1  ;  ARRAYS(BPS02,BPSPAYER,BPSSEGMENT) ; Build BPSPAYER array and BPSSEGMENT array.  ;  ; Build an array listing the fields already on the payer sheet and  ; an array listing all segments on the payer sheet. Include all  ; segments, though some may be excluded later.  ;  N BPSFIELD,BPSORDER  ;  S BPSPAYER=$$GET1^DIQ(9002313.02,BPS02,.02,"I") ; Payer Sheet.  I 'BPSPAYER Q  F BPSSEGMENT=100:10:300 D  . I '$D(^BPSF(9002313.92,BPSPAYER,BPSSEGMENT)) Q  . S BPSSEGMENT(BPSSEGMENT)=""  . S BPSORDER=0  . F S BPSORDER=$O(^BPSF(9002313.92,BPSPAYER,BPSSEGMENT,BPSORDER)) Q:'BPSORDER D  . . S BPSFIELD=$P($G(^BPSF(9002313.92,BPSPAYER,BPSSEGMENT,BPSORDER,0)),"^",2) ; Field IEN  . . I BPSFIELD'="" S BPSPAYER(BPSFIELD)=""  . . Q  . Q  ;  Q | | | | |

| Subroutine Name | **CHECK^BPSRES1** | | | |
| --- | --- | --- | --- | --- |
| Enhancement Category | New | Modify | Delete | No Change |
| Current Logic | | | | |
| n/a | | | | |
| Modified Logic (Changes are highlighted) | | | | |
| CHECK(BPSY,BPSPAYER,BPSSEGMENT) ; Screen for BPS NCPDP FIELD DEFS lookup. See ADDLFLDS above.  ; This function is called for a given entry in the file BPS  ; NCPDP FIELD DEFS, where +Y will be the IEN. If this function  ; returns a 1, then this entry is a valid choice. If this  ; function returns a 0, then this entry will not be displayed to  ; the user when listing possible choices and this entry will not  ; be a valid choice for the user.  ;  ; Disallow if already on the payer sheet.  ;  I $D(BPSPAYER(+BPSY)) Q 0  ;  ; Disallow if this field is not on a request segment or if this  ; field is on a segment not on the payer sheet.  ;  S BPSSEGMENT=$P($G(^BPSF(9002313.91,+BPSY,5)),"^",4) ; Request Segment.  S BPSSEGMENT=$P($G(^BPSF(9002313.9,+BPSSEGMENT,0)),"^",2)  I BPSSEGMENT="" Q 0  I '$D(BPSSEGMENT(BPSSEGMENT)) Q 0  ;  ; There are many segments the VA does not send, even if that  ; segment is on a payer sheet. Disallow any fields that are  ; on one of those segments.  ;  I ",140,170,200,210,220,230,240,250,260,270,280,290,300,"[(","\_BPSSEGMENT\_",") Q 0  ;  ; Disallow if Get Code is simply Setting BPS("X") to "".  ;  I $G(^BPSF(9002313.91,+BPSY,10,1,0))["S BPS(""X"")=""""" Q 0  ;  Q 1 | | | | |

The subroutine SAVE^BPSRES1 will be added. Formerly the override values were saved at the end of the subroutine PROMPTS^BPSRES. More logic will be added to that section of code, and due to space considerations in ^BPSRES and out of a desire to segregate the code, that logic will be moved into SAVE^BPSRES1 where the new logic will be added.

| Subroutine Name | **SAVE^BPSRES1** | | | |
| --- | --- | --- | --- | --- |
| Enhancement Category | New | Modify | Delete | No Change |
| Current Logic | | | | |
| n/a | | | | |
| Modified Logic (Changes are highlighted) | | | | |
| SAVE(BPSACTION,BPS59,BPSADDLFLDS) ; Save into BPS NCPDP OVERRIDES (#9002313.511)  ;  ; If the user chooses to add any fields to the claim, each field  ; will be listed as BPSADDLFLDS(Field IEN).  ;  ; Input: BPSACTION = Action selected by user (e.g. RED, PRO)  ; BPS59 = Pointer to BPS TRANSACTIONS  ; BPSADDLFLDS = Passed by reference, array listing the  ; NCPDP fields to be added to the claim.  ; BPSADDLFLDS(NCPDP Field) = ""  ;  ; BPOVRIEN is Newed in DOSELCTD^BPSRES and should not be Newed here.  ;  N BPSFDA,BPSFIELD,BPSMSG,BPSCNT,BPSFIELD  ;  S BPSFDA(9002313.511,"+1,",.01)=BPS59  D NOW^%DTC  S BPSFDA(9002313.511,"+1,",.02)=%  ;  ; Store the fields for which the user was prompted.  ;  S BPSCNT=1  I BPSACTION="RED" D  . S BPSFIELD=$O(^BPSF(9002313.91,"B",303,"")) I BPSFIELD]"" S BPSFDA(9002313.5111,"+2,+1,",.01)=BPSFIELD,BPSFDA(9002313.511  1,"+2,+1,",.02)=BPPSNCD  . S BPSFIELD=$O(^BPSF(9002313.91,"B",306,"")) I BPSFIELD]"" S BPSFDA(9002313.5111,"+3,+1,",.01)=BPSFIELD,BPSFDA(9002313.511  1,"+3,+1,",.02)=BPRELCD  . S BPSFIELD=$O(^BPSF(9002313.91,"B",462,"")) I BPSFIELD]"" S BPSFDA(9002313.5111,"+4,+1,",.01)=BPSFIELD,BPSFDA(9002313.511  1,"+4,+1,",.02)=BPPREAUT  . S BPSFIELD=$O(^BPSF(9002313.91,"B",461,"")) I BPSFIELD]"" S BPSFDA(9002313.5111,"+5,+1,",.01)=BPSFIELD,BPSFDA(9002313.511  1,"+5,+1,",.02)=BPPRETYP  . S BPSFIELD=$O(^BPSF(9002313.91,"B",420,"")) I BPSFIELD]"" S BPSFDA(9002313.5111,"+6,+1,",.01)=BPSFIELD,BPSFDA(9002313.511  1,"+6,+1,",.02)=BPCLCD1\_"~"\_$G(BPCLCD2)\_"~"\_$G(BPCLCD3)  . S BPSFIELD=$O(^BPSF(9002313.91,"B",384,"")) I BPSFIELD]"" S BPSFDA(9002313.5111,"+7,+1,",.01)=BPSFIELD,BPSFDA(9002313.511  1,"+7,+1,",.02)=BPPTRES  . S BPSFIELD=$O(^BPSF(9002313.91,"B",147,"")) I BPSFIELD]"" S BPSFDA(9002313.5111,"+8,+1,",.01)=BPSFIELD,BPSFDA(9002313.511  1,"+8,+1,",.02)=BPPHSRV  . S BPSFIELD=$O(^BPSF(9002313.91,"B",357,"")) I BPSFIELD]"" S BPSFDA(9002313.5111,"+9,+1,",.01)=BPSFIELD,BPSFDA(9002313.511  1,"+9,+1,",.02)=BPDLYRS  . S BPSCNT=9  . Q  ;  ; Store additional NCPDP fields which the user chose to add to the  ; the resubmitted claim.  ;  S BPSFIELD=0  F S BPSFIELD=$O(BPSADDLFLDS(BPSFIELD)) Q:'BPSFIELD D  . S BPSCNT=BPSCNT+1  . S BPSFDA(9002313.5112,"+"\_BPSCNT\_",+1,",.01)=BPSFIELD ; Field#  . S BPSFDA(9002313.5112,"+"\_BPSCNT\_",+1,",.02)=$$GET1^DIQ(9002313.91,BPSFIELD,2,"I") ; Segment#  . Q  ;  D UPDATE^DIE("","BPSFDA","BPOVRIEN","BPSMSG")  ;  I $D(BPSMSG("DIERR")) W !!,"Could Not Save Override information into BPS NCPDP OVERRIDES FILES",! Q -1  ;  Q 1 | | | | |

During the claim submission process, the entry in the file BPS CLAIMS is created before the claim is sent. The BPS CLAIMS entry is created mainly by the subroutine XLOOP^BPSOSCF. This subroutine loops through each field on each segment on the payer sheet, and for each field it pulls the data, formats the data, and stores the data in the appropriate field in BPS CLAIMS.

XLOOP^BPSOSCF will be modified so that, after looping through each field on a segment, it will check for the presence of additional fields to be added to that segment of the claim.

| Subroutine Name | **XLOOP^BPSOSCF** | | | |
| --- | --- | --- | --- | --- |
| Enhancement Category | New | Modify | Delete | No Change |
| Current Logic | | | | |
| BPSOSCF ;BHAM ISC/FCS/DRS/DLF - Low-level format of .02 ;06/01/2004  ;;1.0;E CLAIMS MGMT ENGINE;\*\*1,5,8,10,15,19\*\*;JUN 2004;Build 18  ;;Per VA Directive 6402, this routine should not be modified.  ;  ; 100 (Transaction Header Segment)  ; 110 (Patient Segment)  ; 120 (Insurance Segment)  ; 130 (Claim Segment)  ; 140 (Pharmacy Provider Segment)  ; 150 (Prescriber Segment)  ; 160 (COB/Other Payments Segment)  ; 170 (Worker's Compensation Segment)  ; 180 (DUR/PPS Segment)  ; 190 (Pricing Segment)  ; 200 (Coupon Segment)  ; 210 (Compound Segment)  ; 220 (Prior Authorization Segment)  ; 230 (Clinical Segment)  ; 240 (Additional Documentation Segment)  ; 250 (Facility Segment)  ; 260 (Narrative Segment)  ; 270 (Purchaser Segment)  ; 280 (Service Provider Segment)  ; 290 (Intermediary Segment)  ; 300 (Last Known 4Rx Segment)  ;  ; FORMAT = IEN in BPS NCPDP FORMATS (#9002313.92)  ; NODE = Segment Node  ; MEDN = Transaction multiple in BPS Claims  XLOOP(FORMAT,NODE,MEDN) ; format claim record  ;  Q:$G(FORMAT)="" Q:$G(NODE)="" ; FORMAT, NODE required  ;  N FLAG,FLDIEN,FLDINFO,MDATA,NCPVERS,ORDER,OVERRIDE,PMODE,RECMIEN,X  ; quit If the payer sheet doesn't have the segment  I '$D(^BPSF(9002313.92,FORMAT,NODE,0)) Q  ;  ; VA doesn't do these segments  I ",300,290,280,270,260,250,240,230,220,210,200,170,140,"[(","\_NODE\_",") Q  ;  ; Per NCPDP standard, eligibility doesn't support segments listed below  I BPS("Transaction Code")=  "E1",",290,280,270,260,250,230,220,210,200,190,180,170,160,130,"[(","\_NODE\_",") Q  ;  ; For COB, if the payer sequence is primary, then quit and don't output the COB fields  I NODE=160,$$COB59^BPSUTIL2(+$G(BPS("RX",BPS(9002313.0201),"IEN59")))=1 Q  ;  ; COB processing is handled differently  I NODE=160 D COB^BPSOSHF(FORMAT,NODE,MEDN) Q  ;  ; DUR is handled differently since it is repeating  I NODE=180 D DURPPS^BPSOSHF(FORMAT,NODE,MEDN) Q  ;  ; Loop through the fields in the segment  S ORDER=0  F S ORDER=$O(^BPSF(9002313.92,FORMAT,NODE,"B",ORDER)) Q:'ORDER D  . ; Get the pointer to the BPS NCPDP FIELD DEFS table  . S RECMIEN=$O(^BPSF(9002313.92,FORMAT,NODE,"B",ORDER,0))  . I 'RECMIEN D IMPOSS^BPSOSUE("DB","TI","NODE="\_NODE,"ORDER="\_ORDER,2,$T(+0)) Q  . S MDATA=^BPSF(9002313.92,FORMAT,NODE,RECMIEN,0),FLDIEN=$P(MDATA,U,2)  .; Corrupt or erroneous format file  . I 'FLDIEN Q  . S FLDINFO=$G(^BPSF(9002313.91,FLDIEN,0)) ; BPS NCPDP FIELD DEFS (#9002313.91)  . I FLDINFO="" D IMPOSS^BPSOSUE("DB,P","TI","FLDIEN="\_FLDIEN,,"XLOOP",$T(+0)) Q  .; Quit for 111-AM Segment Identification  .; 478-H7 Other Amount Claimed Submitted Count  .; 479-H8 Other Amount Claimed Submitted Qualifier  .; 478 and 479 are handled by 480 and 111 is standard field for each segment  . S X=$P(FLDINFO,U) I ",111,478,479,"[(","\_X\_",") Q  .;  .; Set override value (may not be defined so override will be null)  . I $D(MEDN) S OVERRIDE=$G(BPS("OVERRIDE","RX",MEDN,FLDIEN))  . E S OVERRIDE=$G(BPS("OVERRIDE",FLDIEN))  .;  .; Get processing mode (S-Standard (default), X-Special Code)  . S PMODE=$P(MDATA,U,3)  . I PMODE="" S PMODE="S" ;default it  .;  .; Default FLAG and value being computed  . S FLAG="GFS"  . S BPS("X")=""  . ;  . ; If there is an override, set BPS("X") to it and  . ; only do Format and Set code  . I OVERRIDE]"" S FLAG="FS",BPS("X")=OVERRIDE  . ;  . ; If Special Code mode is set, execute special code instead  . ; of field's Get code and change Flag to FS so Format and  . ; Set code is still done but not GET code  . I PMODE="X",OVERRIDE="" D  .. S FLAG="FS"  .. D XSPCCODE(FORMAT,NODE,RECMIEN)  . ;  . ; Call XFLDCODE to do processing based on FLAG setting  . D XFLDCODE(NODE,FLDIEN,FLAG)  ;  Q | | | | |
| Modified Logic (Changes are highlighted) | | | | |
| BPSOSCF ;BHAM ISC/FCS/DRS/DLF - Low-level format of .02 ;06/01/2004  ;;1.0;E CLAIMS MGMT ENGINE;\*\*1,5,8,10,15,19,23\*\*;JUN 2004;Build 18  ;;Per VA Directive 6402, this routine should not be modified.  ;  ; 100 (Transaction Header Segment)  ; 110 (Patient Segment)  ; 120 (Insurance Segment)  ; 130 (Claim Segment)  ; 140 (Pharmacy Provider Segment)  ; 150 (Prescriber Segment)  ; 160 (COB/Other Payments Segment)  ; 170 (Worker's Compensation Segment)  ; 180 (DUR/PPS Segment)  ; 190 (Pricing Segment)  ; 200 (Coupon Segment)  ; 210 (Compound Segment)  ; 220 (Prior Authorization Segment)  ; 230 (Clinical Segment)  ; 240 (Additional Documentation Segment)  ; 250 (Facility Segment)  ; 260 (Narrative Segment)  ; 270 (Purchaser Segment)  ; 280 (Service Provider Segment)  ; 290 (Intermediary Segment)  ; 300 (Last Known 4Rx Segment)  ;  ; FORMAT = IEN in BPS NCPDP FORMATS (#9002313.92)  ; NODE = Segment Node  ; MEDN = Transaction multiple in BPS Claims  XLOOP(FORMAT,NODE,MEDN) ; format claim record  ;  Q:$G(FORMAT)="" Q:$G(NODE)="" ; FORMAT, NODE required  ;  N FLAG,FLDIEN,FLDINFO,IEN511,IEN59,MDATA,NCPVERS,NODEIEN,ORDER,OVERRIDE,PMODE,RECMIEN,BPSX  ; quit If the payer sheet doesn't have the segment  I '$D(^BPSF(9002313.92,FORMAT,NODE,0)) Q  ;  ; VA doesn't do these segments  I ",300,290,280,270,260,250,240,230,220,210,200,170,140,"[(","\_NODE\_",") Q  ;  ; Per NCPDP standard, eligibility doesn't support segments listed below  I BPS("Transaction Code")=  "E1",",290,280,270,260,250,230,220,210,200,190,180,170,160,130,"[(","\_NODE\_",") Q  ;  ; For COB, if the payer sequence is primary, then quit and don't output the COB fields  I NODE=160,$$COB59^BPSUTIL2(+$G(BPS("RX",BPS(9002313.0201),"IEN59")))=1 Q  ;  ; COB processing is handled differently  I NODE=160 D COB^BPSOSHF(FORMAT,NODE,MEDN) Q  ;  ; DUR is handled differently since it is repeating  I NODE=180 D DURPPS^BPSOSHF(FORMAT,NODE,MEDN) Q  ;  ; Loop through the fields in the segment  S ORDER=0  F S ORDER=$O(^BPSF(9002313.92,FORMAT,NODE,"B",ORDER)) Q:'ORDER D  . ; Get the pointer to the BPS NCPDP FIELD DEFS table  . S RECMIEN=$O(^BPSF(9002313.92,FORMAT,NODE,"B",ORDER,0))  . I 'RECMIEN D IMPOSS^BPSOSUE("DB","TI","NODE="\_NODE,"ORDER="\_ORDER,2,$T(+0)) Q  . S MDATA=^BPSF(9002313.92,FORMAT,NODE,RECMIEN,0),FLDIEN=$P(MDATA,U,2)  .; Corrupt or erroneous format file  . I 'FLDIEN Q  . S FLDINFO=$G(^BPSF(9002313.91,FLDIEN,0)) ; BPS NCPDP FIELD DEFS (#9002313.91)  . I FLDINFO="" D IMPOSS^BPSOSUE("DB,P","TI","FLDIEN="\_FLDIEN,,"XLOOP",$T(+0)) Q  .; Quit for 111-AM Segment Identification  .; 478-H7 Other Amount Claimed Submitted Count  .; 479-H8 Other Amount Claimed Submitted Qualifier  .; 478 and 479 are handled by 480 and 111 is standard field for each segment  . S BPSX=$P(FLDINFO,U) I ",111,478,479,"[(","\_BPSX\_",") Q  .;  .; Set override value (may not be defined so override will be null)  . I $D(MEDN) S OVERRIDE=$G(BPS("OVERRIDE","RX",MEDN,FLDIEN))  . E S OVERRIDE=$G(BPS("OVERRIDE",FLDIEN))  .;  .; Get processing mode (S-Standard (default), X-Special Code)  . S PMODE=$P(MDATA,U,3)  . I PMODE="" S PMODE="S" ;default it  .;  .; Default FLAG and value being computed  . S FLAG="GFS"  . S BPS("X")=""  . ;  . ; If there is an override, set BPS("X") to it and  . ; only do Format and Set code  . I OVERRIDE]"" S FLAG="FS",BPS("X")=OVERRIDE  . ;  . ; If Special Code mode is set, execute special code instead  . ; of field's Get code and change Flag to FS so Format and  . ; Set code is still done but not GET code  . I PMODE="X",OVERRIDE="" D  .. S FLAG="FS"  .. D XSPCCODE(FORMAT,NODE,RECMIEN)  . ;  . ; Call XFLDCODE to do processing based on FLAG setting  . D XFLDCODE(NODE,FLDIEN,FLAG)  ;  ; The user has the ability, via the action RED / Resubmit with  ; Edits, to add to the claim fields not on the payer sheet.  ; Any fields to be added to the claim are stored in the file  ; BPS NCPDP OVERRIDE.  ;  ; Determine the transaction from the claim. Determine  ; override, and Quit if none. Field 1.13 is NCPDP OVERRIDES,  ; ptr to 9002313.511.  ;  S IEN59=$$CLAIM59^BPSUTIL2(BPS(9002313.02))  S IEN511=$$GET1^DIQ(9002313.59,IEN59,1.13,"I")  I 'IEN511 Q  ;  ; Loop through additional fields for the current segment (NODE) and  ; call XFLDCODE to add to the claim.  ;  S NODEIEN=$O(^BPSF(9002313.9,"C",NODE,""))  I 'NODEIEN Q  S BPSX=""  F S BPSX=$O(^BPS(9002313.511,IEN511,2,"SEG",NODEIEN,BPSX)) Q:BPSX="" D  . S FLDIEN=$$GET1^DIQ(9002313.5112,BPSX\_","\_IEN511\_",",.01,"I")  . S BPS("X")=""  . D XFLDCODE(NODE,FLDIEN,"GFS")  ;  Q | | | | |

After the entry in BPS CLAIMS is created, the system uses the data in that entry to build the claim to be sent to the payer. Before looping through the segments and fields on the payer sheet, the system copies most of the fields from the BPS CLAIMS record into a local array, BPS. The code which does this, ^BPSECX0, does not pull all of the more recently added NCPDP fields. The subroutine GETBPS3^BPSECX0 will be modified to pull fields not currently pulled.

| Subroutine Name | **GETBPS3^BPSECX0** | | | |
| --- | --- | --- | --- | --- |
| Enhancement Category | New | Modify | Delete | No Change |
| Current Logic | | | | |
| BPSECX0 ;BHAM ISC/FCS/DRS/VA/DLF - Retrieve Claim submission record ;05/17/2004  ;;1.0;E CLAIMS MGMT ENGINE;\*\*1,5,8,10,15,19\*\*;JUN 2004;Build 18  ...  ;Retrieve data in TRANSACTIONS multiple in BPS CLAIMS  ; CLAIMIEN = ien in BPS CLAIMS (#9002313.02)  ; TRXIEN = ien in TRANSACTIONS (#9002313.0201)  ; BPS - Passed by reference  ; returns: BPS(9002313.0201,TRXIEN,field #,"I") = internal format value  GETBPS3(CLAIMIEN,TRXIEN,BPS) ;called from BPSECA1  ;  Q:$G(CLAIMIEN)="" Q:$G(TRXIEN)="" ; must have both  ;  N D0,DA,DIC,DIQ,DIQ2,DR  ;  ; There are other alphanumeric fields that could be added but since they are for segments that are not  ; supported by E1, B1, B3 transactions and/or not segments not used by VA. These can be added later, if  ; needed for those segments.  S DIC=9002313.02,DR="400",DR(9002313.0201)="113:996" ; all TRANSACTION fields  S DR(9002313.0201)=DR(9002313.0201)\_";1023:1032" ;Get alphanumeric NCPDP fields 1023 through 1032 - BPS\*1\*15  S DR(9002313.0201)=DR(9002313.0201)\_";2026:2028;2040:2042" ;Get alphanumeric NCPDP fields 2026-2028 and 2040-2042 - BPS\*1\*19  S DA=CLAIMIEN,DA(9002313.0201)=TRXIEN ; IEN and sub-file IEN  S DIQ="BPS",DIQ(0)="I" ; "I" for internal format  D EN^DIQ1  ;  ; Copy Prescriber Phone Number (498.12) to field 498 as this is where BPSOSH2  ; expects to find it. This works for now but if we implement the Prior Auth  ; segment (which has multiple field labelled 498), a more complete solution  ; will need to be found  S BPS(9002313.0201,TRXIEN,498,"I")=$G(BPS(9002313.0201,TRXIEN,498.12,"I"))  Q  ...  S DR(9002313.0401)=".01;338;339;340;341;443;471;353;392" ; fields | | | | |
| Modified Logic (Changes are highlighted) | | | | |
| BPSECX0 ;BHAM ISC/FCS/DRS/VA/DLF - Retrieve Claim submission record ;05/17/2004  ;;1.0;E CLAIMS MGMT ENGINE;\*\*1,5,8,10,15,19,23\*\*;JUN 2004;Build 18  ...  ;Retrieve data in TRANSACTIONS multiple in BPS CLAIMS  ; CLAIMIEN = ien in BPS CLAIMS (#9002313.02)  ; TRXIEN = ien in TRANSACTIONS (#9002313.0201)  ; BPS - Passed by reference  ; returns: BPS(9002313.0201,TRXIEN,field #,"I") = internal format value  GETBPS3(CLAIMIEN,TRXIEN,BPS) ;called from BPSECA1  ;  Q:$G(CLAIMIEN)="" Q:$G(TRXIEN)="" ; must have both  ;  N D0,DA,DIC,DIQ,DIQ2,DR  ;  ; There are other alphanumeric fields that could be added but since they are for segments that are not  ; supported by E1, B1, B3 transactions and/or not segments not used by VA. These can be added later, if  ; needed for those segments.  S DIC=9002313.02,DR="400",DR(9002313.0201)="113:996" ; all TRANSACTION fields  S DR(9002313.0201)=DR(9002313.0201)\_";1023:1032" ~~;Get alphanumeric NCPDP fields 1023 through 1032 - BPS\*1\*15~~  S DR(9002313.0201)=DR(9002313.0201)\_";2024:2032;2039:2043"  S DR=DR\_“;1093;2013:2021;2034;2035;2037”  S DR=DR\_“;2056:2061;2095:2097;2101;2102”  S DA=CLAIMIEN,DA(9002313.0201)=TRXIEN ; IEN and sub-file IEN  S DIQ="BPS",DIQ(0)="I" ; "I" for internal format  D EN^DIQ1  ;  ; Copy Prescriber Phone Number (498.12) to field 498 as this is where BPSOSH2  ; expects to find it. This works for now but if we implement the Prior Auth  ; segment (which has multiple field labelled 498), a more complete solution  ; will need to be found  S BPS(9002313.0201,TRXIEN,498,"I")=$G(BPS(9002313.0201,TRXIEN,498.12,"I"))  Q  ...  S DR(9002313.0401)=".01;338;339;340;341;443;471;353;392;393" ; fields | | | | |

After the entry in BPS CLAIMS is created, the system uses the data in that entry to build the claim to be sent to the payer. This is largely done in XLOOP^BPSOSH2. This subroutine loops through each field on each segment on the payer sheet, and for each field it pulls the data from BPS CLAIMS and adds it to the segment.

XLOOP^BPSOSCF will be modified so that, after looping through each field on a segment, it will check for the presence of additional fields to be added to that segment of the claim.

| Subroutine Name | **XLOOP^BPSOSH2** | | | |
| --- | --- | --- | --- | --- |
| Enhancement Category | New | Modify | Delete | No Change |
| Current Logic | | | | |
| BPSOSH2 ;BHAM ISC/SD/lwj/DLF - Assemble formatted claim ;06/01/2004  ;;1.0;E CLAIMS MGMT ENGINE;\*\*1,5,8,10,15,19,20\*\*;JUN 2004;Build 27  ...  ; Put together ASCII formatted record per the Payer Sheet Definition  ; Input:  ; NODES - "100^110^120" or "130^140^150^160^170^180^190^200^210^220^230^240^250^260^270^280^290^300"  ; Passed by Ref:  ; .IEN - Internal Entry Number array  ; .BPS - Formatted Data Array with claim and transaction data  ; .REC - Formatted ASCII record (result)  ;  XLOOP(NODES,IEN,BPS,REC) ;EP - from BPSECA1  ;  N DATAFND,FDATA,FLAG,FLDDATA,FLDID,FLDIEN,FLDNUM,INDEX,MDATA,NODE,ORDER,PMODE,RECMIEN,SEGREC  N VER,TYPE  ;  ; Get payer sheet version and transaction type  S VER=$P($G(^BPSF(9002313.92,+$G(IEN(9002313.92)),1)),U,2)  S TYPE=$G(BPS(9002313.02,+$G(IEN(9002313.02)),103,"I"))  ;  ; Loop through the NODES variable delimited by U  F INDEX=1:1:$L(NODES,U) D  . S NODE=$P(NODES,U,INDEX) Q:NODE=""  . ;  . ; VA does not support these segments  . Q:",300,290,280,270,260,250,240,230,220,210,200,170,140,"[NODE  . ;  . ; Quit if the payer sheet does not have the node  . Q:'$D(^BPSF(9002313.92,+IEN(9002313.92),NODE,0))  . ;  . ; Per NCPDP standard, reversals do not support segments listed below  . I TYPE="B2",",300,290,280,270,260,250,240,230,220,210,200,170,150,140,"[NODE Q  . I TYPE="B2",VER="D0",NODE=110 Q ;Patient segment not supported in a D0 reversal  . ;  . ; Per NCPDP standard, eligibility does not support segments listed below  . I TYPE="E1",",290,280,270,260,250,230,220,210,200,190,180,170,160,130,"[NODE Q  . ;  . S DATAFND=0 ; indicates if data is on the segment for us to send  . S SEGREC="" ; segment's information  . ;  . D:NODE=180 PROCDUR  . ;  . ;COB fields  . D:NODE=160 PROCCOB  . ;  . S ORDER=""  . F D Q:'ORDER  .. ;  .. Q:NODE=180 ; DUR/PPS section (repeating), already processed  .. Q:NODE=160 ; COB data processed earlier  .. S ORDER=$O(^BPSF(9002313.92,+IEN(9002313.92),NODE,"B",ORDER))  .. Q:'ORDER  .. S RECMIEN=""  .. S RECMIEN=$O(^BPSF(9002313.92,+IEN(9002313.92),NODE,"B",ORDER,RECMIEN))  .. Q:RECMIEN=""  .. ;  .. S MDATA=$G(^BPSF(9002313.92,+IEN(9002313.92),NODE,RECMIEN,0))  .. Q:MDATA=""  .. ;  .. S FLDIEN=$P(MDATA,U,2)  .. Q:FLDIEN=""  .. ;  .. S FDATA=$G(^BPSF(9002313.91,FLDIEN,0))  .. Q:FDATA=""  .. S FLDNUM=$P(FDATA,U,1)  .. Q:FLDNUM=""  .. ;Check for alphanumeric NCPDP numbers - BPS\*1\*15  .. I $P(FLDNUM,".")'?3N S FLDNUM=$$VNUM^BPSECMPS(FLDNUM) Q:'FLDNUM  .. ;  .. S FLDID=$P($G(^BPSF(9002313.91,FLDIEN,5)),U) ; BPS NCPDP FIELD DEFS, (#.06) ID  .. ;  .. ;header data  .. S:NODE<130 FLDDATA=$G(BPS(9002313.02,IEN(9002313.02),FLDNUM,"I"))  .. ;  .. ;transaction data  .. S:NODE>120 FLDDATA=$G(BPS(9002313.0201,IEN(9002313.0201),FLDNUM,"I"))  .. I $TR($E(FLDDATA,3,999),"0 {}")'="" S DATAFND=1 ;BPS\*1\*15 - allow for zero in NCPDP ID  .. ;  .. ;check if this is the seg id - call this after fld chk since  .. ;we don't want to send the segment if this is all there is  .. I (NODE>100)&(FLDNUM=111) S FLDDATA=$$SEGID(NODE)  .. ;  .. ; Special code to handle the Submission Clarification Code (420) repeating group  .. I FLDNUM=420 D SUBCLAR(.DATAFND,.IEN,.SEGREC) Q  .. ;  .. ; Special code to handle the Other Amount Claimed repeating group  .. I FLDNUM=480 D OAMTCLMD(.DATAFND,.IEN,.SEGREC) Q  .. I FLDNUM=479 Q ; fields 479 & 480 handled as a pair in OAMTCLMD  .. ;  .. Q:FLDDATA="" ;lje;7/23/03; don't want extra field separators when field is blank for testing for WebMD.  .. ;  .. S:NODE=100 SEGREC=SEGREC\_FLDDATA ;no FS on the header rec  .. S:NODE>100 SEGREC=SEGREC\_$C(28)\_FLDDATA ;FS always proceeds fld  ..;  . I (DATAFND)&(NODE=100) S REC(NODE)=SEGREC ;no SS when it's the header  . I (DATAFND)&(NODE>100) D  .. I '$D(REC(NODE)) S REC(NODE)=REC I REC[$C(29) S REC=""  .. S REC(NODE)=REC(NODE)\_$C(30)\_SEGREC ;SS before the seg  ;  Q | | | | |
| Modified Logic (Changes are highlighted) | | | | |
| BPSOSH2 ;BHAM ISC/SD/lwj/DLF - Assemble formatted claim ;06/01/2004  ;;1.0;E CLAIMS MGMT ENGINE;\*\*1,5,8,10,15,19,20,23\*\*;JUN 2004;Build 27  ...  ; Put together ASCII formatted record per the Payer Sheet Definition  ; Input:  ; NODES - "100^110^120" or "130^140^150^160^170^180^190^200^210^220^230^240^250^260^270^280^290^300"  ; Passed by Ref:  ; .IEN - Internal Entry Number array  ; .BPS - Formatted Data Array with claim and transaction data  ; .REC - Formatted ASCII record (result)  ;  XLOOP(NODES,IEN,BPS,REC) ;EP - from BPSECA1  ;  N DATAFND,FDATA,FLAG,FLDDATA,FLDID,FLDIEN,FLDNUM,IEN511,IEN59,INDEX,MDATA,NODE,ORDER,PMODE,RECMIEN,SEGREC  N VER,TYPE,BPSX  ;  ; Get payer sheet version and transaction type  S VER=$P($G(^BPSF(9002313.92,+$G(IEN(9002313.92)),1)),U,2)  S TYPE=$G(BPS(9002313.02,+$G(IEN(9002313.02)),103,"I"))  ;  ; Loop through the NODES variable delimited by U  F INDEX=1:1:$L(NODES,U) D  . S NODE=$P(NODES,U,INDEX) Q:NODE=""  . ;  . ; VA does not support these segments  . Q:",300,290,280,270,260,250,240,230,220,210,200,170,140,"[NODE  . ;  . ; Quit if the payer sheet does not have the node  . Q:'$D(^BPSF(9002313.92,+IEN(9002313.92),NODE,0))  . ;  . ; Per NCPDP standard, reversals do not support segments listed below  . I TYPE="B2",",300,290,280,270,260,250,240,230,220,210,200,170,150,140,"[NODE Q  . I TYPE="B2",VER="D0",NODE=110 Q ;Patient segment not supported in a D0 reversal  . ;  . ; Per NCPDP standard, eligibility does not support segments listed below  . I TYPE="E1",",290,280,270,260,250,230,220,210,200,190,180,170,160,130,"[NODE Q  . ;  . S DATAFND=0 ; indicates if data is on the segment for us to send  . S SEGREC="" ; segment's information  . ;  . D:NODE=180 PROCDUR  . ;  . ;COB fields  . D:NODE=160 PROCCOB  . ;  . S ORDER=""  . F D Q:'ORDER  .. ;  .. Q:NODE=180 ; DUR/PPS section (repeating), already processed  .. Q:NODE=160 ; COB data processed earlier  .. S ORDER=$O(^BPSF(9002313.92,+IEN(9002313.92),NODE,"B",ORDER))  .. Q:'ORDER  .. S RECMIEN=""  .. S RECMIEN=$O(^BPSF(9002313.92,+IEN(9002313.92),NODE,"B",ORDER,RECMIEN))  .. Q:RECMIEN=""  .. ;  .. S MDATA=$G(^BPSF(9002313.92,+IEN(9002313.92),NODE,RECMIEN,0))  .. Q:MDATA=""  .. ;  .. S FLDIEN=$P(MDATA,U,2)  .. Q:FLDIEN=""  .. ;  .. S FDATA=$G(^BPSF(9002313.91,FLDIEN,0))  .. Q:FDATA=""  .. S FLDNUM=$P(FDATA,U,1)  .. Q:FLDNUM=""  .. ;Check for alphanumeric NCPDP numbers - BPS\*1\*15  .. I $P(FLDNUM,".")'?3N S FLDNUM=$$VNUM^BPSECMPS(FLDNUM) Q:'FLDNUM  .. ;  .. S FLDID=$P($G(^BPSF(9002313.91,FLDIEN,5)),U) ; BPS NCPDP FIELD DEFS, (#.06) ID  .. ;  .. ;header data  .. S:NODE<130 FLDDATA=$G(BPS(9002313.02,IEN(9002313.02),FLDNUM,"I"))  .. ;  .. ;transaction data  .. S:NODE>120 FLDDATA=$G(BPS(9002313.0201,IEN(9002313.0201),FLDNUM,"I"))  .. I $TR($E(FLDDATA,3,999),"0 {}")'="" S DATAFND=1 ;BPS\*1\*15 - allow for zero in NCPDP ID  .. ;  .. ;check if this is the seg id - call this after fld chk since  .. ;we don't want to send the segment if this is all there is  .. I (NODE>100)&(FLDNUM=111) S FLDDATA=$$SEGID(NODE)  .. ;  .. ; Special code to handle the Submission Clarification Code (420) repeating group  .. I FLDNUM=420 D SUBCLAR(.DATAFND,.IEN,.SEGREC) Q  .. ;  .. ; Special code to handle the Other Amount Claimed repeating group  .. I FLDNUM=480 D OAMTCLMD(.DATAFND,.IEN,.SEGREC) Q  .. I FLDNUM=479 Q ; fields 479 & 480 handled as a pair in OAMTCLMD  .. ;  .. Q:FLDDATA="" ;lje;7/23/03; don't want extra field separators when field is blank for testing for WebMD.  .. ;  .. S:NODE=100 SEGREC=SEGREC\_FLDDATA ;no FS on the header rec  .. S:NODE>100 SEGREC=SEGREC\_$C(28)\_FLDDATA ;FS always proceeds fld  ..;  . ; If no data on this segment, Quit, don’t check for addl. fields.  . ;  . I 'DATAFND Q  . ;  . ; The user has the ability, via the action RED / Resubmit with  . ; Edits, to add claim fields not on the payer sheet. Any  . ; fields to be added to the claim are stored in the file BPS NCPDP  . ; OVERRIDE. Pull the BPS TRANSACTION from the BPS CLAIMS file,  . ; then pull the field NCPDP OVERRIDES. If populated, pull any  . ; additional fields.  . ;  . S IEN59=$$GET1^DIQ(9002313.02,IEN(9002313.02),.08,"I")  . S IEN511=$$GET1^DIQ(9002313.59,IEN59,1.13,"I")  . I IEN511 D  . . ;  . . ; Loop through additional fields for the current segment  . . ; (NODE) and add to the claim.  . . ;  . . S NODEIEN=$O(^BPSF(9002313.9,"C",NODE,""))  . . I 'NODEIEN Q  . . S BPSX=""  . . F S BPSX=$O(^BPS(9002313.511,IEN511,2,"SEG",NODEIEN,BPSX)) Q:BPSX="" D  . . . S FLDIEN=$$GET1^DIQ(9002313.5112,BPSX\_","\_IEN511\_",",.01,"I")  . . . ;  . . . ; The data in the BPS array is stored according to the number  . . . ; of each field in BPS CLAIMS. That number corresponds to the  . . . ; NCPDP field number when the NCPDP number is all numeric. For  . . . ; alphanumeric field numbers, such as "B95", we must call  . . . ; $$VNUM^BPSECMPS to pull the BPS CLAIMS field number from BPS  . . . ; NCPDP FIELD DEFS.  . . . ;  . . . S FLDNUM=$$GET1^DIQ(9002313.91,FLDIEN,.01,"E")  . . . I FLDNUM="" Q  . . . I $P(FLDNUM,".")'?3N S FLDNUM=$$VNUM^BPSECMPS(FLDNUM) I 'FLDNUM Q  . . . ;  . . . I NODE<130 S FLDDATA=$G(BPS(9002313.02,IEN(9002313.02),FLDNUM,"I"))  . . . I NODE>120 S FLDDATA=$G(BPS(9002313.0201,IEN(9002313.0201),FLDNUM,"I"))  . . . ;  . . . I FLDDATA="" Q  . . . ;  . . . ; $C(28) = File Separator. On all segments except the Header,  . . . ; FS precedes each field.  . . . ;  . . . I NODE=100 S SEGREC=SEGREC\_FLDDATA  . . . I NODE>100 S SEGREC=SEGREC\_$C(28)\_FLDDATA  . . . ;  . . . Q  . . Q  . ;  . I ~~(DATAFND)&(~~NODE=100~~)~~ S REC(NODE)=SEGREC ;no SS when it's the header  . I ~~(DATAFND)&(~~NODE>100~~)~~ D  .. I '$D(REC(NODE)) S REC(NODE)=REC I REC[$C(29) S REC=""  .. S REC(NODE)=REC(NODE)\_$C(30)\_SEGREC ;SS before the seg  ;  Q | | | | |

The length of the comment field in the Activity Log was changed from 75 to 100 (please see PSO\*7.0\*386, from 2012, for more information). The subroutines ECMEACT^PSOBPSU1 and RXACT^PSOBPSU2 should have been updated at that time to reflect this change. They were not. They are being updated as part of this enhancement, along with two other minor changes described below.

| Subroutine Name | **ECMEACT^PSOBPSU1** | | | |
| --- | --- | --- | --- | --- |
| Enhancement Category | New | Modify |  |  |
| Current Logic | | | | |
| ECMEACT(RX,RFL,COMM,USR) ; - Add an Act to the ECME Act Log (FILE 52)  ;Input: (r) RX - Rx IEN (#52)  ; (o) RFL - Refill # (Default: most recent)  ; (r) COMM - Comments (up to 75 characters)  ; (o) USR - User logging the comments (Default: DUZ)  S:'$D(RFL) RFL=$$LSTRFL^PSOBPSU1(RX)  D RXACT^PSOBPSU2(RX,RFL,COMM,"M",+$G(USR))  Q | | | | |
| Modified Logic (Changes are highlighted) | | | | |
| ECMEACT(RX,RFL,COMM,USR) ; - Add an Act to the ECME Act Log (FILE 52)  ;Input: (r) RX - Rx IEN (#52)  ; (o) RFL - Refill # (Default: most recent)  ; (r) COMM - Comments (up to 100 characters)  ; (o) USR - User logging the comments (Default: DUZ)  S:'$D(RFL) RFL=$$LSTRFL^PSOBPSU1(RX)  D RXACT^PSOBPSU2(RX,RFL,COMM,"M",+$G(USR))  Q | | | | |

Add a <space> to make the activity log look better.

| Subroutine Name | **ECMESND^PSOBPSU1** | | | |
| --- | --- | --- | --- | --- |
| Enhancement Category | New | Modify |  |  |
| Current Logic | | | | |
| ...  ;  ; - Creating ECME Act Log in file 52  S ACT="" I $$STATUS^PSOBPSUT(RX,RFL)="E PAYABLE" S ACT="Rev/Resubmit"  S ACT=ACT\_" ECME:"  ;  ... | | | | |
| Modified Logic (Changes are highlighted) | | | | |
| ...  ;  ; - Creating ECME Act Log in file 52  S ACT=""  I $$STATUS^PSOBPSUT(RX,RFL)="E PAYABLE" S ACT="Rev/Resubmit "  S ACT=ACT\_"ECME:"  ;  ... | | | | |

| Subroutine Name | **RXACT^PSOBPSU2** | | | |
| --- | --- | --- | --- | --- |
| Enhancement Category | New | Modify |  |  |
| Current Logic | | | | |
| RXACT(RX,RFL,COMM,TYPE,USR) ; - Add an Activity to the ECME Activity Log (PRESCRIPTION file)  ;Input: (r) RX - Rx IEN (#52)  ; (o) RFL - Refill # (Default: most recent)  ; (r) COMM - Comments (up to 75 characters)  ; (r) TYPE - Comments type: (M-ECME,E-Edit, etc...) See file #52 DD for all values  ; (o) USR - User logging the comments (Default: DUZ)  ;  S:'$D(RFL) RFL=$$LSTRFL^PSOBPSU1(RX) S:'$D(USR) USR=DUZ  S:'$D(^VA(200,+USR,0)) USR=DUZ S COMM=$E($G(COMM),1,75)  ;  I COMM="" Q  I '$D(^PSRX(RX)) Q  ;  N PSOTRIC S PSOTRIC="",PSOTRIC=$$TRIC^PSOREJP1(RX,RFL,PSOTRIC)  I PSOTRIC=1,$E(COMM,1,7)'="TRICARE" S COMM=$E("TRICARE-"\_COMM,1,75)  I PSOTRIC=2,$E(COMM,1,7)'="CHAMPVA" S COMM=$E("CHAMPVA-"\_COMM,1,75)  N X,DIC,DA,DD,DO,DR,DINUM,Y,DLAYGO  S DA(1)=RX,DIC="^PSRX("\_RX\_",""A"",",DLAYGO=52.3,DIC(0)="L"  S DIC("DR")=".02///"\_TYPE\_";.03////"\_USR\_";.04///"\_$S(TYPE'="M"&(RFL>5):RFL+1,1:RFL)\_";.05///"\_COMM  S X=$$NOW^XLFDT() D FILE^DICN  Q | | | | |
| Modified Logic (Changes are highlighted) | | | | |
| RXACT(RX,RFL,COMM,TYPE,USR) ; - Add an Activity to the ECME Activity Log (PRESCRIPTION file)  ;Input: (r) RX - Rx IEN (#52)  ; (o) RFL - Refill # (Default: most recent)  ; (r) COMM - Comments (up to 100 characters)  ; (r) TYPE - Comments type: (M-ECME,E-Edit, etc...) See file #52 DD for all values  ; (o) USR - User logging the comments (Default: DUZ)  ;  S:'$D(RFL) RFL=$$LSTRFL^PSOBPSU1(RX) S:'$D(USR) USR=DUZ  S:'$D(^VA(200,+USR,0)) USR=DUZ S COMM=$E($G(COMM),1,100)  ;  I COMM="" Q  I '$D(^PSRX(RX)) Q  ;  N PSOTRIC S PSOTRIC="",PSOTRIC=$$TRIC^PSOREJP1(RX,RFL,PSOTRIC)  I PSOTRIC=1,$E(COMM,1,7)'="TRICARE" S COMM=$E("TRICARE-"\_COMM,1,100)  I PSOTRIC=2,$E(COMM,1,7)'="CHAMPVA" S COMM=$E("CHAMPVA-"\_COMM,1,100)  N X,DIC,DA,DD,DO,DR,DINUM,Y,DLAYGO  S DA(1)=RX,DIC="^PSRX("\_RX\_",""A"",",DLAYGO=52.3,DIC(0)="L"  S DIC("DR")=".02///"\_TYPE\_";.03////"\_USR\_";.04///"\_$S(TYPE'="M"&(RFL>5):RFL+1,1:RFL)\_";.05///"\_COMM  S X=$$NOW^XLFDT() D FILE^DICN  Q | | | | |

New two additional variables to clean up the results of XINDEX.

| Subroutine Name | **UPDFL^PSOBPSU2** | | | |
| --- | --- | --- | --- | --- |
| Enhancement Category | New | Modify |  |  |
| Current Logic | | | | |
| UPDFL(RXREC,SUB,INDT) ;update fill date with release date when NDC changes at CMOP and OPAI auto-release  ;Input: RXREC = Prescription File IEN  ; SUB = Refill  ; INDT = Release date  N DA,DIE,DR,PSOX,SFN,DEAD,XOK,OLD,X,II,EXDAT,OFILLD,COM,CNT,RFCNT,RF ;\*459 - REMOVED SUB | | | | |
| Modified Logic (Changes are highlighted) | | | | |
| UPDFL(RXREC,SUB,INDT) ;update fill date with release date when NDC changes at CMOP and OPAI auto-release  ;Input: RXREC = Prescription File IEN  ; SUB = Refill  ; INDT = Release date  N DA,DIE,DR,DTOUT,DUOUT,PSOX,SFN,DEAD,XOK,OLD,X,II,EXDAT,OFILLD,COM,CNT,RFCNT,RF ;\*459 - REMOVED SUB | | | | |

The building of each claim request is directed in part by the Payer Sheet, which is stored in the file BPS NCPDP FORMATS. In that file, the fields to be included in the claim request are grouped by segment. Since this story calls for allowing the user to add to a claim request one or more fields that are not on the payer sheet, the system cannot use the payer sheet to determine which segment the additional fields should be a part of.

The system needs to know what segment each field goes on, so a data field (REQUEST SEGMENT) will be added to the file BPS NCPDP FIELD DEFS linking each entry in that file to one segment on the claim request. That field will be a pointer to a new file, BPS NCPDP SEGMENTS, which will list the possible NCPDP segments.

NB: Although this story does not call for any changes to the incoming claim response, the claim response segments will be included in certain changes prescribed in this design.

* The file BPS NCPDP FIELD DEFS lists both outgoing claim request fields and incoming claim response fields. A field, RESPONSE SEGMENT, will be added to this file to indicate which segment each response fields is a part of. Some fields are a part of both a request segment and a response segment.
* The new file BPS NCPDP SEGMENTS will list each segment on the outgoing claim request and the incoming claim response. A field on this file will indicate whether a segment is a part of the request, response, or both.

No changes will be made to the parsing, processing, or filing of the incoming claim response. The above two changes make sense at this time, based on adding segment-based data related to the outgoing claim request.

The file BPS NCPDP SEGMENTS is new and will be created with the following attributes.

|  |  |
| --- | --- |
| File Name | BPS NCPDP SEGMENTS |
| File Number | 9002313.9 |
| Location | ^BPSF(9002313.9, |
| Description | Each transaction described in the NCPDP Telecommunication Standard is comprised of one or more segments. This file contains a list of segments. The file BPS NCPDP FIELD DEFS lists each field that may be on an outgoing claim request or an incoming claim response. Each entry in that file will have a field that points to this file to indicate which segment that field is a part of for the claim request (field REQUEST SEGMENT) and/or the claim response (field RESPONSE SEGMENT). |

The new file BPS NCPDP SEGMENTS, file# 9002313.9, will have the following fields.

|  |  |
| --- | --- |
| **Field Attributes** | **Values** |
| FIELD NAME | NAME |
| FIELD NUMBER | .01 |
| NODE;PIECE | 0;1 |
| FIELD TYPE | Free Text |
| FIELD LENGTH | 50 |
| INPUT TRANSFORM | K:$L(X)>50!($L(X)<5) X |
| HELP PROMPT | Answer must be 5-50 characters in length. |
| DESCRIPTION | This is the name of an NCPDP segment. |
|  |  |
| FIELD NAME | VISTA SEGMENT NUMBER |
| FIELD NUMBER | 1 |
| NODE;PIECE | 0;2 |
| FIELD TYPE | Free Text |
| FIELD LENGTH | 50 |
| INPUT TRANSFORM | K:$L(X)>5!($L(X)<1) X |
| HELP PROMPT | Answer must be 1-5 characters in length |
| DESCRIPTION | This field is the VistA segment number assigned to this segment. |
|  |  |
| FIELD NAME | TYPE |
| FIELD NUMBER | 2 |
| NODE;PIECE | 0;3 |
| FIELD TYPE | Set of Codes |
| FIELD LENGTH | 1 |
| CODES | ‘C’ FOR CLAIM; ‘R’ FOR RESPONSE; ‘B’ FOR BOTH |
| INPUT TRANSFORM | n/a |
| HELP PROMPT | Enter a value to indicate which type(s) of transaction this segment is a part of. |
| DESCRIPTION | Each NCPDP segment is a part of one or more types of transaction. |

A new multiple will be added to file# 9002313.511, BPS NCPDP OVERRIDE. It will list any fields to be added to the claim. It will also have a cross-reference of those fields to index the fields by their segment.

|  |  |
| --- | --- |
| **Field Attributes** | **Values** |
| FIELD NAME | ADDL NCPDP FIELD |
| FIELD NUMBER | 2 |
| SUB-FILE NUMBER | 9002313.5112 |
| FIELD TYPE | Multiple |
| FIELD LENGTH | n/a |
| INPUT TRANSFORM | n/a |
| HELP PROMPT | n/a |
| DESCRIPTION | Additional NCPDP fields to be added to the claim. |

|  |  |
| --- | --- |
| **Field Attributes** | **Values** |
| FIELD NAME | ADDL NCPDP FIELD |
| FIELD NUMBER | .01 |
| NODE;PIECE | 0;1 |
| FIELD TYPE | Pointer to file# 9002313.91, |
| FIELD LENGTH | n/a |
| INPUT TRANSFORM | Q |
| HELP PROMPT | Select the NCPDP field to be added to the claim. |
| DESCRIPTION | This data field is an NCPDP field to be added to the claim. |

|  |  |
| --- | --- |
| **Field Attributes** | **Values** |
| FIELD NAME | SEGMENT |
| FIELD NUMBER | .02 |
| NODE;PIECE | 0;2 |
| FIELD TYPE | Pointer to file 9002313.9, BPS NCPDP SEGMENTS |
| FIELD LENGTH | n/a |
| INPUT TRANSFORM | Q |
| HELP PROMPT | Select the Segment on which this field should appear. |
| DESCRIPTION | This data field indicates the segment of a claim response on which this NCPDP field should appear. |

|  |  |
| --- | --- |
| File | 9002313.511, BPS NCPDP OVERRIDE |
| Sub-File | 9002313.5111, NCPDP FIELD |
| Type of Indexing | REGULAR |
| To Be Used for Lookup? | YES |
| Description | This cross-reference lists each field by its SEGMENT. This index is used during the claim submission process to add any additional fields to the appropriate segment (see XLOOP^BPSOSCF and XLOOP^BPSOSH2). |
| Name of Index | “SEG” |

The following fields will be added to file# 9002313.91, BPS NCPDP FIELD DEFS.

|  |  |
| --- | --- |
| **Field Attributes** | **Values** |
| FIELD NAME | REQUEST SEGMENT |
| FIELD NUMBER | 2 |
| NODE;PIECE | 5;4 |
| FIELD TYPE | Pointer to file 9002313.9, BPS NCPDP SEGMENTS |
| HELP PROMPT | Select the Request Segment which this field would appear on. |
| DESCRIPTION | This data field indicates the segment of a claim request on which this NCPDP field would appear. If this NCPDP field appears only on the incoming response, then this field would remain blank. |

|  |  |
| --- | --- |
| **Field Attributes** | **Values** |
| FIELD NAME | RESPONSE SEGMENT |
| FIELD NUMBER | 3 |
| NODE;PIECE | 5;5 |
| FIELD TYPE | Pointer to file 9002313.9, BPS NCPDP SEGMENTS |
| HELP PROMPT | Select the Response Segment which this field would appear on. |
| DESCRIPTION | This data field indicates the segment of a claim response on which this NCPDP field would appear. If this NCPDP field appears only on the outgoing claim request, then this field would remain blank. |

Two new cross-references on file# 9002313.91, BPS NCPDP FIELD DEFS, will be created to index fields by request segment and response segment.

|  |  |
| --- | --- |
| File | 9002313.91 BPS NCPDP FIELD DEFS |
| Type of Indexing | REGULAR |
| To Be Used for Lookup? | NO |
| Description | This cross-reference lists each field by its REQUEST SEGMENT. |
| Name of Index | “AC” |

|  |  |
| --- | --- |
| File | 9002313.91 BPS NCPDP FIELD DEFS |
| Type of Indexing | REGULAR |
| To Be Used for Lookup? | NO |
| Description | This cross-reference lists each field by its RESPONSE SEGMENT. |
| Name of Index | “AD” |

In the file BPS NCPDP FIELD DEFS, the new fields REQUEST SEGMENT and RESPONSE SEGMENT will be populated in the development environment. When this file is added to the build, it will be indicated that the data will come with the file. Please refer to the latest NCPDP Telecommunication Standard Implementation Guide, sections 10.4 and 10.5 for a description of which fields are on which segments.

These files must be added to the build in the manner described in the following tables

* BPS NCPDP FIELD DEFS
* BPS NCPDP OVERRIDE
* BPS NCPDP SEGMENTS

|  |  |
| --- | --- |
| File Name | BPS NCPDP FIELD DEFS |
| Send Full or Partial DD | Full |
| Update the Data Dictionary | Yes |
| Send Security Codes | No |
| Screen to Determine DD Update | n/a |
| Data Comes with File | Yes |
| Site’s Data | Merge |
| Resolve Pointers | Yes |
| May User Override Data Update | No |
| Data List | n/a |
| Screen to Select Data | n/a |

|  |  |
| --- | --- |
| File Name | BPS NCPDP OVERRIDE |
| Send Full or Partial DD | Partial |
| Update the Data Dictionary | Yes |
| Send Security Codes | No |
| Screen to Determine DD Update | n/a |
| Data Comes with File | No |
| Site’s Data | n/a |
| Field | Sub-file# 9002313.5111, NCPDP FIELD, field# .03, SEGMENT |
| Resolve Pointers | n/a |
| May User Override Data Update | n/a |
| Data List | n/a |
| Screen to Select Data | n/a |

|  |  |
| --- | --- |
| File Name | BPS NCPDP SEGMENTS |
| Send Full or Partial DD | Full |
| Update the Data Dictionary | Yes |
| Send Security Codes | No |
| Screen to Determine DD Update | No |
| Data Comes with File | Yes |
| Site’s Data | Overwrite |
| Resolve Pointers | n/a |
| May User Override Data Update | No |
| Data List | n/a |
| Screen to Select Data | n/a |