

!*****

! DDP001: SAMPLE PROGRAM - may need modifications to meet specific *

! needs. *

! Write Direct Deposit Transmittal file *

! Discription: Creates one file containing direct deposit *

! and Prenotification records for each Company/Bank Code *

! combination within RunID. To create a report of all *

! MULTI CO prenotification records run DDP005 before executing *

! this program. To create prenotification memos run *

! DDP006 prior to this program. *

!*****

!*****

! *

! *

! *

! *

! This software and related documentation are provided under a *

! license agreement containing restrictions on use and *

! disclosure and are protected by intellectual property *

! laws. Except as expressly permitted in your license agreement *

! or allowed by law, you may not use, copy, reproduce, *

! translate, broadcast, modify, license, transmit, distribute, *

! exhibit, perform, publish or display any part, in any form or *

! by any means. Reverse engineering, disassembly, or *

! decompilation of this software, unless required by law for *

! interoperability, is prohibited. *

! The information contained herein is subject to change without *
! notice and is not warranted to be error-free. If you find any *
! errors, please report them to us in writing. *

! *
! *

! Copyright (C) 1988, 2014, Oracle and/or its affiliates. *
! All Rights Reserved. *

!*****

!*****

! *
! \$Date: 2014/08/18:16:39:04 !
! \$Release: HR92 !
! \$Revision: 111 !

! *
! *

!*****

!*****

! Project Port Authority CHANGE LOG *

! *

! DO NOT DELETE ANYTHING UNLESS OTHERWISE NOTED!!! *

! *

! All modifications to this SQR module must be documented. *

! Modifications should be preceded by the following line: *

! ! Modification Start *

! Modifications should be followed by the following line: *

! ! Modification End *

! For changes that do not have easily determined start and end points, *
! the information in the change log will suffice. *

!*****

! SIR Programmer Start Dt End Date Orig. Pgm *
! ---- - - - - - - - - - - *

! 339 Flaherty,Andrew 10/16/98 XX/XX/XX XXXX *
! *

!Description of Change: *

!The file path for Direct Deposit Transmittal needed to be changed. It *

!Curently reads {FILEPREFIX}||\$FileName||{FILESUFFIX}. It needed to *
!be changed to {INTERHOME}/citibank/{OUTBOUND}/\$Filename *

!*****

!*****

! SIR Programmer Start Dt End Date Orig. Pgm *
! ---- - - - - - - - - - - *

! 370 Logan,Danielle 10/21/98 10/21/98 PSOFT *
! *

! Description of Change: *

! The file path for Direct Deposit Transmittal was changed to *
! concatenate \$Filename to the literal. *

!*****

! SIR Programmer Start Dt End Date Orig. Pgm *
! ---- - - - - - - - - - - *

! 0750 Logan,Danielle 11/25/98 11/25/98 PSOFT *
! Description of Change *

! *

! The batch header was altered to reflect Citibank's requested *

! changes and processing for offset transactions was suppressed. *

!*****

! SIR Programmer Start Dt End Date Orig. Pgm *

! ---- - - - - - - - - - - - - - - - *

! 0999 Logan,Danielle 12/07/98 12/07/98 PSOFT *

! *

! Description of Change *

! *

! The batch-control and file-control were altered so that the *

! \$TotBatchDebitAmt and \$TotFileDebitAmt print zero amounts to the *

! interface file. *

!*****

! SIR Programmer Start Dt End Date Orig. Pgm *

! ---- - - - - - - - - - - - - - - - *

! PATH Crestone 12/10/01 12/10/01 PSOFT *

! *

! Description of Change *

! *

!*****

! SIR Programmer Start Dt End Date Orig. Pgm *

! ---- - - - - - - - - - - - - - - - *

! 1453 AF - Crestone 09/09/04 09/09/04 PSOFT *

! *

! Description of Change *

! *

! - Add code for the creation of supplemental file. *

! - Add code to send summary report via email. *

!

! 2351 KB - Crestone 11/06/12 11/06/04 PSOFT *

! *

! Description of Change *

! *

! - Modify email wording. *

!*****

! PHIRE Programmer Start Dt End Date Orig. Pgm *

! ---- ----- *

! 0435 SIERRA CEDAR 06/16/21 06/16/21 PSOFT *

! *

! Description of Change *

! *

! - Add total amount for debits *

! *

! PHIRE Programmer Start Dt End Date Orig. Pgm *

! ---- ----- *

! 0828 SIERRA CEDAR 08/17/23 08/17/23 PSOFT *

! *

! Description of Change *

! *

! - PATH is changing banks to CITI. Remove industrial bank references*

! *

!*****

!*****

```
#Include 'setenv.sqc'  !Set environment

#Include 'setup31.sqc' !Printer and page-size initialization

#Include 'getcodta.sqc' !Get-Company-Data procedure

#Include 'curdtim.sqc' !Get-Current-DateTime procedure

#Include 'payinit.sqc' !Report Initialization and Timing

#Include 'datetime.sqc' !Routines for date and time formatting

#Include 'number.sqc' !Routines to format numbers

#Include 'stdapi.sqc' !Update Process API

!*****PATH Modification Begin

!#Include 'payrnctl.sqc' !Get-Run-Control procedure

!*****PATH Modification End

#Include 'getaddr.sqc' !Get Person Current Address

#include 'rotname1.sqc' !Rotate-Name procedure


begin-report

  do Init-Report

  do Stdapi-Term

end-report


begin-procedure Init-Report


  display 'Creating Direct Deposit Transmittal Files'
```

do Init-DateTime

do Init-Number

do Stdapi-Init

do Create-Array

do Format-DateTime('20090917',\$IATStrtDt,{DEFCMP},"','native')

do Format-DateTime(\$IATStrtDt,\$CMP-IATStrtDt,{DEFCMP},"','")

do Format-DateTime(\$AsOfToday,\$CMP-AsOfToday,{DEFCMP},"','")

if \$CMP-AsOfToday > \$CMP-IATStrtDt

 let \$IAT-Yes = 'Y'

else

 let \$IAT-Yes = 'N'

end-if

move 'N' to \$CyclePrompts !Don't prompt for On/Off cycle in PAYINIT

let \$Batch1_last_rec_IAT = 'N'

do Payroll-Report-Initialization

end-procedure

!*** SR1453 - Modification Begin

begin-procedure getEmailAddresses

let \$email = "

BEGIN-SELECT

EMAIL_ADDR &OUTPUT_EMAIL_ADDR

if \$email = "

let \$email = &OUTPUT_EMAIL_ADDR

else

let \$email = &OUTPUT_EMAIL_ADDR || ' ' || \$email

end-if

FROM PS_Y_RC_DDP001B

WHERE OPRID = \$PRCS_OPRID

AND RUN_CNTL_ID = \$PRCS_RUN_CNTL_ID

END-SELECT

end-procedure !getEmailAddresses

!*** SR1453 - Modification End

!*****PATH Modification Begin

begin-procedure Select-Parameters

begin-select

RC_PAYINIT.RUN_ID

RC_PAYINIT.COMPANY

RC_PAYINIT.PAYGROUP

RC_PAYINIT.PAY_END_DT

RC_PAYINIT.CYCLE_SELECT

RC_PAYINIT.PRENOTE_DT

RC_PAYINIT.TEXT20

RC_PAYINIT.PAGE_NUM

RC_PAYINIT.PAGE_NUM_THRU

RC_PAYINIT.FILE_PATH

!*** SR1453 - Modification Begin

substr(rc_payinit.run_id, 5, 2) &FILETYPE

!*** SR1453 - Modification End

FROM PS_Y_RC_PAY RC_PAYINIT

WHERE RC_PAYINIT.OPRID = \$Prcs_OprID

AND RC_PAYINIT.RUN_CNTL_ID = \$Prcs_Run_Cntl_ID

end-select

let \$SRC_PAYINIT.Run_ID = RTRIM(&RC_PAYINIT.Run_ID, ' ')

let \$SRC_PAYINIT.Company = RTRIM(&RC_PAYINIT.Company, ' ')

let \$SRC_PAYINIT.PayGroup = RTRIM(&RC_PAYINIT.PayGroup, ' ')

let \$SRC_PAYINIT.Pay_End_Dt = &RC_PAYINIT.Pay_End_Dt

let \$SRC_PAYINIT.Cycle_Select = RTRIM(&RC_PAYINIT.Cycle_Select, ' ')

let \$FILE_PATH = RTRIM(&RC_PAYINIT.FILE_PATH, ' ')

let \$SRC_PAYINIT.Text20 = RTRIM(&RC_PAYINIT.Text20, ' ')

let \$SRC_PAYINIT.Prenote_Dt = &RC_PAYINIT.Prenote_Dt

!*** SR1453 - Modification Begin

let \$fileType = &FILETYPE

!*** SR1453 - Modification End

end-procedure

!*****PATH Modification End

begin-PROCEDURE Create-Array

move 8 to #lastIndex !Index of last digit in Transit Routing Number

create-array name=Weights size=1 field=Factor:number:9

put 3 7 1 3 7 1 3 7 into Weights(0) -

Factor(0) Factor(1) Factor(2) Factor(3) -

Factor(4) Factor(5) Factor(6) Factor(7)

end-PROCEDURE !Create-Array

begin-PROCEDURE Report

if \$PrCs_Process_Instance = "

input \$Answer 'Enter Prenote Date ({Prompt-Date})' type=char

move '1' to \$MMLZero

do Format-DateTime(\$Answer, \$out, {Prompt-Mask}, ", ")

if {Prompt-Mask} = {DEFMDY}

let \$mm = substr(\$out, 1, 2)

else !DEFYMD or DEFDMY

let \$mm = substr(\$out, 4, 2)

end-if

do Format-DateTime(\$Answer, \$Prenote_Date, {Prompt-Mask}, ", 'native')

display 'PRENOTE DATE ' noline

display \$Prenote_Date

input \$CustDiscrData 'Enter Description (20 character max)'

else

move \$RC_PAYINIT.Prenote_Dt to \$Prenote_Date

move \$RC_PAYINIT.Text20 to \$CustDiscrData

end-if

move 'X' to \$Details_Processed

move 'N' to \$FileOpen

move 'N' to \$FileOpenIAT

move 'N' to \$BatchOpen

move 'N' to \$BatchOpenIAT

move 'N' to \$NewFile

move 'N' to \$NewFileIAT

move 'N' to \$NewBatch

move 'N' to \$NewBatchIAT

move 0 to #filenum

move 0 to #Batch#

move 0 to #File_Count

move ' ' to \$Prev_IAT_Ind

move 'N' to \$Offset

move 'N' to \$OffsetIAT

move ' ' to \$PriorPaygroup

let \$Verify-New-batch-header = 'N'

move 'N' to \$FromPrenotes

Show 'Selection: ' \$SlctCalendar

if \$SlctRunID <> '*'

move 'AA.RUN_ID = '' to \$DdovbankID

concat \$SlctRunID with \$DdovbankID

concat '' with \$DdovbankID

else

move 'AA.COMPANY = '' to \$DdovbankID

concat \$SlctCompany with \$DdovbankID

concat '' AND AA.PAYGROUP = '' with \$DdovbankID

concat \$SlctPayGroup with \$DdovbankID

concat '' AND AA.PAY_END_DT = '' with \$DdovbankID

concat \$SlctPayEndDt with \$DdovbankID

concat '' with \$DdovbankID

end-if

begin-SELECT

AA.COMPANY

AA.PAYGROUP

AA.PAY_END_DT

AA.CHECK_DT

BB.SRC_BANK_ID

BB.DD_SRC_BANK_ID

if isblank(&BB.DD_Src_bank_id)

move &BB.Src_bank_id to \$Bank_ID

move 'B.SRC_BANK_ID' to \$Bank_ID_to_Use_B

move 'BS.SRC_BANK_ID' to \$Bank_ID_to_Use_BS

move 'PT.SRC_BANK_ID' to \$Bank_ID_to_Use_PT

move 'PT1.SRC_BANK_ID' to \$Bank_ID_to_Use_PT1

move 'PT1S.SRC_BANK_ID' to \$Bank_ID_to_Use_PT1S

else

move &BB.DD_Src_bank_id to \$Bank_ID

move 'B.DD_SRC_BANK_ID' to \$Bank_ID_to_Use_B

move 'BS.DD_SRC_BANK_ID' to \$Bank_ID_to_Use_BS

move 'PT.DD_SRC_BANK_ID' to \$Bank_ID_to_Use_PT

move 'PT1.DD_SRC_BANK_ID' to \$Bank_ID_to_Use_PT1

move 'PT1S.DD_SRC_BANK_ID' to \$Bank_ID_to_Use_PT1S

end-if

FROM PS_PAY_CALEDAR AA,

PS_PAYGROUP_TBL BB

WHERE [\$DdovbankID]

AND AA.PAYGROUP = BB.PAYGROUP

```

AND AA.COMPANY      = BB.COMPANY

AND BB.EFFDT        =

  (SELECT MAX(EFFDT)

    FROM PS_PAYGROUP_TBL

   WHERE COMPANY    = AA.COMPANY

        AND PAYGROUP = AA.PAYGROUP

        AND EFF_STATUS = 'A'

        AND EFFDT    <= AA.PAY_END_DT)

AND AA.PAY_CONFIRM_RUN = 'Y'

AND AA.PAY_CHECKS_RUN  = 'N'

ORDER BY AA.COMPANY,

        AA.PAYGROUP,

        AA.PAY_END_DT

end-SELECT

```

```

if isblank($Bank_ID_to_Use_B)

  move "" to $Bank_ID_to_Use_B

end-if

```

```

if isblank($Bank_ID_to_Use_BS)

  move "" to $Bank_ID_to_Use_BS

end-if

```

```

if isblank($Bank_ID_to_Use_PT)

  move "" to $Bank_ID_to_Use_PT

end-if

```

```
if isblank($Bank_ID_to_Use_PT1)

  move "" to $Bank_ID_to_Use_PT1

end-if
```

```
if isblank($Bank_ID_to_Use_PT1S)

  move "" to $Bank_ID_to_Use_PT1S

end-if
```

!Port Change

```
! if $RC_PAYINIT.Py_ddp_sep_files = 'N'

  do Process-Combined-Files

! else

!   do Process-Separate-Files

! end-if
```

end-PROCEDURE

begin-PROCEDURE Process-Combined-Files

begin-SELECT

A.COMPANY

A.PAYGROUP

A.PAY_END_DT

A.CHECK_DT

S.BANK_CD

S.ACCOUNT_EC_ID

S.DD_ACCOUNT_NBR

S.PRENOTE_WAIT_DAYS

S.IAT_IND

BT.ALT_BANK_CD

Move &A.Company to \$A.Company

Move &A.Paygroup to \$A.Paygroup

Move &A.Pay_End_Dt to \$A.Pay_End_Dt

Move &A.Check_Dt to \$A.Check_Dt

Move 'N' to \$Offset-IAT-Written

move &S.PRENOTE_WAIT_DAYS to #WaitDays

let \$ComplATIND = rtrim(&S.IAT_IND, ' ')

if isblank(&BT.Alt_bank_cd)

move &S.Bank_cd to \$Transit#

move 'SB.BANK_CD' to \$Transit_to_Use

move 'SB1.BANK_CD' to \$Transit_to_Use_IAT

move 'SB1S.BANK_CD' to \$Transit_to_Use_IATS

else

move &BT.Alt_bank_cd to \$Transit#

move 'BTB.ALT_BANK_CD' to \$Transit_to_Use


```
move 'BTB1.ALT_BANK_CD' to $Transit_to_Use_IAT
move 'BTB1S.ALT_BANK_CD' to $Transit_to_Use_IATS
end-if
```

```
if isblank(&S.DD_Account_Nbr)
    move &S.Account_ec_id to $Account#
else
    move &S.DD_Account_Nbr to $Account#
end-if
```

```
do Get-Transit-Descr
```

```
Move $PriorCompany to $CCD_Batch_Company
```

```
if $PriorCompany <> $A.Company
```

```
    if $FileOpen ='Y'
```

```
! SR 1347 - 01/28/2004 - ADDED    If &A.COMPANY = 'TH'
```

```
    If &A.COMPANY = 'NO'
```

```
        if #BatchEntryCount > 0
```

```
            do Write-Offset-Record-Sep
```

```
        end-if
```

```
    END-IF
```

```
    if $Details_Processed <> 'X'
```

if \$BatchOpen ='Y'

if(\$ComplATIND = 'Y' or \$DD1.IAT_ind = 'Y') and

\$IAT-Yes = 'Y'

do Write-Batch-Control

end-if

end-if

do Write-Prenotes

end-if

if \$BatchOpen ='Y'

do Write-Batch-Control

end-if

if #ComplATEntry > 0 or

#CompEntry > 0

if \$Offset-IAT-Written <> 'Y'

if \$RC_PAYINIT.Py_ddp_excl_offs = 'N'

!Port Change

! do Write-Offset-Record-IAT

move 'Y' to \$Offset-IAT-Written

end-if

end-if

move 0 to #ComplATEntry

move 0 to #CompEntry

end-if

else

if \$Details_Processed = 'N' !First time equals X

if \$BatchOpen ='Y'

if(\$ComplATIND = 'Y' or \$DD1.IAT_ind = 'Y') and

\$IAT-Yes = 'Y'

do Write-Batch-Control

end-if

end-if

do Write-Prenotes

if \$BatchOpen ='Y'

do Write-Batch-Control

end-if

end-if

end-if

move 'Y' to \$NewBatch

move \$A.Company to \$PriorCompany

move \$A.Company to \$CCD_Batch_Company

move \$A.Check_Dt to \$priorCHECK_DT

move \$A.Pay_End_Dt to \$priorPay_End_Dt

move \$A.Company to \$Company

```

do Get-Company-Data

!*PATH bank change

Do Get-PA-Company-Data

move $CompanyName to $Batch_CompanyName

UPPERCASE $Batch_CompanyName

move &CT.Federal_EIN to $Federal_EIN 099999999

let $Batch_CompanyID = '1' || $Federal_EIN

!*** SR1453 - Modification Begin

if $fileType = 'MS'

    let $Batch_CompanyID = '9' || $Federal_EIN

end-if

!*** SR1453 - Modification End

end-if

if $Transit# <> $priorTransit#

if $FileOpen ='Y'

! SR 1347 - 01/28/2004 - ADDED If &A.COMPANY = 'TH'

If &A.COMPANY = 'NO'

    if #BatchEntryCount > 0

        do Write-Offset-Record-Sep

    end-if

END-IF

if $Details_Processed <> 'X'

    if ($ComplATIND = 'Y' or $DD1.IAT_ind = 'Y') and

        $IAT-Yes = 'Y'

```

```
if $BatchOpen ='Y'  
    do Write-Batch-Control  
end-if
```

```
do Write-Prenotes  
else
```

```
do Write-Prenotes
```

```
if $BatchOpen ='Y'  
    do Write-Batch-Control  
end-if  
end-if
```

```
end-if
```

```
if $BatchOpen ='Y'  
    do Write-Batch-Control  
end-if
```

```
if #ComplATEntry > 0 or  
    #CompEntry > 0  
if $Offset-IAT-Written <> 'Y'  
    if $RC_PAYINIT.Py_ddp_excl_offs = 'N'
```

!Port Change

```
!      do Write-Offset-Record-IAT
      move 'Y' to $Offset-IAT-Written
      end-if
    end-if

    move 0 to #CompIATEntry
    move 0 to #CompEntry
  end-if

  do Write-File-Control
  do Close-File
else
  if $Details_Processed = 'N'      !First time equals X

  do Write-Prenotes

  if $FileOpen ='Y'

    if $BatchOpen ='Y'

      do Write-Batch-Control
    end-if

    if $RC_PAYINIT.Py_ddp_excl_offs = 'N'
```

!Port Change

```
!      do Write-Offset-Record-IAT
```

end-if

do Write-File-Control

do Close-File

end-if

end-if

end-if

move 'Y' to \$NewFile

move 'Y' to \$NewBatch

move \$Transit# to \$priorTransit#

move \$A.Check_Dt to \$priorCHECK_DT

move \$A.Pay_End_Dt to \$priorPay_End_Dt

else

if \$PriorPaygroup <> \$A.Paygroup and

\$PriorPaygroup <> ' ' and

\$PriorCompany = \$A.Company

if \$FileOpen ='Y'

if \$BatchOpen ='Y'

let \$Verify-New-batch-header = 'Y'

end-if

end-if

end-if

end-if

if \$A.Check_Dt <> \$priorCHECK_DT

if \$FileOpen ='Y'

! SR 1347 - 01/28/2004 - ADDED If &A.COMPANY = 'TH'

If &A.COMPANY = 'NO'

if #BatchEntryCount > 0

do Write-Offset-Record-Sep

end-if

end-if

if \$Details_Processed <> 'X'

if \$BatchOpen ='Y' and

\$DD1.IAT_ind = 'Y'

do Write-Batch-Control

end-if

do Write-Prenotes

end-if

if \$BatchOpen ='Y'

do Write-Batch-Control

end-if

if #ComplATEntry > 0 or

#CompEntry > 0

if \$Offset-IAT-Written <> 'Y'

if \$RC_PAYINIT.Py_ddp_excl_offs = 'N'

!Port Change

! do Write-Offset-Record-IAT

move 'Y' to \$Offset-IAT-Written

end-if

end-if

move 0 to #CompIATEntry

move 0 to #CompEntry

end-if

else

if \$Details_Processed = 'N' !First time equals X

do Write-Prenotes

if \$FileOpen ='Y'

if \$BatchOpen ='Y'

do Write-Batch-Control

end-if

end-if

end-if

end-if

move 'Y' to \$NewBatch

move \$A.Check_Dt to \$priorCHECK_DT

move \$A.Pay_End_Dt to \$priorPay_End_Dt

end-if

move \$Account# to \$priorAccount#

move \$A.Paygroup to \$PriorPaygroup

move \$A.Check_Dt to \$priorCHECK_DT

move \$A.Pay_End_Dt to \$priorPay_End_Dt

do Process-Entry-Detail

do Write-Prenotes

FROM PS_PAY_CALENDAR A,

PS_PAYGROUP_TBL B,

PS_SRC_BANK S,

PS_BANK_EC_TBL BT

WHERE [\$SlctCalendar]

AND A.PAYGROUP = B.PAYGROUP

AND A.COMPANY = B.COMPANY

AND B.EFFDT =

(SELECT MAX(EFFDT)

```

FROM PS_PAYGROUP_TBL

WHERE COMPANY   = A.COMPANY

AND PAYGROUP   = A.PAYGROUP

AND EFF_STATUS = 'A'

AND EFFDT      <= A.PAY_END_DT)

AND A.PAY_CONFIRM_RUN = 'Y'

AND A.PAY_CHECKS_RUN  = 'N'

AND S.SRC_BANK_ID     = [$Bank_ID_to_Use_B]

AND BT.BANK_CD        = S.BANK_CD

AND BT.COUNTRY_CD     = 'USA'

ORDER BY A.COMPANY,

        S.BANK_CD,

        A.PAYGROUP

end-SELECT

```

```

if $FileOpen ='Y'

```

```

! ** SIR0750 Modification Begin **

```

```

!*****

```

```

! This procedure was used to write a debit to the interface file in the *

```

```

! amount of the total for all employee transactions. Citibank does not *

```

```

! wish this debit line to print. They will either sum the transactions *

```

```

! themselves or pull the totals line from the file control data.      *

```

```

!*****

```

```

! 12/12/2001 - City National wants the offset record

```

If &A.COMPANY = 'NO'

if #BatchEntryCount > 0

do Write-Offset-Record-Sep

end-if

End-If

!** SIR0750 Modification End **

if \$BatchOpen = 'Y' and

(\$ComplATIND = 'Y' or \$DD1.IAT_ind = 'Y') and

\$IAT-Yes = 'Y'

do Write-Batch-Control

move 0 to #TotBatchCreditAmt

move 0 to #BatchEntryHash

move 0 to #BatchEntryCount !Batch control

move 0 to #Temp_TotBatchCreditAmt

do Write-Prenotes

end-if

if \$BatchOpen = 'Y'

do Write-Batch-Control

end-if

if #ComplATEntry > 0 or

#CompEntry > 0

if \$RC_PAYINIT.Py_ddp_excl_offs = 'N'

!Port Change

```
!    do Write-Offset-Record-IAT
```

```
    end-if
```

```
end-if
```

```
do Write-File-Control
```

```
do Close-File
```

```
else
```

```
if $NewFile = 'Y'
```

```
    move 'N' to $NewFile
```

```
    do Start-New-File
```

```
    do Start-New-Batch
```

```
    do Write-Prenotes
```

```
    do Write-Batch-Control
```

```
    if #ComplATEntry > 0 or
```

```
        #CompEntry > 0
```

```
        if $RC_PAYINIT.Py_ddp_excl_offs = 'N'
```

```
            do Write-Offset-Record-IAT
```

```
        end-if
```

```
    end-if
```

```
do Write-File-Control
```

```
do Close-File
```

```
end-if
```

```
end-if
```

```
if #File_Count = 0
```

```
    display ' '
```

```
    display 'No direct deposit output file has been generated.'
```

display ' '

end-if

end-PROCEDURE

!*PATH BANK CHANGE

Begin-Procedure Get-PA-Company-Data

begin-SELECT

CTPA.FEDERAL_EIN

FROM PS_COMPANY_TBL CTPA

WHERE CTPA.COMPANY = 'PA'

AND CTPA.EFFDT =

(SELECT MAX(EFFDT)

FROM PS_COMPANY_TBL

WHERE COMPANY = \$Company

AND EFFDT <= \$AsOfDate)

end-SELECT

End-Procedure

begin-PROCEDURE Process-Separate-Files

if \$SlctRunID <> '*'

move 'Z.RUN_ID = '' to \$SlctCalendarS

concat \$SlctRunID with \$SlctCalendarS

```
concat '' with $SlctCalendarS

else

move 'Z.COMPANY = '' to $SlctCalendarS

concat $SlctCompany with $SlctCalendarS

concat '' AND Z.PAYGROUP = '' with $SlctCalendarS

concat $SlctPayGroup with $SlctCalendarS

concat '' AND Z.PAY_END_DT = '' with $SlctCalendarS

concat $SlctPayEndDt with $SlctCalendarS

concat '' with $SlctCalendarS

end-if
```

begin-SELECT

```
Z.COMPANY
Z.PAYGROUP
Z.PAY_END_DT
Z.CHECK_DT
SS.BANK_CD
SS.ACCOUNT_EC_ID
SS.DD_ACCOUNT_NBR
SS.IAT_IND
BTS.ALT_BANK_CD
```

```
Move &Z.Company to $A.Company
Move &Z.Paygroup to $A.Paygroup
Move &Z.Pay_End_Dt to $A.Pay_End_Dt
```

Move &Z.Check_Dt to \$A.Check_Dt

let \$ComplATIND = rtrim(&SS.IAT_IND,' ')

if isblank(&BTS.Alt_bank_cd)

move &SS.Bank_cd to \$Transit#

move 'SB.BANK_CD' to \$Transit_to_Use

move 'SB1.BANK_CD' to \$Transit_to_Use_IAT

move 'SB1S.BANK_CD' to \$Transit_to_Use_IATS

else

move &BTS.Alt_bank_cd to \$Transit#

move 'BTB.ALT_BANK_CD' to \$Transit_to_Use

move 'BTB1.ALT_BANK_CD' to \$Transit_to_Use_IAT

move 'BTB1S.ALT_BANK_CD' to \$Transit_to_Use_IATS

end-if

if isblank(&SS.DD_Account_Nbr)

move &SS.Account_ec_id to \$Account#

else

move &SS.DD_Account_Nbr to \$Account#

end-if

do Get-Transit-Descr

if \$PriorCompany <> \$A.Company

if \$FileOpen ='Y' or \$FileOpenIAT ='Y'

if \$FileOpen ='Y' !Domestic transactions

if #CompEntry > 0

if \$SRC_PAYINIT.Py_ddp_excl_offs = 'N'

do Write-Offset-Record-Sep

move 0 to #CompEntry

end-if

end-if

if \$BatchOpen ='Y'

do Write-Batch-Control

end-if

if \$Details_Processed <> 'X'

do Write-Prenotes

end-if

end-if

if \$FileOpenIAT ='Y' !!IAT transactions

if \$BatchOpenIAT ='Y'

do Write-Batch-Control-IAT-Sep

end-if

```
if #ComplATEntry > 0
  if $RC_PAYINIT.Py_ddp_excl_offs = 'N'
    do Write-Offset-Record-IAT-Sep
    move 0 to #ComplATEntry
  end-if
end-if
```

```
if $BatchOpenIAT ='Y'
  do Write-Batch-Control-IAT-Sep
end-if
```

```
end-if
```

```
else
```

```
if $Details_Processed = 'N'      !First time equals X
  do Write-Prenotes
```

```
  if $BatchOpen ='Y'
    do Write-Batch-Control
  end-if
```

```
  if $BatchOpenIAT ='Y'
    do Write-Batch-Control-IAT-Sep
  end-if
```

```
end-if
```

end-if

move 'Y' to \$NewBatch

move 'Y' to \$NewBatchIAT

move \$A.Company to \$PriorCompany

move \$A.Company to \$Company

move \$A.Check_Dt to \$priorCHECK_DT

move \$A.Pay_End_Dt to \$priorPay_End_Dt

do Get-Company-Data

move \$CompanyName to \$Batch_CompanyName

UPPERCASE \$Batch_CompanyName

move &CT.Federal_EIN to \$Federal_EIN 099999999

let \$Batch_CompanyID = '1' || \$Federal_EIN

end-if

if \$Transit# <> \$priorTransit#

if \$FileOpen ='Y' or \$FileOpenIAT ='Y'

if \$FileOpen ='Y'

if #CompEntry > 0

if \$RC_PAYINIT.Py_ddp_excl_offs = 'N'

do Write-Offset-Record-Sep

end-if

move 0 to #CompEntry

end-if

if \$Details_Processed <> 'X'

do Write-Prenotes

end-if

if \$BatchOpen ='Y'

do Write-Batch-Control

end-if

do Write-File-Control

do Close-File

end-if

if \$FileOpenIAT ='Y'

if \$BatchOpenIAT ='Y'

do Write-Batch-Control-IAT-Sep

end-if

if #ComplATEntry > 0

if \$RC_PAYINIT.Py_ddp_excl_offs = 'N'

do Write-Offset-Record-IAT-Sep

end-if

move 0 to #ComplATEntry

end-if

do Write-File-Control-IAT-Sep

do Close-File-IAT

end-if

else

if \$Details_Processed = 'N' !First time equals X

if \$FileOpen ='Y'

do Write-Prenotes

if \$BatchOpen ='Y'

do Write-Batch-Control

end-if

if #ComplATEntry > 0 or

#CompEntry > 0

if \$SRC_PAYINIT.Py_ddp_excl_offs = 'N'

do Write-Offset-Record-IAT

end-if

end-if

do Write-File-Control

do Close-File

end-if

end-if

end-if

move 'Y' to \$NewFile

move 'Y' to \$NewBatch

move 'Y' to \$NewFileIAT

move 'Y' to \$NewBatchIAT

move \$Transit# to \$priorTransit#

move \$A.CHECK_DT to \$priorCHECK_DT

move \$A.Pay_End_Dt to \$priorPay_End_Dt

end-if

if \$A.CHECK_DT <> \$priorCHECK_DT

if \$FileOpen ='Y' or \$FileOpenIAT ='Y'

if \$FileOpen ='Y'

if #CompEntry > 0

if \$RC_PAYINIT.Py_ddp_excl_offs = 'N'

do Write-Offset-Record-Sep

end-if

move 0 to #CompEntry

end-if

if \$Details_Processed <> 'X'

do Write-Prenotes

end-if

if \$BatchOpen ='Y'

do Write-Batch-Control

move 'Y' to \$NewBatch

end-if

end-if

if \$FileOpenIAT ='Y'

if \$BatchOpenIAT ='Y'

do Write-Batch-Control-IAT-Sep

move 'Y' to \$NewBatchIAT

end-if

if #ComplATEntry > 0

if \$SRC_PAYINIT.Py_ddp_excl_offs = 'N'

do Write-Offset-Record-IAT-Sep

end-if

move 0 to #ComplATEntry

end-if

end-if

else

if \$Details_Processed = 'N' !First time equals X

if \$FileOpen ='Y'

do Write-Prenotes

if \$BatchOpen ='Y'

do Write-Batch-Control

move 'Y' to \$NewBatch

end-if

end-if

end-if

end-if

move \$A.CHECK_DT to \$priorCHECK_DT

move \$A.Pay_End_Dt to \$priorPay_End_Dt

end-if

move \$Account# to \$priorAccount#

move \$A.Paygroup to \$PriorPaygroup

do Process-Entry-Detail-Separate

do Write-Prenotes

FROM PS_PAY_CALENDAR Z,


```

    PS_PAYGROUP_TBL BS,

    PS_SRC_BANK    SS,

    PS_BANK_EC_TBL BTS

WHERE [$SlctCalendarS]

    AND Z.PAYGROUP      = BS.PAYGROUP

    AND Z.COMPANY       = BS.COMPANY

    AND BS.EFFDT        =

        (SELECT MAX(AS1.EFFDT)

            FROM PS_PAYGROUP_TBL AS1

            WHERE AS1.COMPANY = Z.COMPANY

                AND AS1.PAYGROUP = Z.PAYGROUP

                AND AS1.EFF_STATUS = 'A'

                AND AS1.EFFDT    <= Z.PAY_END_DT)

    AND Z.PAY_CONFIRM_RUN = 'Y'

    AND Z.PAY_CHECKS_RUN  = 'N'

    AND SS.SRC_BANK_ID    = [$Bank_ID_to_Use_BS]

    AND BTS.BANK_CD       = SS.BANK_CD

    AND BTS.COUNTRY_CD    = 'USA'

ORDER BY Z.COMPANY,

        SS.BANK_CD,

        Z.PAYGROUP

end-SELECT

```

if \$FileOpen = 'Y' or \$FileOpenIAT = 'Y'

if \$FileOpenIAT = 'Y' and

\$IAT-Yes = 'Y'

if \$BatchOpenIAT = 'Y'

do Write-Batch-Control-IAT-Sep

end-if

if #ComplATEntry > 0

if \$RC_PAYINIT.Py_ddp_excl_offs = 'N'

do Write-Offset-Record-IAT-Sep

end-if

end-if

do Write-File-Control-IAT-Sep

do Close-File-IAT

end-if

if \$FileOpen = 'Y'

if \$BatchOpen = 'N' and

\$Details_Processed = 'N' and

#FileBatchCount = 0

move 0 to #TotBatchCreditAmt

move 0 to #BatchEntryHash

move 0 to #BatchEntryCount !Batch control

move 0 to #Temp_TotBatchCreditAmt

move 'Y' to \$BatchOpen

do Write-Batch-Header

do Write-Prenotes

end-if

if \$BatchOpen = 'Y'

if #CompEntry > 0

if \$SRC_PAYINIT.Py_ddp_excl_offs = 'N'

do Write-Offset-Record-Sep

end-if

do Write-Batch-Control

end-if

end-if

do Write-File-Control

do Close-File

end-if

else !No files are open

if \$NewFile = 'Y'

move 'N' to \$NewFile

do Start-New-File

do Start-New-Batch

do Write-Prenotes

```
if #CompEntry > 0
    if $RC_PAYINIT.Py_ddp_excl_offs = 'N'
        do Write-Offset-Record-Sep
    end-if
end-if
```

```
do Write-Batch-Control
do Write-File-Control
do Close-File
end-if
```

```
end-if
```

```
if #File_Count = 0
    display ' '
    display 'No direct deposit output file has been generated.'
    display ' '
end-if
```

```
end-PROCEDURE
```

```
begin-procedure Process-Entry-Detail
```

```
! ** SIR0750 Modification Begin **
```

```
!*****
```

! Citibank will not process offset transactions as part of the regular *
! regular DDP run. Instead, the Port Authority should use their on-line *
! access to Citibank's system to process reversals. The select for *
! employee check data has been altered to not pick up reversed checks. *

!*****

move 'N' to \$Details_Processed

begin-SELECT

E.COMPANY

E.EMPLID

E.NAME

F.ACCOUNT_TYPE

F.PAYCHECK_NBR

F.ACCOUNT_NUM

F.PRIORITY

F.BANK_CD

F.DEPOSIT_AMOUNT

E.EMPL_RCD

E.PAY_END_DT

F.PAYGROUP

F.PAGE_NUM

F.LINE_NUM

move &E.Company to \$CCD_Batch_Company

move &E.EMPLID to \$E.Emplid

move &E.NAME to \$E.Name

move &F.ACCOUNT_TYPE to \$F.Account_type

move &F.ACCOUNT_NUM to \$F.Account_num

move &F.PRIORITY to #F.Priority

move &F.BANK_CD to \$F.Bank_cd

move &F.DEPOSIT_AMOUNT to #F_Deposit_Amount

move &E.PAY_END_DT to \$E.Pay_end_dt

move &F.BANK_CD to \$DD1.Bank_cd

do Get-IAT-indicator

move 'Y' to \$Details_Processed

if \$Verify-New-batch-header = 'Y'

let \$Verify-New-batch-header = 'N'

if \$BatchOpen = 'Y'

if \$StandardEntryClass = 'IAT' and

\$DD1.IAT_ind <> 'Y'

do Write-Batch-Control

move 'Y' to \$NewBatch

end-if

do Write-Prenotes

end-if

end-if

!Commented out - Port Authority

! move &F.Deposit_Amount to #Temp_Amount

! multiply 100 times #Temp_Amount !Remove decimal

! let #Temp_TotBatchCreditAmt = #Temp_Amount + #TotBatchCreditAmt

! if #Temp_TotBatchCreditAmt > 9999999999

!

! if \$FileOpen ='Y'

!

! if #BatchEntryCount > 0

! do Write-Offset-Record-sep

! end-if

!

! do Write-Prenotes

!

! if \$BatchOpen = 'Y'

! do Write-Batch-Control

! end-if

!

! move 'Y' to \$NewBatch

! end-if

!

! end-if

! end-if

! end-if

! if ((\$DD1.IAT_ind <> \$Prev_IAT_Ind) and

! \$IAT-Yes = 'Y') and

! \$BatchOpen = 'N'

! move 'Y' to \$NewBatch

! end-if

if \$NewFile = 'Y'

move 'N' to \$NewFile

do Start-New-File

end-if

if \$NewBatch = 'Y'

move 'N' to \$NewBatch

if \$BatchOpen = 'Y'

do Write-Batch-Control

end-if

do Start-New-Batch

end-if

if (\$DD1.IAT_ind = 'Y' or

\$ComplATIND = 'Y') and

\$IAT-Yes = 'Y'


```
do Write-Entry-Detail-IAT
else
do Write-Entry-Detail
end-if
```

```
do Check-Prenote-Status
move $DD1.IAT_ind to $Prev_IAT_Ind
```

```
if $DD1.IAT_ind = 'Y'
add 1 to #ComplATEntry
else
add 1 to #CompEntry
end-if
```

```
FROM PS_PAY_CHECK    E,
      PS_PAY_DISTRIBUTN F
WHERE E.COMPANY      = $A.Company
      AND F.COMPANY   = $A.Company
      AND E.PAYGROUP  = $A.Paygroup
      AND F.PAYGROUP  = $A.Paygroup
      AND E.PAY_END_DT = $A.Pay_End_Dt
      AND F.PAY_END_DT = $A.Pay_End_Dt
      AND F.PAGE_NUM   = E.PAGE_NUM
      AND F.LINE_NUM   = E.LINE_NUM
      AND F.OFF_CYCLE  = E.OFF_CYCLE
      AND F.SEPCHK     = E.SEPCHK
```

AND F.ACCOUNT_TYPE <> '\$' !not ISSUE CHECK

ORDER BY F.PAYCHECK_NBR

end-SELECT

if \$Prev_IAT_Ind = 'Y'

let \$Batch1_last_rec_IAT = 'Y'

else

let \$Batch1_last_rec_IAT = 'N'

end-if

end-procedure

begin-procedure Process-Entry-Detail-Separate

move 'N' to \$Details_Processed

move ' ' to \$Prev_IAT_Ind

begin-SELECT

ES.COMPANY

ES.EMPLID

ES.NAME

FS.ACCOUNT_TYPE

FS.PAYCHECK_NBR

FS.ACCOUNT_NUM

FS.PRIORITY

FS.BANK_CD

FS.DEPOSIT_AMOUNT

ES.EMPL_RCD

ES.PAY_END_DT

FS.PAYGROUP

FS.PAGE_NUM

FS.LINE_NUM

move &ES.Company to \$CCD_Batch_Company

move &ES.EMPLID to \$E.Emplid

move &ES.NAME to \$E.Name

move &FS.ACCOUNT_TYPE to \$F.Account_type

move &FS.ACCOUNT_NUM to \$F.Account_num

move &FS.PRIORITY to #F.Priority

move &FS.BANK_CD to \$F.Bank_cd

move &FS.DEPOSIT_AMOUNT to #F_Deposit_Amount

move &ES.PAY_END_DT to \$E.Pay_end_dt

move &FS.BANK_CD to \$DD1.Bank_cd

do Get-IAT-indicator

if \$Details_Processed = 'N'

move \$DD1.IAT_ind to \$Prev_IAT_Ind

end-if

if \$DD1.IAT_ind = 'N' !Domestic transactions

move &FS.Deposit_Amount to #Temp_Amount

multiply 100 times #Temp_Amount !Remove decimal

let #Temp_TotlATBatchCreditAmt = #Temp_Amount + #TotlATBatchCreditAmt

if #Temp_TotlATBatchCreditAmt > 9999999999

if \$FileOpen ='Y'

if \$BatchOpen = 'Y'

do Write-Batch-Control

end-if

move 'Y' to \$NewBatch

end-if

end-if

if \$NewFile = 'Y'

move 'N' to \$NewFile

do Start-New-File

end-if

if \$NewBatch = 'Y'

move 'N' to \$NewBatch

if \$BatchOpen ='Y'

do Write-Batch-Control

end-if

do Start-New-Batch

end-if

do Write-Entry-Detail

add 1 to #CompEntry

else !!IAT transactions

if \$NewFileIAT = 'Y' and

 \$IAT-Yes = 'Y'

 move 'N' to \$NewFileIAT

 do Start-New-File-IAT

end-if

if \$NewBatchIAT = 'Y' and

 \$IAT-Yes = 'Y'

 move 'N' to \$NewBatchIAT

 if \$BatchOpenIAT = 'Y'

 do Write-Batch-Control-IAT-Sep

 end-if

 do Start-New-Batch-IAT

end-if

if \$IAT-Yes = 'Y'

do Write-Entry-Detail-IAT-Sep

end-if

add 1 to #ComplATEntry

end-if

do Check-Prenote-Status

move \$DD1.IAT_ind to \$Prev_IAT_Ind

FROM PS_PAY_CHECK ES,

PS_PAY_DISTRIBUTN FS

WHERE ES.COMPANY = \$A.Company

AND FS.COMPANY = \$A.Company

AND ES.PAYGROUP = \$A.Paygroup

AND FS.PAYGROUP = \$A.Paygroup

AND ES.PAY_END_DT = \$A.Pay_End_Dt

AND FS.PAY_END_DT = \$A.Pay_End_Dt

AND FS.PAGE_NUM = ES.PAGE_NUM

AND FS.LINE_NUM = ES.LINE_NUM

AND FS.OFF_CYCLE = ES.OFF_CYCLE

AND FS.SEPCHK = ES.SEPCHK

AND FS.ACCOUNT_TYPE <> '\$' !not ISSUE CHECK

ORDER BY FS.PAYCHECK_NBR

end-SELECT

end-procedure

begin-procedure Get-IAT-indicator

move 'N' to \$DD1.IAT_ind

move 'USA' to \$DD1.Country_cd

begin-SELECT

DD1.IAT_IND

DD1.COUNTRY_CD

move &DD1.IAT_IND to \$DD1.IAT_ind

move &DD1.COUNTRY_CD to \$DD1.Country_cd

FROM PS_DIR_DEP_DISTRIB DD1

WHERE DD1.EMPLID = \$E.Emplid

AND DD1.ACCOUNT_NUM = \$F.Account_num

AND DD1.ACCOUNT_TYPE = \$F.Account_type

AND DD1.BANK_CD = \$F.Bank_cd

AND DD1.PRIORITY = #F.Priority

AND DD1.EFFDT = (SELECT MAX(DD1D.EFFDT)

FROM PS_DIR_DEP_DISTRIB DD1D

WHERE DD1D.EMPLID = DD1.EMPLID

AND DD1D.ACCOUNT_NUM = DD1.ACCOUNT_NUM

AND DD1D.ACCOUNT_TYPE = DD1.ACCOUNT_TYPE

AND DD1D.BANK_CD = DD1.BANK_CD

AND DD1D.PRIORITY = DD1.PRIORITY)

end-SELECT

end-procedure

begin-procedure Start-New-File

move 0 to #TotBatchCreditAmt

move 0 to #TotBatchDebitAmt

move 0 to #BatchEntryHash

move 0 to #TotMultBatchCreditAmt

move 0 to #TotMultBatchDebitAmt

move 0 to #TotFileDebitAmt

move 0 to #TotFileCreditAmt

move 0 to #FileEntryHash

move 0 to #BlockCount

move 0 to #BatchEntryCount !Batch control

move 0 to #Temp_TotBatchCreditAmt

move 0 to #FileEntryCount !file control

move 0 to #FileBatchCount !file control

move 0 to #FileBlockCount !file control

move 0 to #Batch#

move 0 to #CompEntry

do Get-File-header-Company

do Open-File

do Write-File-Header

end-procedure

begin-procedure Start-New-File-IAT

move 0 to #TOTIATBATCHCreditAmt

move 0 to #TotIATBatchDebitAmt

move 0 to #BatchIATEntryHash

move 0 to #TotIATFileDebitAmt

move 0 to #TotIATFileCreditAmt

move 0 to #FileIATEntryHash

move 0 to #TotIATMultBatchCreditAmt

move 0 to #TotIATMultBatchDebitAmt

move 0 to #BlockCountIAT

move 0 to #BatchIATEntryCount !Batch control

move 0 to #Temp_TotIATBatchCreditAmt

move 0 to #FileIATEntryCount !file control

move 0 to #FileIATBatchCount !file control

move 0 to #FileIATBlockCount !file control

move 0 to #IATBatch#

move 0 to #ComplATEntry

do Get-File-header-Company

do Open-File-IAT

do Write-File-Header-IAT

end-procedure

begin-procedure Start-New-Batch

move 0 to #TotBatchCreditAmt

move 0 to #TotBatchDebitAmt

move 0 to #BatchEntryHash

move 0 to #BatchEntryCount !Batch control

move 0 to #Temp_TotBatchCreditAmt

move 'Y' to \$BatchOpen

move 'N' to \$NewBatch

move 0 to #CompEntry

if (\$ComplATIND = 'Y' or \$DD1.IAT_ind = 'Y') and

\$IAT-Yes = 'Y' and

\$FromPrenotes = 'N'

do Write-Batch-Header-IAT

else

do Write-Batch-Header

end-if

move 'N' to \$FromPrenotes

end-procedure

begin-procedure Start-New-Batch-IAT

move 0 to #TotIATBatchCreditAmt

move 0 to #TotIATBatchDebitAmt

move 0 to #BatchIATEntryHash

move 0 to #BatchIATEntryCount !Batch control

move 0 to #Temp_TotIATBatchCreditAmt

move 'Y' to \$BatchOpenIAT

move 0 to #ComplATEntry

do Write-Batch-Header-IAT-Sep

end-procedure

begin-PROCEDURE Get-File-header-Company

if \$PrCs_Process_Instance = "

input \$File_Co 'Enter 3 Digit Company Code for File Header' type=char

uppercase \$File_Co

else

do Select-Parameters-File-Co

end-if

move \$File_Co to \$Company

```

do Get-Company-Data

move $CompanyName to $File_CompanyName

uppercase $File_CompanyName

move &CT.Federal_EIN   to $Federal_EIN 099999999

!*PATH Bank change

move &CTPA.Federal_EIN   to $PAFederal_EIN 099999999

let $PAFile_CompanyID = '1' || $PAFederal_EIN

let $File_CompanyID = '1' || $Federal_EIN

!*** SR1453 - Modification Begin

if $fileType = 'MS'

    let $PAFile_CompanyID = '9' || $PAFederal_EIN

    let $File_CompanyID = '9' || $Federal_EIN

end-if

!*** SR1453 - Modification End


end-procedure


begin-procedure Select-Parameters-File-Co

begin-select

P1.COMPANY


FROM PS_RC_DDP001 P1

WHERE P1.OPRID      = $PrCs_OprID

```

AND P1.RUN_CNTL_ID = \$PrCs_Run_Cntl_ID

AND P1.BANK_CD = \$PriorTransit#

end-select

let \$File_Co = &P1.Company

end-procedure

begin-PROCEDURE Open-File

add 1 to #filenum

do Format-Number(#filenum, \$filenum, '009')

#ifdef OS400

let \$FileName = 'DDPUS' || '(D' || \$FileNum || ')

#else

#ifdef MVS

let \$FileName = 'DDPUS' || \$FileNum

#Else

!*****PATH Modification Begin

let \$FileName = \$FILE_PATH || '.' || \$FileNum

!*****PATH Modification End

#endif

#endif

!** SIR0370 Modification Begin **

!**** Modification Begin SIR0339 *****

!!let \$FileName = '{FILEPREFIX}' || \$FileName || '{FILESUFFIX}'

!!let \$FileName = '{INTERHOME}/citibank/{OUTBOUND}/\${Filename}'

!**** Modification End SIR0339 *****

!*****PATH Modification Begin

! let \$FileName = '{INTERHOME}/citibank/{OUTBOUND}/' || \$FileName

!*****PATH Modification End

!** SIR0370 Modification End **

display 'file'

display \$filename

open \$filename as 1 for-writing record=94:fixed

move 'Y' to \$FileOpen

!*** SR1453 - Modification Begin

open '/tmp/ddp001msg.txt' as 2 for-writing record = 100:vary

!*** SR1453 - Modification End

end-PROCEDURE

begin-PROCEDURE Open-File-IAT

add 1 to #IATfilenum

do Format-Number(#IATfilenum, \$IATfilenum, '009')

#ifdef OS400

```

    let $IATFileName = 'DDPUSIAT' || '(D' || $IATFileNum || ')'
#else

#ifdef MVS

    let $IATFileName = 'DDPUSIAT' || $IATFileNum
#else

    let $IATFileName = 'DDPIAT' || '.' || $IATFileNum
#endif

#endif

let $IATFileName = '{IMPORTPREFIX}' || $IATFileName || '{IMPORTSUFFIX}'

open $IATfilename as 2 for-writing record=94:fixed

move 'Y' to $FileOpenIAT

end-PROCEDURE

begin-procedure Write-File-Header

    move '1'          to $RecordType
    move '01'         to $PriorityCode
    move ' '          to $TransitRouting#
    concat $Transit#   with $TransitRouting#

    do Format-DateTime($AsOfToday, $out, {DEFCMP}, ", ")

    let $TransmitDate = substr($out, 3, 2) || substr($out, 5, 2) || substr($out, 7, 2)

```

move '0' to \$Time24

move '1' to \$TLZero

move " to \$TDelimiter

do Format-DateTime(\$SysDateTime, \$out, {DEFDATE}, 'time', ")

let \$Time1 = substr(\$out,12, 1)

if \$Time1 = ' '

move '0' to \$Time1

end-if

let \$Time2 = substr(\$out,13, 1)

let \$Time3 = substr(\$out,14, 2)

let \$TransmitTime = \$Time1 || \$Time2 || \$Time3

move 'A' to \$FileModifier

move '094' to \$RecordSize

move '10' to \$BlockingFactor

move '1' to \$FormatCode

move &C.BANK_NM to \$OrigBank

move ' ' to \$Reference ! 8 spaces

uppercase \$OrigBank

add 1 to #BlockCount

if #BlockCount = 10


```
move 0 to #BlockCount  
add 1 to #FileBlockCount  
end-if
```

```
write 1 from $RecordType:1      -  
    $PriorityCode:2              -  
    $TransitRouting#:10         -    ! Immediate Destination
```

! **PATH Bank Change

```
    $PAFile_CompanyID:10        -  
    $TransmitDate:6              -  
    $TransmitTime:4              -  
    $FileModifier:1              -  
    $RecordSize:3                -  
    $BlockingFactor:2            -  
    $FormatCode:1                -  
    $OrigBank:23                 -  
    $File_CompanyName:23         -  
    $Reference:8
```

! *** SR1453 - Modification Begin

```
begin-select  
to_char(sysdate, 'MM/DD/YY') &DISPDATE
```

```
let $dispDate = &DISPDATE  
if &A.COMPANY = 'TH'
```

```
write 2 from 'PATH CITIBANK DIRECT DEPOSIT FILE CONTROL TOTALS AS OF ':55
```

\$dispDate:8

Else

write 2 from 'CITIBANK DIRECT DEPOSIT FILE CONTROL TOTALS AS OF ':50

\$dispDate:8

End-If

from psdbowner

end-select

!*** SR1453 - Modification End

end-procedure

begin-procedure Write-File-Header-IAT

move '1' to \$RecordType

move '01' to \$PriorityCode

move ' ' to \$TransitRouting#

concat \$Transit# with \$TransitRouting#

do Format-DateTime(\$AsOfToday, \$out, {DEFCMP}, ", ")

let \$TransmitDate = substr(\$out, 3, 2) || substr(\$out, 5, 2) || substr(\$out, 7, 2)

move '0' to \$Time24

move '1' to \$TLZero

move " to \$TDelimiter

do Format-DateTime(\$SysDateTime, \$out, {DEFDATE}, 'time', ")

let \$Time1 = substr(\$out,12, 1)

if \$Time1 = ' '

 move '0' to \$Time1

end-if

let \$Time2 = substr(\$out,13, 1)

let \$Time3 = substr(\$out,14, 2)

let \$TransmitTime = \$Time1 || \$Time2 || \$Time3

move 'A' to \$FileModifier

move '094' to \$RecordSize

move '10' to \$BlockingFactor

move '1' to \$FormatCode

move &C.BANK_NM to \$OrigBank

move ' ' to \$Reference ! 8 spaces

uppercase \$OrigBank

add 1 to #BlockCountIAT

if #BlockCountIAT = 10

 move 0 to #BlockCountIAT

 add 1 to #FileIATBlockCount

end-if

write 2 from \$RecordType:1 -

 \$PriorityCode:2 -

\$TransitRouting#:10 - ! Immediate Destination

\$File_CompanyID:10 -

\$TransmitDate:6 -

\$TransmitTime:4 -

\$FileModifier:1 -

\$RecordSize:3 -

\$BlockingFactor:2 -

\$FormatCode:1 -

\$OrigBank:23 -

\$File_CompanyName:23 -

\$Reference:8

end-procedure

begin-procedure Write-Batch-Header

move '5' to \$RecordType

move '200' to \$ServiceClassCode

move 'PPD' to \$StandardEntryClass

move 'DIR DEP' to \$CoEntryDescr

do Format-DateTime(\$priorPay_End_Dt, \$out, {DEFCMP}, ", ")

let \$CoDescrDt = substr(\$out, 3, 2) || substr(\$out, 5, 2) || substr(\$out, 7, 2)

do Format-DateTime(\$priorCheck_Dt, \$out, {DEFCMP}, ", ")

let \$EffEntryDt = substr(\$out, 3, 2) || substr(\$out, 5, 2) || substr(\$out, 7, 2)

move '000' to \$SettlementData

move '1' to \$StatusCode

move \$Transit# to \$TransitRouting#

add 1 to #Batch#

do Format-Number(#Batch#, \$Batch#, '0000009')

add 1 to #FileBatchCount

add 1 to #BlockCount

if #BlockCount = 10

move 0 to #BlockCount

add 1 to #FileBlockCount

end-if

! ** SIR0750 Modification Begin **

if &A.COMPANY = 'TH'

let \$Citibank_CompanyName = 'PORT AUTH-PATH '

ELSE

let \$Citibank_CompanyName = 'PORT AUTH-NYNJ '

END-IF

let \$Citibank_CompanyID = substr(\$Batch_CompanyID,2,9)

! let \$Citibank_CompanyID = '9' || \$Citibank_CompanyID

let \$Citibank_CompanyID = '1' || \$Citibank_CompanyID

!*** SR1453 - Modification Begin

if \$fileType = 'MS'

let \$Citibank_CompanyID = '9' || \$Federal_EIN

end-if

!*** SR1453 - Modification End

let \$CitibankEntryDescr = 'PAYROLL'

write 1 from \$RecordType:1 -

\$ServiceClassCode:3 -

! \$Batch_CompanyName:16 -

\$Citibank_CompanyName:16 -

\$CustDiscrData:20 -

! \$Batch_CompanyID:10 -

\$Citibank_CompanyID:10 -

\$StandardEntryClass:3 -

! \$CoEntryDescr:10 -

\$CitibankEntryDescr:10 -

\$CoDescrDt:6 -

\$EffEntryDt:6 -

\$SettlementData:3 -

\$StatusCode:1 -

\$TransitRouting#:8 - ! originating DFI ID

\$Batch#:7

!** SIR0750 Modification End **

!*** SR1453 - Modification Begin

let #count = #count + 1

if #count = 1

write 2 from ' ':1

write 2 from ' ':1

write 2 from 'START-END DATES':30

\$CustDiscrData:20

end-if

!*** SR1453 - Modification End

end-procedure

begin-procedure Write-Batch-Header-CCD

move '5' to \$RecordType

move '200' to \$ServiceClassCode

move 'CCD' to \$StandardEntryClass

move 'DIR DEP' to \$CoEntryDescr

move \$CCD_Batch_Company to \$Company

do Get-Company-Data

move \$CompanyName to \$CCD_Batch_CompanyName

UPPERCASE \$CCD_Batch_CompanyName

move &CT.Federal_EIN to \$Federal_EIN 099999999

let \$CCD_Batch_Company = '1' || \$Federal_EIN

do Format-DateTime(\$priorPay_End_Dt, \$out, {DEFCMP}, ", ")

let \$CoDescrDt = substr(\$out, 3, 2) || substr(\$out, 5, 2) || substr(\$out, 7, 2)

do Format-DateTime(\$priorCheck_Dt, \$out, {DEFCMP}, ", ")

let \$EffEntryDt = substr(\$out, 3, 2) || substr(\$out, 5, 2) || substr(\$out, 7, 2)

move ' ' to \$SettlementData

move '1' to \$StatusCode

move \$Transit# to \$TransitRouting#

add 1 to #Batch#

do Format-Number(#Batch#, \$Batch#, '0000009')

add 1 to #FileBatchCount

add 1 to #BlockCount

if #BlockCount = 10

move 0 to #BlockCount

add 1 to #FileBlockCount

end-if

write 1 from \$RecordType:1 -

\$ServiceClassCode:3 -

\$CCD_Batch_CompanyName:16 -

\$CustDiscrData:20 -

\$CCD_Batch_Company:10 -

\$StandardEntryClass:3 -

\$CoEntryDescr:10 -
\$CoDescrDt:6 -
\$EffEntryDt:6 -
\$SettlementData:3 -
\$StatusCode:1 -
\$TransitRouting#:8 - ! originating DFI ID
\$Batch#:7

end-procedure

begin-procedure Write-Batch-Header-CCD-Sep

move '5' to \$RecordType
move '200' to \$ServiceClassCode
move 'CCD' to \$StandardEntryClass
move 'DIR DEP' to \$CoEntryDescr
move \$CCD_Batch_Company to \$Company
do Get-Company-Data
move \$CompanyName to \$CCD_Batch_CompanyName
UPPERCASE \$CCD_Batch_CompanyName
move &CT.Federal_EIN to \$Federal_EIN 099999999
let \$CCD_Batch_Company = '1' || \$Federal_EIN

do Format-DateTime(\$priorPay_End_Dt, \$out, {DEFCMP}, ", ")
let \$CoDescrDt = substr(\$out, 3, 2) || substr(\$out, 5, 2) || substr(\$out, 7, 2)

do Format-DateTime(\$priorCheck_Dt, \$out, {DEFCMP}, ", ")

let \$EffEntryDt = substr(\$out, 3, 2) || substr(\$out, 5, 2) || substr(\$out, 7, 2)

move ' ' to \$SettlementData

move '1' to \$StatusCode

move \$Transit# to \$TransitRouting#

add 1 to #IATBatch#

do Format-Number(#IATBatch#, \$IATBatch#, '0000009')

add 1 to #FileIATBatchCount

add 1 to #BlockCountIAT

if #BlockCountIAT = 10

move 0 to #BlockCountIAT

add 1 to #FileIATBlockCount

end-if

write 2 from \$RecordType:1 -

\$ServiceClassCode:3 -

\$CCD_Batch_CompanyName:16 -

\$CustDiscrData:20 -

\$CCD_Batch_Company:10 -

\$StandardEntryClass:3 -

\$CoEntryDescr:10 -

\$CoDescrDt:6 -

\$EffEntryDt:6 -

\$SettlementData:3 -

\$StatusCode:1 -
\$TransitRouting#:8 - ! originating DFI ID
\$IATBatch#:7

end-procedure

begin-procedure Write-Batch-Header-IAT

move '5' to \$RecordType
move '200' to \$ServiceClassCode
move 'IAT' to \$StandardEntryClass
move 'DIR DEP' to \$CoEntryDescr
move ' ' to \$Space16
move 'FF' to \$FXInd
move '3' to \$FXRefInd
move ' ' to \$FXRef
move 'US' to \$OrigCntry
move 'USD' to \$ISOUSD

do Format-DateTime(\$priorPay_End_Dt, \$out, {DEFCMP}, ", ")

let \$CoDescrDt = substr(\$out, 3, 2) || substr(\$out, 5, 2) || substr(\$out, 7, 2)

do Format-DateTime(\$priorCheck_Dt, \$out, {DEFCMP}, ", ")

let \$EffEntryDt = substr(\$out, 3, 2) || substr(\$out, 5, 2) || substr(\$out, 7, 2)

```
move ' ' to $SettlementData
move '1' to $StatusCode
move $Transit# to $TransitRouting#
```

```
add 1 to #Batch#
```

```
do Format-Number(#Batch#, $Batch#, '0000009')
```

```
add 1 to #FileBatchCount
```

```
add 1 to #BlockCount
```

```
if #BlockCount = 10
```

```
    move 0 to #BlockCount
```

```
    add 1 to #FileBlockCount
```

```
end-if
```

```
write 1 from $RecordType:1 -
```

```
    $ServiceClassCode:3 -
```

```
    $Space16:16 -
```

```
    $FXInd:2 -
```

```
    $FXRefInd:1 -
```

```
    $FXRef:15 -
```

```
    $OrigCntry:2 -
```

```
    $Batch_CompanyID:10 -
```

```
    $StandardEntryClass:3 -
```

```
    $CoEntryDescr:10 -
```

```
    $ISOUSD:3 -
```

\$ISOUSD:3 -
\$EffEntryDt:6 -
\$SettlementData:3 -
\$StatusCode:1 -
\$TransitRouting#:8 - ! originating DFI ID
\$Batch#:7

end-procedure

begin-procedure Write-Batch-Header-IAT-Sep

move '5' to \$RecordType
move '200' to \$ServiceClassCode
move 'IAT' to \$StandardEntryClass
move 'DIR DEP' to \$CoEntryDescr
move ' ' to \$Space16
move 'FF' to \$FXInd
move '3' to \$FXRefInd
move ' ' to \$FXRef
move 'US' to \$OrigCntry
move 'USD' to \$ISOUSD

do Format-DateTime(\$priorPay_End_Dt, \$out, {DEFCMP}, ", ")

let \$CoDescrDt = substr(\$out, 3, 2) || substr(\$out, 5, 2) || substr(\$out, 7, 2)

do Format-DateTime(\$priorCheck_Dt, \$out, {DEFCMP}, ", ")

let \$EffEntryDt = substr(\$out, 3, 2) || substr(\$out, 5, 2) || substr(\$out, 7, 2)

move ' ' to \$SettlementData

move '1' to \$StatusCode

move \$Transit# to \$TransitRouting#

add 1 to #IATBatch#

do Format-Number(#IATBatch#, \$IATBatch#, '0000009')

add 1 to #FileIATBatchCount

add 1 to #BlockCountIAT

if #BlockCountIAT = 10

move 0 to #BlockCountIAT

add 1 to #FileIATBlockCount

end-if

write 2 from \$RecordType:1 -

\$ServiceClassCode:3 -

\$Space16:16 -

\$FXInd:2 -

\$FXRefInd:1 -

\$FXRef:15 -

\$OrigCntry:2 -

\$Batch_CompanyID:10 -

\$StandardEntryClass:3 -
\$CoEntryDescr:10 -
\$ISOUSD:3 -
\$ISOUSD:3 -
\$EffEntryDt:6 -
\$SettlementData:3 -
\$StatusCode:1 -
\$TransitRouting#:8 - ! originating DFI ID
\$IATBatch#:7

end-procedure

begin-procedure Write-Entry-Detail

move '6' to \$RecordType
move '0' to \$CheckDigit
move ' ' to \$DiscrData
move '0' to \$AddendaIndicator
move ' ' to \$Trace#
move \$E.Emplid to \$EmplID ! or SSN ?
move \$E.Name to \$Name
uppercase \$Name

if \$F.Account_type = 'C'

```
    move '22' to $TransactionCode      !DDA Automatic Credit
else  !Savings
    move '32' to $TransactionCode      !TDA Automatic Credit
end-if
```

```
move $F.Bank_cd to $TransitRouting#
do Calc-DFI-CheckDigit
move $F.Bank_cd to $hashTransit# xxxxxxxx !Only first eight numbers
move $hashTransit# to #hashTransit#
add #hashTransit# to #BatchEntryHash
```

```
move #F_Deposit_Amount to #Amount
multiply 100 times #Amount           !Remove decimal
```

```
do Format-Number(#Amount, $Amount, '0000000009')
add #Amount to #TotBatchCreditAmt
add #Amount to #TotBatchDebitAmt      ! for offset record
```

```
add 1 to #BatchEntryCount
add 1 to #BlockCount
```

```
if #BlockCount = 10
    move 0 to #BlockCount
    add 1 to #FileBlockCount
end-if
```


add 1 to #Trace#

move #Trace# to \$Trace# 0999999999999999

let \$Transit_Temp = substr(\$Transit#, 0, 8)

do Calc-Alt-Trace

write 1 from \$RecordType:1 - ! 01

\$TransactionCode:2 - ! 02

\$F.Bank_cd:8 - ! 04

\$CheckDigit:1 - ! 12

\$F.Account_num:17 - ! 13

\$Amount:10 - ! 30

\$EmplID:15 - ! 40

\$Name:22 - ! 55

\$DiscrData:2 - ! 77

\$AddendaIndicator:1 - ! 79

! \$Trace#:15 ! 80

\$Alt_Trace:15 ! 80

end-procedure

begin-procedure Get-Bank-name

begin-SELECT

B.BANK_TYPE

B.BANK_NM

FROM PS_BANK_EC_TBL B

WHERE B.BANK_CD = \$DD1.Bank_cd

AND B.COUNTRY_CD = \$DD1.Country_cd

end-SELECT

end-procedure

begin-procedure Write-Entry-Detail-IAT

do Get-Bank-name

move '6' to \$RecordType

move '0' to \$CheckDigit

move ' ' to \$Space1

move ' ' to \$Space2

move ' ' to \$Space13

move ' ' to \$DiscrData

move '1' to \$AddendaIndicator

move ' ' to \$Trace#

move \$E.Emplid to \$EmplID ! or SSN ?

move \$E.Name to \$Name

uppercase \$Name

```
if $F.Account_type = 'C'

    move '22' to $TransactionCode      !DDA Automatic Credit

else !Savings

    move '32' to $TransactionCode      !TDA Automatic Credit

end-if


move $F.Bank_cd to $TransitRouting#

do Calc-DFI-CheckDigit

move $F.Bank_cd to $hashTransit# xxxxxxxx !Only first eight numbers

move $hashTransit# to #hashTransit#

add #hashTransit# to #BatchEntryHash


move #F_Deposit_Amount to #Amount

multiply 100 times #Amount            !Remove decimal

do Format-Number(#Amount, $Amount, '0000000009')

add #Amount to #TotBatchCreditAmt

add #Amount to #TotBatchDebitAmt      ! for offset record


add 1 to #BatchEntryCount

add 1 to #BlockCount


if #BlockCount = 10

    move 0 to #BlockCount

    add 1 to #FileBlockCount

end-if
```

do Format-Number(7,\$Addenda_Count,'0999')

do Format-Number(1,\$Addenda_Ind,'9')

add 1 to #Trace#

move #Trace# to \$Trace# 0999999999999999

let \$Transit_Temp = substr(\$Transit#, 0, 8)

do Calc-Alt-Trace

write 1 from \$RecordType:1	-	! 01
\$TransactionCode:2	-	! 02
\$F.Bank_cd:8	-	! 04
\$CheckDigit:1	-	! 12
\$Addenda_Count:4	-	! 13
\$Space13:13	-	! 17
\$Amount:10	-	! 30
\$F.Account_num:35	-	! 40
\$Space2:2	-	! 75
\$Space1:1	-	! 77
\$Space1:1	-	! 78
\$Addenda_Ind:1	-	! 79
\$Alt_Trace:15		! 80

```

move '7' to $RecordType

move '10' to $AddendaType

move 'SAL' to $Trans_type

do Format-Number(#Amount, $Amount, '000000000000000009')

move '          ' to $Space22

move '      ' to $Space6

move &B.BANK_NM to $BankName

uppercase $BankName

move $Name to $HomeName

do Rotate-Name

move $Name to $HomeName

uppercase $Name

let $EntrySeq = substr($Alt_Trace, 9, 7)

add 1 to #BatchEntryCount

add 1 to #BlockCount

```

```

if #BlockCount = 10

    move 0 to #BlockCount

    add 1 to #FileBlockCount

end-if

```

```

write 1 from $RecordType:1      -          ! 01

      $AddendaType:2          -          ! 02

      $Trans_type:3          -          ! 04

      $Amount:18              -          ! 07

      $Space22:22              -          ! 25

```

\$Name:35	-	! 47
\$Space6:6	-	! 82
\$EntrySeq:7		! 88

move \$HomeName to \$Name

move '11' to \$AddendaType

move ' ' to \$Space14

move \$A.Company to \$Company

do Get-Company-Data

Do Get-Two-Char-Country(&CT.COUNTRY, \$Country_2Char)

move \$CompanyName to \$IAT_CompanyName

uppercase \$IAT_CompanyName

uppercase \$Compnyadd1

add 1 to #BatchEntryCount

add 1 to #BlockCount

if #BlockCount = 10

move 0 to #BlockCount

add 1 to #FileBlockCount

end-if

write 1 from \$RecordType:1	-	! 01
-----------------------------	---	------

\$AddendaType:2	-	! 02
-----------------	---	------

\$IAT_CompanyName:35	-	! 04
----------------------	---	------

\$Compnyadd1:35	-	! 39
-----------------	---	------

\$Space14:14 - ! 74

\$EntrySeq:7 ! 88

move '12' to \$AddendaType

if rtrim(&CT.City,' ') = " and

rtrim(&CT.State,' ') = "

move '\' to \$CityState

else

if rtrim(&CT.City,' ') = "

let \$CityState = '*' || rtrim(&CT.State,' ') || '\'

else

if rtrim(&CT.State,' ') = "

let \$CityState = \$Compnycity || '\'

else

let \$CityState = \$Compnycity || '*' || rtrim(&CT.State,' ') || '\'

end-if

end-if

end-if

uppercase \$CityState

if rtrim(\$Country_2Char,' ') = " and

rtrim(&CT.Postal,' ') = "

move '\' to \$CntryPstl

else

if rtrim(\$Country_2Char,' ') = "

```

    let $CntryPstl = '*' || rtrim(&CT.Postal,' ') || '\'
else
    if rtrim(&CT.Postal,' ') = ''
        let $CntryPstl = rtrim($Country_2Char,' ') || '\'
    else
        let $CntryPstl = rtrim($Country_2Char,' ') || '*' || rtrim(&CT.Postal,' ') || '\'
    end-if
end-if

end-if

uppercase $CntryPstl

add 1 to #BatchEntryCount

add 1 to #BlockCount

if #BlockCount = 10
    move 0 to #BlockCount
    add 1 to #FileBlockCount
end-if

write 1 from $RecordType:1      -      ! 01
    $AddendaType:2      -      ! 02
    $CityState:35      -      ! 04
    $CntryPstl:35      -      ! 39
    $Space14:14      -      ! 74
    $EntrySeq:7      ! 88

```


move '13' to \$AddendaType

move '01' to \$DFIQualif

move 'US' to \$DFICountry

move ' ' to \$Space10

add 1 to #BatchEntryCount

add 1 to #BlockCount

uppercase \$OrigBank

if #BlockCount = 10

move 0 to #BlockCount

add 1 to #FileBlockCount

end-if

write 1 from \$RecordType:1 - ! 01

\$AddendaType:2 - ! 02

\$OrigBank:35 - ! 04

\$DFIQualif:2 -

\$Transit#:34 - ! 39

\$DFICountry:3 - ! 74

\$Space10:10 -

\$EntrySeq:7 ! 88

move '14' to \$AddendaType

let \$RecBankNm = rtrim(&B.BANK_NM,' ')

uppercase \$RecBankNm

move '01' to \$DFIQualif

move 'US' to \$DFICountry

move ' ' to \$Space10

add 1 to #BatchEntryCount

add 1 to #BlockCount

if #BlockCount = 10

move 0 to #BlockCount

add 1 to #FileBlockCount

end-if

write 1 from \$RecordType:1 - ! 01

\$AddendaType:2 - ! 02

\$RecBankNm:35 - ! 04

\$DFIQualif:2 -

\$F.Bank_cd:34 - ! 04

\$DFICountry:3 - ! 74

\$Space10:10 -

\$EntrySeq:7 ! 88

move '15' to \$AddendaType

move ' ' to \$Space34

let \$EmplAddr1 = \"

let \$EmplCityState = \"

let \$EmplCntryPstl = \"

Do Get-Empl-Address (\$E.Emplid, 'HOME', \$AddressFound)

if \$AddressFound = 'Y'

let \$EmplAddr1 = \$GETADDR_Address1

let \$CITY = \$GETADDR_CITY

let \$STATE = \$GETADDR_STATE

let \$POSTAL = \$GETADDR_POSTAL

let \$COUNTRY = \$GETADDR_COUNTRY

Do Get-Two-Char-Country(\$COUNTRY, \$Country_2Char)

if \$CITY = " and

\$STATE = "

move \" to \$EmplCityState

else

if \$CITY = "

let \$EmplCityState = '*' || \$STATE || \"

else

if \$STATE = "

let \$EmplCityState = \$CITY || \"

else

let \$EmplCityState = \$CITY || '*' || \$STATE || \"

end-if

end-if

end-if

if rtrim(\$Country_2Char,' ') = " and

rtrim(\$POSTAL,' ') = "

move '\' to \$EmplCntryPstl

else

if rtrim(\$Country_2Char,' ') = "

let \$EmplCntryPstl = '*' || rtrim(\$POSTAL,' ') || '\'

else

if rtrim(\$POSTAL,' ') = "

let \$EmplCntryPstl = rtrim(\$Country_2Char,' ') || '\'

else

let \$EmplCntryPstl = rtrim(\$Country_2Char,' ') || '*' || rtrim(\$POSTAL,' ') || '\'

end-if

end-if

end-if

end-if

uppercase \$EmplAddr1

uppercase \$EmplCityState

uppercase \$EmplCntryPstl

add 1 to #BatchEntryCount

add 1 to #BlockCount

if #BlockCount = 10

move 0 to #BlockCount
add 1 to #FileBlockCount
end-if

write 1 from \$RecordType:1	-	! 01
\$AddendaType:2	-	! 02
\$EmplID:15	-	! 40
\$EmplAddr1:35	-	
\$Space34:34	-	
\$EntrySeq:7		! 88

move '16' to \$AddendaType
add 1 to #BatchEntryCount
add 1 to #BlockCount

if #BlockCount = 10
move 0 to #BlockCount
add 1 to #FileBlockCount
end-if

write 1 from \$RecordType:1	-	! 01
\$AddendaType:2	-	! 02
\$EmplCityState:35	-	
\$EmplCntryPstl:35	-	

\$Space14:14 -

\$EntrySeq:7 ! 88

end-procedure

begin-procedure Write-Entry-Detail-IAT-Sep

do Get-Bank-name

move '6' to \$RecordType

move '0' to \$CheckDigit

move ' ' to \$Space1

move ' ' to \$Space2

move ' ' to \$Space13

move ' ' to \$DiscrData

move '1' to \$AddendaIndicator

move ' ' to \$Trace#

move \$E.Emplid to \$EmplID ! or SSN ?

move \$E.Name to \$Name

uppercase \$Name

if \$F.Account_type = 'C'

 move '22' to \$TransactionCode !DDA Automatic Credit

else !Savings

 move '32' to \$TransactionCode !TDA Automatic Credit

end-if

move \$F.Bank_cd to \$TransitRouting#

do Calc-DFI-CheckDigit

move \$F.Bank_cd to \$hashTransit# xxxxxxxx !Only first eight numbers

move \$hashTransit# to #hashTransit#

add #hashTransit# to #BatchIATEntryHash

move #F_Deposit_Amount to #Amount

multiply 100 times #Amount !Remove decimal

do Format-Number(#Amount, \$Amount, '0000000009')

add #Amount to #TotIATBatchCreditAmt

add #Amount to #TotIATBatchDebitAmt ! for offset record

add 1 to #BatchIATEntryCount

add 1 to #BlockCountIAT

if #BlockCountIAT = 10

 move 0 to #BlockCountIAT

 add 1 to #FileIATBlockCount

end-if

do Format-Number(7,\$Addenda_Count,'0999')

do Format-Number(1,\$Addenda_Ind,'9')

add 1 to #Trace#

move #Trace# to \$Trace# 0999999999999999

let \$Transit_Temp = substr(\$Transit#, 0, 8)

do Calc-Alt-Trace

write 2 from \$RecordType:1 - ! 01

\$TransactionCode:2 - ! 02

\$F.Bank_cd:8 - ! 04

\$CheckDigit:1 - ! 12

\$Addenda_Count:4 - ! 13

\$Space13:13 - ! 17

\$Amount:10 - ! 30

\$F.Account_num:35 - ! 40

\$Space2:2 - ! 75

\$Space1:1 - ! 77

\$Space1:1 - ! 78

\$Addenda_Ind:1 - ! 79

\$Alt_Trace:15 ! 80

move '7' to \$RecordType

move '10' to \$AddendaType


```

move 'SAL' to $Trans_type

do Format-Number(#Amount, $Amount, '000000000000000009')

move '          ' to $Space22

move '      ' to $Space6

move &B.BANK_NM to $BankName

uppercase $BankName

move $Name to $HomeName

do Rotate-Name

move $Name to $HomeName

uppercase $Name

let $EntrySeq = substr($Alt_Trace, 9, 7)

add 1 to #BatchIATEntryCount

add 1 to #BlockCountIAT


if #BlockCountIAT = 10

    move 0 to #BlockCountIAT

    add 1 to #FileIATBlockCount

end-if

write 2 from $RecordType:1      -          ! 01

    $AddendaType:2      -          ! 02

    $Trans_type:3      -          ! 04

    $Amount:18      -          ! 07

    $Space22:22      -          ! 25

    $Name:35      -          ! 47

    $Space6:6      -          ! 82

    $EntrySeq:7          ! 88

```

```

move $HomeName to $Name
move '11' to $AddendaType
move '      ' to $Space14
move $A.Company to $Company
do Get-Company-Data
Do Get-Two-Char-Country(&CT.COUNTRY, $Country_2Char)
move $CompanyName to $IAT_CompanyName
uppercase $IAT_CompanyName
uppercase $Compnyadd1
add 1 to #BatchIATEntryCount
add 1 to #BlockCountIAT

if #BlockCountIAT = 10
    move 0 to #BlockCountIAT
    add 1 to #FileIATBlockCount
end-if

```

```

write 2 from $RecordType:1      -      ! 01
    $AddendaType:2      -      ! 02
    $IAT_CompanyName:35      -      ! 04
    $Compnyadd1:35      -      ! 39
    $Space14:14      -      ! 74
    $EntrySeq:7      ! 88

```

move '12' to \$AddendaType

if rtrim(&CT.City,' ') = " and

rtrim(&CT.State,' ') = "

move '\' to \$CityState

else

if rtrim(&CT.City,' ') = "

let \$CityState = '*' || rtrim(&CT.State,' ') || '\'

else

if rtrim(&CT.State,' ') = "

let \$CityState = \$Compnycity || '\'

else

let \$CityState = \$Compnycity || '*' || rtrim(&CT.State,' ') || '\'

end-if

end-if

end-if

uppercase \$CityState

if rtrim(\$Country_2Char,' ') = " and

rtrim(&CT.Postal,' ') = "

move '\' to \$CntryPstl

else

if rtrim(\$Country_2Char,' ') = "

let \$CntryPstl = '*' || rtrim(&CT.Postal,' ') || '\'

else

if rtrim(&CT.Postal,' ') = "

let \$CntryPstl = rtrim(\$Country_2Char,' ') || '\'

```

else

    let $CntryPstl = rtrim($Country_2Char,' ') || '*' || rtrim(&CT.Postal,' ') || '\'

end-if

end-if

end-if

uppercase $CntryPstl

add 1 to #BatchIATEntryCount

add 1 to #BlockCountIAT

if #BlockCountIAT = 10

    move 0 to #BlockCountIAT

    add 1 to #FileIATBlockCount

end-if

write 2 from $RecordType:1      -      ! 01

    $AddendaType:2      -      ! 02

    $CityState:35      -      ! 04

    $CntryPstl:35      -      ! 39

    $Space14:14      -      ! 74

    $EntrySeq:7      ! 88

move '13' to $AddendaType

move '01' to $DFIQualif

move 'US' to $DFICountry

move '      ' to $Space10

add 1 to #BatchIATEntryCount

```

add 1 to #BlockCountIAT

uppercase \$OrigBank

if #BlockCountIAT = 10

move 0 to #BlockCountIAT

add 1 to #FileIATBlockCount

end-if

write 2 from \$RecordType:1 - ! 01

\$AddendaType:2 - ! 02

\$OrigBank:35 - ! 04

\$DFIQualif:2 -

\$Transit#:34 - ! 39

\$DFICountry:3 - ! 74

\$Space10:10 -

\$EntrySeq:7 ! 88

move '14' to \$AddendaType

let \$RecBankNm = rtrim(&B.BANK_NM,' ')

uppercase \$RecBankNm

move '01' to \$DFIQualif

move 'US' to \$DFICountry

move ' ' to \$Space10

add 1 to #BatchIATEntryCount

add 1 to #BlockCountIAT

if #BlockCountIAT = 10

move 0 to #BlockCountIAT

add 1 to #FileIATBlockCount

end-if

write 2 from \$RecordType:1 - ! 01

\$AddendaType:2 - ! 02

\$RecBankNm:35 - ! 04

\$DFIQualif:2 -

\$F.Bank_cd:34 - ! 04

\$DFICountry:3 - ! 74

\$Space10:10 -

\$EntrySeq:7 ! 88

move '15' to \$AddendaType

move ' ' to \$Space34

let \$EmplAddr1 = ''

let \$EmplCityState = ''

let \$EmplCntryPstl = ''

Do Get-Empl-Address (\$E.EmplID, 'HOME', \$AddressFound)

if \$AddressFound = 'Y'

let \$EmplAddr1 = \$GETADDR_Address1

let \$CITY = \$GETADDR_CITY

let \$STATE = \$GETADDR_STATE

let \$POSTAL = \$GETADDR_POSTAL

let \$COUNTRY = \$GETADDR_COUNTRY

Do Get-Two-Char-Country(\$COUNTRY, \$Country_2Char)

if \$CITY = " and

\$STATE = "

move '\' to \$EmplCityState

else

if \$CITY = "

let \$EmplCityState = '*' || \$STATE || '\'

else

if \$STATE = "

let \$EmplCityState = \$CITY || '\'

else

let \$EmplCityState = \$CITY || '*' || \$STATE || '\'

end-if

end-if

end-if

if rtrim(\$Country_2Char,' ') = " and

rtrim(\$POSTAL,' ') = "

move '\' to \$EmplCntryPstl

```

else

    if rtrim($Country_2Char,' ') = "

        let $EmplCntryPstl = '*' || rtrim($POSTAL,' ') || '\'

    else

        if rtrim($POSTAL,' ') = "

            let $EmplCntryPstl = rtrim($Country_2Char,' ') || '\'

        else

            let $EmplCntryPstl = rtrim($Country_2Char,' ') || '*' || rtrim($POSTAL,' ') || '\'

        end-if

    end-if

end-if

end-if

uppercase $EmplAddr1

uppercase $EmplCityState

uppercase $EmplCntryPstl

add 1 to #BatchIATEntryCount

add 1 to #BlockCountIAT


if #BlockCountIAT = 10

    move 0 to #BlockCountIAT

    add 1 to #FileIATBlockCount

end-if


write 2 from $RecordType:1      -      ! 01

    $AddendaType:2      -      ! 02

    $EmplID:15      -      ! 40

```


\$EmplAddr1:35	-	
\$Space34:34	-	
\$EntrySeq:7		! 88

move '16' to \$AddendaType

add 1 to #BatchIATEntryCount

add 1 to #BlockCountIAT

if #BlockCountIAT = 10

move 0 to #BlockCountIAT

add 1 to #FileIATBlockCount

end-if

write 2 from \$RecordType:1	-	! 01
-----------------------------	---	------

\$AddendaType:2	-	! 02
-----------------	---	------

\$EmplCityState:35	-	
--------------------	---	--

\$EmplCntryPstl:35	-	
--------------------	---	--

\$Space14:14	-	
--------------	---	--

\$EntrySeq:7		! 88
--------------	--	------

end-procedure

begin-procedure Check-Prenote-Status

begin-SELECT

D.EFFDT

D.PRENOTE_STATUS

D.PRENOTE_DATE

D.PRENOTE_REQD

D.ACCOUNT_TYPE

```
if &D.Prenote_Status = 'S'          ! if submitted in previous payroll
    and $E.Pay_end_dt >= &D.Prenote_Date    ! and prenotification period has passed
do Complete-Prenote-Status
end-if
```

!Port Change - no prenote

```
! if &D.Prenote_Reqd = 'Y'
!     AND #WaitDays = 0
! do Update-Prenote-Reqd
! end-if
```

!End Port Change

FROM PS_DIR_DEP_DISTRIB D

WHERE EMPLID = \$E.Emplid

AND EFFDT <= \$E.Pay_end_dt

AND BANK_CD = \$F.Bank_cd

AND ACCOUNT_NUM = \$F.Account_num

ORDER BY EFFDT DESC

end-SELECT

end-procedure

begin-procedure Complete-Prenote-Status

begin-SQL

UPDATE PS_DIR_DEP_DISTRIB

SET PRENOTE_STATUS = 'C',

LAST_ACTION = 'C',

LAST_UPDATE_DATE = \$AsOfToday

WHERE EMPLID = \$E.Emplid

AND EFFDT = &D.Effdt

AND BANK_CD = \$F.Bank_cd

AND ACCOUNT_NUM = \$F.Account_num

AND ACCOUNT_TYPE = &D.Account_type

end-SQL

end-procedure

begin-PROCEDURE Update-Prenote-Reqd

begin-SQL

UPDATE PS_DIR_DEP_DISTRIB

SET PRENOTE_REQD = 'N',

LAST_ACTION = 'C',

LAST_UPDATE_DATE = \$AsOfToday

```
WHERE EMPLID      = $E.Emplid

AND EFFDT        = &D.Effdt

AND BANK_CD      = $F.BANK_CD

AND ACCOUNT_NUM  = $F.ACCOUNT_NUM
```

```
end-SQL
```

```
end-PROCEDURE
```

```
begin-PROCEDURE Close-File
```

```
if #BlockCount <> 0 and #BlockCount <> 10
```

```
do Complete-Block
```

```
end-if
```

```
close 1
```

```
move 'N' to $FileOpen
```

```
add 1 to #File_Count
```

```
!*** SR1453 - Modification Begin
```

```
close 2
```

```
do getEmailAddresses
```

```
if &A.COMPANY = 'TH'
```

```
LET $SYSTEM_CALL = 'mailx -s "PATH CitiBank Direct Deposit Summary" ' || $email || ' <
/tmp/ddp001msg.txt'
```

Else

```
    LET $SYSTEM_CALL = 'mailx -s "CITIBANK Direct Deposit Summary" ' || $email || ' <
/tmp/ddp001msg.txt'
```

End-If

CALL SYSTEM USING \$SYSTEM_CALL #RETVAL WAIT

```
LET $SYSTEM_CALL = 'rm /tmp/ddp001msg.txt'
```

CALL SYSTEM USING \$SYSTEM_CALL #RETVAL WAIT

!*** SR1453 - Modification End

end-PROCEDURE

begin-PROCEDURE Close-File-IAT

if #BlockCountIAT <> 0 and #BlockCountIAT <> 10

do Complete-Block-IAT

end-if

close 2

move 'N' to \$FileOpenIAT

add 1 to #File_Count

end-PROCEDURE

begin-procedure Write-Offset-Record-Sep

move '6' to \$RecordType

move '27' to \$TransactionCode !DDA Automatic Debit

move '0' to \$CheckDigit

do Format-Number(#TotBatchDebitAmt, \$Amount, '0000000009')

move ' ' to \$DiscrData

move '0' to \$AddendaIndicator

add 1 to #Trace#

let \$Transit_Temp = substr(\$priorTransit#, 0, 8)

do Calc-Alt-Trace

add 1 to #BatchEntryCount

move \$priorAccount# to \$CoAccount#

move \$priorTransit# to \$TransitRouting#

do Calc-DFI-CheckDigit

move \$priorTransit# to \$hashTransit# xxxxxxxx !Only first eight numbers

move \$hashTransit# to #hashTransit#

add #hashTransit# to #BatchEntryHash

add 1 to #BlockCount

```

if #BlockCount = 10
    move 0 to #BlockCount
    add 1 to #FileBlockCount
end-if

write 1 from $RecordType:1      -
    $TransactionCode:2      -
    $priorTransit#:8      -      ! company
    $CheckDigit:1      -
    $CoAccount#:17      -      ! company
    $Amount:10      -
    $Batch_CompanyID:15      -
    $Batch_CompanyName:22      -
    $DiscrData:2      -
    $AddendaIndicator:1      -
!    $Trace#:15
    $Alt_Trace:15

```

end-procedure

begin-procedure Write-Offset-Record-IAT

```

    move 0 to #TotBatchCreditAmt
    move 0 to #BatchEntryHash
!    move 0 to #BatchEntryCount    !Batch control
    move 0 to #Temp_TotBatchCreditAmt

```

move 'Y' to \$BatchOpenIAT

do Write-Batch-Header-CCD

move '6' to \$RecordType

move '27' to \$TransactionCode !DDA Automatic Debit

move '0' to \$CheckDigit

do Format-Number(#TotMultBatchDebitAmt, \$Amount, '0000000009')

move ' ' to \$DiscrData

move '0' to \$AddendaIndicator

add 1 to #Trace#

let \$Transit_Temp = substr(\$priorTransit#, 0, 8)

do Calc-Alt-Trace

add 1 to #BatchEntryCount

move \$priorAccount# to \$CoAccount#

move \$priorTransit# to \$TransitRouting#

do Calc-DFI-CheckDigit

move \$priorTransit# to \$hashTransit# xxxxxxxx !Only first eight numbers

move \$hashTransit# to #hashTransit#

add #hashTransit# to #BatchEntryHash

add 1 to #BlockCount


```
if #BlockCount = 10
    move 0 to #BlockCount
    add 1 to #FileBlockCount
end-if
```

```
write 1 from $RecordType:1      -
    $TransactionCode:2          -
    $priorTransit#:8            -      ! company
    $CheckDigit:1              -
    $CoAccount#:17              -      ! company
    $Amount:10                  -
    $CCD_Batch_Company:15       -
    $CCD_Batch_CompanyName:22   -
    $DiscrData:2                -
    $AddendaIndicator:1         -
!    $Trace#:15
    $Alt_Trace:15
```

```
move 'Y' to $Offset
do Write-Batch-Control
move 'N' to $BatchOpen
```

```
end-procedure
```

```
begin-procedure Write-Offset-Record-IAT-Sep
```

move 0 to #TotlATBatchCreditAmt
move 0 to #BatchIATEntryHash
move 0 to #BatchIATEntryCount !Batch control
move 0 to #Temp_TotlATBatchCreditAmt
move 'Y' to \$BatchOpenIAT
do Write-Batch-Header-CCD-Sep

move '6' to \$RecordType
move '27' to \$TransactionCode !DDA Automatic Debit
move '0' to \$CheckDigit
do Format-Number(#TotlATMultBatchDebitAmt, \$Amount, '0000000009')
move ' ' to \$DiscrData
move '0' to \$AddendaIndicator
add 1 to #Trace#

let \$Transit_Temp = substr(\$priorTransit#, 0, 8)

do Calc-Alt-Trace

move 1 to #BatchIATEntryCount

move \$priorAccount# to \$CoAccount#
move \$priorTransit# to \$TransitRouting#
do Calc-DFI-CheckDigit

move \$priorTransit# to \$hashTransit# xxxxxxxx !Only first eight numbers

move \$hashTransit# to #hashTransit#

add #hashTransit# to #BatchIATEntryHash

add 1 to #BlockCountIAT

if #BlockCountIAT = 10

move 0 to #BlockCountIAT

add 1 to #FileIATBlockCount

end-if

write 2 from \$RecordType:1 -

\$TransactionCode:2 -

\$priorTransit#:8 - ! company

\$CheckDigit:1 -

\$CoAccount#:17 - ! company

\$Amount:10 -

\$CCD_Batch_Company:15 -

\$CCD_Batch_CompanyName:22 -

\$DiscrData:2 -

\$AddendaIndicator:1 -

! \$Trace#:15

\$Alt_Trace:15

move 'Y' to \$OffsetIAT

do Write-Batch-Control-IAT-Sep

move 'N' to \$BatchOpenIAT

end-procedure

begin-procedure Write-Prenotes

move '6' to \$RecordType

move '0' to \$CheckDigit

move ' ' to \$DiscrData

move '0' to \$AddendaIndicator

move '0000000000' to \$Amount

move ' ' to \$Prev_Emplid

move ' ' to \$Prev_AcctType

move ' ' to \$Prev_AcctNum

move ' ' to \$Prev_BankCd

begin-SELECT

DD.EMPLID

DD.EFFDT

DD.PRIORITY

PD.NAME

DD.ACCOUNT_TYPE

DD.ACCOUNT_NUM

DD.BANK_CD

SB.BANK_CD

BTB.ALT_BANK_CD

```
if $Prev_Emplid <> &DD.EMPLID  
    or $Prev_AcctType <> &DD.ACCOUNT_TYPE  
    or $Prev_AcctNum <> &DD.ACCOUNT_NUM  
    or $Prev_BankCd <> substr(&DD.BANK_CD, 1, 8)
```

```
move &DD.EMPLID      to $Prev_Emplid  
move &DD.ACCOUNT_TYPE to $Prev_AcctType  
move &DD.ACCOUNT_NUM to $Prev_AcctNum  
let $Prev_BankCd = substr(&DD.BANK_CD, 1, 8)
```

```
if isblank(&BTB.ALT_BANK_CD)  
    move &SB.Bank_CD      to $TransitNbr  
else  
    move &BTB.Alt_bank_cd to $TransitNbr  
end-if
```

Display \$Transit_to_use

Display \$priorTransit#

```
if $SRC_PAYINIT.Py_ddp_sep_files = 'N'  
    if $Prev_IAT_Ind = 'Y' and  
        $IAT-Yes = 'Y' and  
        $BatchOpen = 'Y'  
        do Write-Batch-Control
```

move 'N' to \$Prev_IAT_Ind

end-if

end-if

if \$NewFile = 'Y'

move 'N' to \$NewFile

do Start-New-File

move '6' to \$RecordType

move '0' to \$CheckDigit

move ' ' to \$DiscrData

move '0' to \$AddendaIndicator

move '0000000000' to \$Amount

end-if

if \$BatchOpen = 'N'

move 'Y' to \$FromPrenotes

move 'N' to \$Prev_IAT_Ind

do Start-New-Batch

move '6' to \$RecordType

move '0' to \$CheckDigit

move ' ' to \$DiscrData

move '0' to \$AddendaIndicator

move '0000000000' to \$Amount

end-if

move &DD.EmplID to \$EmplID ! or SSN ?

move &DD.Effdt to \$DD_Effdt

move &DD.Priority to #DD_Priority

move &PD.Name to \$Name

UPPERCASE \$Name

if &DD.Account_Type = 'C'

 move '23' to \$TransactionCode !DDA Prenote Credit

else

 move '33' to \$TransactionCode !TDA Prenote Credit

end-if

do Update-Prenote-Status !Change prenote-status to submitted

move &DD.BANK_CD to \$TransitRouting#

do Calc-DFI-CheckDigit

move &DD.BANK_CD to \$hashTransit# xxxxxxxx !Only first eight numbers

move \$hashTransit# to #hashTransit#

add #hashTransit# to #BatchEntryHash

add 1 to #BatchEntryCount

add 1 to #BlockCount

if #BlockCount = 10

 move 0 to #BlockCount

 add 1 to #FileBlockCount

end-if

add 1 to #Trace#

move #Trace# to \$Trace# 0999999999999999

let \$Transit_Temp = substr(\$TransitNbr, 0, 8)

do Calc-Alt-Trace

WRITE 1 from \$RecordType:1 -

\$TransactionCode:2 -

&DD.BANK_CD:8 -

\$CheckDigit:1 -

&DD.ACCOUNT_NUM:17 -

\$Amount:10 -

\$EmplID:15 -

\$Name:22 -

\$DiscrData:2 -

\$AddendaIndicator:1 -

! \$Trace#:15

\$Alt_Trace:15

add 1 to #CompEntry

move 'N' to \$Prev_IAT_Ind

else

do Update-Prenote-Status !Change prenote-status to submitted

end-if


```

FROM PS_DIR_DEP_DISTRIB DD,

     PS_NAMES      PD,

     PS_PAYGROUP_TBL PT,

     PS_JOB        J,

     PS_DIRECT_DEPOSIT PS,

     PS_SRC_BANK    SB,

     PS_BANK_EC_TBL BTB

WHERE DD.EMPLID      = PD.EMPLID

     AND PS.EMPLID    = PD.EMPLID

     AND PT.PAYGROUP   = $PriorPaygroup

     AND DD.PRENOTE_STATUS = 'N'      !Not Submitted

     AND DD.PRENOTE_REQD  = 'Y'

     AND DD.IAT_IND       = 'N'      !Domestic prenotes only

     AND DD.ACCOUNT_TYPE  <> '$'      !not ISSUE CHECK

     AND DD.EFFDT =

         (SELECT MAX(DD1.EFFDT)

          FROM PS_DIR_DEP_DISTRIB DD1

          WHERE DD.EMPLID  = DD1.EMPLID

               AND DD1.EFFDT <= $A.Pay_End_Dt)

     AND PD.NAME_TYPE = 'PRI'

     AND PD.EFFDT =

         (SELECT MAX(PX.EFFDT)

          FROM PS_NAMES PX

          WHERE PX.EMPLID = PD.EMPLID

               AND PX.NAME_TYPE = PD.NAME_TYPE

```

```

AND PX.EFFDT    <= $A.Pay_End_Dt)

AND J.EMPLID    = PS.EMPLID

AND J.EMPLID    = DD.EMPLID

AND J.EFFDT =

(SELECT MAX(EFFDT)

FROM PS_JOB

WHERE EMPLID    = J.EMPLID

AND EMPL_RCD    = J.EMPL_RCD

AND EFFDT    <= $A.Pay_End_Dt)

AND J.EFFSEQ =

(SELECT MAX(EFFSEQ)

FROM PS_JOB

WHERE EMPLID    = J.EMPLID

AND EMPL_RCD    = J.EMPL_RCD

AND EFFDT    = J.EFFDT)

AND PS.EFF_STATUS    = 'A'

AND PS.EFFDT    = DD.EFFDT

AND PT.COMPANY    = J.COMPANY

AND PT.PAYGROUP    = J.PAYGROUP

AND PT.EFFDT =

(SELECT MAX(EFFDT)

FROM PS_PAYGROUP_TBL

WHERE COMPANY    = PT.COMPANY

AND PAYGROUP    = PT.PAYGROUP

AND EFFDT    <= $A.Pay_End_Dt)

AND PT.COMPANY    = $PriorCompany

```

```
AND SB.SRC_BANK_ID    = [$Bank_ID_to_Use_PT]

AND BTB.BANK_CD       = SB.BANK_CD

AND BTB.COUNTRY_CD    = 'USA'

AND [$Transit_to_Use] = $priorTransit#

ORDER BY DD.EMPLID, DD.ACCOUNT_TYPE, DD.ACCOUNT_NUM, DD.BANK_CD

end-SELECT
```

```
end-PROCEDURE
```

```
begin-procedure Write-Batch-Control
```

```
    move '8'                to $RecordType
```

```
    move '200'              to $ServiceClassCode
```

```
    add #BatchEntryCount    to #FileEntryCount
```

```
    do Format-Number(#BatchEntryCount, $BatchEntryCount, '000009')
```

```
    move 0 to #BatchEntryCount
```

```
    add #BatchEntryHash     to #FileEntryHash
```

```
    do Format-Number(#BatchEntryHash, $TestEntryHash, '0999999999999999')
```

```
    concat '*' with $TestEntryHash
```

```
    find '*' in $TestEntryHash 0 #Loc
```

```
    if #Loc > 10
```

```
        subtract 10 from #loc
```

```
        extract $BatchEntryHash from $TestEntryHash #loc 10
```

end-if

! ** SIR0999 Modification Begin **

! ** use debit for PATH - HCM00435

if &A.COMPANY = 'NO'

else

let #TotBatchDebitAmt = 0

end-if

do Format-Number(#TotBatchDebitAmt, \$TotBatchDebitAmt, '000000000009')

! ** SIR0999 Modification End **

if \$Offset = 'N'

add #TotBatchDebitAmt to #TotFileDebitAmt

add #TotBatchCreditAmt to #TotFileCreditAmt

add #TotBatchDebitAmt to #TotMultBatchDebitAmt

add #TotBatchCreditAmt to #TotMultBatchCreditAmt

! ** use debit for PATH - HCM00435

if &A.COMPANY = 'NO'

else

move '000000000000' to \$TotBatchDebitAmt

END-IF

do Format-Number(#TotBatchCreditAmt, \$TotBatchCreditAmt, '000000000009')

else

do Format-Number(#TotMultBatchDebitAmt, \$TotBatchDebitAmt, '000000000009')

move '000000000000' to \$TotBatchCreditAmt

move 0 to #TotMultBatchDebitAmt

move 0 to #TotMultBatchCreditAmt

end-if

move ' ' to \$Reserved1

move ' ' to \$Reserved2

move \$priorTransit# to \$TransitRouting#

do Format-Number(#Batch#, \$Batch#, '0000009')

add 1 to #BlockCount

if #BlockCount = 10

move 0 to #BlockCount

add 1 to #FileBlockCount

end-if

write 1 from \$RecordType:1 -

\$ServiceClassCode:3 -

\$BatchEntryCount:6 -

\$BatchEntryHash:10 -

\$TotBatchDebitAmt:12 -

\$TotBatchCreditAmt:12 -

\$Batch_CompanyID:10 -

\$Reserved1:19 -

\$Reserved2:6 -

\$TransitRouting#:8 -

\$Batch#:7

move 'N' to \$BatchOpen

!*** SR1453 - Modification Begin

let \$dispDetailRec = ltrim(edit(\$BatchEntryCount, '999,999'), '0')

let \$dispDetailRec = ltrim(\$dispDetailRec, ',')

write 2 from ' ':1

write 2 from 'NUMBER OF DETAIL RECORDS':38

\$dispDetailRec:7

!*** SR1453 - Modification End

move 'N' to \$Offset

end-procedure

begin-procedure Write-Batch-Control-IAT-Sep

move '8' to \$RecordType

move '200' to \$ServiceClassCode

add #BatchIATEntryCount to #FileIATEntryCount

do Format-Number(#BatchIATEntryCount, \$BatchIATEntryCount, '000009')

move 0 to #BatchIATEntryCount

add #BatchIATEntryHash to #FileIATEntryHash

do Format-Number(#BatchIATEntryHash, \$TestIATEntryHash, '0999999999999999')

concat '*' with \$TestIATEntryHash

find '*' in \$TestIATEntryHash 0 #Loc

if #Loc > 10

subtract 10 from #loc

extract \$BatchIATEntryHash from \$TestIATEntryHash #loc 10

end-if

do Format-Number(#TotIATBatchDebitAmt, \$TotIATBatchDebitAmt, '000000000009')

if \$OffsetIAT = 'N'

add #TotIATBatchDebitAmt to #TotIATFileDebitAmt

add #TotIATBatchCreditAmt to #TotIATFileCreditAmt

add #TotIATBatchDebitAmt to #TotIATMultBatchDebitAmt

add #TotIATBatchCreditAmt to #TotIATMultBatchCreditAmt

move '000000000000' to \$TotIATBatchDebitAmt

do Format-Number(#TotIATBatchCreditAmt, \$TotIATBatchCreditAmt, '000000000009')

else

do Format-Number(#TotIATMultBatchDebitAmt, \$TotIATBatchDebitAmt, '000000000009')

move '000000000000' to \$TotIATBatchCreditAmt

move 0 to #TotIATMultBatchDebitAmt

move 0 to #TotIATMultBatchCreditAmt

end-if

move ' ' to \$Reserved1

move ' ' to \$Reserved2

move \$priorTransit# to \$TransitRouting#

do Format-Number(#IATBatch#, \$IATBatch#, '0000009')

add 1 to #BlockCountIAT

if #BlockCountIAT = 10

 move 0 to #BlockCountIAT

 add 1 to #FileIATBlockCount

end-if

write 2 from \$RecordType:1 -

 \$ServiceClassCode:3 -

 \$BatchIATEntryCount:6 -

 \$BatchIATEntryHash:10 -

 \$TotIATBatchDebitAmt:12 -

 \$TotIATBatchCreditAmt:12 -

 \$Batch_CompanyID:10 -

 \$Reserved1:19 -

 \$Reserved2:6 -

 \$TransitRouting#:8 -

 \$IATBatch#:7

move 'N' to \$BatchOpenIAT

move 'N' to \$OffsetIAT

end-procedure

begin-procedure Write-File-Control

move '9' to \$RecordType

do Format-Number(#FileBatchCount, \$FileBatchCount, '000009')

do Format-Number(#FileEntryCount, \$FileEntryCount, '00000009')

do Format-Number(#FileEntryHash, \$TestEntryHash, '0999999999999999')

concat '*' with \$TestEntryHash

find '*' in \$TestEntryHash 0 #Loc

if #Loc > 10

subtract 10 from #loc

extract \$FileEntryHash from \$TestEntryHash #loc 10

else

do Format-Number(#FileEntryHash, \$FileEntryHash, '0999999999')

end-if

if \$SRC_PAYINIT.Py_ddp_excl_offs = 'Y'

move 0 to #TotFileDebitAmt

end-if

!** SIR0999 Modification Begin **

!** Use debit for PATH - HCM00435

if &A.COMPANY = 'NO'

ELSE

let #TotFileDebitAmt = 0

END-IF

do Format-Number(#TotFileDebitAmt, \$TotFileDebitAmt, '000000000009')

!** SIR0999 Modification End **

do Format-Number(#TotFileCreditAmt, \$TotFileCreditAmt, '000000000009')

move ' ' to \$Reserved1

add 1 to #BlockCount

if #BlockCount = 10

move 0 to #BlockCount

add 1 to #FileBlockCount

else

if #BlockCount > 0

add 1 to #FileBlockCount

end-if

end-if

do Format-Number(#FileBlockCount, \$FileBlockCount, '000009')

write 1 from \$RecordType:1 -

\$FileBatchCount:6 -

\$FileBlockCount:6 -

\$FileEntryCount:8 -

\$FileEntryHash:10 -

\$TotFileDebitAmt:12 -

\$TotFileCreditAmt:12 -

\$Reserved1:39

!*** SR1453 - Modification Begin

let \$dispTotAmt = #TotFileCreditAmt / 100

let \$dispTotAmt = edit(\$dispTotAmt, '9,999,999,999.99')

write 2 from ' ':1

write 2 from 'TOTAL CREDIT AMOUNT':30

\$dispTotAmt:16

write 2 from ' ':1

write 2 from ' ':1

write 2 from ' ':40

'END OF REPORT':13

!*** SR1453 - Modification End

end-procedure

begin-procedure Write-File-Control-IAT

move '9' to \$RecordType

do Format-Number(#FileIATBatchCount, \$FileIATBatchCount, '000009')

do Format-Number(#FileIATEntryCount, \$FileIATEntryCount, '00000009')

do Format-Number(#FileIATEntryHash, \$TestIATEntryHash, '0999999999999999')

concat '*' with \$TestIATEntryHash

find '*' in \$TestIATEntryHash 0 #Loc

if #Loc > 10

 subtract 10 from #loc

 extract \$FileIATEntryHash from \$TestIATEntryHash #loc 10

else

 do Format-Number(#FileIATEntryHash, \$FileIATEntryHash, '0999999999')

end-if

if \$SRC_PAYINIT.Py_ddp_excl_offs = 'Y'

 move 0 to #TotIATFileDebitAmt

end-if

do Format-Number(#TotIATFileDebitAmt, \$TotIATFileDebitAmt, '000000000009')

do Format-Number(#TotIATFileCreditAmt, \$TotIATFileCreditAmt, '000000000009')

move ' ' to \$Reserved1

add 1 to #BlockCountIAT

if #BlockCountIAT = 10

 move 0 to #BlockCountIAT

 add 1 to #FileIATBlockCount

else

 if #BlockCountIAT > 0

 add 1 to #FileIATBlockCount

 end-if

end-if

do Format-Number(#FileIATBlockCount, \$FileIATBlockCount, '000009')

write 1 from \$RecordType:1 -
\$FileIATBatchCount:6 -
\$FileIATBlockCount:6 -
\$FileIATEntryCount:8 -
\$FileIATEntryHash:10 -
\$TotIATFileDebitAmt:12 -
\$TotIATFileCreditAmt:12 -
\$Reserved1:39

end-procedure

begin-procedure Write-File-Control-IAT-Sep

move '9' to \$RecordType

do Format-Number(#FileIATBatchCount, \$FileIATBatchCount, '000009')

do Format-Number(#FileIATEntryCount, \$FileIATEntryCount, '00000009')

do Format-Number(#FileIATEntryHash, \$TestIATEntryHash, '0999999999999999')

concat '*' with \$TestIATEntryHash

find '*' in \$TestIATEntryHash 0 #Loc

if #Loc > 10

```
subtract 10 from #loc

extract $FileIATEntryHash from $TestIATEntryHash #loc 10

else

do Format-Number(#FileIATEntryHash, $FileIATEntryHash, '0999999999')

end-if


if $RC_PAYINIT.Py_ddp_excl_offs = 'Y'

move 0 to #TotIATFileDebitAmt

end-if


do Format-Number(#TotIATFileDebitAmt, $TotIATFileDebitAmt, '000000000009')
do Format-Number(#TotIATFileCreditAmt, $TotIATFileCreditAmt, '000000000009')
move ' ' to $Reserved1


add 1 to #BlockCountIAT


if #BlockCountIAT = 10

move 0 to #BlockCountIAT

add 1 to #FileIATBlockCount

else

if #BlockCountIAT > 0

add 1 to #FileIATBlockCount

end-if

end-if
```

do Format-Number(#FileIATBlockCount, \$FileIATBlockCount, '000009')

write 2 from \$RecordType:1 -

\$FileIATBatchCount:6 -

\$FileIATBlockCount:6 -

\$FileIATEntryCount:8 -

\$FileIATEntryHash:10 -

\$TotIATFileDebitAmt:12 -

\$TotIATFileCreditAmt:12 -

\$Reserved1:39

end-procedure

begin-procedure Complete-Block

move '99999999999999999999999999999999' to \$ThirtyNines

while #BlockCount < 10

write 1 from \$ThirtyNines:30 -

\$ThirtyNines:30 -

\$ThirtyNines:30 -

\$ThirtyNines:4

add 1 to #BlockCount

end-while

end-procedure

begin-procedure Complete-Block-IAT

```
move '999999999999999999999999999999' to $ThirtyNines
```

```
while #BlockCountIAT < 10
```

```
write 2 from $ThirtyNines:30 -
```

\$ThirtyNines:30 -

\$ThirtyNines:30 -

\$ThirtyNines:4

```
add 1 to #BlockCountIAT
```

end-while

end-procedure

begin-PROCEDURE Calc-DFI-CheckDigit

```
move 0 to #CheckDigit
```

```
move 0 to #i
```

```
while #i <= #lastIndex
```

```
extract $digit from $TransitRouting# #i 1
```

```
move $digit to #digit
```

```
get #factor from Weights(0) Factor(#i)
```

multiply #factor times #digit

add #digit to #CheckDigit

add 1 to #i

end-while

while #CheckDigit > 9

subtract 10 from #CheckDigit

end-while

do Format-Number(#CheckDigit, \$CheckDigit, '9')

move \$CheckDigit to #tempDigit

move 10 to #CheckDigit

subtract #tempDigit from #CheckDigit

while #CheckDigit > 9

subtract 10 from #CheckDigit

end-while

do Format-Number(#CheckDigit, \$CheckDigit, '9')

end-PROCEDURE !Calc-DFI-CheckDigit

begin-PROCEDURE Calc-Alt-Trace

if length(\$Transit_Temp) < 8

let #diff = 8 - length(\$Transit_Temp)

end-if

```
do Format-Number(#Trace#, $Trace_Temp, '0999999')
```

```
let $Alt_Trace = $Transit_Temp || $Trace_Temp
```

```
if #diff = 1
```

```
    let $Alt_Trace = $Transit_Temp || ' ' || $Trace_Temp
```

```
else
```

```
    if #diff = 2
```

```
        let $Alt_Trace = $Transit_Temp || ' ' || $Trace_Temp
```

```
    else
```

```
        if #diff = 3
```

```
            let $Alt_Trace = $Transit_Temp || ' ' || $Trace_Temp
```

```
        else
```

```
            if #diff = 4
```

```
                let $Alt_Trace = $Transit_Temp || ' ' || $Trace_Temp
```

```
            else
```

```
                if #diff = 5
```

```
                    let $Alt_Trace = $Transit_Temp || ' ' || $Trace_Temp
```

```
                else
```

```
                    if #diff = 6
```

```
                        let $Alt_Trace = $Transit_Temp || ' ' || $Trace_Temp
```

```
                    end-if
```

```
                end-if
```

```
            end-if
```

```
        end-if
```

end-if

end-if

end-PROCEDURE !Calc-Alt-Trace

begin-PROCEDURE Update-Prenote-Status

begin-SQL

UPDATE PS_DIR_DEP_DISTRIB

SET PRENOTE_STATUS = 'S',

PRENOTE_DATE = \$Prenote_Date,

LAST_ACTION = 'C',

LAST_UPDATE_DATE = \$AsOfToday

WHERE EMPLID = \$EmplID

AND EFFDT = \$DD_Effdt

AND PRIORITY = #DD_Priority

end-SQL

end-PROCEDURE

begin-procedure Get-Transit-Descr

begin-SELECT

C.BANK_NM

FROM PS_BANK_EC_TBL C

WHERE (C.BANK_CD = \$Transit# OR

C.ALT_BANK_CD = \$Transit#)

AND C.COUNTRY_CD = 'USA'

end-SELECT

end-procedure

Begin-Procedure Get-Two-Char-Country(\$Three-Char-Country, :\$Two-Char-Country)

Let \$Two-Char-Country = '??'

Begin-Select

CTB.COUNTRY_2CHAR

move &CTB.COUNTRY_2CHAR to \$Two-Char-Country

FROM PS_COUNTRY_TBL CTB

WHERE CTB.COUNTRY = \$Three-Char-Country

End-Select

End-Procedure

Begin-Procedure Lower-To-Upper(:\$Name)

UpperCase \$Name

Let \$Name = translate(\$Name, 'á', 'Á')

Let \$Name = translate(\$Name, 'â', 'Â')

Let \$Name = translate(\$Name, 'ä', 'Ä')

Let \$Name = translate(\$Name, 'à', 'À')

Let \$Name = translate(\$Name, 'å', 'Å')

Let \$Name = translate(\$Name, 'ã', 'Ã')

Let \$Name = translate(\$Name, 'æ', 'Æ')

Let \$Name = translate(\$Name, 'ç', 'Ç')

Let \$Name = translate(\$Name, 'é', 'É')

Let \$Name = translate(\$Name, 'ê', 'Ê')

Let \$Name = translate(\$Name, 'ë', 'Ë')

Let \$Name = translate(\$Name, 'è', 'È')

Let \$Name = translate(\$Name, 'í', 'Í')

Let \$Name = translate(\$Name, 'î', 'Î')

Let \$Name = translate(\$Name, 'ï', 'Ï')

Let \$Name = translate(\$Name, 'ì', 'Ì')

Let \$Name = translate(\$Name, 'ñ', 'Ñ')

Let \$Name = translate(\$Name, 'ó', 'Ó')

Let \$Name = translate(\$Name, 'ô', 'Ô')

Let \$Name = translate(\$Name, 'ö', 'Ö')

Let \$Name = translate(\$Name, 'ò', 'Ò')

Let \$Name = translate(\$Name, 'õ', 'Õ')

Let \$Name = translate(\$Name, 'ú', 'Ú')

Let \$Name = translate(\$Name, 'û', 'Û')

Let \$Name = translate(\$Name, 'ü', 'Ü')

Let \$Name = translate(\$Name, 'ù', 'Ù')

Let \$Name = translate(\$Name, 'ý', 'Ý')

End-Procedure