```
!*****************************
!  PAY002:  Payroll Register  *
!*****************************
```

```
!***********************************************************************
!                                                                      *
!                                                                      *
!                                                                      *
!                                                                      *
! 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, 2018, Oracle and/or its affiliates.              *
! All Rights Reserved.                                                 *
!***********************************************************************
```

```
!**********************************************************************
!                                                     *
!      $Release:  HR92                                          !
!         $Bug:  27002643                                       !
!                                                     *
!**********************************************************************
!
!**********************************************************************
! Modified for Education & Government                              *
! HP99999       Release 8 Technical Merge                           *
!**********************************************************************

#include 'setenv.sqc' !Set environment
#Include 'setup32.sqc'  !Printer and page-size initialization

begin-report
  do Init-Report
  do Stdapi-Term
end-report

begin-procedure Init-Report
  move 'PAY002' to $ReportID

  do Init-DateTime
  do Init-Number
  do Stdapi-Init
  do Get-Report-Language   ! Procedure that decides the report language

                           ! and then calls 'Report-Translation' procedure

                           ! that reads the language dep. text.

  move $Pay002_ReportTitle to $ReportTitle


  do Array-Initialization


  do Payroll-Report-Initialization


end-procedure


! This procedure is used to read language-dependent text from the database.

begin-procedure Report-Translation


! Read from PAY002 program id

  do Init_Report_Translation($ReportID,$curr_language_cd)

  do Append_Report_Translation('HR')


  if $Program_Read = 'Y'

    do Get_Field_Information ('PAY002', 'REPORT_TITLE', $Pay002_ReportTitle, #dummy_width)

    do Get_Field_Information ('PAY002', 'GR_TOT',       $Pay002_Gr_Tot,      #dummy_width)

    do Get_Field_Information ('PAY002', 'COMP_TOT',     $Pay002_Comp_Tot,    #dummy_width)

    do Get_Field_Information ('PAY002', 'PAYPER_TOT',   $Pay002_PayPer_Tot,  #dummy_width)

    do Get_Field_Information ('PAY002', 'PAYGRP_TOT',   $Pay002_PayGrp_Tot,  #dummy_width)

    do Get_Field_Information ('PAY002', 'DEPT_TOT',     $Pay002_Dept_Tot,    #dummy_width)
    do Get_Field_Information ('PAY002', 'EMPNM_ID',     $Pay002_EmpNm_Id,    #dummy_width)
    do Get_Field_Information ('PAY002', 'FORM_ID',      $Pay002_Form_Id,     #dummy_width)
    do Get_Field_Information ('PAY002', 'REGULAR',      $Pay002_Regular,     #dummy_width)
    do Get_Field_Information ('PAY002', 'OVERTIME',     $Pay002_Overtime,    #dummy_width)
    do Get_Field_Information ('PAY002', 'OTHER',        $Pay002_Other,       #dummy_width)
    do Get_Field_Information ('PAY002', 'DEPT_ID',      $Pay002_Dept_Id,     #dummy_width)
    do Get_Field_Information ('PAY002', 'CHECK_DT',     $Pay002_Check_Dt,    #dummy_width)
    do Get_Field_Information ('PAY002', 'CHECK_NO',     $Pay002_Check_No,    #dummy_width)
    do Get_Field_Information ('PAY002', 'HOURS',        $Pay002_Hours,       #dummy_width)
    do Get_Field_Information ('PAY002', 'EARNINGS',     $Pay002_Earnings,    #dummy_width)
    do Get_Field_Information ('PAY002', 'TYPE',         $Pay002_Type,        #dummy_width)
    do Get_Field_Information ('PAY002', 'GROSS_PAY',    $Pay002_Gross_Pay,   #dummy_width)
    do Get_Field_Information ('PAY002', 'TAXES',        $Pay002_Taxes,       #dummy_width)
    do Get_Field_Information ('PAY002', 'DEDUCTIONS',   $Pay002_Deductions,  #dummy_width)
    do Get_Field_Information ('PAY002', 'CODE',         $Pay002_Code,        #dummy_width)
    do Get_Field_Information ('PAY002', 'NET_PAY',      $Pay002_Net_Pay,     #dummy_width)
    do Get_Field_Information ('PAY002', 'LEGEND',       $Pay002_Legend,      #dummy_width)

    do Get_Field_Information ('HR',     'BUS_UNIT',     $HR_Bus_Unit,        #dummy_width)



    let $Pay002_Hours      = lpad($Pay002_Hours,6,' ')
    let $Pay002_Earnings   = lpad($Pay002_Earnings,8,' ')
    let $Pay002_Gross_Pay  = lpad($Pay002_Gross_Pay,9,' ')
    let $Pay002_Taxes      = lpad($Pay002_Taxes,6,' ')
    let $Pay002_Deductions = lpad($Pay002_Deductions,10,' ')
```

```
    let $Pay002_Net_Pay    = lpad($Pay002_Net_Pay,8,' ')
  end-if

end-procedure

begin-procedure Array-Initialization

  create-array name=Totals              -
               size=6                   -
               field=TotalName:char     -
               field=total:number:13    -
               field=startCol:number:13

  put $Pay002_Gr_Tot      into Totals(1) TotalName
  put $Pay002_Comp_Tot    into Totals(2) TotalName
  put $Pay002_PayPer_Tot  into Totals(3) TotalName
  put $Pay002_PayGrp_Tot  into Totals(4) TotalName
  put $Pay002_Dept_Tot    into Totals(5) TotalName

  put  27 into Totals(0) startCol(0)
  put  41 into Totals(0) startCol(1)
  put  50 into Totals(0) startCol(2)
  put  61 into Totals(0) startCol(3)
  put  74 into Totals(0) startCol(4)
  put  88 into Totals(0) startCol(5)
  put 102 into Totals(0) startCol(6)
  put 115 into Totals(0) startCol(7)

  put 138 into Totals(0) startCol(8)

  put 162 into Totals(0) startCol(9)

  put 162 into Totals(0) startCol(10)

  put 162 into Totals(0) startCol(11)

  put 162 into Totals(0) startCol(12)



! increase array size 12/12/18

  create-array name=ETDarray          -

              size=500               -

              field=RegEnHrs:number  -

              field=RegEarn:number   -

              field=OTHrs:number     -

              field=OTEarn:number    -

              field=EarnType:char    -

              field=EarnHours:number -

              field=EarnGross:number -

              field=AddToGrs:char    -

              field=TaxAmt:number    -

              field=TaxType:char     -

              field=DedAmt:number    -

              field=DedCode:char


end-procedure
```

```
begin-heading 8

 #Include 'stdhtrpi.sqc'
 #Include 'stdhtr06.sqc'

 print $Pay002_EmpNm_Id         (+2,1)
 print $Pay002_Form_Id          (0,25)
 print $pay002_Regular          (0,35)
 print $Pay002_Overtime         (0,58)
 print $Pay002_Other            (0,78)
 print $Pay002_Dept_Id          (+1,1)
 print $Hr_Bus_Unit             (0,13)
 print $Pay002_Hours            (0,35)
 print $Pay002_Earnings         (0,47)
 print $Pay002_Hours            (0,58)
 print $Pay002_Earnings         (0,67)
 print $Pay002_Type             (0,77)
 print $Pay002_Hours            (0,82)
 print $Pay002_Earnings         (0,94)
 print $Pay002_Gross_Pay        (0,107)
 print $Pay002_Taxes            (0,123)
 print $Pay002_Type             (0,130)
 print $Pay002_Deductions       (0,142)
 print $Pay002_Code             (0,153)
 print $Pay002_Net_Pay          (0,168)
 print $Pay002_Check_Dt         (+1,1)
```

```
  print $Pay002_Check_No          (0,13)


end-heading


begin-procedure Report


begin-SELECT

A.COMPANY    () on-break print=never  level=1 procedure=Company-Change

A.PAY_END_DT () on-break print=never  level=2 procedure=PayEndDate-Change

A.PAYGROUP   () on-break print=never  level=3 procedure=PayGroup-Change

B.DEPTID     () on-break print=never  level=4 procedure=Department-Change

B.NAME

B.EMPLID

A.CHECK_DT

B.CHECK_DT

B.PAYCHECK_OPTION

B.PAYCHECK_STATUS

B.FORM_ID

B.PAYCHECK_NBR

B.TOTAL_GROSS

B.NET_PAY

B.TOTAL_TAXES

B.TOTAL_DEDUCTIONS

B.OFF_CYCLE

B.PAGE_NUM

B.LINE_NUM
```

```
B.SEPCHK

B.BUSINESS_UNIT

P.COUNTRY

B.PAY_SHEET_SRC


  move &A.Company     to $Company

  move &A.PayGroup    to $PayGroup

  move &A.Pay_End_Dt  to $PayEndDate

  move &B.OFF_CYCLE   to $Off-Cycle

  move &A.Check_Dt    to $AsOfDate  !for included procedures


  do Initialize-ETD-Array


  do Get-Pay-Earnings


  if $Pay_Status_OK = 'Y'


    if #Current-Line < 12

        move 'N' to $Overflow_SW

    else

        move 'Y' to $Overflow_SW

    end-if

      let $Name = RTRIM(&B.Name, ' ')

    print $Name                   (3,1,21)

    print &B.Form_ID              (3,25)

    print &B.EmplID               (4,1)
    print &B.DeptID              (5,1)

    print &B.Business_unit        (5,13)


    if not IsNULL(&B.Check_Dt)

      do Format-DateTime(&B.Check_Dt, $out, {DEFDATE}, '', '')

      print $out  (6,1)

    end-if


    evaluate &B.PayCheck_Option

      when = 'R'

      when = 'M'

      when = 'A'

        print &B.PayCheck_Option  (6,12)

    end-evaluate


    if &B.PayCheck_Option = '&' or &B.PayCheck_Option = 'A'

      do Get-Pay-Distribution

    end-if


    if &B.PAYCHECK_NBR <> 0

      do Format-Number(&B.PAYCHECK_NBR, $out, '099999999999999')

      print $out         (6,13)

    end-if


    if &B.Pay_Sheet_Src = 'O' or &B.Pay_Sheet_Src = 'K' or &B.Pay_Sheet_Src = 'L'

       print &B.Pay_Sheet_Src   (6,28)
```

```
end-if


if &B.PayCheck_Option = '&'                    ! Advice and Check

  print 'A'          (7,13)

  do Format-Number(#Advice#, $out, '099999999999999')

  print $out         (7,15)

end-if


do Format-Number(&B.Total_Gross, $out, '99,999,999.99mi')

print $out         (3,103)


if &B.PayCheck_Option <> '&'

   or #Check_Amount = 0

   or #Deposit_Total = 0


  do Format-Number(&B.Net_Pay, $out, '99,999,999.99mi')

  print $out         (3,163)

end-if


evaluate &B.PayCheck_Option

  when = 'C'

  when = 'M'


    if &B.PayCheck_Status = 'F'

            or &B.PayCheck_Status = 'A'

            or &B.PayCheck_Status = 'R'
```

```
      move &B.Net_Pay to #Check_Amount

      print 'C'           (+0,0)

    else

      move &B.Net_Pay to #UnConfirm_Total

      print 'U'           (+0,0)

    end-if


  when = 'A'

    print 'A'           (+0,0)

  when = '&'

    move 2 to #LINE_NUM


    if #Check_Amount <> 0

      do Format-Number(#Check_Amount, $out, '99,999,999.99mi')

      print $out  (#LINE_NUM,163)

      print 'C'           (+0,0)

      add 1 to #LINE_NUM

    end-if


    if #Deposit_Total <> 0

      do Format-Number(#Deposit_Total, $out, '99,999,999.99mi')

      print $out (#LINE_NUM,163)

      print 'A'           (+0,0)

      add 1 to #LINE_NUM

    end-if
  when = 'R'
    if &B.PayCheck_Status = 'F'
            or &B.PayCheck_Status = 'A'
        do Process-Reversal
    else
        move &B.Net_Pay to #UnConfirm_Total
    end-if
end-evaluate

ARRAY-ADD &B.Total_Gross      -
          &B.Total_Taxes      -
          &B.Total_Deductions -
          &B.Net_Pay          -
          #Check_Amount       -
          #Deposit_Total      -
          #UnConfirm_Total    -
  to Totals(5) total(6)       -
               TOTAL(7)       -
               TOTAL(8)       -
               TOTAL(9)       -
               TOTAL(10)      -
               TOTAL(11)      -
               TOTAL(12)

if &P.Country = 'USA'
  do Get-USA-Taxes
```

```
      let $Ded_Class_Clause = ' <= "K"'
    else
      if &P.Country = 'CAN'
        do Get-CAN-Taxes
        let $Ded_Class_Clause = ' IN ("A","B","L","T")'
      end-if
    end-if

    do Get-Deductions
    do Print-Detail
  end-if

  move 0 to #Advice#
  move 0 to #UnConfirm_Total
  move 0 to #Deposit_Total
  move 0 to #Check_Amount
  move 0 to #Cash_Total
  NEXT-LISTING NEED=7

FROM  PS_PAY_CALENDAR A,
      PS_PAY_CHECK    B,
      PS_PAYGROUP_TBL P
WHERE [$SlctCalendar]
  AND B.OFF_CYCLE IN ($SlctOffCycleA, $SlctOffCycleB)
  AND B.PAGE_NUM BETWEEN #SlctPageFrom AND #SlctPageThru
  AND B.COMPANY    = A.COMPANY
  AND B.PAYGROUP   = A.PAYGROUP
  AND B.PAY_END_DT = A.PAY_END_DT
  AND P.COMPANY    = A.COMPANY
  AND P.PAYGROUP   = A.PAYGROUP
  AND P.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)
ORDER BY A.COMPANY,
         A.PAY_END_DT,
         A.PAYGROUP,
         B.DEPTID,
         B.NAME,
         B.EMPLID,
         B.PAGE_NUM,
         B.LINE_NUM,
         B.SEPCHK
end-SELECT

  move 1 to #changeLevel
  move 'Y' to $End
  do Print-Totals
```

```
end-procedure



begin-procedure Initialize-ETD-Array

 move 0 to #i

  while #i <= #ETDmax
  put            0      -
               0      -
               0      -
               0      -
               ' '    -
               0      -
               0      -
               ' '    -
               0      -
               ' '    -
               0      -
               ' '    -
    into ETDarray(#i)
               RegEnHrs  -
               RegEarn   -
               OTHrs     -
               OTEarn    -
               EarnType  -
```

```
                          EarnHours -

                          EarnGross -

                          AddToGrs  -

                          TaxAmt    -

                          TaxType   -

                          DedAmt    -

                          DedCode

    add 1 to #i

  end-while


  move -1 to #ETDmax


end-procedure




begin-procedure Get-Pay-Earnings


  position (+1)

  move 0 to #i

  move 'N' to $Pay_Status_OK


begin-SELECT

C.REG_HRS+C.REG_EARN_HRS     &Hours

C.REG_EARNS+C.REG_HRLY_EARNS &Earnings

C.OT_HRS                     &C.Ot_Hrs

C.OT_HRLY_EARNS              &C.Ot_Hrly_Earns
```

```
C.ADDL_NBR


!*********This logic was added to add Reg earns to the array in order to print from one array **********

!********* and to have one page breaking logic for the report *********************



  PUT               &Hours     -

                  &Earnings  -

                  &C.Ot_Hrs  -

                  &C.Ot_Hrly_Earns -

   INTO ETDARRAY(#i)  RegEnHrs  -

                   RegEarn   -

                   OTHrs     -

                   OTEarn






!*********************************************************************************


  ARRAY-ADD &Hours          -

          &Earnings       -

          &C.OT_Hrs       -

          &C.OT_Hrly_Earns -

   to Totals(5) total(0)    -
```

```
            TOTAL(1)   -
            TOTAL(2)   -
            TOTAL(3)

  move 'Y' to $Add_to_Array
  do Get-Other-Hours-Earnings
  move 'Y' to $Pay_Status_OK

FROM  PS_PAY_EARNINGS C
WHERE COMPANY        = $Company
  AND PAYGROUP       = $PayGroup
  AND PAY_END_DT     = $PayEndDate
  AND OFF_CYCLE      = $Off-Cycle
  AND PAGE_NUM          = &B.PAGE_NUM
  AND LINE_NUM          = &B.LINE_NUM
  AND SEPCHK         = &B.SepChk
  AND OK_TO_PAY      = 'Y'
  AND PAY_LINE_STATUS IN ('C','F','V')
  AND C.SINGLE_CHECK_USE IN ('C', 'N')
ORDER BY C.ADDL_NBR
end-SELECT

end-procedure


begin-procedure Get-Other-Hours-Earnings
```

```
begin-SELECT
ERNCD,
OTH_HRS,
OTH_EARNS,
ADD_GROSS



 PUT            &ErnCd     -
              &Oth_Hrs   -
              &Oth_Earns -
              &Add_Gross -
  INTO ETDARRAY(#I)  EARNTYPE  -
               EARNHOURS -
               EARNGROSS -
               ADDTOGRS



 ARRAY-ADD     &Oth_Hrs   -
           &Oth_Earns -
  to Totals(5) total(4)  -
            TOTAL(5)

  add 1 to #i
```

```
    let $Add_to_Array = 'N'


FROM  PS_PAY_OTH_EARNS
WHERE COMPANY    = $Company
  AND PAYGROUP   = $PayGroup
  AND PAY_END_DT = $PayEndDate
  AND OFF_CYCLE  = &B.Off_Cycle
  AND PAGE_NUM      = &B.PAGE_NUM
  AND LINE_NUM      = &B.LINE_NUM
  AND SEPCHK     = &B.SepChk
  AND ADDL_NBR      = &C.ADDL_NBR
ORDER BY ERNCD
end-SELECT


  if $Add_to_Array = 'Y'
     add 1 to #i
  end-if


  if #i > #ETDmax
    move #i to #ETDmax
  end-if


end-procedure




begin-procedure Process-Reversal
```

```
do Get-Original-Check


evaluate $PayCheck_Option
  when = 'C'
  when = 'M'
    print 'C'           (+0,0)
  when = 'A'
    print 'A'           (+0,0)
  when = '&'
    move 4 to #LINE_NUM

    if #CheckReversalAmount <> 0
      do Format-Number(#CheckReversalAmount, $out, '99,999,999.99mi')
      print $out  (#LINE_NUM,163)
      print 'C'           (+0,0)
      add 1 to #LINE_NUM
    end-if

    if #DepositReversalAmount <> 0
      do Format-Number(#DepositReversalAmount, $out, '99,999,999.99mi')
      print $out (#LINE_NUM,163)
      print 'A'           (+0,0)
      add 1 to #LINE_NUM
    end-if
  end-evaluate


  if #Advice# <> 0

    print 'A'          (6,12)

    do Format-Number(#Advice#, $out, '099999999999999')

    print $out     (6,13)

  end-if


end-procedure




begin-procedure Get-Original-Check


  move 0 to #DepositReversalAmount

  move 0 to #CheckReversalAmount


begin-SELECT

P1.COMPANY

P1.PAYGROUP

P1.PAY_END_DT

P1.PAYCHECK_OPTION

P1.PAYCHECK_STATUS

P1.NET_PAY

P1.OFF_CYCLE

P1.PAGE_NUM

P1.LINE_NUM
```

```
P1.SEPCHK

  move &P1.PayCheck_Option to $PayCheck_Option
  move &P1.PayCheck_Status to $PayCheck_Status

  evaluate &P1.PayCheck_Option
    when = 'C'
    when = 'M'
      subtract &P1.Net_Pay from #CheckReversalAmount

      if $PayCheck_Status <> 'A'
        subtract &P1.Net_Pay from #Check_Amount
      end-if

    when = 'A'
      subtract &P1.Net_Pay from #DepositReversalAmount

      if $PayCheck_Status <> 'A'
        subtract &P1.Net_Pay from #Deposit_Total
      end-if

    when = '&'
      do Get-Original-Distribution
  end-evaluate

FROM  PS_PAY_CHECK P1
```

```
WHERE P1.PAYCHECK_NBR          = &B.PAYCHECK_NBR

  AND P1.FORM_ID         = &B.Form_ID

  AND P1.CHECK_DT        = &B.Check_Dt

  AND P1.PAYCHECK_OPTION <> 'R'

end-SELECT


end-procedure




begin-procedure Get-Original-Distribution


  move 0 to #Advice#


begin-SELECT

PAYCHECK_NBR

ACCOUNT_TYPE

DEPOSIT_AMOUNT


  if &Account_Type = '$'

    subtract &Deposit_Amount from #CheckReversalAmount


    if $PayCheck_Status <> 'A'

        subtract &Deposit_Amount from #Check_Amount

    end-if


  else
```

```
    subtract &Deposit_Amount from #DepositReversalAmount


    if $PayCheck_Status <> 'A'
        subtract &Deposit_Amount from #Deposit_Total
    end-if
  end-if


  move &PAYCHECK_NBR to #Advice#


FROM  PS_PAY_DISTRIBUTN
WHERE COMPANY    = &P1.Company
  AND PAYGROUP   = &P1.PayGroup
  AND PAY_END_DT = &P1.Pay_End_Dt
  AND OFF_CYCLE  = &P1.Off_Cycle
  AND PAGE_NUM      = &P1.PAGE_NUM
  AND LINE_NUM      = &P1.LINE_NUM
  AND SEPCHK     = &P1.SepChk
end-SELECT


end-procedure



begin-procedure Get-USA-Taxes


  move -1 to #i
```

```
begin-SELECT

L.STATE,

L.LOCALITY,

L.TAX_CLASS,

L.TAX_CUR


  add 1 to #i


  let $Tax_Class = RTRIM(&L.Tax_Class, ' ')

  if RTRIM(&L.State, ' ') = '$U'  !if U.S. federal


    evaluate $Tax_Class

      when = 'H'  !FWT

        move 'FWT' to $TaxType

      when = 'C'  !Earned Income Credit

        move 'EIC' to $TaxType

      when = 'D'  !FICA OASDI

        move 'OASDI' to $TaxType

      when = 'G'  !FICA OASDI Tips / EE

        move 'OASDI/Tips' to $TaxType

      when = 'F'  !FICA Medicare

        move 'MEDI' to $TaxType

      when = 'T'  !FICA Medicare Tips / EE

        move 'MEDI/Tips' to $TaxType

      when = 'A'  !Non-resident alien (1042)                              !HP99999

        move '1042' to $TaxType                                          !HP99999
      when = '7'  !Addl Medcr

        move 'ADDL MED' to $TaxType

      when = '5'  !OR Transit

        move 'TRANSIT' to $TaxType

    end-evaluate


  else


    if RTRIM(&L.Locality, ' ') = '' !if a State

     LET $TaxType = RTRIM(&L.State, ' ')


      evaluate $Tax_Class

        when = 'H'  !SWT

          concat 'SWT' with $TaxType

        when = 'A'  !SWT for Non-resident alien (1042)                  !HP99999

          concat '1042' with $TaxType                                   !HP99999

        when = 'C'  !Earned Income Credit

          concat 'EIC' with $TaxType

        when = 'D'  !Employee SDI

          concat 'SDI' with $TaxType

        when = 'L'  !New Jersey SWAF

          concat 'SWAF' with $TaxType

        when = 'M'  !New Jersey WFDP

          concat 'WFDP' with $TaxType

        when = 'N'  !New Jersey HCSF

          concat 'HCSF' with $TaxType
     when = 'V'  !Employee UI
      concat 'UI' with $TaxType
     when = 'W'  !VDI EE
      concat 'VDI' with $TaxType
     when = 'I'  !FLI EE
      concat 'FLI' with $TaxType
     when = '3'  !MLI EE
      concat 'MLI' with $TaxType
     when = 'O'  !VOL FLI EE
      concat 'VFLI' with $TaxType
     when = '1'  !VOL MLI EE
      concat 'VMLI' with $TaxType
     when = '5'  !OR Transit
      concat 'TRANSIT' with $TaxType
    end-evaluate

   else
    move &L.Locality to $TaxType
   end-if
  end-if

  PUT         &L.Tax_Cur -
            $TaxType   -
   INTO ETDARRAY(#I) TAXAMT    -
            TAXTYPE
```

```
FROM  PS_PAY_TAX L
WHERE COMPANY    = $Company
  AND PAYGROUP   = $PayGroup
  AND PAY_END_DT = $PayEndDate
  AND OFF_CYCLE  = &B.Off_Cycle
  AND PAGE_NUM      = &B.PAGE_NUM
  AND LINE_NUM      = &B.LINE_NUM
  AND SEPCHK     = &B.SepChk
  AND TAX_CLASS IN ('H','C','D','F','L','M','N','V','P','B','G','T','W','A','I','O','7','5','1','3')  !HP99999
  AND STATE       NOT IN('$UAS', '$UGU', '$UPR', '$UVI')
  AND TAX_CUR   <> 0
end-SELECT


  if #i > #ETDmax
    move #i to #ETDmax
  end-if


end-procedure




begin-procedure Get-CAN-Taxes


  move -1 to #i


begin-SELECT
M.TAX_CLASS_CAN
```

```
M.TAX_CUR


  if &M.Tax_Cur <> 0
    add 1 to #i
    PUT &M.Tax_Cur &M.TAX_CLASS_CAN INTO ETDARRAY(#I) TAXAMT TAXTYPE
  end-if


FROM  PS_PAY_TAX_CAN M
WHERE M.COMPANY    = $Company
  AND M.PAYGROUP   = $PayGroup
  AND M.PAY_END_DT = $PayEndDate
  AND M.OFF_CYCLE  = &B.Off_Cycle
  AND M.PAGE_NUM      = &B.PAGE_NUM
  AND M.LINE_NUM      = &B.LINE_NUM
  AND M.SEPCHK     = &B.SepChk
  AND M.TAX_CLASS_CAN IN ('CIT','QIT','CPP','QPP','T4A','RV2','PYT','EIE','QIE')
ORDER BY M.TAX_CLASS_CAN
end-SELECT


  if #i > #ETDmax
    move #i to #ETDmax
  end-if


end-procedure
```

```
begin-procedure Get-Deductions

  move -1 to #i

begin-SELECT
F.PLAN_TYPE
F.DEDCD
F.DED_CLASS
F.DED_SLSTX_CLASS

#ifdef DB2ALL
DECIMAL(SUM(F.DED_CUR),15,3)   &F.Ded_Cur
#else
SUM(F.DED_CUR)            &F.Ded_Cur
#endif

G.SPCL_PROCESS

  if &G.SPCL_PROCESS = 'G'
    do Get-Garnishment-Detail
  else
    let #DedAmt = &F.Ded_Cur
    move &F.Plan_Type to $DedCode xx
    concat &F.DedCd with $DedCode -xxxxxx
    add 1 to #i
    PUT #DedAmt $DedCode INTO ETDARRAY(#I) DEDAMT DEDCODE
  end-if


FROM  PS_PAY_DEDUCTION F,
      PS_DEDUCTION_TBL G
WHERE COMPANY    = $Company
  AND PAYGROUP   = $PayGroup
  AND PAY_END_DT = $PayEndDate
  AND OFF_CYCLE  = &B.Off_Cycle
  AND PAGE_NUM      = &B.PAGE_NUM
  AND LINE_NUM      = &B.LINE_NUM
  AND SEPCHK     = &B.SepChk
  AND DED_CLASS [$Ded_Class_Clause]
  AND DED_CUR   <> 0
  AND G.DEDCD    = F.DEDCD
  AND G.PLAN_TYPE = F.PLAN_TYPE
  AND G.EFFDT    =
      (SELECT MAX(EFFDT)
         FROM PS_DEDUCTION_TBL
        WHERE DEDCD    = F.DEDCD
          AND PLAN_TYPE = F.PLAN_TYPE
          AND EFFDT <= $PayEndDate)
GROUP BY F.PLAN_TYPE,
         F.DEDCD,
         F.DED_CLASS,
         F.DED_SLSTX_CLASS,
         G.SPCL_PROCESS
```

```
ORDER BY F.PLAN_TYPE,

         F.DEDCD,

         F.DED_CLASS,

         F.DED_SLSTX_CLASS

end-SELECT


  if #i > #ETDmax

    move #i to #ETDmax

  end-if


end-procedure


!********************************************************************

begin-procedure Get-Garnishment-Detail

!********************************************************************


begin-SELECT

GR.GARNID

GR.DEDCD

GR.DEDUCT_AMT


  if &F.DEDCD = &GR.DEDCD and

     &GR.DEDUCT_AMT <> 0

    move &GR.DEDUCT_AMT to #DedAmt


    let $DedCode = 'GR'
    let $Garn = RTRIM(&GR.GARNID, ' ')
    concat $Garn with $DedCode -xxxxxx
    add 1 to #i
    PUT #DedAmt $DedCode INTO ETDARRAY(#I) DEDAMT DEDCODE
  end-if

FROM PS_PAY_GARNISH GR
WHERE GR.COMPANY = &A.COMPANY
  AND GR.PAYGROUP = &A.PAYGROUP
  AND GR.PAY_END_DT = &A.PAY_END_DT
  AND GR.OFF_CYCLE = &B.OFF_CYCLE
  AND GR.PAGE_NUM = &B.PAGE_NUM
  AND GR.LINE_NUM = &B.LINE_NUM
  AND GR.SEPCHK = &B.SEPCHK
ORDER BY GR.GARNID
end-SELECT
end-procedure

begin-procedure Get-Pay-Distribution

begin-SELECT
PD.ACCOUNT_TYPE
PD.DEPOSIT_AMOUNT
PD.PAYCHECK_NBR

  if &PD.PAYCHECK_NBR = &B.PAYCHECK_NBR
```

```
    if &PD.Account_Type <> '$'
      add &PD.Deposit_Amount to #Deposit_Total
    else
      move &PD.Deposit_Amount to #Check_Amount
    end-if

  else

    if &PD.Account_Type <> '$'
      add &PD.Deposit_Amount to #Deposit_Total
    end-if

    move &PD.PAYCHECK_NBR to #Advice#
  end-if

FROM  PS_PAY_DISTRIBUTN PD
WHERE PD.COMPANY         = &A.Company
  AND PD.PAYGROUP        = &A.PayGroup
  AND PD.PAY_END_DT      = &A.Pay_End_Dt
  AND PD.OFF_CYCLE       = &B.Off_Cycle
  AND PD.PAGE_NUM          = &B.PAGE_NUM
  AND PD.LINE_NUM          = &B.LINE_NUM
  AND PD.SEPCHK          = &B.SepChk
end-SELECT
```

```
end-procedure



begin-procedure Print-Detail


  position (-1)
  move 0 to #i


  while #i <= #ETDmax
    get              #RegEnHrs  -
                    #RegEarn   -
                    #OTHrs     -
                    #OTEarn    -
                    $EarnType   -
                    #EarnHours  -
                    #EarnGross  -
                    $AddToGrs   -
                    #TaxAmt     -
                    $TaxType    -
                    #DedAmt     -
                    $DedCode    -
      from ETDarray(#i) RegEnHrs  -
                     RegEarn   -
                     OTHrs     -
                     OTEarn    -
                     EarnType   -
                     EarnHours  -

                     EarnGross  -

                     AddToGrs   -

                     TaxAmt     -

                     TaxType    -

                     DedAmt     -

                     DedCode


    print $EarnType  (+1,77)

    print $AddtoGrs  (0,103)




    if #Current-Line = 10 and $Overflow_SW = 'Y'

       if length($name) > 14

          let $Name = substr($Name,1,14)||' (cont.)'

       else

          let $Name = $Name||' (cont.)'

       end-if

       print $Name     (0,1)

       move 'N' to $Overflow_SW

    end-if


    do Format-Number(#RegEnHrs, $out, 'b999.99mi')

    print $out         (0,34)

    do Format-Number(#RegEarn,  $out, 'b99,999,999.99mi')

    print $out         (0,42)
```

```
    do Format-Number(#OTHrs, $out, 'b999.99mi')
    print $out         (0,57)
    do Format-Number(#OTEarn, $out, 'b99,999.99mi')
    print $out         (0,65)
    do Format-Number(#EarnHours, $out, 'b999.99mi')
    print $out (0,81)
    do Format-Number(#EarnGross, $out, 'b9,999,999.99mi')
    print $out (0,89)
    do Format-Number(#TaxAmt, $out, 'b9999,999.99mi')
    print $out    (0,117)

    if $TaxType = 'RV2'
      print 'RL-2'     (0,130)
    else
      print $TaxType   (0,130)
    end-if

    do Format-Number(#DedAmt, $out, 'b999,999.99mi')
    print $out    (0,141)
    print $DedCode   (0,153)
    add 1 to #i
  end-while

end-procedure
```

```
begin-procedure Company-Change
  move 2 to #changeLevel
  do Print-Totals
end-procedure


begin-procedure PayEndDate-Change
  move 3 to #changeLevel
  do Print-Totals
end-procedure


begin-procedure PayGroup-Change
  move 4 to #changeLevel
  do Print-Totals
end-procedure


begin-procedure Department-Change
  move 5 to #changeLevel
  do Print-Totals
end-procedure


begin-procedure Print-Totals
```

```
  move 5 to #totalRow

  while #totalRow >= #changeLevel
!   SHOW #current-line
    if #current-line > 51  !don't split a total series across pages
      new-page
      print ' ' (+1,1)
    end-if

    move #totalRow to #rollRow
    subtract 1 from #rollRow
    get $TotalName from Totals(#totalRow) TotalName
    print $TotalName (+1,11)
    position (+1)
    move -1 to #posn
    move 0 to #i

    while #i <= 8
      get #total from Totals(#totalRow) total(#i)
      get #startCol from Totals(0) startCol(#i)
      do Format-Number(#total, $out, '999,999,999.99mi')
      print $out (+#posn,#startCol)
      multiply -1 times #posn
      array-add #total to Totals(#rollRow) total(#i)
      put 0 into Totals(#totalRow) total(#i)
      add 1 to #i
```

```
end-while

position (-1)

while #i <= 12
  get #total from Totals(#totalRow) total(#i)
  get #startCol from Totals(0) startCol(#i)

  if #total <> 0
    do Format-Number(#total, $out, '999,999,999.99mi')
    print $out (+#posn,#startCol)

    evaluate #i
      when = 10
        print 'C' ()
      when = 11
        print 'A' ()
      when = 12
        print 'U' ()
    end-evaluate

    array-add #total to Totals(#rollRow) total(#i)

    put 0 into Totals(#totalRow) total(#i)

  else
```

```
      evaluate #i
        when = 9
        when = 10
          print 0.00 (+#posn,#startCol) edit 999,999,999.99mi
      end-evaluate

    end-if

    add 1 to #i
  end-while

  if #totalRow <> 1
    position (+1)
  end-if

  subtract 1 from #totalRow
end-while

if #changeLevel = 5  !skip page except on Department change
  position (+1)
  next-listing need=7
else

  if $End = 'Y'
    do Reset
    end-if

    if #current-line > 12
        new-page
    end-if

  end-if

end-procedure

!****************************************************
!  RESET: Reset printer, print 'End of Report'        *
!        Note: Uncomment ENCODE statement to force   *
!              a printer reset at end of report      *
!****************************************************

begin-procedure Reset
  move 'Y' to $End_of_Report
  if #MsgFlagW
     print $MsgTextW (+2,1)
  end-if
end-procedure

begin-footing 2
  print $Pay002_Legend (1,1)
  if $End_of_Report = 'Y'
```

```
    !****************************************************************
    !  Use a globalized string if one was gotten from stdhdgtr        *
    !****************************************************************
    if $stdhdg_end_rep = ''
      print 'End of Report' (+1,1) center
    else
      print $stdhdg_end_rep (+1,1) center
    end-if
 !  encode '<27>E' into $Reset
    print $Reset (+0,+1)
  end-if
end-footing



#Include 'getcodta.sqc'  !Get-Company-Data procedure
#Include 'getpgdta.sqc'  !Get-PayGroup-Data procedure
#Include 'payinit.sqc'   !Report Initialization and Timing
#Include 'curdttim.sqc'  !Get-Current-DateTime procedure
#Include 'datetime.sqc'  !Routines for date and time formatting
#Include 'number.sqc'    !Routines to format numbers
#Include 'stdapi.sqc'    !Update Process API
#Include 'payrnctl.sqc'  !Get-Run-Control procedure
#Include 'sqrtrans.sqc'  !Translate SQR strings to a given language
#Include 'getrplng.sqc'  !Gets the report language
#Include 'getmsgct.sqc'  !Gets Message Catalog
```