**MCCF EDI TAS US174**

**System Design Document**



Department of Veterans Affairs

**May 2017**

**Version 1.01**

**User Story Number:** USRX-167A

**User Story Name:** NCPDP Field Definition - Basic Updates

**Product Backlog ID:** 167

**Backlog Priority:** (High, Medium, Low)

**Initial Sizing Estimate:**

**Rational ID:** tbd

**Rally ID:** US174

# Design

In reviewing the latest *NCPDP Implementation Guide* (October 2016), it was determined that updates must be made to the VistA files corresponding to Data Elements. This story concerns the addition of 2 fields – B95-3Z, **Facility ID Qualifier**, and B98-34, **Reconciliation ID**.

The new field definitions will be added to File 9002313.91, BPS NCPDP FIELD DEFS, in our test environment, and that file will be included in our patch, so that the new field will be included in the KIDS Build. When the KIDS Build is installed, the new fields will be automatically added to File 9002313.91.

The data dictionary for file 9002313.91, BPS NCPDP FIELD DEFS, will be sent with data to the target sites. The table below describes how the file is to be included in the KIDS Build.

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

The table below describes how the 2 new field definitions are going to be added to file 9002313.91, BPS NCPDP FIELD DEFS.

|  |  |
| --- | --- |
| **Fields** | **Values** |
| **#.01 - NCPDP FIELD NUMBER** | B95 |
| **#.03 - NAME** | FACILITY ID QUALIFIER |
| **#.04 - FORMAT (N, A/N, D)** | A/N |
| **#.06 - ID** | 3Z |
| **#.07 - LENGTH** | 2 |
| **#.08 - D0 LENGTH** | 2 |
| **#.09 - D0 FORMAT (N, A/N, D)** | A/N |
| **#1 - VISTA FIELD NUMBER** | 2095 |
| **#1.01 - STANDARD NCPDP FIELD NAME** | FACILITY ID QUALIFIER |
| **#10 - GET CODE** | S BPS(“X”)=$G(BPS(""Insurer"",""Facility ID""))"  S BPS(""X"")=$S(BPS(""X"")’="""":""1"",1:"""") |
| **#20 - D0 FORMAT** | S BPS(“X”)=$$ANFF^BPSECFM($G(BPS(“X”)),2) |
| **#30 - SET CODE** | S $P(^BPSC(BPS(9002313.02),400,BPS(9002313.0201),  “B90”),U,1)=BPS("X") |
| **#40 - FORMAT CODE** | S BPS(“X”)=$$ANFF^BPSECFM($G(BPS(“X”)),2) |

|  |  |
| --- | --- |
| **Fields** | **Values** |
| **#.01 - NCPDP FIELD NUMBER** | B98 |
| **#.03 - NAME** | RECONCILIATION ID |
| **#.04 - FORMAT (N, A/N, D)** | A/N |
| **#.06 - ID** | 34 |
| **#.07 - LENGTH** | 30 |
| **#.08 - D0 LENGTH** | 30 |
| **#.09 - D0 FORMAT (N, A/N, D)** | A/N |
| **#1 - VISTA FIELD NUMBER** | 2098 |
| **#1.01 - STANDARD NCPDP FIELD NAME** | RECONCILIATION ID |
| **#10 - GET CODE** | "; THIS IS A RESPONSE-ONLY FIELD WHICH DOES NOT USE THE GET, FORMAT, OR SET CODE" |
| **#20 - D0 FORMAT** | n/a |
| **#30 - SET CODE** | n/a |
| **#40 - FORMAT CODE** | n/a |

The **Facility ID Qualifier** field will need to be added to the File 9002313.02, BPS CLAIMS, so it will need to be stored when we send an outgoing claim. The field will be added to File 9002313.02, BPS CLAIMS, in our test environment, and that file will be included in our patch, so that the new field will be included in the KIDS Build. When the KIDS Build is installed, the new fields will be automatically added to File 9002313.02.

The data dictionary for file 9002313.0201, BPS CLAIMS-TRANSACTIONS sub-file, will be sent without data to the target sites. The table below describes how the file is to be included in the KIDS Build.

|  |  |
| --- | --- |
| File Name | **BPS CLAIMS** |
| Send Full or Partial DD | PARTIAL |
| Data Dictionary Number | 9002313.0201 |
| Field Number | 2095 |
| Update the Data Dictionary | YES |
| Send Security Codes | NO |
| Screen to Determine DD Update | N/A |
| Data Comes with File | NO |

The table below describes how the new field is going to be added to file 9002313.0201, BPS CLAIMS, TRANSACTIONS sub-file.

|  |  |
| --- | --- |
| **Field Attributes** | **Values** |
| **FIELD NAME** | FACILITY ID QUALIFIER |
| **FIELD NUMBER** | 2095 |
| **NODE;PIECE** | B90;5 |
| **FIELD TYPE** | FREE TEXT |
| **FIELD LENGTH** | 2 |
| **INPUT TRANSFORM** | K:$L(X)>2!($L(X)<1) X |
| **HELP PROMPT** | Answer must be 1-2 characters in length. |
| **DESCRIPTION** | Code qualifying the ‘Facility ID’ (336-8C). NCPDP standard field B95-3Z.  Blank=Not Specified, 1=Facility Type 2 NPI, 2= Other |
| **TECHNICAL DESCRIPTION** | FACILITY SEGMENT |

The **Reconciliation ID** field will need to be added to the File 9002313.03, BPS RESPONSES, so it will need to be stored when we receive an incoming response. The field will be added to File 9002313.03, BPS RESPONSES, in our test environment, and that file will be included in our patch, so that the new field will be included in the KIDS Build. When the KIDS Build is installed, the new fields will be automatically added to File 9002313.03.

The data dictionary for file 9002313.0301, BPS RESPONSES-RESPONSES sub-file, will be sent without data to the target sites. The table below describes how the file is to be included in the KIDS Build.

|  |  |
| --- | --- |
| File Name | **BPS RESPONSES** |
| Send Full or Partial DD | PARTIAL |
| Data Dictionary Number | 9002313.0301 |
| Field Number | 2098 |
| Update the Data Dictionary | YES |
| Send Security Codes | NO |
| Screen to Determine DD Update | N/A |
| Data Comes with File | NO |

The table below describes how the new field is going to be added to file 9002313.0301, BPS RESPONSES, RESPONSES sub-file.

|  |  |
| --- | --- |
| **Field Attributes** | **Values** |
| **FIELD NAME** | RECONCILIATION ID |
| **FIELD NUMBER** | 2098 |
| **NODE;PIECE** | B98;1 |
| **FIELD TYPE** | FREE TEXT |
| **FIELD LENGTH** | 30 |
| **INPUT TRANSFORM** | K:$L(X)>30!($L(X)<1) X |
| **HELP PROMPT** | Answer must be 1-30 characters in length. |
| **DESCRIPTION** | This will be used to store NCPDP field B98-34 (Reconciliation ID) |

The **Reconciliation ID** field will need to be added to the File 9002313.32, BPS PAYER RESPONSE OVERRIDES, so the field can be tested on an incoming response. In order to do the incoming response from the payer, the field will need to be added to File 9002313.32, BPS PAYER RESPONSE OVERRIDES, and the routine BPSTEST will need to be modified. The field will be added to File 9002313.32, BPS PAYER RESPONSE OVERRIDES, in our test environment, and that file will be included in our patch, so that the new field will be included in the KIDS Build. When the KIDS Build is installed, the new fields will be automatically added to File 9002313.32.

The data dictionary for file 9002313.32, BPS PAYER RESPONSE OVERRIDES, will be sent without data to the target sites. The table below describes how the file is to be included in the KIDS Build.

|  |  |
| --- | --- |
| File Name | **BPS PAYER RESPONSE OVERRIDES** |
| Send Full or Partial DD | PARTIAL |
| Data Dictionary Number | 9002313.32 |
| Field Number | 2.09 |
| Update the Data Dictionary | YES |
| Send Security Codes | NO |
| Screen to Determine DD Update | N/A |
| Data Comes with File | NO |

The table below describes how the new field is going to be added to file 9002313.32, BPS PAYER RESPONSE OVERRIDES.

|  |  |
| --- | --- |
| **Field Attributes** | **Values** |
| **FIELD NAME** | RECONCILIATION ID |
| **FIELD NUMBER** | 2.09 |
| **NODE;PIECE** | 2;9 |
| **FIELD TYPE** | FREE TEXT |
| **FIELD LENGTH** | 30 |
| **INPUT TRANSFORM** | K:$L(X)>30!($L(X)<1) X |
| **HELP PROMPT** | Answer must be 1-30 characters in length. |
| **DESCRIPTION** | This is the override value that will be used for the payer response for billing requests. The value will be used to populate the RECONCILIATION ID (#2098) field of RESPONSES (#9002313.0301) subfile of the BPS RESPONSES (#9002313.03) file. This corresponds to NCPDP code B98-34. |

The table below describes how the routine BPSTEST will be modified in order for the **Reconciliation ID** field to be included in the testing tool.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Subroutine Name | **GETOVER^BPSTEST** | | | | | | | |
| Enhancement Category | New | | Modify | Delete | | | No Change | |
| Story | US174 (Rally ID) | | | | | | | |
| Related Options | n/a | | | | | | | |
| Related Subroutines | This Subroutine is Called By | | | | This Subroutine Calls | | | |
| REVRESUB^BPSNCPD4, REVERSAL^BPSNCPD5, NEWCLM^BPSNCPD5, REVRESNP^BPSNCPD5, EN^BPSNCPD9 | | | | $$IEN59^BPSOSRX,  ^DIC,^DIR,^DIK,  GET1^DIQ | | | |
| Data Dictionary (DD) References | n/a | | | | | | | |
| Related Protocols | n/a | | | | | | | |
| Related Integration Control Registrations (ICRs) | n/a | | | | | | | |
| Data Passing | Input | Output Reference | | | Both | Global Reference | | Local |
| Input Attribute Name and Definition | Name: KEY1  Definition: Prescription IEN/Patient IEN  New  Modify  Delete  No Change  Name: KEY2  Definition: Fill Number/Policy Number  New  Modify  Delete  No Change  Name: BPSORESP  Definition: Previous response when this claim was processed  New  Modify  Delete  No Change  Name: BPSWHERE  Definition: RX Action passed into BPSNCPDP  New  Modify  Delete  No Change  Name: BPSTYPE  Definition: R (Reversal), S (Submission), E (Eligibility)  New  Modify  Delete  No Change  Name: BPPAYSEQ  Definition: payer sequence 1 - primary, 2 - secondary  New  Modify  Delete  No Change | | | | | | | |
| Output Attribute Name and Definition | Name: n/a - the BPS PAYER RESPONSE OVERRIDE entry is created  Definition:  New  Modify  Delete  No Change | | | | | | | |
| Current Logic | | | | | | | | |
| BPSTEST ;OAK/ELZ - ECME TESTING TOOL ;11/15/07 09:55  ;;1.0;E CLAIMS MGMT ENGINE;\*\*6,7,8,10,11,15,19,20\*\*;JUN 2004;Build 27  ;;Per VA Directive 6402, this routine should not be modified.  ;  ;GETOVER(KEY1,KEY2,BPSORESP,BPSWHERE,BPSTYPE,BPPAYSEQ) ;  ; called by BPSNCPDP to enter overrides for a particular RX  ; INPUT  ; KEY1 - Prescription IEN/Patient IEN  ; KEY2 - Fill Number/Policy Number  ; BPSORESP - Previous response when this claim was processed  ; BPSWHERE - RX Action passed into BPSNCPDP  ; BPSTYPE - R (Reversal), S (Submission), E (Eligibility)  ; BPPAYSEQ - payer sequence 1 - primary, 2 - secondary  ; OUTPUT  ; None - Table BPS PAYER RESPONSE OVERRIDE entry is created.  ;  N BPSTRANS,BPSTIEN,BPSSRESP,DIC,X,Y,DIR,DIK,DA  ;  .  .  .  . ; This section is for new D1-E7 fields and other fields so we can test that t  hey are filed correctly  . ; At some point, these can probably be removed  . I BPSSRESP="P"!(BPSSRESP="D")!(BPSSRESP="R") D  .. ;  .. ; Ask if user wants to enter data for additional response file fields - Quit  if user says no  .. N DIR,DTOUT,DUOUT,DIROUT,DIRUT  .. S DIR(0)="YA",DIR("A")="Populate Additional Response Fields? ",DIR("B")="No"  W ! D ^DIR  .. I Y'=1 Q  .. ;  .. ; Overrides to test functionality of BPS\*1\*20  .. D PROMPT(BPSTIEN,.15,0) ; Ingredient Cost Paid (506)  .. D PROMPT(BPSTIEN,.16,0) ; Dispensing Fee Paid (507)  .. D PROMPT(BPSTIEN,.17,0) ; Remaining Deductible Amount (513)  .. D PROMPT(BPSTIEN,.18,0) ; Amount Applied to Periodic Deductible (517)  .. ;  .. ; Additional overrides for D1-D9 (BPS\*1\*15)  .. D PROMPT(BPSTIEN,.09,"") ; next available fill date  .. D PROMPT(BPSTIEN,.1,"") ; adjudicated payment type  .. ;  .. ; Additional overrides for E0-E6 (BPS\*1\*19)  .. D PROMPT(BPSTIEN,2.01,"04") ; % sales tax basis pd  .. D PROMPT(BPSTIEN,2.02,11) ; other amount paid qualifier  .. D PROMPT(BPSTIEN,2.03,"01") ; payer id qualifier  .. D PROMPT(BPSTIEN,2.04,"") ; help desk phone# ext  .. D PROMPT(BPSTIEN,2.05,"") ; pro service fee cont/reim amt  .. D PROMPT(BPSTIEN,2.06,"") ; other payer help desk phone# ext  .. D PROMPT(BPSTIEN,2.07,"") ; response intermed auth type id  .. D PROMPT(BPSTIEN,2.08,"") ; response intermed auth id  .. D PROMPT(BPSTIEN,3.01,"") ; response intermed message  .. ;  .. ; E7 overrides (BPS\*1\*20)  .. D PROMPT(BPSTIEN,.11,"") ; quan limit per specific time period  .. D PROMPT(BPSTIEN,.12,"") ; quan limit time period  .. D PROMPT(BPSTIEN,.13,"") ; days supp limit per specific time period  .. D PROMPT(BPSTIEN,.14,"") ; days supp limit time period  ;  W ! D PROMPT(BPSTIEN,.07,0)  Q | | | | | | | | |
| Modified Logic (Changes are highlighted) | | | | | | | | |
| BPSTEST ;OAK/ELZ - ECME TESTING TOOL ;11/15/07 09:55  ;;1.0;E CLAIMS MGMT ENGINE;\*\*6,7,8,10,11,15,19,20,22\*\*;JUN 2004;Build 27  ;;Per VA Directive 6402, this routine should not be modified.  ;  GETOVER(KEY1,KEY2,BPSORESP,BPSWHERE,BPSTYPE,BPPAYSEQ) ;  ; called by BPSNCPDP to enter overrides for a particular RX  ; INPUT  ; KEY1 - Prescription IEN/Patient IEN  ; KEY2 - Fill Number/Policy Number  ; BPSORESP - Previous response when this claim was processed  ; BPSWHERE - RX Action passed into BPSNCPDP  ; BPSTYPE - R (Reversal), S (Submission), E (Eligibility)  ; BPPAYSEQ - payer sequence 1 - primary, 2 - secondary  ; OUTPUT  ; None - Table BPS PAYER RESPONSE OVERRIDE entry is created.  ;  N BPSTRANS,BPSTIEN,BPSSRESP,DIC,X,Y,DIR,DIK,DA  ;  .  .  .  ;  ; If BPSTYPE contains 'S', do submission response  I BPSTYPE["S" D  . W !!,"Submission Questions"  . D PROMPT(BPSTIEN,.03,"P")  . S BPSSRESP=$$GET1^DIQ(9002313.32,BPSTIEN\_",",.03,"I")  . I BPSSRESP="P"!(BPSSRESP="D") D PROMPT(BPSTIEN,.04,40) ; total amount paid (509-F9)  . I BPSSRESP="P"!(BPSSRESP="D") D PROMPT(BPSTIEN,.06,9) ; copay amount (518-FI)  . I BPSSRESP="R" D PROMPT(BPSTIEN,1,"07")  . ;  . ; This section is for new D1-E7 fields and other fields so we can test that they are filed correctly  . ; At some point, these can probably be removed  . I BPSSRESP="P"!(BPSSRESP="D")!(BPSSRESP="R") D  .. ;  .. ; Ask if user wants to enter data for additional response file fields - Quit if user says no  .. N DIR,DTOUT,DUOUT,DIROUT,DIRUT  .. S DIR(0)="YA",DIR("A")="Populate Additional Response Fields? ",DIR("B")="No" W ! D ^DIR  .. I Y'=1 Q  .. ;  .. ; Overrides to test functionality of BPS\*1\*20  .. D PROMPT(BPSTIEN,.15,0) ; Ingredient Cost Paid (506)  .. D PROMPT(BPSTIEN,.16,0) ; Dispensing Fee Paid (507)  .. D PROMPT(BPSTIEN,.17,0) ; Remaining Deductible Amount (513)  .. D PROMPT(BPSTIEN,.18,0) ; Amount Applied to Periodic Deductible (517)  .. ;  .. ; Additional overrides for D1-D9 (BPS\*1\*15)  .. D PROMPT(BPSTIEN,.09,"") ; next available fill date  .. D PROMPT(BPSTIEN,.1,"") ; adjudicated payment type  .. ;  .. ; Additional overrides for E0-E6 (BPS\*1\*19)  .. D PROMPT(BPSTIEN,2.01,"04") ; % sales tax basis pd  .. D PROMPT(BPSTIEN,2.02,11) ; other amount paid qualifier  .. D PROMPT(BPSTIEN,2.03,"01") ; payer id qualifier  .. D PROMPT(BPSTIEN,2.04,"") ; help desk phone# ext  .. D PROMPT(BPSTIEN,2.05,"") ; pro service fee cont/reim amt  .. D PROMPT(BPSTIEN,2.06,"") ; other payer help desk phone# ext  .. D PROMPT(BPSTIEN,2.07,"") ; response intermed auth type id  .. D PROMPT(BPSTIEN,2.08,"") ; response intermed auth id  .. D PROMPT(BPSTIEN,3.01,"") ; response intermed message  .. ;  .. ; E7 overrides (BPS\*1\*20)  .. D PROMPT(BPSTIEN,.11,"") ; quan limit per specific time period  .. D PROMPT(BPSTIEN,.12,"") ; quan limit time period  .. D PROMPT(BPSTIEN,.13,"") ; days supp limit per specific time period  .. D PROMPT(BPSTIEN,.14,"") ; days supp limit time period  .. ; Overrides to test functionality - BPS\*1\*22  .. D PROMPT(BPSTIEN,2.09,"") ; reconciliation id  ;  W ! D PROMPT(BPSTIEN,.07,0)  Q | | | | | | | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Subroutine Name | **SETOVER^BPSTEST** | | | | | | | |
| Enhancement Category | New | | Modify | Delete | | No Change | | |
| Story | US174 (Rally ID) | | | | | | | |
| Related Options | n/a | | | | | | | |
| Related Subroutines | This Subroutine is Called By | | | | This Subroutine Calls | | | |
| PARSE^BPSECMPS | | | | GET1^DIQ | | | |
| Data Dictionary (DD) References | n/a | | | | | | | |
| Related Protocols | n/a | | | | | | | |
| Related Integration Control Registrations (ICRs) | n/a | | | | | | | |
| Data Passing | Input | Output Reference | | | Both | | Global Reference | Local |
| Input Attribute Name and Definition | Name: BPSTRANS  Definition: Transaction IEN  New  Modify  Delete  No Change  Name: BPSTYPE  Definition: B1 for submission, B2 for reversals  New  Modify  Delete  No Change | | | | | | | |
| Output Attribute Name and Definition | Name: BPSDATA  Definition: passed by reference and update with appropriate overrides  New  Modify  Delete  No Change | | | | | | | |
| Current Logic | | | | | | | | |
| SETOVER(BPSTRANS,BPSTYPE,BPSDATA) ;  ; called by BPSECMPS to set the override data  ; Input  ; BPSTRANS - Transaction IEN  ; BPSTYPE - B1 for submission, B2 for reversals  ; Output  ; BPSDATA - Passed by reference and updated with appropriate overrides  ;  N BPSTIEN,BPSRRESP,BPSSRESP,BPSPAID,BPSRCNT,BPSRIEN,BPSRCODE,BPSRCD,BPSCOPAY,  BPSXXXX,BPSUNDEF  N BPSAJPAY,BPSNFLDT,BPSX  N BPS506,BPS507,BPS513,BPS517  ;  ; Check the Test Flag in set in BPS SETUP  I '$$CHECK() Q  ;  ; Check if the Transaction Number is defined in BPS RESPONSE OVERRIDES  S BPSTIEN=$O(^BPS(9002313.32,"B",BPSTRANS,""))  I BPSTIEN="" Q  ;  ; If a eligibility, check for specific reversal overrides and set  I BPSTYPE="E1" D Q  . S BPSRRESP=$$GET1^DIQ(9002313.32,BPSTIEN\_",",.08,"I")  . ;  . ; If the response is Stranded, force an <UNDEF> error  . I BPSRRESP="S" S BPSXXXX=BPSUNDEF  . I BPSRRESP]"" S BPSDATA(1,112)=BPSRRESP  . S BPSDATA(9002313.03,9002313.03,"+1,",501)=$S(BPSRRESP="R":"R",1:"A")  . ;  . ; If the response is accepted, delete the reject code count and codes  . I BPSRRESP="A" K BPSDATA(1,510),BPSDATA(1,511)  . ;  . ; If the response is rejected, delete the rejections returned by payers  . ; and put in the ones entered by the user  . I BPSRRESP="R" D  .. K BPSDATA(1,509),BPSDATA(1,511)  .. S BPSRCNT=0  .. S BPSRIEN=0 F S BPSRIEN=$O(^BPS(9002313.32,BPSTIEN,1,BPSRIEN)) Q:+BPSRIEN=0 D  ... S BPSRCODE=$P($G(^BPS(9002313.32,BPSTIEN,1,BPSRIEN,0)),"^",1)  ... ; Increment counter and store  ... I BPSRCODE]"" D  .... S BPSRCD=$$GET1^DIQ(9002313.93,BPSRCODE\_",",.01,"E")  .... I BPSRCD]"" S BPSRCNT=BPSRCNT+1,BPSDATA(1,511,BPSRCNT)=BPSRCD  .. ; Store total number of rejections  .. S BPSDATA(1,510)=BPSRCNT  ;  ; If a reversal, check for specific reversal overrides and set  I BPSTYPE="B2" D  . S BPSRRESP=$$GET1^DIQ(9002313.32,BPSTIEN\_",",.05,"I")  . ;  . ; If the response is Stranded, force an <UNDEF> error  . I BPSRRESP="S" S BPSXXXX=BPSUNDEF  . I BPSRRESP]"" S BPSDATA(1,112)=$S(BPSRRESP="D":"S",1:BPSRRESP)  . S BPSDATA(9002313.03,9002313.03,"+1,",501)=$S(BPSRRESP="R":"R",1:"A")  . ;  . ; If the response is accepted or duplicate, kill the reject code count and codes  . I BPSRRESP="A"!(BPSRRESP="D") K BPSDATA(1,510),BPSDATA(1,511)  . ;  . ; If the response is rejected, set the reject codes  . I BPSRRESP="R" D SETREJ(BPSTRANS)  ;  ; If a submission, check for specific submission overrides and set  I BPSTYPE="B1" D  . ; Get submission response  . S BPSSRESP=$$GET1^DIQ(9002313.32,BPSTIEN\_",",.03,"I")  . ;  . ; If the response is Stranded, force an <UNDEF> error  . I BPSSRESP="S" S BPSXXXX=BPSUNDEF  . ;  . ; If BPSSRESP exists, file it  . I BPSSRESP]"" D  .. S BPSDATA(1,112)=BPSSRESP  .. S BPSDATA(9002313.03,9002313.03,"+1,",501)=$S(BPSSRESP="R":"R",1:"A")  .. ; If payable or duplicate, get the BPSPAID amount and file it if it  .. ; exists. Also delete any reject codes  .. I BPSSRESP="P"!(BPSSRESP="D") D  ... S BPSPAID=$$GET1^DIQ(9002313.32,BPSTIEN\_",",.04,"I")  ... I BPSPAID]"" S BPSDATA(1,509)=$$DFF^BPSECFM(BPSPAID,8) ; 509 Total amount paid  ... ;  ... K BPSDATA(1,510),BPSDATA(1,511) ; kill Reject Count (510) and Reject Code (511)  ... ;  ... S BPSCOPAY=$$GET1^DIQ(9002313.32,BPSTIEN\_",",.06,"I")  ... I BPSCOPAY]"" S BPSDATA(1,518)=$$DFF^BPSECFM(BPSCOPAY,8) ; 518 Copay Amount  ... ;  ... S BPS506=$$GET1^DIQ(9002313.32,BPSTIEN\_",",.15,"I")  ... I BPS506]"" S BPSDATA(1,506)=$$DFF^BPSECFM(BPS506,8) ; 506 Ingredient Cost Paid  ... ;  ... S BPS507=$$GET1^DIQ(9002313.32,BPSTIEN\_",",.16,"I")  ... I BPS507]"" S BPSDATA(1,507)=$$DFF^BPSECFM(BPS507,8) ; 507 Dispensing Fee Paid  ... ;  ... S BPS513=$$GET1^DIQ(9002313.32,BPSTIEN\_",",.17,"I")  ... I BPS513]"" S BPSDATA(1,513)=$$DFF^BPSECFM(BPS513,8) ; 513 Remaining Deductible Amount  ... ;  ... S BPS517=$$GET1^DIQ(9002313.32,BPSTIEN\_",",.18,"I")  ... I BPS517]"" S BPSDATA(1,517)=$$DFF^BPSECFM(BPS517,8) ; 517 Amount Applied to Periodic Deductible  ... Q  .. ;  .. I BPSSRESP="P"!(BPSSRESP="D")!(BPSSRESP="R") D  ... ; D1-D9 fields (BPS\*1\*15)  ... S BPSAJPAY=$$GET1^DIQ(9002313.32,BPSTIEN\_",",.1,"I") ; Adjudicated Payment Type  ... I BPSAJPAY]"" S BPSDATA(1,1028)=$$NFF^BPSECFM(BPSAJPAY,2)  ... S BPSNFLDT=$$GET1^DIQ(9002313.32,BPSTIEN\_",",.09,"I") ; Override Next Available Fill  ... I BPSNFLDT]"" S BPSDATA(1,2004)=$$DTF1^BPSECFM(BPSNFLDT)  ... ;  ... ; E0-E6 overrides (BPS\*1\*19)  ... ; PERCENTAGE SALES TAX BASIS PAID  ... S BPSX=$$GET1^DIQ(9002313.32,BPSTIEN\_",",2.01,"I")  ... I BPSX]"" S BPSDATA(1,561)=BPSX  ... ; OTHER AMOUNT PAID QUALIFIER and associated field  ... S BPSX=$$GET1^DIQ(9002313.32,BPSTIEN\_",",2.02,"I")  ... I BPSX]"" S BPSDATA(1,564,1)=$$NFF^BPSECFM(BPSX,2),BPSDATA(1,565,1)=$$DFF^  BPSECFM(5.64,8),BPSDATA(1,563)=1  ... ; PAYER ID QUALIFIER  ... S BPSX=$$GET1^DIQ(9002313.32,BPSTIEN\_",",2.03,"I")  ... I BPSX]"" S BPSDATA(9002313.03,9002313.03,"+1,",568)=BPSX  ... ; HELP DESK TELEPHONE NUMBER EXTENSION  ... S BPSX=$$GET1^DIQ(9002313.32,BPSTIEN\_",",2.04,"I")  ... I BPSX]"" S BPSDATA(1,"2022")=$$NFF^BPSECFM(BPSX,8)  ... ; PROFESSIONAL SERVICE FEE CONTRACTED/REIMURSEMENT AMOUNT  ... S BPSX=$$GET1^DIQ(9002313.32,BPSTIEN\_",",2.05,"I")  ... I BPSX]"" S BPSDATA(1,"2033")=$$DFF^BPSECFM(BPSX,8)  ... ; OTHER PAYER HELPDESK TELEPHONE EXTENSION  ... S BPSX=$$GET1^DIQ(9002313.32,BPSTIEN\_",",2.06,"I")  ... I BPSX]"" S BPSDATA(1,"2023",1)=$$NFF^BPSECFM(BPSX,8),BPSDATA(1,338,1)="01"  ... ; RESPONSE INTERMEDIARY AUTHORIZATION TYPE ID and associated fields  ... S BPSX=$$GET1^DIQ(9002313.32,BPSTIEN\_",",2.07,"I")  ... I BPSX]"" S BPSDATA(1,"2053",1)=$$NFF^BPSECFM(BPSX,2),BPSDATA(1,2052)=1  ... ; RESPONSE INTERMEDIARY AUTHORIZATION ID and associated fields  ... S BPSX=$$GET1^DIQ(9002313.32,BPSTIEN\_",",2.08,"I")  ... I BPSX]"" S BPSDATA(1,"2054",1)=$$ANFF^BPSECFM(BPSX,20),BPSDATA(1,2052)=1  ... ; PERCENTAGE SALES TAX BASIS PAID  ... S BPSX=$$GET1^DIQ(9002313.32,BPSTIEN\_",",2.01,"I")  ... I BPSX]"" S BPSDATA(1,561)=BPSX  ... ; INTERMEDIARY MESSAGE and associated fields  ... S BPSX=$$GET1^DIQ(9002313.32,BPSTIEN\_",",3.01,"I")  ... I BPSX]"" S BPSDATA(1,"2051",1)=$$ANFF^BPSECFM(BPSX,200),BPSDATA(1,2052)=1  ... ;  ... ; E7 overrides (BPS\*1\*20)  ... S BPSX=$$GET1^DIQ(9002313.32,BPSTIEN\_",",.11,"I") I BPSX'="" D ; B88-3R quantity limit per spec time period  .... S BPSDATA(1,2087)=1 ; count field  .... S BPSDATA(1,2088,1)=$$NFF^BPSECFM(BPSX,10) ; data from override file  .... Q  ... S BPSX=$$GET1^DIQ(9002313.32,BPSTIEN\_",",.12,"I") I BPSX'="" D ; B89-3S quantity limit time period  .... S BPSDATA(1,2087)=1 ; count field  .... S BPSDATA(1,2089,1)=$$NFF^BPSECFM(BPSX,5) ; data from override file  .... Q  ... S BPSX=$$GET1^DIQ(9002313.32,BPSTIEN\_",",.13,"I") I BPSX'="" D ; B91-3W days supply limit per spec time period  .... S BPSDATA(1,2090)=1 ; count field  .... S BPSDATA(1,2091,1)=$$NFF^BPSECFM(BPSX,3) ; data from override file  .... Q  ... S BPSX=$$GET1^DIQ(9002313.32,BPSTIEN\_",",.14,"I") I BPSX'="" D ; B92-3X days supply limit time period  .... S BPSDATA(1,2090)=1 ; count field  .... S BPSDATA(1,2092,1)=$$NFF^BPSECFM(BPSX,5) ; data from override file  .... Q  ... Q  .. ;  .. ; If rejected, get the rejection code and file them  .. ; Also, delete the BPSPAID amount  .. I BPSSRESP="R" D  ... ; Delete old rejections and BPSPAID amount  ... K BPSDATA(1,509),BPSDATA(1,511)  ... ; Loop through rejections and store  ... S BPSRCNT=0  ... S BPSRIEN=0 F S BPSRIEN=$O(^BPS(9002313.32,BPSTIEN,1,BPSRIEN)) Q:+BPSRIEN=0 D  .... S BPSRCODE=$P($G(^BPS(9002313.32,BPSTIEN,1,BPSRIEN,0)),"^",1)  .... ; Increment counter and store  .... I BPSRCODE]"" D  ..... S BPSRCD=$$GET1^DIQ(9002313.93,BPSRCODE\_",",.01,"E")  ..... I BPSRCD]"" S BPSRCNT=BPSRCNT+1,BPSDATA(1,511,BPSRCNT)=BPSRCD  ... ; Store total number of rejections  ... S BPSDATA(1,510)=BPSRCNT  Q | | | | | | | | |
| Modified Logic (Changes are highlighted) | | | | | | | | |
| SETOVER(BPSTRANS,BPSTYPE,BPSDATA) ;  ; called by BPSECMPS to set the override data  ; Input  ; BPSTRANS - Transaction IEN  ; BPSTYPE - B1 for submission, B2 for reversals  ; Output  ; BPSDATA - Passed by reference and updated with appropriate overrides  ;  N BPSTIEN,BPSRRESP,BPSSRESP,BPSPAID,BPSRCNT,BPSRIEN,BPSRCODE,BPSRCD,BPSCOPAY,  BPSXXXX,BPSUNDEF  N BPSAJPAY,BPSNFLDT,BPSX  N BPS506,BPS507,BPS513,BPS517  ;  ; Check the Test Flag in set in BPS SETUP  I '$$CHECK() Q  ;  ; Check if the Transaction Number is defined in BPS RESPONSE OVERRIDES  S BPSTIEN=$O(^BPS(9002313.32,"B",BPSTRANS,""))  I BPSTIEN="" Q  ;  ; If a eligibility, check for specific reversal overrides and set  I BPSTYPE="E1" D Q  . S BPSRRESP=$$GET1^DIQ(9002313.32,BPSTIEN\_",",.08,"I")  . ;  . ; If the response is Stranded, force an <UNDEF> error  . I BPSRRESP="S" S BPSXXXX=BPSUNDEF  . I BPSRRESP]"" S BPSDATA(1,112)=BPSRRESP  . S BPSDATA(9002313.03,9002313.03,"+1,",501)=$S(BPSRRESP="R":"R",1:"A")  . ;  . ; If the response is accepted, delete the reject code count and codes  . I BPSRRESP="A" K BPSDATA(1,510),BPSDATA(1,511)  . ;  . ; If the response is rejected, delete the rejections returned by payers  . ; and put in the ones entered by the user  . I BPSRRESP="R" D  .. K BPSDATA(1,509),BPSDATA(1,511)  .. S BPSRCNT=0  .. S BPSRIEN=0 F S BPSRIEN=$O(^BPS(9002313.32,BPSTIEN,1,BPSRIEN)) Q:+BPSRIEN=0 D  ... S BPSRCODE=$P($G(^BPS(9002313.32,BPSTIEN,1,BPSRIEN,0)),"^",1)  ... ; Increment counter and store  ... I BPSRCODE]"" D  .... S BPSRCD=$$GET1^DIQ(9002313.93,BPSRCODE\_",",.01,"E")  .... I BPSRCD]"" S BPSRCNT=BPSRCNT+1,BPSDATA(1,511,BPSRCNT)=BPSRCD  .. ; Store total number of rejections  .. S BPSDATA(1,510)=BPSRCNT  ;  ; If a reversal, check for specific reversal overrides and set  I BPSTYPE="B2" D  . S BPSRRESP=$$GET1^DIQ(9002313.32,BPSTIEN\_",",.05,"I")  . ;  . ; If the response is Stranded, force an <UNDEF> error  . I BPSRRESP="S" S BPSXXXX=BPSUNDEF  . I BPSRRESP]"" S BPSDATA(1,112)=$S(BPSRRESP="D":"S",1:BPSRRESP)  . S BPSDATA(9002313.03,9002313.03,"+1,",501)=$S(BPSRRESP="R":"R",1:"A")  . ;  . ; If the response is accepted or duplicate, kill the reject code count and codes  . I BPSRRESP="A"!(BPSRRESP="D") K BPSDATA(1,510),BPSDATA(1,511)  . ;  . ; If the response is rejected, set the reject codes  . I BPSRRESP="R" D SETREJ(BPSTRANS)  ;  ; If a submission, check for specific submission overrides and set  I BPSTYPE="B1" D  . ; Get submission response  . S BPSSRESP=$$GET1^DIQ(9002313.32,BPSTIEN\_",",.03,"I")  . ;  . ; If the response is Stranded, force an <UNDEF> error  . I BPSSRESP="S" S BPSXXXX=BPSUNDEF  . ;  . ; If BPSSRESP exists, file it  . I BPSSRESP]"" D  .. S BPSDATA(1,112)=BPSSRESP  .. S BPSDATA(9002313.03,9002313.03,"+1,",501)=$S(BPSSRESP="R":"R",1:"A")  .. ; If payable or duplicate, get the BPSPAID amount and file it if it  .. ; exists. Also delete any reject codes  .. I BPSSRESP="P"!(BPSSRESP="D") D  ... S BPSPAID=$$GET1^DIQ(9002313.32,BPSTIEN\_",",.04,"I")  ... I BPSPAID]"" S BPSDATA(1,509)=$$DFF^BPSECFM(BPSPAID,8) ; 509 Total amount paid  ... ;  ... K BPSDATA(1,510),BPSDATA(1,511) ; kill Reject Count (510) and Reject Code (511)  ... ;  ... S BPSCOPAY=$$GET1^DIQ(9002313.32,BPSTIEN\_",",.06,"I")  ... I BPSCOPAY]"" S BPSDATA(1,518)=$$DFF^BPSECFM(BPSCOPAY,8) ; 518 Copay Amount  ... ;  ... S BPS506=$$GET1^DIQ(9002313.32,BPSTIEN\_",",.15,"I")  ... I BPS506]"" S BPSDATA(1,506)=$$DFF^BPSECFM(BPS506,8) ; 506 Ingredient Cost Paid  ... ;  ... S BPS507=$$GET1^DIQ(9002313.32,BPSTIEN\_",",.16,"I")  ... I BPS507]"" S BPSDATA(1,507)=$$DFF^BPSECFM(BPS507,8) ; 507 Dispensing Fee Paid  ... ;  ... S BPS513=$$GET1^DIQ(9002313.32,BPSTIEN\_",",.17,"I")  ... I BPS513]"" S BPSDATA(1,513)=$$DFF^BPSECFM(BPS513,8) ; 513 Remaining Deductible Amount  ... ;  ... S BPS517=$$GET1^DIQ(9002313.32,BPSTIEN\_",",.18,"I")  ... I BPS517]"" S BPSDATA(1,517)=$$DFF^BPSECFM(BPS517,8) ; 517 Amount Applied to Periodic Deductible  ... Q  .. ;  .. I BPSSRESP="P"!(BPSSRESP="D")!(BPSSRESP="R") D  ... ; D1-D9 fields (BPS\*1\*15)  ... S BPSAJPAY=$$GET1^DIQ(9002313.32,BPSTIEN\_",",.1,"I") ; Adjudicated Payment Type  ... I BPSAJPAY]"" S BPSDATA(1,1028)=$$NFF^BPSECFM(BPSAJPAY,2)  ... S BPSNFLDT=$$GET1^DIQ(9002313.32,BPSTIEN\_",",.09,"I") ; Override Next Available Fill  ... I BPSNFLDT]"" S BPSDATA(1,2004)=$$DTF1^BPSECFM(BPSNFLDT)  ... ;  ... ; E0-E6 overrides (BPS\*1\*19)  ... ; PERCENTAGE SALES TAX BASIS PAID  ... S BPSX=$$GET1^DIQ(9002313.32,BPSTIEN\_",",2.01,"I")  ... I BPSX]"" S BPSDATA(1,561)=BPSX  ... ; OTHER AMOUNT PAID QUALIFIER and associated field  ... S BPSX=$$GET1^DIQ(9002313.32,BPSTIEN\_",",2.02,"I")  ... I BPSX]"" S BPSDATA(1,564,1)=$$NFF^BPSECFM(BPSX,2),BPSDATA(1,565,1)=$$DFF^  BPSECFM(5.64,8),BPSDATA(1,563)=1  ... ; PAYER ID QUALIFIER  ... S BPSX=$$GET1^DIQ(9002313.32,BPSTIEN\_",",2.03,"I")  ... I BPSX]"" S BPSDATA(9002313.03,9002313.03,"+1,",568)=BPSX  ... ; HELP DESK TELEPHONE NUMBER EXTENSION  ... S BPSX=$$GET1^DIQ(9002313.32,BPSTIEN\_",",2.04,"I")  ... I BPSX]"" S BPSDATA(1,"2022")=$$NFF^BPSECFM(BPSX,8)  ... ; PROFESSIONAL SERVICE FEE CONTRACTED/REIMURSEMENT AMOUNT  ... S BPSX=$$GET1^DIQ(9002313.32,BPSTIEN\_",",2.05,"I")  ... I BPSX]"" S BPSDATA(1,"2033")=$$DFF^BPSECFM(BPSX,8)  ... ; OTHER PAYER HELPDESK TELEPHONE EXTENSION  ... S BPSX=$$GET1^DIQ(9002313.32,BPSTIEN\_",",2.06,"I")  ... I BPSX]"" S BPSDATA(1,"2023",1)=$$NFF^BPSECFM(BPSX,8),BPSDATA(1,338,1)="01"  ... ; RESPONSE INTERMEDIARY AUTHORIZATION TYPE ID and associated fields  ... S BPSX=$$GET1^DIQ(9002313.32,BPSTIEN\_",",2.07,"I")  ... I BPSX]"" S BPSDATA(1,"2053",1)=$$NFF^BPSECFM(BPSX,2),BPSDATA(1,2052)=1  ... ; RESPONSE INTERMEDIARY AUTHORIZATION ID and associated fields  ... S BPSX=$$GET1^DIQ(9002313.32,BPSTIEN\_",",2.08,"I")  ... I BPSX]"" S BPSDATA(1,"2054",1)=$$ANFF^BPSECFM(BPSX,20),BPSDATA(1,2052)=1  ... ; INTERMEDIARY MESSAGE and associated fields  ... S BPSX=$$GET1^DIQ(9002313.32,BPSTIEN\_",",3.01,"I")  ... I BPSX]"" S BPSDATA(1,"2051",1)=$$ANFF^BPSECFM(BPSX,200),BPSDATA(1,2052)=1  ... ; (BPS\*1\*22)  ... S BPSX=$$GET1^DIQ(9002313.32,BPSTIEN\_",",2.09,"I") ; B98-34 reconciliation id  ... I BPSX]"" S BPSDATA(1,”2098”)=$$ANFF^BPSECFM(BPSX,30)  ... ;  ... ; E7 overrides (BPS\*1\*20)  ... S BPSX=$$GET1^DIQ(9002313.32,BPSTIEN\_",",.11,"I") I BPSX'="" D ; B88-3R quantity limit per spec time period  .... S BPSDATA(1,2087)=1 ; count field  .... S BPSDATA(1,2088,1)=$$NFF^BPSECFM(BPSX,10) ; data from override file  .... Q  ... S BPSX=$$GET1^DIQ(9002313.32,BPSTIEN\_",",.12,"I") I BPSX'="" D ; B89-3S quantity limit time period  .... S BPSDATA(1,2087)=1 ; count field  .... S BPSDATA(1,2089,1)=$$NFF^BPSECFM(BPSX,5) ; data from override file  .... Q  ... S BPSX=$$GET1^DIQ(9002313.32,BPSTIEN\_",",.13,"I") I BPSX'="" D ; B91-3W days supply limit per spec time period  .... S BPSDATA(1,2090)=1 ; count field  .... S BPSDATA(1,2091,1)=$$NFF^BPSECFM(BPSX,3) ; data from override file  .... Q  ... S BPSX=$$GET1^DIQ(9002313.32,BPSTIEN\_",",.14,"I") I BPSX'="" D ; B92-3X days supply limit time period  .... S BPSDATA(1,2090)=1 ; count field  .... S BPSDATA(1,2092,1)=$$NFF^BPSECFM(BPSX,5) ; data from override file  .... Q  ... Q  .. ;  .. ; If rejected, get the rejection code and file them  .. ; Also, delete the BPSPAID amount  .. I BPSSRESP="R" D  ... ; Delete old rejections and BPSPAID amount  ... K BPSDATA(1,509),BPSDATA(1,511)  ... ; Loop through rejections and store  ... S BPSRCNT=0  ... S BPSRIEN=0 F S BPSRIEN=$O(^BPS(9002313.32,BPSTIEN,1,BPSRIEN)) Q:+BPSRIEN=0 D  .... S BPSRCODE=$P($G(^BPS(9002313.32,BPSTIEN,1,BPSRIEN,0)),"^",1)  .... ; Increment counter and store  .... I BPSRCODE]"" D  ..... S BPSRCD=$$GET1^DIQ(9002313.93,BPSRCODE\_",",.01,"E")  ..... I BPSRCD]"" S BPSRCNT=BPSRCNT+1,BPSDATA(1,511,BPSRCNT)=BPSRCD  ... ; Store total number of rejections  ... S BPSDATA(1,510)=BPSRCNT  Q | | | | | | | | |