

```

/*-----*/
/*---          MAINLINE          PRTM001          (PRTM001)          ---*/
/*---          FROM LIB: RLS94          FILE: QSRC3          MBR: PRTM001          ---*/
/*---          CONV.BY: PRISMDAVE          ---*/
/*--- SECTION TYPE : MAINLINE          ---*/
/*--- Generated PROGRESS via MAKEPROGRESS          Release 1996.02.09          ---*/
/*--- (C) 1993 by EFA Software Services Ltd.          2/13/2006          ---*/
/*-----*/
&GLOBAL-DEFINE Patch 01
&GLOBAL-DEFINE Version 9.4.0

```

```

/*****/
/* RPG SUBROUTINE          PROGRESS CALL          */
/* MAINLINE          TM001_          */
/* $$001          */
/* $$002          */
/* $$003          */
/* $$004          */
/* $$005          */
/* $TIME          */
/*****/

```

```

/* RPG SCREEN          PROGRESS CALL          */
/* EXFMT SCR3          TM001_F4          */
/* EXFMT SCR1          TM001_F1          */
/* EXFMT SCR2          TM001_F3          */
/* EXFMT SCR2D          TM001_F2          */
/*****/

```

```

/*=====*/
/* COPY IN INCLUDE:TM001_99=====*/
/*=====*/

```

```

/*-----*/
/*---          PRTM001          (TM001_99)          ---*/
/*---          FROM LIB: RLS94          FILE: QSRC3          MBR: PRTM001          ---*/
/*---          CONV.BY: PRISMDAVE          ---*/
/*--- SECTION TYPE : VARIABLE DEFINITIONS          ---*/
/*--- Generated PROGRESS via MAKEPROGRESS          Release 1996.02.09          ---*/
/*--- (C) 1993 by EFA Software Services Ltd.          2/13/2006          ---*/
/*-----*/

```

```

DEFINE NEW SHARED VARIABLE IN_IN AS CHAR FORMAT "X" EXTENT 99 INITIAL [ "0" ] NO-UNDO.
DEFINE NEW SHARED VARIABLE DS_WK1 AS CHAR NO-UNDO.
DEFINE NEW SHARED VARIABLE DS_WK2 AS CHAR NO-UNDO.
DEFINE NEW SHARED VARIABLE WK_X AS CHAR NO-UNDO.
DEFINE NEW SHARED VARIABLE DO_COUNT AS INTEGER NO-UNDO.
DEFINE NEW SHARED VARIABLE ARR_I AS INTEGER NO-UNDO.
DEFINE NEW SHARED VARIABLE ARR_J AS INTEGER NO-UNDO.
DEFINE NEW SHARED VARIABLE ARR_K AS INTEGER NO-UNDO.
DEFINE NEW SHARED VARIABLE ARR_I2 AS INTEGER NO-UNDO.
DEFINE NEW SHARED VARIABLE S_RETRN AS CHAR NO-UNDO.
DEFINE NEW SHARED BUFFER PB001R FOR PB001.
DEFINE NEW SHARED VARIABLE PB001R_RECID AS RECID NO-UNDO.
DEFINE NEW SHARED VARIABLE PB001R_LAST_OP AS CHAR NO-UNDO.
DEFINE NEW SHARED BUFFER PB006R FOR PB006.
DEFINE NEW SHARED VARIABLE PB006R_RECID AS RECID NO-UNDO.
DEFINE NEW SHARED VARIABLE PB006R_LAST_OP AS CHAR NO-UNDO.
DEFINE NEW SHARED BUFFER PCTL1R FOR PCTL1.
DEFINE NEW SHARED VARIABLE PCTL1R_LAST_OP AS CHAR NO-UNDO.
DEFINE NEW SHARED BUFFER P0004R FOR P0004.
DEFINE NEW SHARED VARIABLE P0004R_LAST_OP AS CHAR NO-UNDO.
DEFINE NEW SHARED BUFFER PTT02R FOR PTT02.
DEFINE NEW SHARED VARIABLE PTT02R_RECID AS RECID NO-UNDO.
DEFINE NEW SHARED VARIABLE PTT02R_LAST_OP AS CHAR NO-UNDO.
DEFINE NEW SHARED BUFFER PZ001R FOR PZ001.
DEFINE NEW SHARED VARIABLE PZ001R_RECID AS RECID NO-UNDO.
DEFINE NEW SHARED VARIABLE PZ001R_LAST_OP AS CHAR NO-UNDO.
DEFINE NEW SHARED BUFFER PZ002R FOR PZ002.
DEFINE NEW SHARED VARIABLE PZ002R_LAST_OP AS CHAR NO-UNDO.
DEFINE NEW SHARED VARIABLE Q_$$DUM AS CHAR NO-UNDO.
DEFINE NEW SHARED VARIABLE Q_$$ERR# AS CHAR NO-UNDO.
DEFINE NEW SHARED VARIABLE IN_KA AS CHAR INITIAL [ "0" ] NO-UNDO.
DEFINE NEW SHARED VARIABLE IN_KC AS CHAR INITIAL [ "0" ] NO-UNDO.
DEFINE NEW SHARED VARIABLE IN_KL AS CHAR INITIAL [ "0" ] NO-UNDO.
DEFINE NEW SHARED VARIABLE IN_KD AS CHAR INITIAL [ "0" ] NO-UNDO.
DEFINE NEW SHARED VARIABLE IN_KE AS CHAR INITIAL [ "0" ] NO-UNDO.
DEFINE NEW SHARED VARIABLE IN_KF AS CHAR INITIAL [ "0" ] NO-UNDO.
DEFINE NEW SHARED VARIABLE IN_KH AS CHAR INITIAL [ "0" ] NO-UNDO.
DEFINE NEW SHARED VARIABLE IN_KI AS CHAR INITIAL [ "0" ] NO-UNDO.
DEFINE NEW SHARED VARIABLE IN_LR AS CHAR INITIAL [ "0" ] NO-UNDO.
DEFINE NEW SHARED VARIABLE IN_LRT AS CHAR INITIAL [ "0" ] NO-UNDO.
DEFINE NEW SHARED VARIABLE Q_#NUM AS CHAR NO-UNDO.
DEFINE NEW SHARED VARIABLE Q_#BNAM AS CHAR NO-UNDO.
DEFINE NEW SHARED VARIABLE Q_#BNUM AS CHAR NO-UNDO.
DEFINE NEW SHARED VARIABLE Q_#CTP AS CHAR NO-UNDO.
DEFINE NEW SHARED VARIABLE Q_#CTPT AS CHAR NO-UNDO.
DEFINE NEW SHARED VARIABLE Q_#DATE AS CHAR NO-UNDO.
DEFINE NEW SHARED VARIABLE Q_#DELYN AS CHAR NO-UNDO.
DEFINE NEW SHARED VARIABLE Q_#DSID AS CHAR NO-UNDO.

```

```

DEFINE NEW SHARED VARIABLE Q_#DSIDT AS CHAR NO-UNDO.
DEFINE NEW SHARED VARIABLE Q_#GGS AS CHAR NO-UNDO.
DEFINE NEW SHARED VARIABLE Q_#PCOD AS CHAR NO-UNDO.
DEFINE NEW SHARED VARIABLE Q_#REN AS CHAR NO-UNDO.
DEFINE NEW SHARED VARIABLE Q_#USER AS CHAR NO-UNDO.
DEFINE NEW SHARED VARIABLE Q_#WNAM AS CHAR NO-UNDO.
DEFINE NEW SHARED VARIABLE Q_#WNUM AS CHAR NO-UNDO.
DEFINE NEW SHARED VARIABLE B1PNM1 AS CHAR NO-UNDO.
DEFINE NEW SHARED VARIABLE B1PNM2 AS CHAR NO-UNDO.
DEFINE NEW SHARED VARIABLE B1PNM3 AS CHAR NO-UNDO.
DEFINE NEW SHARED VARIABLE CHKQ AS CHAR NO-UNDO.
DEFINE NEW SHARED VARIABLE CTL AS CHAR NO-UNDO.
DEFINE NEW SHARED VARIABLE DTA AS CHAR NO-UNDO.
DEFINE NEW SHARED VARIABLE ERR# AS CHAR NO-UNDO.
DEFINE NEW SHARED VARIABLE FACOPR AS CHAR NO-UNDO.
DEFINE NEW SHARED VARIABLE FLD# AS CHAR NO-UNDO.
DEFINE NEW SHARED VARIABLE LDA AS CHAR NO-UNDO.
DEFINE NEW SHARED VARIABLE NEWFLD AS CHAR NO-UNDO.
DEFINE NEW SHARED VARIABLE PROVA AS CHAR NO-UNDO.
DEFINE NEW SHARED VARIABLE Q AS CHAR NO-UNDO.
DEFINE NEW SHARED VARIABLE RUL# AS CHAR NO-UNDO.
DEFINE NEW SHARED VARIABLE SVNBAT AS CHAR NO-UNDO.
DEFINE NEW SHARED VARIABLE TIME14 AS DECIMAL DECIMALS 00 NO-UNDO.
DEFINE NEW SHARED VARIABLE UDATX AS CHAR NO-UNDO.
DEFINE NEW SHARED VARIABLE UDATZ AS INTEGER NO-UNDO.
DEFINE NEW SHARED VARIABLE UTIME AS INTEGER NO-UNDO.
DEFINE NEW SHARED VARIABLE VAL# AS CHAR NO-UNDO.
DEFINE NEW SHARED VARIABLE VARNAM AS CHAR NO-UNDO.
DEFINE NEW SHARED VARIABLE WHOAMI AS CHAR NO-UNDO.
DEFINE NEW SHARED VARIABLE XCCYY AS CHAR NO-UNDO.
DEFINE NEW SHARED VARIABLE XCTO AS CHAR NO-UNDO.
DEFINE NEW SHARED VARIABLE XFAX1 AS INTEGER NO-UNDO.
DEFINE NEW SHARED VARIABLE XFAX2 AS INTEGER NO-UNDO.
DEFINE NEW SHARED VARIABLE XFAX3 AS INTEGER NO-UNDO.
DEFINE NEW SHARED VARIABLE XMMDD AS CHAR NO-UNDO.
DEFINE NEW SHARED VARIABLE XOID AS CHAR NO-UNDO.
DEFINE NEW SHARED VARIABLE XPAPPL AS CHAR NO-UNDO.
DEFINE NEW SHARED VARIABLE XPDESC AS CHAR NO-UNDO.
DEFINE NEW SHARED VARIABLE XPMSG AS CHAR NO-UNDO.
DEFINE NEW SHARED VARIABLE XPERR# AS CHAR NO-UNDO.
DEFINE NEW SHARED VARIABLE XPFLD AS CHAR NO-UNDO.
DEFINE NEW SHARED VARIABLE XPRAF AS CHAR NO-UNDO.
DEFINE NEW SHARED VARIABLE XPRCA AS CHAR NO-UNDO.
DEFINE NEW SHARED VARIABLE XPRCO AS CHAR NO-UNDO.
DEFINE NEW SHARED VARIABLE XPRCV AS CHAR NO-UNDO.
DEFINE NEW SHARED VARIABLE XPRULE AS CHAR NO-UNDO.
DEFINE NEW SHARED VARIABLE XPTABV AS CHAR NO-UNDO.
DEFINE NEW SHARED VARIABLE XXBNUM AS CHAR NO-UNDO.
/*=====*/

```

```

DEFINE BUTTON S_0016 AUTO-GO LABEL "Exit" TRIGGERS:
ON CHOOSE ASSIGN
    IN_KC = "1".

ON GO ASSIGN
    IN_KC = "1".

END.

```

```

/*=====*/
/*=====*/

```

```

DEFINE BUTTON S_0017 AUTO-GO LABEL "Confirm" TRIGGERS:
ON CHOOSE ASSIGN
    IN_KD = "1".

ON GO ASSIGN
    IN_KD = "1".

END.

```

```

/*=====*/
/*=====*/

```

```

DEFINE BUTTON S_0018 AUTO-GO LABEL "Pr Screen" TRIGGERS:
ON CHOOSE ASSIGN
    IN_KL = "1".

ON GO ASSIGN
    IN_KL = "1".

END.

```

```

/*=====*/
/*=====*/

```

```

DEFINE BUTTON S_0002 AUTO-GO LABEL "Exit" TRIGGERS:

```

```

ON CHOOSE ASSIGN
    IN_KC = "1".

ON GO ASSIGN
    IN_KC = "1".

END.

/*=====*/
DEFINE NEW SHARED VARIABLE S_0003 AS CHAR FORMAT "X(016)" INITIAL [ "Government Codes" ] VIEW-AS TEXT
NO-UNDO.

DEFINE NEW SHARED VARIABLE S_0004 AS CHAR FORMAT "X(017)" INITIAL [ "Report Signatures" ] VIEW-AS TEXT
NO-UNDO.

/*=====*/

DEFINE BUTTON S_0005 AUTO-GO LABEL "Exit" TRIGGERS:
ON CHOOSE ASSIGN
    IN_KC = "1".

ON GO ASSIGN
    IN_KC = "1".

END.

/*=====*/
/*=====*/

DEFINE BUTTON S_0006 AUTO-GO LABEL "Paste" TRIGGERS:
ON CHOOSE ASSIGN
    IN_KH = "1".

ON GO ASSIGN
    IN_KH = "1".

END.

/*=====*/
/*=====*/

DEFINE BUTTON S_0007 AUTO-GO LABEL "Copy" TRIGGERS:
ON CHOOSE ASSIGN
    IN_KI = "1".

ON GO ASSIGN
    IN_KI = "1".

END.

/*=====*/
/*=====*/

DEFINE BUTTON S_0008 AUTO-GO LABEL "Guide 60" TRIGGERS:
ON CHOOSE ASSIGN
    IN_KF = "1".

ON GO ASSIGN
    IN_KF = "1".

END.

/*=====*/
/*=====*/

DEFINE BUTTON S_0010 AUTO-GO LABEL "Exit" TRIGGERS:
ON CHOOSE ASSIGN
    IN_KC = "1".

ON GO ASSIGN
    IN_KC = "1".

END.

/*=====*/
/*=====*/

DEFINE BUTTON S_0011 AUTO-GO LABEL "Delete" TRIGGERS:
ON CHOOSE ASSIGN
    IN_KD = "1".

ON GO ASSIGN
    IN_KD = "1".

END.

/*=====*/

```

```
/*=====*/
DEFINE BUTTON S_0012 AUTO-GO LABEL "Paste" TRIGGERS:
ON CHOOSE ASSIGN
    IN_KH = "1".

ON GO ASSIGN
    IN_KH = "1".

END.
```

```
/*=====*/
/*=====*/
DEFINE BUTTON S_0013 AUTO-GO LABEL "Copy" TRIGGERS:
ON CHOOSE ASSIGN
    IN_KI = "1".

ON GO ASSIGN
    IN_KI = "1".

END.
```

```
/*=====*/
/*=====*/
DEFINE BUTTON S_0014 AUTO-GO LABEL "Guide 60" TRIGGERS:
ON CHOOSE ASSIGN
    IN_KF = "1".

ON GO ASSIGN
    IN_KF = "1".

END.
```

```
/*=====*/
/*=====*/
/*===== STANDARD WINDOW CONTROL AND MENU CONTROL =====*/
DEFINE NEW SHARED VARIABLE PRMT001 AS WIDGET-HANDLE NO-UNDO.
{STDINC01.I "PRMT001 " " Maintain Battery ID Master " " 19"}
```

```
.
{STDINC02.I "NEW"}
```

```
/*=====*/
DEFINE NEW SHARED VARIABLE B1NUM LIKE PB001R.B1NUM NO-UNDO.
DEFINE NEW SHARED VARIABLE B1NAME LIKE PB001R.B1NAME NO-UNDO.
DEFINE NEW SHARED VARIABLE B1OPER LIKE PB001R.B1OPER NO-UNDO.
DEFINE NEW SHARED VARIABLE B1FLD LIKE PB001R.B1FLD NO-UNDO.
DEFINE NEW SHARED VARIABLE B1GBAT LIKE PB001R.B1GBAT NO-UNDO.
DEFINE NEW SHARED VARIABLE B1NOPR LIKE PB001R.B1NOPR NO-UNDO.
DEFINE NEW SHARED VARIABLE B1NBAT LIKE PB001R.B1NBAT NO-UNDO.
DEFINE NEW SHARED VARIABLE B1S1 LIKE PB001R.B1S1 NO-UNDO.
DEFINE NEW SHARED VARIABLE B1S2 LIKE PB001R.B1S2 NO-UNDO.
DEFINE NEW SHARED VARIABLE B1SNAM LIKE PB001R.B1SNAM NO-UNDO.
DEFINE NEW SHARED VARIABLE B1XREF LIKE PB001R.B1XREF NO-UNDO.
DEFINE NEW SHARED VARIABLE B1PROV LIKE PB001R.B1PROV NO-UNDO.
DEFINE NEW SHARED VARIABLE B1GESQ LIKE PB001R.B1GESQ NO-UNDO.
DEFINE NEW SHARED VARIABLE B1GLCX LIKE PB001R.B1GLCX NO-UNDO.
DEFINE NEW SHARED VARIABLE B1GLSD LIKE PB001R.B1GLSD NO-UNDO.
DEFINE NEW SHARED VARIABLE B1GMER LIKE PB001R.B1GMER NO-UNDO.
DEFINE NEW SHARED VARIABLE B1GRNG LIKE PB001R.B1GRNG NO-UNDO.
DEFINE NEW SHARED VARIABLE B1GSCT LIKE PB001R.B1GSCT NO-UNDO.
DEFINE NEW SHARED VARIABLE B1GTWN LIKE PB001R.B1GTWN NO-UNDO.
DEFINE NEW SHARED VARIABLE B1OPR5 LIKE PB001R.B1OPR5 VIEW-AS TEXT NO-UNDO.
DEFINE NEW SHARED VARIABLE B1REF2 LIKE PB001R.B1REF2 NO-UNDO.
DEFINE NEW SHARED VARIABLE B1TXT1 LIKE PB001R.B1TXT1 NO-UNDO.
DEFINE NEW SHARED VARIABLE B1TXT2 LIKE PB001R.B1TXT2 NO-UNDO.
DEFINE NEW SHARED VARIABLE B1PNUM LIKE PB001R.B1PNUM NO-UNDO.
DEFINE NEW SHARED VARIABLE B1TYPE LIKE PB001R.B1TYPE NO-UNDO.
DEFINE NEW SHARED VARIABLE B6DATE LIKE PB006R.B6DATE NO-UNDO.
DEFINE NEW SHARED VARIABLE B6TYPE LIKE PB006R.B6TYPE NO-UNDO.
DEFINE NEW SHARED VARIABLE B6ID LIKE PB006R.B6ID NO-UNDO.
DEFINE NEW SHARED VARIABLE B6GOVT LIKE PB006R.B6GOVT NO-UNDO.
DEFINE NEW SHARED VARIABLE CTL#0B LIKE PCTL1R.CTL#0B NO-UNDO.
DEFINE NEW SHARED VARIABLE CTL#01 LIKE PCTL1R.CTL#01 NO-UNDO.
DEFINE NEW SHARED VARIABLE CTL#30 LIKE PCTL1R.CTL#30 VIEW-AS TEXT NO-UNDO.
DEFINE NEW SHARED VARIABLE CTL#31 LIKE PCTL1R.CTL#31 VIEW-AS TEXT NO-UNDO.
DEFINE NEW SHARED VARIABLE CTL#93 LIKE PCTL1R.CTL#93 NO-UNDO.
DEFINE NEW SHARED VARIABLE CTL#94 LIKE PCTL1R.CTL#94 NO-UNDO.
DEFINE NEW SHARED VARIABLE O4GOVT LIKE P0004R.O4GOVT NO-UNDO.
DEFINE NEW SHARED VARIABLE O4DATE LIKE P0004R.O4DATE NO-UNDO.
DEFINE NEW SHARED VARIABLE T2FLD LIKE PTT02R.T2FLD NO-UNDO.
DEFINE NEW SHARED VARIABLE T2RL1C LIKE PTT02R.T2RL1C NO-UNDO.
```

```

DEFINE NEW SHARED VARIABLE T2ENVC LIKE PTT02R.T2ENVC NO-UNDO.
DEFINE NEW SHARED VARIABLE T2TBEV LIKE PTT02R.T2TBEV NO-UNDO.
DEFINE NEW SHARED VARIABLE T2TBED LIKE PTT02R.T2TBED NO-UNDO.
DEFINE NEW SHARED VARIABLE T2RET LIKE PTT02R.T2RET NO-UNDO.
DEFINE NEW SHARED VARIABLE T2SORT LIKE PTT02R.T2SORT NO-UNDO.
DEFINE NEW SHARED VARIABLE Z1TYPE LIKE PZ001R.Z1TYPE NO-UNDO.
DEFINE NEW SHARED VARIABLE Z1ID LIKE PZ001R.Z1ID NO-UNDO.
DEFINE NEW SHARED VARIABLE Z1TEXT LIKE PZ001R.Z1TEXT NO-UNDO.
DEFINE NEW SHARED VARIABLE Z1XREF LIKE PZ001R.Z1XREF NO-UNDO.
DEFINE NEW SHARED VARIABLE Z1GOVT LIKE PZ001R.Z1GOVT NO-UNDO.
DEFINE NEW SHARED VARIABLE Z1REF2 LIKE PZ001R.Z1REF2 NO-UNDO.
DEFINE NEW SHARED VARIABLE Z1TXT1 LIKE PZ001R.Z1TXT1 NO-UNDO.
DEFINE NEW SHARED VARIABLE Z1TXT2 LIKE PZ001R.Z1TXT2 NO-UNDO.
DEFINE NEW SHARED VARIABLE Z1OPER LIKE PZ001R.Z1OPER NO-UNDO.
DEFINE NEW SHARED VARIABLE Z1CON LIKE PZ001R.Z1CON NO-UNDO.
DEFINE NEW SHARED VARIABLE Z1PNUM LIKE PZ001R.Z1PNUM NO-UNDO.
DEFINE NEW SHARED VARIABLE Z1DCOD LIKE PZ001R.Z1DCOD NO-UNDO.
DEFINE NEW SHARED VARIABLE Z1FAX1 LIKE PZ001R.Z1FAX1 NO-UNDO.
DEFINE NEW SHARED VARIABLE Z1FAX2 LIKE PZ001R.Z1FAX2 NO-UNDO.
DEFINE NEW SHARED VARIABLE Z1FAX3 LIKE PZ001R.Z1FAX3 NO-UNDO.
DEFINE NEW SHARED VARIABLE Z1OID LIKE PZ001R.Z1OID NO-UNDO.
DEFINE NEW SHARED VARIABLE Z1EUB LIKE PZ001R.Z1EUB NO-UNDO.
DEFINE NEW SHARED VARIABLE Z1INTF LIKE PZ001R.Z1INTF NO-UNDO.
DEFINE NEW SHARED VARIABLE Z1REF3 LIKE PZ001R.Z1REF3 NO-UNDO.
DEFINE NEW SHARED VARIABLE Z1CTO LIKE PZ001R.Z1CTO NO-UNDO.
DEFINE NEW SHARED VARIABLE Z1PRAF LIKE PZ001R.Z1PRAF NO-UNDO.
DEFINE NEW SHARED VARIABLE Z1PRCV LIKE PZ001R.Z1PRCV NO-UNDO.
DEFINE NEW SHARED VARIABLE Z1PRCO LIKE PZ001R.Z1PRCO NO-UNDO.
DEFINE NEW SHARED VARIABLE Z1PRCA LIKE PZ001R.Z1PRCA NO-UNDO.
DEFINE NEW SHARED VARIABLE Z1RMS LIKE PZ001R.Z1RMS NO-UNDO.
DEFINE NEW SHARED VARIABLE Z1LEGA LIKE PZ001R.Z1LEGA NO-UNDO.
DEFINE NEW SHARED VARIABLE Z1SLPT LIKE PZ001R.Z1SLPT NO-UNDO.
DEFINE NEW SHARED VARIABLE Z1USER LIKE PZ001R.Z1USER NO-UNDO.
DEFINE NEW SHARED VARIABLE Z1UDAT LIKE PZ001R.Z1UDAT NO-UNDO.
DEFINE NEW SHARED VARIABLE Z1UTIM LIKE PZ001R.Z1UTIM NO-UNDO.
DEFINE NEW SHARED VARIABLE Z1EOSF LIKE PZ001R.Z1EOSF NO-UNDO.
DEFINE NEW SHARED VARIABLE Z1CALC LIKE PZ001R.Z1CALC NO-UNDO.
DEFINE NEW SHARED VARIABLE Z1RECP LIKE PZ001R.Z1RECP NO-UNDO.
DEFINE NEW SHARED VARIABLE Z2ERR LIKE PZ002R.Z2ERR NO-UNDO.
DEFINE NEW SHARED VARIABLE DOU_01 AS CHAR NO-UNDO.
PAUSE 0 BEFORE-HIDE.
DEFINE INPUT-OUTPUT PARAMETER PARM_Q_#DELYN LIKE Q_#DELYN NO-UNDO.
/*=====*/
ASSIGN
  Q_#DELYN = PARM_Q_#DELYN.

/* F*-----*/
/* F*-
/* F*-   PRM001 - BATTERY MASTER ID MAINTENANCE
/* F*-
/* F*-   APR 01 89 KJH
/* F*-
/* F*-   MODIFICATIONS:
/* F*-   JAN 30 98 JB - Add System Parameter to see if Operator
/* F*-                   Code is Mandatory
/* F*-
/* F*-   MAR 16 99 JB - Add Operator Fax Number for Saskatchewan
/* F*---
/* F*--- AUG 01 02 AE - MAKE GOVERNMENT CODE DATE SPECIFIC
/* F*-
/* F*--- AUG 01 03 AE - ADD BATTERY TYPE (OIL GAS) B1TYPE
/* F*-
/* F*-                   ADD GUIDE 60 INQUIRY ON F6
/* F*-                   GUIDE 60 ONLY APPLICABLE IN AB - IF OTHER
/* F*-                   PROVINCES TRY TO USE F6(GUIDE 60), DISPLAY
/* F*-                   MESSAGE
/* F*-   OCT 18 01 GAS- SAVE REF2 INFO IN PZ001
/* F*-   DEC 04 01 GAS- IF GOVT. BATTERY NUMBER CHANGES ON PB001 UPDATE
/* F*-                   THEN CHANGE APPROPRIATE RECORDS IN PB006
/* F*-
/* F*-   02 05 21 - DP - ADD PRA FACILTIY INDICATOR
/* F*-
/* F*-   02 09 03 - DP - SPLIT PRA PROCESSING TO OAF AND VOLUMETRICS
/* F*-   02 11 22 - DP - ALLOW 'W' TYPE BATTERIES
/* F*-   03 01 29 - DP - ADD APMC PROCESSING FLAG
/* F*-   DP 09 2003 - Add User Date Time Stamp on Update Write
/* F*-   JB 02 2004 - Add Custody Transfer Override
/* F*-   2004 12 15 - KM ADDED CONFIRM DELETE SCREEN WHICH INCLUDES WARNING
/* F*-                   MESSAGE ON CONSEQUENCES OF DELETING AN ID MASTER
/* F*-   2006 02 10 DWP - REMOVE THE PRA PRAFAC CODE FOR PTT02,
/* F*-                   REPLACED BY PZ001 FLAGS (MOD 15033)
/* F*-----*/
/*=====*/
/* COPY IN INCLUDE:TM001_S1=====*/
/*=====*/
/*-----*/
/*---          PRM001          (TM001_S1)          ---*/
/*--- Generated PROGRESS via MAKEPROGRESS          ---*/

```

```

/*--- (C) 1993 by EFA Software Services Ltd.                2/13/2006 ---*/
/*-----*/
ON RETURN GO.
DEFINE NEW SHARED FRAME SCR3.
FORM
/*=====*/

/*===STANDARD XTRA BUTTONS,FOLLOWED BY: BUTTONS POSITION===*/
{STDINC05.I "SCR3"}

S_0017 AT ROW 1.1
/*3*/
COL 31 "**** Warning - You have selected to delete the Battery Master ID" AT
12.15 VIEW-AS TEXT "record, by doing so, Production Records for any month cannot be" AT
12.15 VIEW-AS TEXT "accessed Press Confirm, to verify delete " AT 12.15 VIEW-AS TEXT
"
" Battery" AT 18.90 VIEW-AS TEXT B1NUM AT 29.70 FORMAT "X(5)" VIEW-AS TEXT B1NAME AT 37.80 FORMAT
"X(36)" VIEW-AS TEXT
/*= ABOUT TO COMPLETE FRAME.. KJH*/

/*= ADD SOME THING AT END SO ALL FRAMES SAME SIZE =====*/
{STDINC07.I "19"}

WITH{STDINC16.I}

NO-LABELS THREE-D WIDTH 108.00 ROW 1 FRAME SCR3.
ASSIGN FRAME
SCR3:MOVABLE = FALSE FRAME
SCR3:SCROLLABLE = TRUE FRAME
SCR3:RESIZABLE = FALSE FRAME
SCR3:FONT = 8.

/*-----*/
DEFINE NEW SHARED FRAME SCR1.
FORM
/*=====*/

/*===STANDARD XTRA BUTTONS,FOLLOWED BY: BUTTONS POSITION===*/
{STDINC05.I "SCR1"}

" Battery ID " AT 2.70 VIEW-AS TEXT B1NUM AT 18.90 FORMAT "X(5)"
/*= ABOUT TO COMPLETE FRAME.. KJH*/

/*= ADD SOME THING AT END SO ALL FRAMES SAME SIZE =====*/
{STDINC07.I "19"}

WITH{STDINC16.I}

NO-LABELS THREE-D WIDTH 108.00 ROW 1 FRAME SCR1.
ASSIGN FRAME
SCR1:MOVABLE = FALSE FRAME
SCR1:SCROLLABLE = TRUE FRAME
SCR1:RESIZABLE = FALSE FRAME
SCR1:FONT = 8.

/*-----*/
DEFINE NEW SHARED FRAME SCR2.
FORM
/*=====*/

/*===STANDARD XTRA BUTTONS,FOLLOWED BY: BUTTONS POSITION===*/
{STDINC05.I "SCR2"}

S_0006 AT ROW 1.1
/*3*/
COL 31 S_0007 AT ROW 1.1
/*3*/
COL 39 S_0008 AT ROW 1.1
/*3*/
COL 46 "Battery Id " AT 2.70 VIEW-AS TEXT
B1NUM AT 18.90 FORMAT "X(5)" VIEW-AS TEXT NEWFLD AT 33.75 FORMAT "X(3)" VIEW-AS TEXT
"Last Saved for Copying " AT 59.40 VIEW-AS TEXT XXBNUM AT 91.80 FORMAT "X(5)" VIEW-AS TEXT
"Province " AT 2.70 VIEW-AS TEXT B1PROV AT 37.80 FORMAT "X(1)" {STDINC24.I}

PROVA AT 41.85 FORMAT "X(16)" VIEW-AS TEXT S_0003 AT 2.70 VIEW-AS TEXT "Battery Number " AT
5.40 VIEW-AS TEXT B1NBAT AT 37.80 FORMAT "X(7)" "Battery Operator " AT 5.40 VIEW-AS TEXT
Z1OID AT 37.80 FORMAT "X(8)" B1OPR5 AT 51.30 FORMAT "X(5)" VIEW-AS TEXT "Search Name " AT 89.10
VIEW-AS TEXT "Battery Name " AT 5.40 VIEW-AS TEXT B1NAME AT 37.80 FORMAT "X(36)" B1SNAM
AT 89.10 FORMAT "X(10)" "Type (Gas/Oil/Water) " AT 5.40 VIEW-AS TEXT B1TYPE AT 37.80 FORMAT

```

"X(1)">{STDINC24.I}

"(G/O/W)" AT 41.85 VIEW-AS TEXT "Location" AT 5.40 VIEW-AS TEXT "Sub" AT 20.25 VIEW-AS TEXT "Event Seq" AT 75.60 VIEW-AS TEXT "Exception" AT 5.40 VIEW-AS TEXT "Division" AT 20.25 VIEW-AS TEXT "Section" AT 32.40 VIEW-AS TEXT "Township" AT 43.20 VIEW-AS TEXT "Range" AT 55.35 VIEW-AS TEXT "Meridian" AT 63.45 VIEW-AS TEXT "(zone)" AT 75.60 VIEW-AS TEXT B1GLCX AT 9.45 FORMAT ">>" B1GLSD AT 22.95 FORMAT ">>" B1GSCT AT 35.10 FORMAT ">>" B1GTWN AT 45.90 FORMAT ">>>" B1GRNG AT 56.70 FORMAT ">>" B1GMER AT 67.50 FORMAT ">"{STDINC24.I}

B1GESQ AT 78.30 FORMAT ">"{STDINC24.I}

S\_0004 AT 2.70 VIEW-AS TEXT "Wells (S1/EM18)" AT 9.45 VIEW-AS TEXT B1S1 AT 37.80 FORMAT "X(30)" "Battery (S2/EM61)" AT 9.45 VIEW-AS TEXT B1S2 AT 37.80 FORMAT "X(30)" "Phone" AT 9.45 VIEW-AS TEXT "Number" AT 17.55 VIEW-AS TEXT B1PNM1 AT 37.80 FORMAT "X(3)" B1PNM2 AT 45.90 FORMAT "X(3)" B1PNM3 AT 54.00 FORMAT "X(4)" "Fax Number" AT 63.45 VIEW-AS TEXT Z1FAX1 AT 79.65 FORMAT "999" Z1FAX2 AT 87.75 FORMAT "999" Z1FAX3 AT 95.85 FORMAT "9999" CTL#30 AT 9.45 FORMAT "X(20)" VIEW-AS TEXT B1XREF AT 37.80 FORMAT "X(15)" CTL#31 AT 9.45 FORMAT "X(20)" VIEW-AS TEXT B1REF2 AT 37.80 FORMAT "X(15)" "PRA Facility (Y/N)" AT 9.45 VIEW-AS TEXT Z1PRAF AT 37.80 FORMAT "X(1)">{STDINC24.I}

"Process OAF/SAF" AT 43.20 VIEW-AS TEXT Z1PRCV AT 66.15 FORMAT "X(1)">{STDINC24.I}

"Volumetrics" AT 71.55 VIEW-AS TEXT Z1PRCO AT 89.10 FORMAT "X(1)">{STDINC24.I}

"APMC" AT 94.50 VIEW-AS TEXT Z1PRCA AT 102.60 FORMAT "X(1)">{STDINC24.I}

"APMC Custody Transfer Override" AT 9.45 VIEW-AS TEXT Z1CTO AT 52.65 FORMAT "X(11)"  
/\*= ABOUT TO COMPLETE FRAME.. KJH\*/

/\*= ADD SOME THING AT END SO ALL FRAMES SAME SIZE =====\*/  
{STDINC07.I  
"19"}

WITH{STDINC16.I}

NO-LABELS THREE-D WIDTH 108.00 ROW 1 FRAME SCR2.

ASSIGN FRAME  
SCR2:MOVABLE = FALSE FRAME  
SCR2:SCROLLABLE = TRUE FRAME  
SCR2:RESIZABLE = FALSE FRAME  
SCR2:FONT = 8.

/\*-----\*/  
DEFINE NEW SHARED FRAME SCR2D.  
FORM  
/\*=====\*/  
/\*===STANDARD XTRA BUTTONS,FOLLOWED BY: BUTTONS POSITION===\*/  
{STDINC05.I "SCR2D"}

S\_0011 AT ROW 1.1  
/\*3\*/  
COL 31 S\_0012 AT ROW 1.1  
/\*3\*/  
COL 40 S\_0013 AT ROW 1.1  
/\*3\*/  
COL 48 S\_0014 AT ROW 1.1  
/\*3\*/  
COL 55 "Battery Id"  
AT 2.70 VIEW-AS TEXT B1NUM AT 18.90 FORMAT "X(5)" VIEW-AS TEXT NEWFLD AT 33.75 FORMAT "X(3)" VIEW-AS TEXT "Last Saved for Copying" AT 59.40 VIEW-AS TEXT XXBNUM AT 91.80 FORMAT "X(5)" VIEW-AS TEXT "Province" AT 2.70 VIEW-AS TEXT B1PROV AT 37.80 FORMAT "X(1)">{STDINC24.I}

PROVA AT 41.85 FORMAT "X(16)" VIEW-AS TEXT S\_0003 AT 2.70 VIEW-AS TEXT "Battery Number" AT 5.40 VIEW-AS TEXT B1NBAT AT 37.80 FORMAT "X(7)" "Battery Operator" AT 5.40 VIEW-AS TEXT Z1OID AT 37.80 FORMAT "X(8)" B1OPR5 AT 51.30 FORMAT "X(5)" VIEW-AS TEXT "Search Name" AT 89.10 VIEW-AS TEXT "Battery Name" AT 5.40 VIEW-AS TEXT B1NAME AT 37.80 FORMAT "X(36)" B1SNAM AT 89.10 FORMAT "X(10)" "Type (Gas/Oil/Water)" AT 5.40 VIEW-AS TEXT B1TYPE AT 37.80 FORMAT "X(1)">{STDINC24.I}

"(G/O/W)" AT 41.85 VIEW-AS TEXT "Location" AT 5.40 VIEW-AS TEXT "Sub" AT 20.25 VIEW-AS TEXT "Event Seq" AT 75.60 VIEW-AS TEXT "Exception" AT 5.40 VIEW-AS TEXT "Division" AT 20.25 VIEW-AS TEXT "Section" AT 32.40 VIEW-AS TEXT "Township" AT 43.20 VIEW-AS TEXT "Range" AT 55.35 VIEW-AS TEXT "Meridian" AT 63.45 VIEW-AS TEXT "(zone)" AT 75.60 VIEW-AS TEXT B1GLCX AT 9.45 FORMAT ">>" B1GLSD AT 22.95 FORMAT ">>" B1GSCT AT 35.10 FORMAT ">>" B1GTWN AT 45.90 FORMAT ">>>" B1GRNG AT 56.70 FORMAT ">>" B1GMER AT 67.50 FORMAT ">"{STDINC24.I}

B1GESQ AT 78.30 FORMAT ">"{STDINC24.I}

S\_0004 AT 2.70 VIEW-AS TEXT "Wells (S1/EM18) " AT 9.45 VIEW-AS TEXT B1S1 AT 37.80 FORMAT "X(30)"  
"Battery (S2/EM61) " AT 9.45 VIEW-AS TEXT B1S2 AT 37.80 FORMAT "X(30)" "Phone Number "  
AT 9.45 VIEW-AS TEXT B1PNM1 AT 37.80 FORMAT "X(3)" B1PNM2 AT 45.90 FORMAT "X(3)" B1PNM3 AT 54.00 FORMAT  
"X(4)" "Fax Number " AT 9.45 VIEW-AS TEXT Z1FAX1 AT 37.80 FORMAT "999" Z1FAX2 AT 45.90  
FORMAT "999" Z1FAX3 AT 54.00 FORMAT "9999" CTL#30 AT 9.45 FORMAT "X(20)" VIEW-AS TEXT B1XREF AT 37.80  
FORMAT "X(15)" CTL#31 AT 9.45 FORMAT "X(20)" VIEW-AS TEXT B1REF2 AT 37.80 FORMAT "X(15)"  
/\*= ABOUT TO COMPLETE FRAME.. KJH\*/

/\*= ADD SOME THING AT END SO ALL FRAMES SAME SIZE =====\*/  
{STDINC07.I  
"19"}

WITH{STDINC16.I}

NO-LABELS THREE-D WIDTH 108.00 ROW 1 FRAME SCR2D.

ASSIGN FRAME  
SCR2D:MOVABLE = FALSE FRAME  
SCR2D:SCROLLABLE = TRUE FRAME  
SCR2D:RESIZABLE = FALSE FRAME  
SCR2D:FONT = 8.

/\*-----\*/  
/\*=====\*/  
{STDINC11.I}

/\* F\* COPY QSRC,PRCM906 \*/  
/\* F\*-----\*/  
/\* F\* AE 01 08 02 \*/  
/\* F\*PO004A IF E K DISK \*/  
/\* F\* AE 01 08 02 \*/  
/\* F\*DTA1 IF E DISK \*/  
/\* F\*-----\*/  
/\* F\*--- GET DATE USER TIME FOR FILE UPDATE STAMP \*/  
/\* F\*-----\*/  
/\* F\*-----\*/  
/\* I\*-----\*/  
/\* I\* COPY QSRC,PRCM900 \*/  
/\* I\*-----\*/  
/\* I\*- PRCM900 - PRODUCTION REVENUE \*/  
/\* I\*- - COPY MODULE \*/  
/\* I\*- APR 20 89 BEY - (C) ENERGY APPLICATION SYSTEMS LTD. \*/  
/\* I\*-----\*/  
/\* I\*-----\*/  
/\* I\*- LOCAL DATA AREA \*/  
/\* I\*-----\*/  
/\* I\*- FIELD \*/  
/\* I\*- RULE \*/  
/\* I\*- TABLE VALUE \*/  
/\* I\*- ERROR DESC \*/  
/\* I\*- ERROR DESC \*/  
/\* I\*- ERROR # \*/  
/\* I\* COPY QSRC,PRCM953 \*/  
/\* C\*-----\*/  
/\* C\*-----\*/  
/\* C\*-----\*/  
/\* C\*- \*/  
/\* C\*-----\*/  
/\* C\*- STANDARD TABLE SEARCH CALL \*/  
/\* C\*-----\*/  
/\* C\*- \*/  
ASSIGN  
/\* C\* COPY QSRC,PRCM903 \*/  
.

FIND FIRST PCTL1R NO-LOCK NO-ERROR.

/\*-----\*/  
ASSIGN  
CTL#0B = PCTL1R.CTL#0B  
CTL#01 = PCTL1R.CTL#01  
CTL#30 = PCTL1R.CTL#30  
CTL#31 = PCTL1R.CTL#31  
CTL#93 = PCTL1R.CTL#93  
CTL#94 = PCTL1R.CTL#94.

/\*-----\*/  
/\* C\*-----\*/  
/\* C\*- \*/  
/\* C\*- \*/  
/\* C\* AE 01 08 02 START \*/



```

/* C* AE 01 08 02 END */
/* C*----- */
/* C*----- */
/*-----IN LDA --*/
FIND FIRST PL001{STDINC30.I}

NO-LOCK NO-ERROR.
ASSIGN
  XPFLD = SUBSTR(L1DATA, 0256, 0006)
  XPRULE = SUBSTR(L1DATA, 0262, 0006)
  XPTABV = SUBSTR(L1DATA, 0268, 0015)
  XPDESC = SUBSTR(L1DATA, 0283, 0050)
  Q_$$DUM = SUBSTR(L1DATA, 0333, 0060)
  Q_$$ERR# = SUBSTR(L1DATA, 0393, 0004)
  LDA = SUBSTR(L1DATA, 0001, 0396).

/* C*-- */
ASSIGN
  FACOPR = "Y"
  FLD# = "SC001P"
  RUL# = ""
  VAL# = "" OVERLAY(VAL#, 1, 0006) = "FACOPR".

/* ----- */
/* #LOKUP */
RUN DT100_00.P(INPUT-OUTPUT FLD#, INPUT-OUTPUT RUL#, INPUT-OUTPUT VAL#, INPUT-OUTPUT XPDESC,
  INPUT-OUTPUT Q_$$DUM, INPUT-OUTPUT XPERR#).

IF XPERR# <> "" THEN DO:
  ASSIGN
    XPERR# = "000" + XPERR#
    XPEMSG = ".
  FIND FIRST PZ002 USE-INDEX PZ002A WHERE Z2ERR# = XPERR# NO-LOCK NO-ERROR.
  IF AVAILABLE PZ002 THEN ASSIGN
    XPEMSG = Z2ERR
    ERR# = "Y".
END.
ELSE ASSIGN
  ERR# = "N"
  XPERR# = ""
  XPEMSG = ".
/* ----- */

IF XPERR# = "" THEN DO:
  ASSIGN
    FACOPR = SUBSTR(XPDESC, 1, 0001).
END.

/* C*-- */
ASSIGN
  ERR# = "N"
  XPEMSG = ""
  XPERR# = ".

DO ARR_I = 21 TO 29:
  IN_IN [ ARR_I ] = "0".
END.

/* C*----- */
/* C*- SET UP 1ST SCREEN */
/* C*----- */
$T001:

REPEAT:
  /* C*----- */
  /* C*- KEY INPUT SCREEN */
  /* C*- */
  /*=====*/
  /*= STANDARD MENU HANDLING =*/
  {STDINC03.I}

  /*-----*/

CREATE MENU-ITEM sub_2_03 ASSIGN PARENT = sub_2 LABEL = "Exit" ACCELERATOR = "F3" TRIGGERS:
  ON CHOOSE DO:
  ASSIGN
    IN_KC = "1".

```

```

| APPLY "GO".
| END.
END.

RUN TM001_F1.
/* EXFMT SCR1 */

IF IN_KA = "1" THEN DO:
| NEXT $T001.
| END.

ASSIGN
    ERR# = "N"
    XPEMSG = ""
    XPERR# = ""

DO ARR_I = 21 TO 29:
| IN_IN [ ARR_I ] = "0".
| END.

/* C*----- */
/* C*-  CMD 3 - RETURN */
/* C*- */

IF IN_KC = "1" THEN DO:
| ASSIGN
|     IN_LR = "1".

| ASSIGN
|     PARM_Q_#DELYN = Q_#DELYN.

| {STDINC12.I}

| DELETE WIDGET PRTM001.
| RETURN.
| END.

/* C*----- */
/* C*-  Search For BATTERY */
/* C*- */

IF B1NUM = "" THEN DO:
| ASSIGN OVERLAY(B1NUM, 1, 0001) = "+".
| END.

/* C*----- */
/* C*-- STANDARD SEARCH */
/* C*----- */
ASSIGN
    Q = SUBSTR(B1NUM, 1, 0001).

IF Q = "+" THEN DO:
| ASSIGN
|     VAL# = ""
|     FLD# = "BNUM  "
|     RUL# = ""

| /* ----- */
| /* #SERCH */
| IF FLD# = "WNUM" THEN RUL# = "1  ".
| RUN DT101_00.P(INPUT-OUTPUT FLD#, INPUT-OUTPUT RUL#, INPUT-OUTPUT VAL#, INPUT-OUTPUT XPDESC,
|     INPUT-OUTPUT Q_$$DUM, INPUT-OUTPUT Q_$$ERR#).

| /* ----- */
| ASSIGN
|     B1NUM = SUBSTR(VAL#, 1, 0005)
|     /* C*----- */
|     /* C*- */
|     .

| IF XPERR# <> "" THEN DO:
|     RUN G2GETERR.P(INPUT XPERR#, OUTPUT XPEMSG, OUTPUT ERR#).
| END.
| ELSE DO:
|     ASSIGN
|         ERR# = "N"
|         XPEMSG = ""
|         XPERR# = ""

| DO ARR_I = 21 TO 29:

```

```

| IN_IN [ ARR_I ] = "0".
END.

END.

/* C*****          GOTO $T001          */
/* C*-             -----          */

IF B1NUM = "" THEN DO:
| NEXT $T001.
END.

END.

/* C*-----          */
/* C*-   GET RECORD          */
/* C*-          */
ASSIGN
    SVNBAT = "".

FIND FIRST PB001R USE-INDEX PB001A WHERE PB001R.B1NUM = B1NUM NO-LOCK NO-ERROR.

IF AVAILABLE PB001R THEN DO:
PB001R_RECID = RECID(PB001R).
/*-----*/
ASSIGN
    B1NUM = PB001R.B1NUM
    B1NAME = PB001R.B1NAME
    B1OPER = PB001R.B1OPER
    B1FLD = PB001R.B1FLD
    B1GBAT = PB001R.B1GBAT
    B1NOPR = PB001R.B1NOPR
    B1NBAT = PB001R.B1NBAT
    B1S1 = PB001R.B1S1
    B1S2 = PB001R.B1S2
    B1SNAM = PB001R.B1SNAM
    B1XREF = PB001R.B1XREF
    B1PROV = PB001R.B1PROV
    B1GESQ = PB001R.B1GESQ
    B1GLCX = PB001R.B1GLCX
    B1GLSD = PB001R.B1GLSD
    B1GMER = PB001R.B1GMER
    B1GRNG = PB001R.B1GRNG
    B1GSCT = PB001R.B1GSCT
    B1GTWN = PB001R.B1GTWN
    B1OPR5 = PB001R.B1OPR5
    B1REF2 = PB001R.B1REF2
    B1TXT1 = PB001R.B1TXT1
    B1TXT2 = PB001R.B1TXT2
    B1PNUM = PB001R.B1PNUM
    B1TYPE = PB001R.B1TYPE.

/*-----*/
ASSIGN
    B1PNM1 = SUBSTR(B1PNUM, 0001, 0003)
    B1PNM2 = SUBSTR(B1PNUM, 0004, 0003)
    B1PNM3 = SUBSTR(B1PNUM, 0007, 0004).

    IN_IN [ 90 ] = "0".
END.
ELSE DO:
PB001R_RECID = ?.
    IN_IN [ 90 ] = "1".
END.

/* C* REMOVE PRAFAC PROCESSING          */
/* C*-          */

IF IN_IN [ 90 ] = "1" THEN DO:
ASSIGN
    NEWFLD = "NEW".

    RUN Q_$$S001.
    /* $S001 */
END.
ELSE DO:
ASSIGN
    SVNBAT = B1NBAT
    NEWFLD = ""
    Z1TYPE = "B"
    Z1ID = ""
    Z1ID = B1NUM.

FIND FIRST PZ001R USE-INDEX PZ001A WHERE PZ001R.Z1TYPE = Z1TYPE AND PZ001R.Z1ID = Z1ID NO-LOCK
NO-ERROR.

```

```

IF AVAILABLE PZ001R THEN DO:
PZ001R_RECID = RECID(PZ001R).
/*-----*/
ASSIGN
    Z1TYPE = PZ001R.Z1TYPE
    Z1ID = PZ001R.Z1ID
    Z1TEXT = PZ001R.Z1TEXT
    Z1XREF = PZ001R.Z1XREF
    Z1GOVT = PZ001R.Z1GOVT
    Z1REF2 = PZ001R.Z1REF2
    Z1TXT1 = PZ001R.Z1TXT1
    Z1TXT2 = PZ001R.Z1TXT2
    Z1OPER = PZ001R.Z1OPER
    Z1CON = PZ001R.Z1CON
    Z1PNUM = PZ001R.Z1PNUM
    Z1DCOD = PZ001R.Z1DCOD
    Z1FAX1 = PZ001R.Z1FAX1
    Z1FAX2 = PZ001R.Z1FAX2
    Z1FAX3 = PZ001R.Z1FAX3
    Z1OID = PZ001R.Z1OID
    Z1EUB = PZ001R.Z1EUB
    Z1INTF = PZ001R.Z1INTF
    Z1REF3 = PZ001R.Z1REF3
    Z1CTO = PZ001R.Z1CTO
    Z1PRAF = PZ001R.Z1PRAF
    Z1PRCV = PZ001R.Z1PRCV
    Z1PRCO = PZ001R.Z1PRCO
    Z1PRCA = PZ001R.Z1PRCA
    Z1RMS = PZ001R.Z1RMS
    Z1LEGA = PZ001R.Z1LEGA
    Z1SLPT = PZ001R.Z1SLPT
    Z1USER = PZ001R.Z1USER
    Z1UDAT = PZ001R.Z1UDAT
    Z1UTIM = PZ001R.Z1UTIM
    Z1EOSF = PZ001R.Z1EOSF
    Z1CALC = PZ001R.Z1CALC
    Z1RECP = PZ001R.Z1RECP.

/*-----*/
IN_IN [ 44 ] = "0".
END.
ELSE DO:
    PZ001R_RECID = ?.
    IN_IN [ 44 ] = "1".
END.

/* C*-- USER DATE TIME STAMP                                     */
RUN Q_$TIME.
/* $TIME */
ASSIGN
    Z1USER = WHOAMI
    Z1UTIM = UTIME
    Z1UDAT = TRUNCATE(DECIMAL(UDATX), 00)
/* C* AE 01 08 03                                               */

DO TRANSACTION:
FIND PZ001 WHERE RECID(PZ001) = PZ001R_RECID EXCLUSIVE-LOCK.
/*-----*/
ASSIGN
    PZ001.Z1TYPE = TRIM(Z1TYPE)
    PZ001.Z1ID = TRIM(Z1ID)
    PZ001.Z1TEXT = TRIM(Z1TEXT)
    PZ001.Z1XREF = TRIM(Z1XREF)
    PZ001.Z1GOVT = TRIM(Z1GOVT)
    PZ001.Z1REF2 = TRIM(Z1REF2)
    PZ001.Z1TXT1 = TRIM(Z1TXT1)
    PZ001.Z1TXT2 = TRIM(Z1TXT2)
    PZ001.Z1OPER = TRIM(Z1OPER)
    PZ001.Z1CON = TRIM(Z1CON)
    PZ001.Z1PNUM = TRIM(Z1PNUM)
    PZ001.Z1DCOD = TRIM(Z1DCOD)
    PZ001.Z1FAX1 = Z1FAX1
    PZ001.Z1FAX2 = Z1FAX2
    PZ001.Z1FAX3 = Z1FAX3
    PZ001.Z1OID = TRIM(Z1OID)
    PZ001.Z1EUB = TRIM(Z1EUB)
    PZ001.Z1INTF = TRIM(Z1INTF)
    PZ001.Z1REF3 = TRIM(Z1REF3)
    PZ001.Z1CTO = TRIM(Z1CTO)
    PZ001.Z1PRAF = TRIM(Z1PRAF)
    PZ001.Z1PRCV = TRIM(Z1PRCV)
    PZ001.Z1PRCO = TRIM(Z1PRCO)
    PZ001.Z1PRCA = TRIM(Z1PRCA)
    PZ001.Z1RMS = TRIM(Z1RMS)
    PZ001.Z1LEGA = TRIM(Z1LEGA)

```

```

        PZ001.Z1SLPT = TRIM(Z1SLPT)
        PZ001.Z1USER = TRIM(Z1USER)
        PZ001.Z1UDAT = Z1UDAT
        PZ001.Z1UTIM = Z1UTIM
        PZ001.Z1EOSF = TRIM(Z1EOSF)
        PZ001.Z1CALC = TRIM(Z1CALC)
        PZ001.Z1RECP = TRIM(Z1RECP) NO-ERROR.
    /*-----*/
END.
/* C* AE 01 08 03 */
END.
/* C*----- */
/* C*- ===== */
IF IN_IN [ 90 ] = "0" THEN DO:
    IF B1OPR5 = "" THEN DO:
        ASSIGN OVERLAY(B1OPR5, 1, 0004) = B1NOPR.
    END.
END.
/* C*--- GET PROVINCE */
ASSIGN
    FLD# = ""
    FLD# = "PROV "
    RUL# = ""
    VAL# = "" OVERLAY(VAL#, 1, 0001) = B1PROV.
/* ----- */
/* #LOKUP */
RUN DT100_00.P(INPUT-OUTPUT FLD#, INPUT-OUTPUT RUL#, INPUT-OUTPUT VAL#, INPUT-OUTPUT XPDESC,
    INPUT-OUTPUT Q_$$DUM, INPUT-OUTPUT XPERR#).
IF XPERR# <> "" THEN DO:
    ASSIGN
        XPERR# = "000" + XPERR#
        XPMSG = ".
    FIND FIRST PZ002 USE-INDEX PZ002A WHERE Z2ERR# = XPERR# NO-LOCK NO-ERROR.
    IF AVAILABLE PZ002 THEN ASSIGN
        XPMSG = Z2ERR
        ERR# = "Y".
END.
ELSE ASSIGN
    ERR# = "N"
    XPERR# = ""
    XPMSG = ".
/* ----- */
/* C*--- */
IF ERR# = "Y" THEN DO:
    ASSIGN
        PROVA = "*****".
END.
ELSE DO:
    ASSIGN
        PROVA = SUBSTR(XPDESC, 1, 0016).
END.
/* C*----- */
$T003:
REPEAT:
    /* C*----- */
    /* C*- DATA SCREEN */
    /* C*- */
    IF Q_#DELYN = "Y" THEN DO:
        /*=====*/
        /*= STANDARD MENU HANDLING =*/
        {STDINC03.I}
        /*-----*/
        CREATE MENU-ITEM sub_2_03 ASSIGN PARENT = sub_2 LABEL = "Exit" ACCELERATOR = "F3" TRIGGERS:
        ON CHOOSE DO:

```

```

    ASSIGN
        IN_KC = "1".
    APPLY "GO".
END.
END.

CREATE MENU-ITEM sub_2_04 ASSIGN PARENT = sub_2 LABEL = "Delete" ACCELERATOR = "F4" TRIGGERS:
    ON CHOOSE DO:
        ASSIGN
            IN_KD = "1".
        APPLY "GO".
    END.
END.

CREATE MENU-ITEM sub_2_05 ASSIGN PARENT = sub_2 LABEL = "Paste" ACCELERATOR = "F8" TRIGGERS:
    ON CHOOSE DO:
        ASSIGN
            IN_KH = "1".
        APPLY "GO".
    END.
END.

CREATE MENU-ITEM sub_2_06 ASSIGN PARENT = sub_2 LABEL = "Copy" ACCELERATOR = "F9" TRIGGERS:
    ON CHOOSE DO:
        ASSIGN
            IN_KI = "1".
        APPLY "GO".
    END.
END.

CREATE MENU-ITEM sub_2_07 ASSIGN PARENT = sub_2 LABEL = "Guide 60" ACCELERATOR = "F6" TRIGGERS:
    ON CHOOSE DO:
        ASSIGN
            IN_KF = "1".
        APPLY "GO".
    END.
END.

RUN TM001_F2.
/* EXFMT SCR2D */
END.
ELSE DO:
/*=====*/
/*= STANDARD MENU HANDLING =*/
{STDINC03.I}

/*-----*/

CREATE MENU-ITEM sub_2_03 ASSIGN PARENT = sub_2 LABEL = "Exit" ACCELERATOR = "F3" TRIGGERS:
    ON CHOOSE DO:
        ASSIGN
            IN_KC = "1".
        APPLY "GO".
    END.
END.

CREATE MENU-ITEM sub_2_04 ASSIGN PARENT = sub_2 LABEL = "Paste" ACCELERATOR = "F8" TRIGGERS:
    ON CHOOSE DO:
        ASSIGN
            IN_KH = "1".
        APPLY "GO".
    END.
END.

```

```

END.

CREATE MENU-ITEM sub_2_05 ASSIGN PARENT = sub_2 LABEL = "Copy" ACCELERATOR = "F9" TRIGGERS:
  ON CHOOSE DO:
  | ASSIGN
  |           IN_KI = "1".
  |
  | APPLY "GO".
  |
  | END.
END.

CREATE MENU-ITEM sub_2_06 ASSIGN PARENT = sub_2 LABEL = "Guide 60" ACCELERATOR = "F6" TRIGGERS:
  ON CHOOSE DO:
  | ASSIGN
  |           IN_KF = "1".
  |
  | APPLY "GO".
  |
  | END.
END.

RUN TM001_F3.
/* EXFMT SCR2 */
END.

IF IN_KA = "1" THEN DO:
| NEXT $T003.
END.

ASSIGN
  ERR# = "N"
  XPMSG = ""
  XPERR# = ""

DO ARR_I = 21 TO 29:
| IN_IN [ ARR_I ] = "0".
END.

/* C* AE 01 08 03 */
ASSIGN IN_IN [ 30 ] = "0"
/* C* AE 01 08 03 */
/* C*----- */
/* C*- F3 - BYPASS UPDATE */
/* C*- */
.

IF IN_KC = "1" THEN DO:
| NEXT $T001.
END.

/* C*- ----- */
/* C*----- */
/* C*- F8-COPY */
/* C*- */

IF IN_KH = "1" THEN DO:
  IF XXBNUM <> "" THEN DO:
  | ASSIGN
  |           Q_#NUM = B1NUM.

  IF IN_IN [ 90 ] = "0" THEN DO:
  | DO TRANSACTION:
  | | FIND PB001 WHERE RECID(PB001) = PB001R_RECID EXCLUSIVE-LOCK.
  | | DELETE PB001.
  | | END.
  |
  | END.

  FIND FIRST PB001R USE-INDEX PB001A WHERE PB001R.B1NUM = XXBNUM NO-LOCK NO-ERROR.

  IF AVAILABLE PB001R THEN DO:

```

```

PB001R_RECID = RECID(PB001R).
/*-----*/
ASSIGN
    B1NUM = PB001R.B1NUM
    B1NAME = PB001R.B1NAME
    B1OPER = PB001R.B1OPER
    B1FLD = PB001R.B1FLD
    B1GBAT = PB001R.B1GBAT
    B1NOPR = PB001R.B1NOPR
    B1NBAT = PB001R.B1NBAT
    B1S1 = PB001R.B1S1
    B1S2 = PB001R.B1S2
    B1SNAM = PB001R.B1SNAM
    B1XREF = PB001R.B1XREF
    B1PROV = PB001R.B1PROV
    B1GESQ = PB001R.B1GESQ
    B1GLCX = PB001R.B1GLCX
    B1GLSD = PB001R.B1GLSD
    B1GMER = PB001R.B1GMER
    B1GRNG = PB001R.B1GRNG
    B1GSCT = PB001R.B1GSCT
    B1GTWN = PB001R.B1GTWN
    B1OPR5 = PB001R.B1OPR5
    B1REF2 = PB001R.B1REF2
    B1TXT1 = PB001R.B1TXT1
    B1TXT2 = PB001R.B1TXT2
    B1PNUM = PB001R.B1PNUM
    B1TYPE = PB001R.B1TYPE.

/*-----*/
ASSIGN
    B1PNM1 = SUBSTR(B1PNUM, 0001, 0003)
    B1PNM2 = SUBSTR(B1PNUM, 0004, 0003)
    B1PNM3 = SUBSTR(B1PNUM, 0007, 0004).

IN_IN [ 90 ] = "0".
END.
ELSE DO:
    PB001R_RECID = ?.
    IN_IN [ 90 ] = "1".
END.

ASSIGN
    B1NUM = Q_#NUM.

DO TRANSACTION:
    CREATE PB001 NO-ERROR.
    /*-----*/
    ASSIGN
        PB001.B1NUM = TRIM(B1NUM)
        PB001.B1NAME = TRIM(B1NAME)
        PB001.B1OPER = TRIM(B1OPER)
        PB001.B1FLD = B1FLD
        PB001.B1GBAT = TRIM(B1GBAT)
        PB001.B1NOPR = TRIM(B1NOPR)
        PB001.B1NBAT = TRIM(B1NBAT)
        PB001.B1S1 = TRIM(B1S1)
        PB001.B1S2 = TRIM(B1S2)
        PB001.B1SNAM = TRIM(B1SNAM)
        PB001.B1XREF = TRIM(B1XREF)
        PB001.B1PROV = TRIM(B1PROV)
        PB001.B1GESQ = B1GESQ
        PB001.B1GLCX = B1GLCX
        PB001.B1GLSD = B1GLSD
        PB001.B1GMER = B1GMER
        PB001.B1GRNG = B1GRNG
        PB001.B1GSCT = B1GSCT
        PB001.B1GTWN = B1GTWN
        PB001.B1OPR5 = TRIM(B1OPR5)
        PB001.B1REF2 = TRIM(B1REF2)
        PB001.B1TXT1 = TRIM(B1TXT1)
        PB001.B1TXT2 = TRIM(B1TXT2)
        PB001.B1PNUM = TRIM(B1PNUM)
        PB001.B1TYPE = TRIM(B1TYPE) NO-ERROR.

/*-----*/
END.

FIND FIRST PB001R USE-INDEX PB001A WHERE PB001R.B1NUM = B1NUM NO-LOCK NO-ERROR.

IF AVAILABLE PB001R THEN DO:
    PB001R_RECID = RECID(PB001R).
    /*-----*/
    ASSIGN
        B1NUM = PB001R.B1NUM
        B1NAME = PB001R.B1NAME

```



```

BLOPER = PB001R.BLOPER
B1FLD = PB001R.B1FLD
B1GBAT = PB001R.B1GBAT
B1NOPR = PB001R.B1NOPR
B1NBAT = PB001R.B1NBAT
B1S1 = PB001R.B1S1
B1S2 = PB001R.B1S2
B1SNAM = PB001R.B1SNAM
B1XREF = PB001R.B1XREF
B1PROV = PB001R.B1PROV
B1GESQ = PB001R.B1GESQ
B1GLCX = PB001R.B1GLCX
B1GLSD = PB001R.B1GLSD
B1GMER = PB001R.B1GMER
B1GRNG = PB001R.B1GRNG
B1G SCT = PB001R.B1G SCT
B1GTWN = PB001R.B1GTWN
B1OPR5 = PB001R.B1OPR5
B1REF2 = PB001R.B1REF2
B1TXT1 = PB001R.B1TXT1
B1TXT2 = PB001R.B1TXT2
B1PNUM = PB001R.B1PNUM
B1TYPE = PB001R.B1TYPE.

/*-----*/
ASSIGN
    B1PNM1 = SUBSTR(B1PNUM, 0001, 0003)
    B1PNM2 = SUBSTR(B1PNUM, 0004, 0003)
    B1PNM3 = SUBSTR(B1PNUM, 0007, 0004).

IN_IN [ 90 ] = "0".
END.
ELSE DO:
    PB001R_RECID = ?.
    IN_IN [ 90 ] = "1".
END.
END.
END.

/* C*-----*/
/* C*- F4 - DELETE ( IF ARMED ) */
*/
*/

IF Q_#DELYN = "Y" THEN DO:
IF IN_KD = "1" THEN DO:
$TOOX:
REPEAT:
ASSIGN
    IN_KD = "0"
/* C*--- DISPLAY DELETE CONFIRMATION SCREEN */
.

/*=====*/
/*= STANDARD MENU HANDLING =*/
{STDINC03.I}

/*-----*/

CREATE MENU-ITEM sub_2_03 ASSIGN PARENT = sub_2 LABEL = "Exit" ACCELERATOR = "F3" TRIGGERS:
ON CHOOSE DO:
ASSIGN
    IN_KC = "1".
APPLY "GO".
END.
END.

CREATE MENU-ITEM sub_2_04 ASSIGN PARENT = sub_2 LABEL = "Confirm" ACCELERATOR = "F4"
TRIGGERS:
ON CHOOSE DO:
ASSIGN
    IN_KD = "1".
APPLY "GO".
END.
END.

```

```

CREATE MENU-ITEM sub_2_05 ASSIGN PARENT = sub_2 LABEL = "Pr Screen" ACCELERATOR = "F12"
      TRIGGERS:

      ON CHOOSE DO:
      | ASSIGN
      |           IN_KL = "1".
      |
      | APPLY "GO".
      | END.
      END.

      RUN TM001_F4.
      /* EXFMT SCR33 */

      IF IN_KA = "1" THEN DO:
      | NEXT $T00X.
      | END.
      LEAVE $T00X.
      END.

      /*$T00X*/
      /* C*--- F12 PREVIOUS SCREEN */

      IF IN_KL = "1" THEN DO:
      | NEXT $T003.
      | END.

      /* C*----- */
      /* C*--- F3 EXIT */

      IF IN_KC = "1" THEN DO:
      | NEXT $T003.
      | END.

      /* C*----- */
      /* C*--- F4 DELETE */

      IF IN_KD = "1" THEN DO:
      | RUN Q_$$S002.
      | /* $S002 */
      | NEXT $T001.
      | END.
      END.

      END.

      END.

      /* C* AE 01 08 03 */
      /* C*----- */
      /* C*- F6 - GUIDE 60 - CALL INQUIRY PROGRAM */

      IF IN_KF = "1" THEN DO:
      /* C*** ONLY APPLICABLE IN AB - DISPLAY MESSAGE IF NOT AB */

      IF CTL#0B <> "1" THEN DO:
      ASSIGN
      |           XPERR# = "TM00103".
      |
      | RUN G2GETERR.P(INPUT XPERR#, OUTPUT XPEMSG, OUTPUT ERR#).
      | END.
      ELSE DO:
      /* C*** */
      ASSIGN
      |           Z1TYPE = "B".
      |
      | RUN PRM015.P(INPUT-OUTPUT Z1TYPE, INPUT-OUTPUT B1NUM).
      | END.
      NEXT $T001.
      END.

      END.

      /* C* AE 01 08 03 */
      /* C*----- */
      /* C*- PROVINCE CODE VALIDATION */
      /* C*- */
      /* C*- MOVE1 XPDESC IF REQUIRED */
      /* C*----- */
      /* C*--- PROV CODE 27 */
      ASSIGN
      |           CHKQ = B1PROV.

      IF CHKQ = "+" THEN DO:

```

```

ASSIGN
    FLD# = "PROV ".

/* ----- */
/* #SERCH */
IF FLD# = "WNUM" THEN RUL# = "1 ".
RUN DT101_00.P(INPUT-OUTPUT FLD#, INPUT-OUTPUT RUL#, INPUT-OUTPUT VAL#, INPUT-OUTPUT XPDESC
, INPUT-OUTPUT Q_$$DUM, INPUT-OUTPUT Q_$$ERR#).

/* ----- */
ASSIGN
    IN_IN [ 40 ] = "1"
    B1PROV = SUBSTR(VAL#, 1, 0001).

END.

/* C*----- */
ASSIGN
    FLD# = ""
    FLD# = "PROV "
    RUL# = ""
    VAL# = "" OVERLAY(VAL#, 1, 0001) = B1PROV.

/* ----- */
/* #LOKUP */
RUN DT100_00.P(INPUT-OUTPUT FLD#, INPUT-OUTPUT RUL#, INPUT-OUTPUT VAL#, INPUT-OUTPUT XPDESC,
INPUT-OUTPUT Q_$$DUM, INPUT-OUTPUT XPERR#).

IF XPERR# <> "" THEN DO:
    ASSIGN
        XPERR# = "000" + XPERR#
        XPEMSG = "".

    FIND FIRST PZ002 USE-INDEX PZ002A WHERE Z2ERR# = XPERR# NO-LOCK NO-ERROR.
    IF AVAILABLE PZ002 THEN ASSIGN
        XPEMSG = Z2ERR
        ERR# = "Y".

END.
ELSE ASSIGN
    ERR# = "N"
    XPERR# = ""
    XPEMSG = "".

/* ----- */
/* C*---- */
IF ERR# = "Y" THEN DO:
    ASSIGN
        B1PROV = "+"
        IN_IN [ 27 ] = "1"
        XPERR# = "TM00101".

    RUN G2GETERR.P(INPUT XPERR#, OUTPUT XPEMSG, OUTPUT ERR#).
    NEXT $T003.
END.

ASSIGN
    PROVA = SUBSTR(XPDESC, 1, 0016)
/* C*----- */
/* C*- GOVERNMENT BATTERY NUMBER CANNOT BE ZERO */
/* C*- */
.

IF B1NBAT = "" THEN DO:
    ASSIGN
        IN_IN [ 21 ] = "1"
        XPERR# = "0001008".

    RUN G2GETERR.P(INPUT XPERR#, OUTPUT XPEMSG, OUTPUT ERR#).
    NEXT $T003.
/* C*----- */
END.

/* C*----- */
/* C*- GOVERNMENT OPERATOR FIELD CANNOT BE BLANK - EXCEPT ONTARIO */
/* C*----- */
ASSIGN
    CHKQ = SUBSTR(Z1OID, 1, 0001).

IF CHKQ = "+" THEN DO:
    ASSIGN

```

```

        FLD# = "OID      "
        RUL# = " ".

/* ----- */
/* #SERCH      */
IF FLD# = "WNUM" THEN RUL# = "1      ".
RUN DT101_00.P(INPUT-OUTPUT FLD#, INPUT-OUTPUT RUL#, INPUT-OUTPUT VAL#, INPUT-OUTPUT XPDESC
, INPUT-OUTPUT Q_$$DUM, INPUT-OUTPUT Q_$$ERR#).

/* ----- */
ASSIGN
        IN_IN [ 40 ] = "1"
        Z1OID = SUBSTR(VAL#, 1, 0008).

END.

/* C*--- */
IF FACOPR = "Y" THEN DO:
    IF Z1OID = "" THEN DO:
        ASSIGN
            XPERR# = "0001009"
            BLOPR5 = " " OVERLAY(Z1OID, 1, 0001) = "+".

        RUN G2GETERR.P(INPUT XPERR#, OUTPUT XPEMSG, OUTPUT ERR#).
        NEXT $T003.
    END.
END.

/* C*-- */
IF Z1OID <> "" THEN DO:
    /* C* AE 01 08 02 */
    ASSIGN
        O4DATE = TRUNCATE(DECIMAL(CTL#01), 00)
    /* C* AE 01 08 02 */
    .

    FIND FIRST PO004R USE-INDEX PO004C WHERE PO004R.O4DATE = O4DATE AND PO004R.O4OID = Z1OID AND
    PO004R.O4PROV = CTL#0B NO-LOCK NO-ERROR.

    IF AVAILABLE PO004R THEN DO:
        /*-----*/
        ASSIGN
            O4GOVT = PO004R.O4GOVT
            O4DATE = PO004R.O4DATE.

        /*-----*/
        IN_IN [ 22 ] = "0".
    END.
    ELSE DO:
        IN_IN [ 22 ] = "1".
    END.

    IF IN_IN [ 22 ] = "1" THEN DO:
        ASSIGN
            XPERR# = "0001009"
            BLOPR5 = " " OVERLAY(Z1OID, 1, 0001) = "+".

        RUN G2GETERR.P(INPUT XPERR#, OUTPUT XPEMSG, OUTPUT ERR#).
        NEXT $T003.
    END.
END.

/* C*----- */
/* C*- */
ASSIGN
    BLOPR5 = SUBSTR(O4GOVT, 1, 0005)
    BLOPER = " "
    B1NOPR = " "
    BLOPER = SUBSTR(BLOPR5, 1, 0003)
    B1NOPR = SUBSTR(BLOPR5, 1, 0004)
/* C*----- */
/* C*- GOVERNMENT BATTERY NAME CANNOT BE BLANK */
.

IF B1NAME = "" THEN DO:
    ASSIGN
        IN_IN [ 23 ] = "1"
        XPERR# = "0001010".

```

```

RUN G2GETERR.P(INPUT XPERR#, OUTPUT XPEMSG, OUTPUT ERR#).
NEXT $T003.
/* C*-          -----          */
END.

/* C*-----          */
/* C* AE 01 08 03          */
IF B1TYPE <> "O" AND B1TYPE <> "G" AND B1TYPE <> "W" THEN DO:
  ASSIGN
    IN_IN [ 30 ] = "1"
    XPERR# = "TM00102".

  RUN G2GETERR.P(INPUT XPERR#, OUTPUT XPEMSG, OUTPUT ERR#).
  NEXT $T003.
END.

/* C*-----          */
/* C* AE 01 08 03          */
/* C*- S1 SIGNATURE NAME CANNOT BE BLANK          */
IF B1S1 = "" THEN DO:
  ASSIGN
    IN_IN [ 24 ] = "1"
    XPERR# = "0001011".

  RUN G2GETERR.P(INPUT XPERR#, OUTPUT XPEMSG, OUTPUT ERR#).
  NEXT $T003.
/* C*-          -----          */
END.

/* C*-----          */
/* C*- S2 SIGNATURE NAME CANNOT BE BLANK          */
IF B1S2 = "" THEN DO:
  ASSIGN
    IN_IN [ 25 ] = "1"
    XPERR# = "0001011".

  RUN G2GETERR.P(INPUT XPERR#, OUTPUT XPEMSG, OUTPUT ERR#).
  NEXT $T003.
/* C*-          -----          */
END.

/* C*-----          */
/* C*- SEARCH NAME CANNOT BE BLANK          */
IF B1SNAM = "" THEN DO:
  ASSIGN
    IN_IN [ 26 ] = "1"
    XPERR# = "0001014".

  RUN G2GETERR.P(INPUT XPERR#, OUTPUT XPEMSG, OUTPUT ERR#).
  NEXT $T003.
/* C*-          -----          */
END.

/* C*-----          */
/* C*- EXTERNAL REF EDITS.....          */
IF CTL#93 = "Y" THEN DO:
  IF B1XREF = "" THEN DO:
    ASSIGN
      IN_IN [ 28 ] = "1"
      XPERR# = "0001099".

    RUN G2GETERR.P(INPUT XPERR#, OUTPUT XPEMSG, OUTPUT ERR#).
    NEXT $T003.
  END.
END.

IF CTL#94 = "Y" THEN DO:
  IF B1REF2 = "" THEN DO:
    ASSIGN
      IN_IN [ 29 ] = "1"
      XPERR# = "0001099".

    RUN G2GETERR.P(INPUT XPERR#, OUTPUT XPEMSG, OUTPUT ERR#).
    NEXT $T003.
  END.
END.

```

```

/* C*---
/* C** PRA FACILITY?
*/
*/

IF Z1PRAF <> "" THEN DO:
  IF Z1PRAF <> "Y" AND Z1PRAF <> "N" THEN DO:
    ASSIGN
      IN_IN [ 31 ] = "1"
      XPERR# = "DT30021".
    RUN G2GETERR.P(INPUT XPERR#, OUTPUT XPEMSG, OUTPUT ERR#).
    NEXT $T003.
  END.
END.
ELSE DO:
  ASSIGN
    Z1PRAF = "Y".
END.

/* C*---
/* C** PROCESS SAF OAF?
*/
*/

IF Z1PRCO <> "" THEN DO:
  IF Z1PRCO <> "Y" AND Z1PRCO <> "N" THEN DO:
    ASSIGN
      IN_IN [ 32 ] = "1"
      XPERR# = "DT30021".
    RUN G2GETERR.P(INPUT XPERR#, OUTPUT XPEMSG, OUTPUT ERR#).
    NEXT $T003.
  END.
END.
ELSE DO:
  ASSIGN
    Z1PRCO = "Y".
END.

/* C*---
/* C** PROCESS VOLUMETRICS?
*/
*/

IF Z1PRCV <> "" THEN DO:
  IF Z1PRCV <> "Y" AND Z1PRCV <> "N" THEN DO:
    ASSIGN
      IN_IN [ 33 ] = "1"
      XPERR# = "DT30021".
    RUN G2GETERR.P(INPUT XPERR#, OUTPUT XPEMSG, OUTPUT ERR#).
    NEXT $T003.
  END.
END.
ELSE DO:
  ASSIGN
    Z1PRCV = "Y".
END.

/* C*---
/* C** PROCESS APMC?
*/
*/

IF Z1PRCA <> "" THEN DO:
  IF Z1PRCA <> "Y" AND Z1PRCA <> "N" THEN DO:
    ASSIGN
      IN_IN [ 34 ] = "1"
      XPERR# = "DT30021".
    RUN G2GETERR.P(INPUT XPERR#, OUTPUT XPEMSG, OUTPUT ERR#).
    NEXT $T003.
  END.
END.
ELSE DO:
  ASSIGN
    Z1PRCA = "Y".
END.

/* C*-----
/* C*- UPDATE RECORD
*/
*/

```

```
IF IN_IN [ 90 ] = "0" THEN DO:
```

```
DO TRANSACTION:
```

```
FIND PB001 WHERE RECID(PB001) = PB001R_RECID EXCLUSIVE-LOCK.
```

```
/*-----*/
```

```
ASSIGN
```

```
PB001.B1NUM = TRIM(B1NUM)
PB001.B1NAME = TRIM(B1NAME)
PB001.B1OPER = TRIM(B1OPER)
PB001.B1FLD = B1FLD
PB001.B1GBAT = TRIM(B1GBAT)
PB001.B1NOPR = TRIM(B1NOPR)
PB001.B1NBAT = TRIM(B1NBAT)
PB001.B1S1 = TRIM(B1S1)
PB001.B1S2 = TRIM(B1S2)
PB001.B1SNAM = TRIM(B1SNAM)
PB001.B1XREF = TRIM(B1XREF)
PB001.B1PROV = TRIM(B1PROV)
PB001.B1GESQ = B1GESQ
PB001.B1GLCX = B1GLCX
PB001.B1GLSD = B1GLSD
PB001.B1GMER = B1GMER
PB001.B1GRNG = B1GRNG
PB001.B1GSCT = B1GSCT
PB001.B1GTWN = B1GTWN
PB001.B1OPR5 = TRIM(B1OPR5)
PB001.B1REF2 = TRIM(B1REF2)
PB001.B1TXT1 = TRIM(B1TXT1)
PB001.B1TXT2 = TRIM(B1TXT2)
PB001.B1PNUM = TRIM(B1PNUM)
PB001.B1TYPE = TRIM(B1TYPE) NO-ERROR.
```

```
/*-----*/
```

```
END.
```

```
IF B1NBAT <> SVNBAT THEN DO:
```

```
RUN Q_$$S003.
```

```
/* $$S003 */
```

```
END.
```

```
END.
```

```
ELSE DO:
```

```
/* C*-----*/
```

```
/* C*- ADD RECORD
```

```
*/
```

```
*/
```

```
DO TRANSACTION:
```

```
CREATE PB001 NO-ERROR.
```

```
/*-----*/
```

```
ASSIGN
```

```
PB001.B1NUM = TRIM(B1NUM)
PB001.B1NAME = TRIM(B1NAME)
PB001.B1OPER = TRIM(B1OPER)
PB001.B1FLD = B1FLD
PB001.B1GBAT = TRIM(B1GBAT)
PB001.B1NOPR = TRIM(B1NOPR)
PB001.B1NBAT = TRIM(B1NBAT)
PB001.B1S1 = TRIM(B1S1)
PB001.B1S2 = TRIM(B1S2)
PB001.B1SNAM = TRIM(B1SNAM)
PB001.B1XREF = TRIM(B1XREF)
PB001.B1PROV = TRIM(B1PROV)
PB001.B1GESQ = B1GESQ
PB001.B1GLCX = B1GLCX
PB001.B1GLSD = B1GLSD
PB001.B1GMER = B1GMER
PB001.B1GRNG = B1GRNG
PB001.B1GSCT = B1GSCT
PB001.B1GTWN = B1GTWN
PB001.B1OPR5 = TRIM(B1OPR5)
PB001.B1REF2 = TRIM(B1REF2)
PB001.B1TXT1 = TRIM(B1TXT1)
PB001.B1TXT2 = TRIM(B1TXT2)
PB001.B1PNUM = TRIM(B1PNUM)
PB001.B1TYPE = TRIM(B1TYPE) NO-ERROR.
```

```
/*-----*/
```

```
END.
```

```
END.
```

```
/* C*-----*/
```

```
/* C*- F9-SAVE
```

```
/* C*-
```

```
*/
```

```
*/
```

```
*/
```

```
IF IN_KI = "1" THEN DO:
```

```

ASSIGN
      XXBNUM = B1NUM.
END.

/* C*----- */
/* C*--- ALSO UPDATE PZ001 SYS ID GENERAL */
/* C*          Z1TYPE          ' ) */
/* C*          Z1ID            ' ) */
/* C*          Z1TEXT         e' ) */
/* C*          Z1XREF          ' ) */
/* C*          Z1OPER          ' ) */
/* C*          Z1FAX1,2,3     ' ) */
ASSIGN
      Z1TYPE = "B"
      Z1ID = " "
      Z1ID = B1NUM
      XOID = Z1OID
      XFAX1 = Z1FAX1
      XFAX2 = Z1FAX2
      XFAX3 = Z1FAX3
      XCTO = Z1CTO
/* C** SAVE PRA FLAGS */
      XPRAF = Z1PRAF
      XPRCO = Z1PRCO
      XPRCV = Z1PRCV
      XPRCA = Z1PRCA
/* C*--- */

FIND FIRST PZ001R USE-INDEX PZ001A WHERE PZ001R.Z1TYPE = Z1TYPE AND PZ001R.Z1ID = Z1ID NO-LOCK
NO-ERROR.

IF AVAILABLE PZ001R THEN DO:
  PZ001R_RECID = RECID(PZ001R).
  /*-----*/
  ASSIGN
    Z1TYPE = PZ001R.Z1TYPE
    Z1ID = PZ001R.Z1ID
    Z1TEXT = PZ001R.Z1TEXT
    Z1XREF = PZ001R.Z1XREF
    Z1GOVT = PZ001R.Z1GOVT
    Z1REF2 = PZ001R.Z1REF2
    Z1TXT1 = PZ001R.Z1TXT1
    Z1TXT2 = PZ001R.Z1TXT2
    Z1OPER = PZ001R.Z1OPER
    Z1CON = PZ001R.Z1CON
    Z1PNUM = PZ001R.Z1PNUM
    Z1DCOD = PZ001R.Z1DCOD
    Z1FAX1 = PZ001R.Z1FAX1
    Z1FAX2 = PZ001R.Z1FAX2
    Z1FAX3 = PZ001R.Z1FAX3
    Z1OID = PZ001R.Z1OID
    Z1EUB = PZ001R.Z1EUB
    Z1INTF = PZ001R.Z1INTF
    Z1REF3 = PZ001R.Z1REF3
    Z1CTO = PZ001R.Z1CTO
    Z1PRAF = PZ001R.Z1PRAF
    Z1PRCV = PZ001R.Z1PRCV
    Z1PRCO = PZ001R.Z1PRCO
    Z1PRCA = PZ001R.Z1PRCA
    Z1RMS = PZ001R.Z1RMS
    Z1LEGA = PZ001R.Z1LEGA
    Z1SLPT = PZ001R.Z1SLPT
    Z1USER = PZ001R.Z1USER
    Z1UDAT = PZ001R.Z1UDAT
    Z1UTIM = PZ001R.Z1UTIM
    Z1EOSF = PZ001R.Z1EOSF
    Z1CALC = PZ001R.Z1CALC
    Z1RECP = PZ001R.Z1RECP.
  /*-----*/
  IN_IN [ 44 ] = "0".
END.
ELSE DO:
  PZ001R_RECID = ?.
  IN_IN [ 44 ] = "1".
END.

ASSIGN
  Z1XREF = B1XREF
  Z1REF2 = B1REF2
  Z1TEXT = SUBSTR(B1NAME, 1, 0030)
  Z1OPER = B1OPR5
  Z1GOVT = " "

```



```

Z1GOVT = B1NBAT
Z1OID = XOID
Z1FAX1 = XFAX1
Z1FAX2 = XFAX2
Z1FAX3 = XFAX3
Z1CTO = XCTO
Z1PRAF = XPRAF
Z1PRCO = XPRCO
Z1PRCV = XPRCV
Z1PRCA = XPRCA
/* C*-- USER DATE TIME STAMP */
.

RUN Q_$TIME.
/* $TIME */
ASSIGN
    Z1USER = WHOAMI
    Z1UTIM = UTIME
    Z1UDAT = TRUNCATE(DECIMAL(UDATX), 00)
/* C*- */
.

IF IN_IN [ 44 ] = "0" THEN DO:
DO TRANSACTION:
FIND PZ001 WHERE RECID(PZ001) = PZ001R_RECID EXCLUSIVE-LOCK.
/*-----*/
ASSIGN
    PZ001.Z1TYPE = TRIM(Z1TYPE)
    PZ001.Z1ID = TRIM(Z1ID)
    PZ001.Z1TEXT = TRIM(Z1TEXT)
    PZ001.Z1XREF = TRIM(Z1XREF)
    PZ001.Z1GOVT = TRIM(Z1GOVT)
    PZ001.Z1REF2 = TRIM(Z1REF2)
    PZ001.Z1TXT1 = TRIM(Z1TXT1)
    PZ001.Z1TXT2 = TRIM(Z1TXT2)
    PZ001.Z1OPER = TRIM(Z1OPER)
    PZ001.Z1CON = TRIM(Z1CON)
    PZ001.Z1PNUM = TRIM(Z1PNUM)
    PZ001.Z1DCOD = TRIM(Z1DCOD)
    PZ001.Z1FAX1 = Z1FAX1
    PZ001.Z1FAX2 = Z1FAX2
    PZ001.Z1FAX3 = Z1FAX3
    PZ001.Z1OID = TRIM(Z1OID)
    PZ001.Z1EUB = TRIM(Z1EUB)
    PZ001.Z1INTF = TRIM(Z1INTF)
    PZ001.Z1REF3 = TRIM(Z1REF3)
    PZ001.Z1CTO = TRIM(Z1CTO)
    PZ001.Z1PRAF = TRIM(Z1PRAF)
    PZ001.Z1PRCV = TRIM(Z1PRCV)
    PZ001.Z1PRCO = TRIM(Z1PRCO)
    PZ001.Z1PRCA = TRIM(Z1PRCA)
    PZ001.Z1RMS = TRIM(Z1RMS)
    PZ001.Z1LEGA = TRIM(Z1LEGA)
    PZ001.Z1SLPT = TRIM(Z1SLPT)
    PZ001.Z1USER = TRIM(Z1USER)
    PZ001.Z1UDAT = Z1UDAT
    PZ001.Z1UTIM = Z1UTIM
    PZ001.Z1EOSF = TRIM(Z1EOSF)
    PZ001.Z1CALC = TRIM(Z1CALC)
    PZ001.Z1RECP = TRIM(Z1RECP) NO-ERROR.
/*-----*/
END.

END.
ELSE DO:
/* C*----- */
/* C*- ADD RECORD */
.

DO TRANSACTION:
CREATE PZ001 NO-ERROR.
/*-----*/
ASSIGN
    PZ001.Z1TYPE = TRIM(Z1TYPE)
    PZ001.Z1ID = TRIM(Z1ID)
    PZ001.Z1TEXT = TRIM(Z1TEXT)
    PZ001.Z1XREF = TRIM(Z1XREF)
    PZ001.Z1GOVT = TRIM(Z1GOVT)
    PZ001.Z1REF2 = TRIM(Z1REF2)
    PZ001.Z1TXT1 = TRIM(Z1TXT1)
    PZ001.Z1TXT2 = TRIM(Z1TXT2)
    PZ001.Z1OPER = TRIM(Z1OPER)
    PZ001.Z1CON = TRIM(Z1CON)
    PZ001.Z1PNUM = TRIM(Z1PNUM)
    PZ001.Z1DCOD = TRIM(Z1DCOD)

```

```
PZ001.Z1FAX1 = Z1FAX1
PZ001.Z1FAX2 = Z1FAX2
PZ001.Z1FAX3 = Z1FAX3
PZ001.Z1OID = TRIM(Z1OID)
PZ001.Z1EUB = TRIM(Z1EUB)
PZ001.Z1INTF = TRIM(Z1INTF)
PZ001.Z1REF3 = TRIM(Z1REF3)
PZ001.Z1CTO = TRIM(Z1CTO)
PZ001.Z1PRAF = TRIM(Z1PRAF)
PZ001.Z1PRCV = TRIM(Z1PRCV)
PZ001.Z1PRCO = TRIM(Z1PRCO)
PZ001.Z1PRCA = TRIM(Z1PRCA)
PZ001.Z1RMS = TRIM(Z1RMS)
PZ001.Z1LEGA = TRIM(Z1LEGA)
PZ001.Z1SLPT = TRIM(Z1SLPT)
PZ001.Z1USER = TRIM(Z1USER)
PZ001.Z1UDAT = Z1UDAT
PZ001.Z1UTIM = Z1UTIM
PZ001.Z1EOSF = TRIM(Z1EOSF)
PZ001.Z1CALC = TRIM(Z1CALC)
PZ001.Z1RECP = TRIM(Z1RECP) NO-ERROR.
```

```
/*-----*/
```

```
END.
```

```
END.
```

```
/* C*--- */
```

```
RUN Q_$$S005.
```

```
/* $S005 */
```

```
/* C*--- */
```

```
FIND FIRST PB001R USE-INDEX PB001A WHERE PB001R.B1NUM = B1NUM NO-LOCK NO-ERROR.
```

```
IF AVAILABLE PB001R THEN DO:
```

```
PB001R_RECID = RECID(PB001R).
```

```
/*-----*/
```

```
ASSIGN
```

```
B1NUM = PB001R.B1NUM
B1NAME = PB001R.B1NAME
B1OPER = PB001R.B1OPER
B1FLD = PB001R.B1FLD
B1GBAT = PB001R.B1GBAT
B1NOPR = PB001R.B1NOPR
B1NBAT = PB001R.B1NBAT
B1S1 = PB001R.B1S1
B1S2 = PB001R.B1S2
B1SNAM = PB001R.B1SNAM
B1XREF = PB001R.B1XREF
B1PROV = PB001R.B1PROV
B1GESQ = PB001R.B1GESQ
B1GLCX = PB001R.B1GLCX
B1GLSD = PB001R.B1GLSD
B1GMER = PB001R.B1GMER
B1GRNG = PB001R.B1GRNG
B1GSCT = PB001R.B1GSCT
B1GTWN = PB001R.B1GTWN
B1OPR5 = PB001R.B1OPR5
B1REF2 = PB001R.B1REF2
B1TXT1 = PB001R.B1TXT1
B1TXT2 = PB001R.B1TXT2
B1PNUM = PB001R.B1PNUM
B1TYPE = PB001R.B1TYPE.
```

```
/*-----*/
```

```
ASSIGN
```

```
B1PNM1 = SUBSTR(B1PNUM, 0001, 0003)
B1PNM2 = SUBSTR(B1PNUM, 0004, 0003)
B1PNM3 = SUBSTR(B1PNUM, 0007, 0004).
```

```
IN_IN [ 90 ] = "0".
```

```
END.
```

```
ELSE DO:
```

```
PB001R_RECID = ?.
```

```
IN_IN [ 90 ] = "1".
```

```
END.
```

```
NEXT $T003.
```

```
LEAVE $T003.
```

```
END.
```

```
/*$T003*/
```

```
LEAVE $T001.
```

```
END.
```

```
/*$T001*/
```

```
ASSIGN
```

```
PARM_Q_#DELYN = Q_#DELYN.
```

{STDINC12.I}

DELETE WIDGET PRM001.  
RETURN.  
{STDINC20.I}

/\*=====\*/

PROCEDURE Q\_\$TIME:

```
/*-- SECTION TYPE : SUBROUTINE          ---*/
RUN GETTIME.P(OUTPUT UTIME).
RUN GETTIME.P(OUTPUT TIME14).
TIME14 = DEC(STRING(TIME14, "999999") + STRING(TODAY, "99999999")).
ASSIGN
  UDATZ = TIME14 - TRUNCATE(TIME14 / 100000000, 0) * 100000000
  XCCYY = SUBSTR(STRING(UDATZ, "99999999-"), 005)
  XMMDD = SUBSTRING(STRING(UDATZ, "99999999-"), 1, 0004) OVERLAY(UDATX, 1, 0004) = XCCYY SUBSTR(
  UDATX, 005, 0004) = XMMDD
/* C*-----*/
```

```
WHOAMI = ""
VARNAM = "" OVERLAY(VARNAM, 1, 0004) = "USER".
```

```
RUN RTVJOBA.P(INPUT-OUTPUT VARNAM, INPUT-OUTPUT WHOAMI).
END PROCEDURE.
```

/\*=====\*/

PROCEDURE Q\_\$S001:

```
/*-- SECTION TYPE : SUBROUTINE          ---*/
/* C*-----*/
/* C*-      =====*/
ASSIGN
  B1NAME = ""
  B1NOPR = ""
  B1OPER = ""
  B1OPR5 = ""
  Z1OPER = ""
  Z1OID = ""
  B1FLD = 0
  B1NBAT = ""
  B1S1 = ""
  B1S2 = ""
  B1SNAM = ""
  B1XREF = ""
/* C*-                                           */
```

```
B1GESQ = 0
B1GLCX = 0
B1GLSD = 0
B1GMER = 0
B1GRNG = 0
B1GSCT = 0
B1GTWN = 0
/* C*-                                           */
```

```
B1PNM1 = "".
```

```
ASSIGN OVERLAY(B1PNUM, 0001, 0003) = B1PNM1.
ASSIGN
```

```
B1PNM2 = "".
```

```
ASSIGN OVERLAY(B1PNUM, 0004, 0003) = B1PNM2.
ASSIGN
```

```
B1PNM3 = "".
```

```
ASSIGN OVERLAY(B1PNUM, 0007, 0004) = B1PNM3.
/* C*-                                           */
```

END PROCEDURE.

/\*=====\*/

PROCEDURE Q\_\$S002:

```
/*-- SECTION TYPE : SUBROUTINE          ---*/
/* C*-----*/
/* C*-      BATTERY MASTER*/
/* C*-----*/
FIND FIRST PB001R USE-INDEX PB001A WHERE PB001R.B1NUM = B1NUM NO-LOCK NO-ERROR.
```

```
IF AVAILABLE PB001R THEN DO:
  PB001R_RECID = RECID(PB001R).
```

```
/*-----*/
```

```
ASSIGN
```

```
B1NUM = PB001R.B1NUM
B1NAME = PB001R.B1NAME
B1OPER = PB001R.B1OPER
B1FLD = PB001R.B1FLD
B1GBAT = PB001R.B1GBAT
B1NOPR = PB001R.B1NOPR
B1NBAT = PB001R.B1NBAT
B1S1 = PB001R.B1S1
B1S2 = PB001R.B1S2
B1SNAM = PB001R.B1SNAM
B1XREF = PB001R.B1XREF
B1PROV = PB001R.B1PROV
B1GESQ = PB001R.B1GESQ
B1GLCX = PB001R.B1GLCX
B1GLSD = PB001R.B1GLSD
B1GMER = PB001R.B1GMER
B1GRNG = PB001R.B1GRNG
B1GSCT = PB001R.B1GSCT
B1GTWN = PB001R.B1GTWN
B1OPR5 = PB001R.B1OPR5
B1REF2 = PB001R.B1REF2
B1TXT1 = PB001R.B1TXT1
B1TXT2 = PB001R.B1TXT2
B1PNUM = PB001R.B1PNUM
B1TYPE = PB001R.B1TYPE.
```

```
/*-----*/
```

```
ASSIGN
```

```
B1PNM1 = SUBSTR(B1PNUM, 0001, 0003)
B1PNM2 = SUBSTR(B1PNUM, 0004, 0003)
B1PNM3 = SUBSTR(B1PNUM, 0007, 0004).
```

```
IN_IN [ 44 ] = "0".
```

```
END.
```

```
ELSE DO:
```

```
PB001R_RECID = ?.
IN_IN [ 44 ] = "1".
```

```
END.
```

```
IF IN_IN [ 44 ] = "0" THEN DO:
```

```
DO TRANSACTION:
```

```
  FIND PB001 WHERE RECID(PB001) = PB001R_RECID EXCLUSIVE-LOCK.
  DELETE PB001.
END.
```

```
END.
```

```
/* C*-----
/* C*- FACILITY MASTER
/* C*-----
```

```
*/
*/
*/
```

```
ASSIGN
```

```
Z1TYPE = "B"
Z1ID = ""
Z1ID = B1NUM.
```

```
FIND FIRST PZ001R USE-INDEX PZ001A WHERE PZ001R.Z1TYPE = Z1TYPE AND PZ001R.Z1ID = Z1ID NO-LOCK
NO-ERROR.
```

```
IF AVAILABLE PZ001R THEN DO:
```

```
PZ001R_RECID = RECID(PZ001R).
```

```
/*-----*/
```

```
ASSIGN
```

```
Z1TYPE = PZ001R.Z1TYPE
Z1ID = PZ001R.Z1ID
Z1TEXT = PZ001R.Z1TEXT
Z1XREF = PZ001R.Z1XREF
Z1GOVT = PZ001R.Z1GOVT
Z1REF2 = PZ001R.Z1REF2
Z1TXT1 = PZ001R.Z1TXT1
Z1TXT2 = PZ001R.Z1TXT2
Z1OPER = PZ001R.Z1OPER
Z1CON = PZ001R.Z1CON
Z1PNUM = PZ001R.Z1PNUM
Z1DCOD = PZ001R.Z1DCOD
Z1FAX1 = PZ001R.Z1FAX1
Z1FAX2 = PZ001R.Z1FAX2
Z1FAX3 = PZ001R.Z1FAX3
Z1OID = PZ001R.Z1OID
Z1EUB = PZ001R.Z1EUB
Z1INTF = PZ001R.Z1INTF
Z1REF3 = PZ001R.Z1REF3
Z1CTO = PZ001R.Z1CTO
Z1PRAF = PZ001R.Z1PRAF
Z1PRCV = PZ001R.Z1PRCV
```

```
Z1PRCO = PZ001R.Z1PRCO
Z1PRCA = PZ001R.Z1PRCA
Z1RMS = PZ001R.Z1RMS
Z1LEGA = PZ001R.Z1LEGA
Z1SLPT = PZ001R.Z1SLPT
Z1USER = PZ001R.Z1USER
Z1UDAT = PZ001R.Z1UDAT
Z1UTIM = PZ001R.Z1UTIM
Z1EOSF = PZ001R.Z1EOSF
Z1CALC = PZ001R.Z1CALC
Z1RECP = PZ001R.Z1RECP.
```

```
/*-----*/
```

```
IN_IN [ 44 ] = "0".
```

```
END.
```

```
ELSE DO:
```

```
  PZ001R_RECID = ?.
```

```
  IN_IN [ 44 ] = "1".
```

```
END.
```

```
IF IN_IN [ 44 ] = "0" THEN DO:
```

```
  DO TRANSACTION:
```

```
  FIND PZ001 WHERE RECID(PZ001) = PZ001R_RECID EXCLUSIVE-LOCK.
```

```
  DELETE PZ001.
```

```
END.
```

```
END.
```

```
/* C*-----*/
```

```
/* C*- TABLE RECORD S
```

```
/* C*-----*/
```

```
RUN Q_$$S004.
```

```
/* $$S004 */
```

```
/* C*-----*/
```

```
END PROCEDURE.
```

```
/*=====*/
```

```
PROCEDURE Q_$$S003:
```

```
/*-- SECTION TYPE : SUBROUTINE
```

```
/* C*-
```

```
/* C*-----*/
```

```
/* C*- CHANGE PB006 RECORDS AS REQUIRED
```

```
/* C*-----*/
```

```
/* C*-
```

```
/* C*- =====
```

```
ASSIGN
```

```
  B6DATE = 0.
```

```
FIND FIRST PB006R USE-INDEX PB006A WHERE PB006R.B6DATE = B6DATE NO-LOCK NO-ERROR.
```

```
IF AVAILABLE PB006R THEN DO:
```

```
  PB006R_RECID = RECID(PB006R).
```

```
  /*-----*/
```

```
  ASSIGN
```

```
    B6DATE = PB006R.B6DATE
```

```
    B6TYPE = PB006R.B6TYPE
```

```
    B6ID = PB006R.B6ID
```

```
    B6GOVT = PB006R.B6GOVT.
```

```
  /*-----*/
```

```
  IN_IN [ 98 ] = "0".
```

```
END.
```

```
ELSE DO:
```

```
  PB006R_RECID = ?.
```

```
  IN_IN [ 98 ] = "1".
```

```
END.
```

```
FIND NEXT PB006R USE-INDEX PB006A NO-LOCK NO-ERROR.
```

```
IF AVAILABLE PB006R THEN DO:
```

```
  PB006R_RECID = RECID(PB006R).
```

```
  /*-----*/
```

```
  ASSIGN
```

```
    B6DATE = PB006R.B6DATE
```

```
    B6TYPE = PB006R.B6TYPE
```

```
    B6ID = PB006R.B6ID
```

```
    B6GOVT = PB006R.B6GOVT.
```

```
  /*-----*/
```

```
  IN_IN [ 98 ] = "0".
```

```
END.
```

```
ELSE DO:
```

```
  PB006R_RECID = ?.
```

```
  IN_IN [ 98 ] = "1".
```

```

END.

DOW_02:

DO WHILE IN_IN [ 98 ] = "0":

  IF B6TYPE = "B" AND B6ID = B1NUM THEN DO:
    ASSIGN
      B6GOVT = B1NBAT.

    DO TRANSACTION:
      FIND PB006 WHERE RECID(PB006) = PB006R_RECID EXCLUSIVE-LOCK.
      /*-----*/
      ASSIGN
        PB006.B6DATE = B6DATE
        PB006.B6TYPE = TRIM(B6TYPE)
        PB006.B6ID = TRIM(B6ID)
        PB006.B6GOVT = TRIM(B6GOVT) NO-ERROR.
      /*-----*/
    END.
  END.

  FIND NEXT PB006R USE-INDEX PB006A NO-LOCK NO-ERROR.

  IF AVAILABLE PB006R THEN DO:
    PB006R_RECID = RECID(PB006R).
    /*-----*/
    ASSIGN
      B6DATE = PB006R.B6DATE
      B6TYPE = PB006R.B6TYPE
      B6ID = PB006R.B6ID
      B6GOVT = PB006R.B6GOVT.

    /*-----*/
    IN_IN [ 98 ] = "0".
  END.
  ELSE DO:
    PB006R_RECID = ?.
    IN_IN [ 98 ] = "1".
  END.
END.

/* C*-
END PROCEDURE.
*/

/*=====*/

PROCEDURE Q_$$S004:
/*-- SECTION TYPE : SUBROUTINE -----*/
/* C*-----*/
/* C*---- DELETE ANY EXISTING TABLES FOR THIS ID */
ASSIGN
  T2ENVC = CTL#0B
/* C*- BAT ID */

  T2FLD = "BNUM "
  T2RL1C = ""
  T2RET = "" OVERLAY(T2RET, 1, 0005) = B1NUM.

FIND FIRST PTT02R USE-INDEX PTT02C WHERE PTT02R.T2FLD = T2FLD AND PTT02R.T2RL1C = T2RL1C AND PTT02R
.T2ENVC = T2ENVC AND PTT02R.T2RET = T2RET NO-LOCK NO-ERROR.

IF AVAILABLE PTT02R THEN DO:
  PTT02R_RECID = RECID(PTT02R).
  /*-----*/
  ASSIGN
    T2FLD = PTT02R.T2FLD
    T2RL1C = PTT02R.T2RL1C
    T2ENVC = PTT02R.T2ENVC
    T2TBEV = PTT02R.T2TBEV
    T2TBED = PTT02R.T2TBED
    T2RET = PTT02R.T2RET
    T2SORT = PTT02R.T2SORT.

  /*-----*/
  IN_IN [ 91 ] = "0".
END.
ELSE DO:
  PTT02R_RECID = ?.
  IN_IN [ 91 ] = "1".
END.

```

```

IF IN_IN [ 91 ] = "0" THEN DO:
DO TRANSACTION:
  FIND PTT02 WHERE RECID(PTT02) = PTT02R_RECID EXCLUSIVE-LOCK.
  DELETE PTT02.
END.
END.

/* C*-- BAT SEARCH NAME */
ASSIGN
  T2FLD = "BNUM "
  T2RL1C = "NAME "
  T2ENVC = ""
  T2ENVC = CTL#0B
  T2RET = "" OVERLAY(T2RET, 1, 0005) = B1NUM.

FIND FIRST PTT02R USE-INDEX PTT02C WHERE PTT02R.T2FLD = T2FLD AND PTT02R.T2RL1C = T2RL1C AND PTT02R
.T2ENVC = T2ENVC AND PTT02R.T2RET = T2RET NO-LOCK NO-ERROR.

IF AVAILABLE PTT02R THEN DO:
  PTT02R_RECID = RECID(PTT02R).
  /*-----*/
  ASSIGN
    T2FLD = PTT02R.T2FLD
    T2RL1C = PTT02R.T2RL1C
    T2ENVC = PTT02R.T2ENVC
    T2TBEV = PTT02R.T2TBEV
    T2TBED = PTT02R.T2TBED
    T2RET = PTT02R.T2RET
    T2SORT = PTT02R.T2SORT.

  /*-----*/
  IN_IN [ 91 ] = "0".
END.
ELSE DO:
  PTT02R_RECID = ?.
  IN_IN [ 91 ] = "1".
END.

IF IN_IN [ 91 ] = "0" THEN DO:
DO TRANSACTION:
  FIND PTT02 WHERE RECID(PTT02) = PTT02R_RECID EXCLUSIVE-LOCK.
  DELETE PTT02.
END.
END.

/* C*----- */
/* C*-- BAT GOV'T CODE */
ASSIGN
  T2FLD = "BNUM "
  T2RL1C = "GOVT "
  T2ENVC = ""
  T2ENVC = CTL#0B
  T2RET = "" OVERLAY(T2RET, 1, 0005) = B1NUM.

FIND FIRST PTT02R USE-INDEX PTT02C WHERE PTT02R.T2FLD = T2FLD AND PTT02R.T2RL1C = T2RL1C AND PTT02R
.T2ENVC = T2ENVC AND PTT02R.T2RET = T2RET NO-LOCK NO-ERROR.

IF AVAILABLE PTT02R THEN DO:
  PTT02R_RECID = RECID(PTT02R).
  /*-----*/
  ASSIGN
    T2FLD = PTT02R.T2FLD
    T2RL1C = PTT02R.T2RL1C
    T2ENVC = PTT02R.T2ENVC
    T2TBEV = PTT02R.T2TBEV
    T2TBED = PTT02R.T2TBED
    T2RET = PTT02R.T2RET
    T2SORT = PTT02R.T2SORT.

  /*-----*/
  IN_IN [ 91 ] = "0".
END.
ELSE DO:
  PTT02R_RECID = ?.
  IN_IN [ 91 ] = "1".
END.

IF IN_IN [ 91 ] = "0" THEN DO:

```

```
DO TRANSACTION:
  FIND PTT02 WHERE RECID(PTT02) = PTT02R_RECID EXCLUSIVE-LOCK.
  DELETE PTT02.
END.
```

END.

```
/* C* REMOVE PRAFAC PROCESSING */
/* C*----- */
```

END PROCEDURE.

```
/*-----*/
```

PROCEDURE Q\_\$\$S005:

```
/*-- SECTION TYPE : SUBROUTINE ---*/
```

```
/* C*----- */
```

```
/* C*---- WRITE NEW TABLES FOR THIS ID */
```

```
RUN Q_$$S004.
```

```
/* $$S004 */
```

```
/* C*---- BY ID */
```

ASSIGN

T2FLD = "BNUM "

T2RL1C = ""

T2ENVC = ""

T2RET = ""

T2TBED = ""

T2TBEV = ""

T2SORT = ""

```
/* C*----- */
```

T2ENVC = CTL#0B OVERLAY(T2TBED, 1, 0036) = B1NAME OVERLAY(T2TBEV, 1, 0005) = B1NUM OVERLAY(T2RET, 1, 0005) = B1NUM OVERLAY(T2SORT, 1, 0005) = B1NUM.

DO TRANSACTION:

CREATE PTT02 NO-ERROR.

```
/*-----*/
```

ASSIGN

PTT02.T2FLD = TRIM(T2FLD)

PTT02.T2RL1C = TRIM(T2RL1C)

PTT02.T2ENVC = TRIM(T2ENVC)

PTT02.T2TBEV = TRIM(T2TBEV)

PTT02.T2TBED = TRIM(T2TBED)

PTT02.T2RET = TRIM(T2RET)

PTT02.T2SORT = TRIM(T2SORT) NO-ERROR.

```
/*-----*/
```

END.

```
/* C*---- BY SEARCH NAME */
```

ASSIGN

T2FLD = "BNUM "

T2RL1C = "NAME "

T2ENVC = ""

T2RET = ""

T2TBED = ""

T2TBEV = ""

T2SORT = ""

```
/* C*----- */
```

T2ENVC = CTL#0B OVERLAY(T2TBED, 1, 0036) = B1NAME OVERLAY(T2TBEV, 1, 0010) = B1SNAM OVERLAY(T2RET, 1, 0005) = B1NUM OVERLAY(T2SORT, 1, 0010) = B1SNAM.

DO TRANSACTION:

CREATE PTT02 NO-ERROR.

```
/*-----*/
```

ASSIGN

PTT02.T2FLD = TRIM(T2FLD)

PTT02.T2RL1C = TRIM(T2RL1C)

PTT02.T2ENVC = TRIM(T2ENVC)

PTT02.T2TBEV = TRIM(T2TBEV)

PTT02.T2TBED = TRIM(T2TBED)

PTT02.T2RET = TRIM(T2RET)

PTT02.T2SORT = TRIM(T2SORT) NO-ERROR.

```
/*-----*/
```

END.

```
/* C*---- BY GOV'T CODE */
```

ASSIGN

T2FLD = "BNUM "

T2RL1C = "GOVT "

T2ENVC = ""

T2RET = ""

T2TBED = ""



```

T2TBEV = ""
T2SORT = ""
/* C*----- */

T2ENVC = CTL#0B OVERLAY(T2TBED, 1, 0036) = B1NAME OVERLAY(T2TBEV, 1, 0007) = B1NBAT OVERLAY(T2RET, 1
, 0005) = B1NUM OVERLAY(T2SORT, 1, 0007) = B1NBAT.

DO TRANSACTION:
CREATE PTT02 NO-ERROR.
/*-----*/
ASSIGN
    PTT02.T2FLD = TRIM(T2FLD)
    PTT02.T2RL1C = TRIM(T2RL1C)
    PTT02.T2ENVC = TRIM(T2ENVC)
    PTT02.T2TBEV = TRIM(T2TBEV)
    PTT02.T2TBED = TRIM(T2TBED)
    PTT02.T2RET = TRIM(T2RET)
    PTT02.T2SORT = TRIM(T2SORT) NO-ERROR.
/*-----*/
END.

/* C* REMOVE PRAFAC PROCESSING */
/* C*----- */
END PROCEDURE.

/*-----*/
/*--- EXFMT SCR1 -----*/

PROCEDURE TM001_F1:
/*-- SECTION TYPE : SCREEN ---*/
/*-----*/
/*===BUTTONS ENABLE===*/
{STDINC06.I "SCR1"}

{STDINC09.I "SCR1"}

/*===BUTTONS ENABLE===*/
{STDINC15.I " XPEMSG " " XPERR# "}

DISPLAY B1NUM WITH FRAME SCR1.
ASSIGN
    IN_KC = "0"
    IN_KD = "0"
    IN_KE = "0"
    IN_KF = "0"
    IN_KH = "0"
    IN_KI = "0"
    IN_KL = "0".

/*==== TRY WAIT-FOR =====*/
SET B1NUM GO-ON(F3) WITH FRAME SCR1.
ASSIGN
    B1NUM = CAPS(B1NUM).

{STDINC10.I}

/*-----*/
END PROCEDURE.

/*-----*/
/*--- EXFMT SCR2D -----*/

PROCEDURE TM001_F2:
/*-- SECTION TYPE : SCREEN ---*/
/*-----*/
/*===BUTTONS ENABLE===*/
{STDINC06.I "SCR2D"}

{STDINC04.I "S_0011" "SCR2D"}

{STDINC14.I "S_0011" "SCR2D"}

{STDINC04.I "S_0012" "SCR2D"}

{STDINC14.I "S_0012" "SCR2D"}

```

```
{STDINC04.I "S_0013" "SCR2D"}
```

```
{STDINC14.I "S_0013" "SCR2D"}
```

```
{STDINC04.I "S_0014" "SCR2D"}
```

```
{STDINC14.I "S_0014" "SCR2D"}
```

```
{STDINC09.I "SCR2D"}
```

```
/*===BUTTONS ENABLE===*/  
{STDINC15.I " XPEMSG " " XPERR# " }
```

```
DISPLAY B1NUM NEWFLD XXBNUM B1PROV PROVA S_0003 B1NBAT Z1OID B1OPR5 B1NAME B1SNAM B1TYPE B1GLCX  
B1GLSD B1GSCT B1GTWN B1GRNG B1GMER B1GESQ S_0004 B1S1 B1S2 B1PNM1 B1PNM2 B1PNM3 Z1FAX1 Z1FAX2 Z1FAX3  
CTL#30 B1XREF CTL#31 B1REF2 WITH FRAME SCR2D.
```

```
IF IN_IN [ 27 ] = "1" THEN NEXT-PROMPT B1PROV WITH FRAME SCR2D.  
ELSE IF IN_IN [ 21 ] = "1" THEN NEXT-PROMPT B1NBAT WITH FRAME SCR2D.  
ELSE IF IN_IN [ 22 ] = "1" THEN NEXT-PROMPT Z1OID WITH FRAME SCR2D.  
ELSE IF IN_IN [ 23 ] = "1" THEN NEXT-PROMPT B1NAME WITH FRAME SCR2D.  
ELSE IF IN_IN [ 26 ] = "1" THEN NEXT-PROMPT B1SNAM WITH FRAME SCR2D.  
ELSE IF IN_IN [ 30 ] = "1" THEN NEXT-PROMPT B1TYPE WITH FRAME SCR2D.  
ELSE IF IN_IN [ 24 ] = "1" THEN NEXT-PROMPT B1S1 WITH FRAME SCR2D.  
ELSE IF IN_IN [ 25 ] = "1" THEN NEXT-PROMPT B1S2 WITH FRAME SCR2D.  
ELSE IF IN_IN [ 28 ] = "1" THEN NEXT-PROMPT B1XREF WITH FRAME SCR2D.  
ELSE IF IN_IN [ 29 ] = "1" THEN NEXT-PROMPT B1REF2 WITH FRAME SCR2D.
```

```
ASSIGN  
IN_KC = "0"  
IN_KD = "0"  
IN_KE = "0"  
IN_KF = "0"  
IN_KH = "0"  
IN_KI = "0"  
IN_KL = "0".
```

```
/*==== TRY WAIT-FOR =====*/  
SET B1PROV B1NBAT Z1OID B1NAME B1SNAM B1TYPE B1GLCX B1GLSD B1GSCT B1GTWN B1GRNG B1GMER B1GESQ B1S1  
B1S2 B1PNM1 B1PNM2 B1PNM3 Z1FAX1 Z1FAX2 Z1FAX3 B1XREF B1REF2 GO-ON(F3 F4 F5 F6 F8 F9) WITH FRAME SCR2D.
```

```
ASSIGN OVERLAY(B1PNUM, 0001, 0003) = B1PNM1.  
ASSIGN OVERLAY(B1PNUM, 0004, 0003) = B1PNM2.  
ASSIGN OVERLAY(B1PNUM, 0007, 0004) = B1PNM3.  
ASSIGN
```

```
B1PROV = CAPS(B1PROV)  
B1NBAT = CAPS(B1NBAT)  
Z1OID = CAPS(Z1OID)  
B1NAME = CAPS(B1NAME)  
B1SNAM = CAPS(B1SNAM)  
B1TYPE = CAPS(B1TYPE)  
B1S1 = CAPS(B1S1)  
B1S2 = CAPS(B1S2)  
B1PNM1 = CAPS(B1PNM1)  
B1PNM2 = CAPS(B1PNM2)  
B1PNM3 = CAPS(B1PNM3)  
B1XREF = CAPS(B1XREF)  
B1REF2 = CAPS(B1REF2).
```

```
{STDINC10.I}
```

```
/*-----*/  
END PROCEDURE.
```

```
/*=====*/  
/*--- EXFMT SCR2 ---*/
```

```
PROCEDURE TM001_F3:
```

```
/*-- SECTION TYPE : SCREEN ---*/  
/*-----*/  
/*===BUTTONS ENABLE===*/  
{STDINC06.I "SCR2"}
```

```
{STDINC04.I "S_0006" "SCR2"}
```

```
{STDINC14.I "S_0006" "SCR2"}
```

```
{STDINC04.I "S_0007" "SCR2"}
```

```
{STDINC14.I "S_0007" "SCR2"}
```

```
{STDINC04.I "S_0008" "SCR2"}
```

```
{STDINC14.I "S_0008" "SCR2"}
```

```
{STDINC09.I "SCR2"}
```

```
/*====BUTTONS ENABLE====*/  
{STDINC15.I " XPEMSG " " XPERR# " }
```

```
DISPLAY B1NUM NEWFLD XXBNUM B1PROV PROVA S_0003 B1NBAT Z1OID B1OPR5 B1NAME B1SNAM B1TYPE B1GLCX  
B1GLSD B1GSCT B1GTWN B1GRNG B1GMER B1GESQ S_0004 B1S1 B1S2 B1PNM1 B1PNM2 B1PNM3 Z1FAX1 Z1FAX2 Z1FAX3  
CTL#30 B1XREF CTL#31 B1REF2 Z1PRAF Z1PRCV Z1PRCO Z1PRCA Z1CTO WITH FRAME SCR2.
```

```
IF IN_IN [ 27 ] = "1" THEN NEXT-PROMPT B1PROV WITH FRAME SCR2.  
ELSE IF IN_IN [ 21 ] = "1" THEN NEXT-PROMPT B1NBAT WITH FRAME SCR2.  
ELSE IF IN_IN [ 22 ] = "1" THEN NEXT-PROMPT Z1OID WITH FRAME SCR2.  
ELSE IF IN_IN [ 23 ] = "1" THEN NEXT-PROMPT B1NAME WITH FRAME SCR2.  
ELSE IF IN_IN [ 26 ] = "1" THEN NEXT-PROMPT B1SNAM WITH FRAME SCR2.  
ELSE IF IN_IN [ 30 ] = "1" THEN NEXT-PROMPT B1TYPE WITH FRAME SCR2.  
ELSE IF IN_IN [ 24 ] = "1" THEN NEXT-PROMPT B1S1 WITH FRAME SCR2.  
ELSE IF IN_IN [ 25 ] = "1" THEN NEXT-PROMPT B1S2 WITH FRAME SCR2.  
ELSE IF IN_IN [ 28 ] = "1" THEN NEXT-PROMPT B1XREF WITH FRAME SCR2.  
ELSE IF IN_IN [ 29 ] = "1" THEN NEXT-PROMPT B1REF2 WITH FRAME SCR2.  
ELSE IF IN_IN [ 31 ] = "1" THEN NEXT-PROMPT Z1PRAF WITH FRAME SCR2.  
ELSE IF IN_IN [ 33 ] = "1" THEN NEXT-PROMPT Z1PRCV WITH FRAME SCR2.  
ELSE IF IN_IN [ 32 ] = "1" THEN NEXT-PROMPT Z1PRCO WITH FRAME SCR2.  
ELSE IF IN_IN [ 34 ] = "1" THEN NEXT-PROMPT Z1PRCA WITH FRAME SCR2.
```

```
ASSIGN  
IN_KC = "0"  
IN_KD = "0"  
IN_KE = "0"  
IN_KF = "0"  
IN_KH = "0"  
IN_KI = "0"  
IN_KL = "0".
```

```
/*==== TRY WAIT-FOR =====*/  
SET B1PROV B1NBAT Z1OID B1NAME B1SNAM B1TYPE B1GLCX B1GLSD B1GSCT B1GTWN B1GRNG B1GMER B1GESQ B1S1  
B1S2 B1PNM1 B1PNM2 B1PNM3 Z1FAX1 Z1FAX2 Z1FAX3 B1XREF B1REF2 Z1PRAF Z1PRCV Z1PRCO Z1PRCA Z1CTO GO-ON  
(F3 F4 F5 F6 F8 F9) WITH FRAME SCR2.
```

```
ASSIGN OVERLAY(B1PNUM, 0001, 0003) = B1PNM1.  
ASSIGN OVERLAY(B1PNUM, 0004, 0003) = B1PNM2.  
ASSIGN OVERLAY(B1PNUM, 0007, 0004) = B1PNM3.  
ASSIGN
```

```
B1PROV = CAPS(B1PROV)  
B1NBAT = CAPS(B1NBAT)  
Z1OID = CAPS(Z1OID)  
B1NAME = CAPS(B1NAME)  
B1SNAM = CAPS(B1SNAM)  
B1TYPE = CAPS(B1TYPE)  
B1S1 = CAPS(B1S1)  
B1S2 = CAPS(B1S2)  
B1PNM1 = CAPS(B1PNM1)  
B1PNM2 = CAPS(B1PNM2)  
B1PNM3 = CAPS(B1PNM3)  
B1XREF = CAPS(B1XREF)  
B1REF2 = CAPS(B1REF2)  
Z1PRAF = CAPS(Z1PRAF)  
Z1PRCV = CAPS(Z1PRCV)  
Z1PRCO = CAPS(Z1PRCO)  
Z1PRCA = CAPS(Z1PRCA)  
Z1CTO = CAPS(Z1CTO).
```

```
{STDINC10.I}
```

```
/*-----*/  
END PROCEDURE.
```

```
/*=====  
/*--- EXFMT SCR3 ---*/
```

```
PROCEDURE TM001_F4:  
/*-- SECTION TYPE : SCREEN ---*/  
/*-----*/  
/*====BUTTONS ENABLE====*/  
{STDINC06.I "SCR3"}
```

```
{STDINC04.I "S_0017" "SCRN3"}
```

```
{STDINC14.I "S_0017" "SCRN3"}
```

```
{STDINC04.I "PRE-BOT" "SCRN3"}
```

```
{STDINC09.I "SCRN3"}
```

```
/*===BUTTONS ENABLE===*/
```

```
{STDINC15.I " XPEMSG " " XPERR# " }
```

```
DISPLAY B1NUM B1NAME WITH FRAME SCRN3.  
ASSIGN
```

```
IN_KC = "0"  
IN_KD = "0"  
IN_KE = "0"  
IN_KF = "0"  
IN_KH = "0"  
IN_KI = "0"  
IN_KL = "0".
```

```
/*===== TRY WAIT-FOR =====*/
```

```
WAIT-FOR GO OF PRM001.
```

```
RETURN.
```

```
/*=====*/
```

```
REPEAT:
```

```
  READKEY.
```

```
  IF KEYLABEL(LASTKEY) <> "RETURN" AND KEYLABEL(LASTKEY) <> "F1" AND KEYLABEL(LASTKEY) <> "F3" AND  
    KEYLABEL(LASTKEY) <> "F4" AND KEYLABEL(LASTKEY) <> "F12" THEN DO:
```

```
    BELL.  
    NEXT.  
  END.  
  ELSE DO:  
    LEAVE.  
  END.
```

```
END.
```

```
{STDINC10.I}
```

```
/*-----*/
```

```
END PROCEDURE.
```