

```
#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

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, '0999999999999999')

  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-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

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'

        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