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

!  AD New:  Send data to active directory for new employees.            *

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

!                          SQR LOG                              *

!                                                       *

! DESCRIPTION: This program extracts employee and non-employee current  *

!           data to be sent to AD.                                *

!                                                       *

! ASSUMPTION:  Always current information when run.                  *

!           Only active, leave and suspended employees.           *

!           Companies PS, TH and JST.                         *

!                                                       *

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

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

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

!                        CHANGE LOG                          *

!                                                      *

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

!                                                      *

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

! 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           *

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

!                                                      *
! DESCRIPTION:                                    *

!                                                 *

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

! SIR  Programmer            Start Dt  End Date  Orig. Pgm

! ---- ---------------------- --------  --------  ---------



#include 'setenv.sqc'    !Set PeopleSoft Environment Constants

#include 'setup32.sqc'   !Report is landscape



begin-report



   do Init-DateTime

   do Init-Number
   do Get-Current-DateTime

 show 'Began:  ' $asofnow


   do Stdapi-Init

   do Init-Report

   do Process-Main

   do Wrap-Up

   do Stdapi-Term


   do Get-Current-DateTime

 show 'Ended:  ' $asofnow


end-report
```

```
!************************************************************************
! Initialise report.                                                    *
!************************************************************************
begin-procedure Init-Report


   move 'YAD_NEW'     to $ReportID
   move 'AD Extract' to $ReportTitle


   display ' '
   display $ReportID
   display $ReportTitle
   display ' '
```

```
#ifdef UNIX

  let $Filepath = '/u06/home/intrfcs/enet/'

#else

  let $Filepath = 'c:\winnt\temp\'

#endif



  let $Filepathftp = $Filepath

  let $Filename = 'AD_LISTING_NEW.TXT'

  let $Filepath = $Filepath || 'AD_LISTING_NEW.TXT'

  open $Filepath as 1 for-writing record=1000:vary



  do Check-Prov-Implemented ($Prov_Implemented)
```

```
end-procedure

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

! Check to see if JOB contains the field Y_PROVISIONAL. If so, then new *

! provisionals process has been implemented so no more EMPL_RCD = 1.   *

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

begin-procedure Check-Prov-Implemented (:$Implemented_Ind)

   let $Implemented_Ind = 'N'

begin-select

FIELDNAME
```

```
   let $Implemented_Ind = 'Y'

FROM  PSRECFIELD

WHERE RECNAME = 'JOB'

  AND FIELDNAME = 'Y_PROVISIONAL'

end-select

end-procedure

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

! Main part of program. Select all non-terminated employees, retrieve   *

! additional information and write to fixed length file.                *
```

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

begin-procedure Process-Main

DO WRITE-HEADER

begin-select

A.EMPLID

REPLACE(A.NAME,',',', ') &A.NAME

A.FIRST_NAME

A.LAST_NAME

A.Y_ORG_UNIT_DS

A.JOBCODE_DESCR

A.Y_DEPARTMENT_DS
```

```
A.Y_ORG_UNIT

A.PER_ORG

A.EMPL_CLASS

A.Y_IS_CURR_PROV_MOB

A.LOCATION

A.Y_MAIL_LOCATION

A.Y_ZIP_FLOOR

A.COMPANY

A.Y_DEPARTMENT

TO_CHAR(A.TERMINATION_DT,'YYYY/MM/DD')  &termination_dt

A.EMPL_STATUS

A.SAL_ADMIN_PLAN

A.GRADE
```

```
A.SUPERVISOR_ID  &A.SUPERVISOR_ID
A.UNION_CD  &UNION_CD
!A.ADDRESS1
!A.ADDRESS2
!A.CITY
!A.STATE
!A.POSTAL
!A.COUNTRY


   let $emplid   = &A.EMPLID
   let $A.EMPLID = &A.EMPLID
   LET $A.FIRST_NAME = &A.FIRST_NAME
   LET $A.LAST_NAME = &A.LAST_NAME
   LET $SUPERVISOR_ID = &A.SUPERVISOR_ID
   LET $EMPL_CLASS = &A.EMPL_CLASS


   LET $MAIL_LOCATION = &A.LOCATION
   LET $UNION_CD = &UNION_CD
   DO GET-REPRESENTED
   IF $REPRESENTED = 'N'
    LET $LICENSE = 'MS-G3'
   ELSE
    LET $LICENSE = ' '
   END-IF
   DO GET-ADDRESS-FOR-LOCATION
```

```
!   LET $ADDRESS1 = &A.ADDRESS1

!   LET $ADDRESS2 = &A.ADDRESS2

!   LET $CITY = &A.CITY

!   LET $STATE = &A.STATE

!   LET $POSTAL = &A.POSTAL

!   LET $COUNTRY = &A.COUNTRY



   let $A.NAME   = &A.NAME

   let $A.PER_ORG = &a.per_org



   ! Get mailing label from Y_ZIP_FLOOR_DTL.

     do  Get-Mail-Label  (&A.EMPLID,  &A.LOCATION,  &A.Y_MAIL_LOCATION,  &A.Y_ZIP_FLOOR,
$AsofToday, $Mail_Label)
```

```
   ! If provisional employee and new provisional process NOT implemented,

   ! get phone details from provisional employment record (since there are two employment records)

   if &A.Y_IS_CURR_PROV_MOB = 'Y' and $Prov_Implemented = 'N'

          do Get-Phone-Details (&A.EMPLID, 1, $Work_Phone1, $Work_Phone2, $Cell_Phone1, $Cell_Phone2, $Pager1, $Pager2, $Fax, $Telex)

   else

          do Get-Phone-Details (&A.EMPLID, 0, $Work_Phone1, $Work_Phone2, $Cell_Phone1, $Cell_Phone2, $Pager1, $Pager2, $Fax, $Telex)

   end-if



   ! If the nickname indicator is set to Y, retrieve preferred first name

   do Get-Preferred-Name (&A.EMPLID, 'PRF', 'F', $Nickname)

   if rtrim($Nickname, ' ') <> ''
   let $Nickname = $A.LAST_NAME || ', ' || $Nickname

else

   let $Nickname = $A.NAME

end-if



! Ensure phone numbers contain only digits.

do RemoveChars($Work_Phone1, $O_Work_Phone1)

do RemoveChars($Work_Phone2, $O_Work_Phone2)

do RemoveChars($Cell_Phone1, $O_Cell_Phone1)

do RemoveChars($Cell_Phone2, $O_Cell_Phone2)

do RemoveChars($Pager1, $O_Pager1)

do RemoveChars($Pager2, $O_Pager2)

do RemoveChars($Fax, $O_Fax)
```

```
  do RemoveChars($Telex, $O_Telex)

let $company      = &A.COMPANY

let $deptid       = &A.Y_DEPARTMENT

let $check_jd     = substr($deptid,1,1)

if $check_jd      = 'J'

 let $deptid = substr($deptid,3,4)

end-if

let $termdt       = &termination_dt

let $status       = &A.EMPL_STATUS

let $salplan      = &A.SAL_ADMIN_PLAN

let $grade        = &A.GRADE

       let $jobcodedescr = rtrim(&A.JOBCODE_DESCR, ' ')
```

```
    if substr($emplid,1,1) = 'C'
     let $jobcodedescr = 'Contract Staff'
     let $salplan      = '  '
     let $grade        = '  '
    end-if
    if substr($emplid,1,2) = 'LC'
     let $jobcodedescr = 'Supplemental Staff'
     let $salplan      = '  '
     let $grade        = '  '
    end-if


!** Begin Mod:SR-1056 **
   !if  &A.PER_ORG  =  'N'  AND  (&A.EMPL_CLASS  =  'U'  OR  &A.EMPL_CLASS  =  'V'  OR
&A.EMPL_CLASS = 'W')

   !if &A.COMPANY = 'JST'

   !** End Mod:SR-1056 **

   !   let $Org_Unit = substr(&A.Y_ORG_UNIT, 2, length(&A.Y_ORG_UNIT) - 1)

   !   do Get-Org-Unit-Descr ('PA', $Org_Unit, $Org_Unit_Desc, $Department_Desc)

   !else

     let $Org_Unit        = &A.Y_ORG_UNIT

     let $Org_Unit_Desc   = &A.Y_ORG_UNIT_DS

     let $Department_Desc = &A.Y_DEPARTMENT_DS

   !end-if

     do get-tie-line (&A.LOCATION, $TIE_LINE)

     DO CHECK-VIP

     do get-oprid
```

```
 if $OPRID <> ''
      do Write-Data
    end-if



  ! Write out to file if all of the phone numbers are not blank
!   if (rtrim($Work_Phone1, ' ') = '' and
!       rtrim($Work_Phone2, ' ') = '' and
!       rtrim($Cell_Phone1, ' ') = '' and
!       rtrim($Cell_Phone2, ' ') = '' and
!       rtrim($Pager1, ' ')      = '' and
!       rtrim($Pager2, ' ')      = '' and
!       rtrim($Fax, ' ')         = '' and
!       rtrim($Telex, ' ')       = '')
!      !Ignore if phone numbers are blank

!   else

!    if $OPRID <> ''

!      do Write-Data

!    end-if

!   end-if



   add 1 to #rows_processed

!   if mod(#rows_processed, 500) = 0

!      display #rows_processed 99999 noline

!      display ' Rows Processed'

!   end-if
```

```
!FROM PS_YQ_QUERY_DATA A

FROM PS_YQ_QUERY_FUT A

WHERE

A.JOBCODE NOT IN ('2600') AND A.EMPLID NOT IN (SELECT SENT.EMPLID FROM
PS_Y_ID_SENT_TO_AD SENT) AND

(A.COMPANY in ('PA','TH','PTA','RT','ERT','LTD','SBC') and (A.TERMINATION_DT >=
'01-JAN-1995') or (A.EMPL_STATUS IN ('A','L','P','S') and A.COMPANY in
('PA','TH','PTA','RT','ERT','LTD','SBC')))

ORDER BY A.EMPLID

end-select




begin-select

ac.EMPLID    &AC_EMPLID

REPLACE(ac.First_NAME,'''','')  &AC_FIRST_NAME

REPLACE(ac.last_name,'''','')  &AC_LAST_NAME
REPLACE(ac.middle_init,'"','')  &AC_MIDDLE_INIT

ac.deptid    &AC_DEPTID

ac.JOBCODE    &AC_JOBCODE

ac.LOCATION    &AC_LOCATION

ac.Y_MAIL_LOCATION   &AC_Y_MAIL_LOCATION

ac.Y_ZIP_FLOOR    &AC_Y_ZIP_FLOOR

ac.EMPL_STATUS    &AC_EMPL_STATUS

ac.VENDOR_ID    &AC_VENDOR_ID



 Let $Nickname       = ' '

 Let $O_Work_Phone1  = ' '

 Let $O_Work_Phone2  = ' '

 Let $O_Cell_Phone1  = ' '
 Let $O_Cell_Phone2  = ' '

 Let $O_Pager1       = ' '

 Let $O_Pager2       = ' '

 Let $O_Fax          = ' '

 Let $O_Telex        = ' '

 Let $Mail_Label     = ' '

 Let $Org_Unit_Desc  = ' '

 Let $jobcodedescr   = ' '

 Let $Department_Desc = ' '

 Let $Org_Unit       = ' '

 Let $A.PER_ORG       = ' '

 Let $deptid          = ' '

 Let $company         = ' '
 Let $termdt         = ' '

 Let $status         = ' '

 Let $salplan        = ' '

 Let $grade          = ' '

 Let $TIE_LINE       = ' '

 LET $OUT_VIP        = 'N'

 LET $SUPERVISOR_ID  = ' '



   let $emplid = &AC_EMPLID

   let $A.EMPLID = &AC_EMPLID

   LET $A.FIRST_NAME = &AC_FIRST_NAME

   LET $A.LAST_NAME = &AC_LAST_NAME

   let $A.NAME   = &AC_LAST_NAME || ', ' || &AC_FIRST_NAME
   let $A.PER_ORG = ' '

   LET $ADDRESS1 = ' '

   LET $ADDRESS2 = ' '

   LET $CITY = ' '

   LET $STATE = ' '

   LET $POSTAL = ' '

   LET $COUNTRY = ' '

   ! Get mailing label from Y_ZIP_FLOOR_DTL.

           do Get-Mail-Label (&AC_EMPLID, &AC_LOCATION, &AC_Y_MAIL_LOCATION,
 &AC_Y_ZIP_FLOOR, $AsofToday, $Mail_Label)
```

```
      do Get-Phone-Details-Consultant (&AC_EMPLID, 0, $Work_Phone1, $Work_Phone2, $Cell_Phone1, $Cell_Phone2, $Pager1, $Pager2, $Fax, $Telex)

  let $Nickname = $A.NAME

  ! Ensure phone numbers contain only digits.
  do RemoveChars($Work_Phone1, $O_Work_Phone1)
  do RemoveChars($Work_Phone2, $O_Work_Phone2)
  do RemoveChars($Cell_Phone1, $O_Cell_Phone1)
  do RemoveChars($Cell_Phone2, $O_Cell_Phone2)
  do RemoveChars($Pager1, $O_Pager1)
  do RemoveChars($Pager2, $O_Pager2)
  do RemoveChars($Fax, $O_Fax)
```

```
do RemoveChars($Telex, $O_Telex)

 let $company      = 'PA'

 LET $AC_VENDOR_ID = &AC_VENDOR_ID

 DO GET-CONSULTANT-COMPANY

 let $deptid       = &AC_DEPTID

 let $check_jd     = substr($deptid,1,1)

 if $check_jd      = 'J'

    let $deptid = substr($deptid,3,4)

 end-if

 let $termdt       = ' '

 let $status       = &AC_EMPL_STATUS
```

```
if substr($emplid,1,1) = 'C'

  let $jobcodedescr = 'Contract Staff'

  let $salplan      = '  '

  let $grade        = '  '

end-if



if substr($emplid,1,2) = 'LC'

  let $jobcodedescr = 'Supplemental Staff'

  let $salplan      = '  '

  let $grade        = '  '

end-if



let $Org_Unit        = &AC_DEPTID
```

```
    do Get-Org-Unit-Descr ('PA', $Org_Unit, $Org_Unit_Desc, $Department_Desc)



    do get-tie-line (&AC_LOCATION, $TIE_LINE)

    do get-oprid

 if $OPRID <> ''

       do Write-Data

    end-if



   add 1 to #rows_processed



   if mod(#rows_processed, 500) = 0

      display #rows_processed 99999 noline

      display ' Rows Processed'
   end-if

FROM PS_Y_consultant AC
WHERE AC.EMPL_STATUS IN ('A','L','P','S')
  and AC.EFFDT = (SELECT MAX(AC1.EFFDT) FROM PS_Y_CONSULTANT AC1
                  WHERE AC1.EMPLID = AC.EMPLID)
  and ac.emplid not in (select sentac.emplid from ps_Y_ID_SENT_TO_AD sentac)
ORDER BY AC.EMPLID
end-select

BEGIN-SELECT
A.EMPLID     &CWR_EMPLID
a.JOBCODE     &CWR_JOBCODE
```

```
a.DEPTID     &CWR_ORG_UNIT

a.LOCATION     &CWR_LOCATION

a.Y_ZIP_FLOOR     &CWR_Y_ZIP_FLOOR

A.DEPTID     &CWR_DEPTID

a.COMPANY     &CWR_COMPANY

a.empl_status     &CWR_EMPL_STATUS

A.SAL_ADMIN_PLAN    &CWR_SALARY_PLAN

A.GRADE      &CWR_GRADE

A.PER_ORG     &CWR_PER_ORG
```

```
 let $emplid = &CWR_EMPLID

 let $A.EMPLID = &CWR_EMPLID
```

```
DO GET-CWR-NAME
DO GET-WORK-PHONE
let $CWR_LOCATION = &CWR_LOCATION
LET $CWR_MAIL_LOCATION = &CWR_LOCATION
LET $CWR_Y_ZIP_FLOOR = &CWR_Y_ZIP_FLOOR
DO GET-CWR-LOCATION
LET $deptid = &CWR_DEPTID
LET $company = &CWR_COMPANY
DO GET-CWR-ORG
LET $termdt = ' '
let $status = &CWR_EMPL_STATUS
LET $salplan = &CWR_SALARY_PLAN
LET $grade = &CWR_GRADE
```

```
 LET $A.PER_ORG = &CWR_PER_ORG

 LET $CWR_JOBCODE = &CWR_JOBCODE

 LET $jobcodedescr = 'Contingent Worker'

 LET $SUPERVISOR_ID = ' '

 LET $OUT_VIP = 'N'



 do get-oprid

 if $OPRID <> ''

        do Write-Data

    end-if



from ps_job a where a.per_org = 'CWR'

and a.empl_status in ('A','L','P','S')
and a.emplid not in (select sentcwr.emplid from ps_Y_ID_SENT_TO_AD sentcwr)
and a.effdt = (select max(b.effdt) from ps_job b
WHERE B.EFFDT <= SYSDATE
and a.emplid = b.emplid
and a.empl_rcd = b.empl_rcd)
AND A.EFFSEQ = (SELECT MAX(JOBES.EFFSEQ) FROM PS_JOB JOBES
WHERE A.EMPLID = JOBES.EMPLID
AND A.EMPL_RCD = JOBES.EMPL_RCD
AND A.EFFDT = JOBES.EFFDT)

END-SELECT

end-procedure
```

```
BEGIN-PROCEDURE GET-REPRESENTED


 LET $REPRESENTED = ' '


BEGIN-SELECT

REPRESENTED.CERTIFIED &CERTIFIED


 LET $REPRESENTED = &CERTIFIED


  FROM PS_UNION_TBL REPRESENTED

  WHERE REPRESENTED.EFFDT =

        (SELECT MAX(REPRESENTED_ED.EFFDT) FROM PS_UNION_TBL REPRESENTED_ED
        WHERE REPRESENTED.UNION_CD = REPRESENTED_ED.UNION_CD

          AND REPRESENTED_ED.EFFDT <= SYSDATE)

     AND REPRESENTED.UNION_CD = $UNION_CD

END-SELECT


END-PROCEDURE


BEGIN-PROCEDURE GET-CONSULTANT-COMPANY


BEGIN-SELECT

REPLACE(SUBSTR(NAME,1,30),';',',') &VENDOR_NAME


 LET $company = &VENDOR_NAME
```

```
FROM PS_Y_VENDOR_VW

WHERE VENDOR_ID = $AC_VENDOR_ID

END-SELECT



END-PROCEDURE



BEGIN-PROCEDURE CHECK-VIP



! DISPLAY 'CHECKING VIP'

! DISPLAY 'COMPANY =  ' NOLINE

! DISPLAY $COMPANY
! DISPLAY 'CLASS = ' NOLINE

! DISPLAY $EMPL_CLASS

! DISPLAY 'GRADE =  ' NOLINE

! DISPLAY $GRADE

! DISPLAY 'SALPLAN = ' NOLINE

! DISPLAY $salplan



 LET $OUT_VIP = 'N'



! IF $COMPANY = 'PA'

!  IF $EMPL_CLASS = 'A' AND $GRADE = '0'

!    LET $OUT_VIP = 'Y'
!  END-IF

!  IF $EMPL_CLASS = 'A' AND SUBSTR($GRADE,1,2) = 'EM'

!    LET $OUT_VIP = 'Y'

!  END-IF

!  IF $EMPL_CLASS = 'A' AND $salplan = 'NRZ' AND $grade = '06'

!    LET $OUT_VIP = 'Y'

!  END-IF

!  IF $EMPL_CLASS = 'A' AND $salplan = 'NRZ' AND $grade = '07'

!    LET $OUT_VIP = 'Y'

!  END-IF

!  IF $EMPL_CLASS = 'A' AND $salplan = 'NRY' AND $grade = '06'

!    LET $OUT_VIP = 'Y'

!  END-IF
!  IF $EMPL_CLASS = 'A' AND $salplan = 'NRY' AND $grade = '07'

!   LET $OUT_VIP = 'Y'

!  END-IF

! END-IF



 IF $COMPANY = 'PA'

  IF $GRADE = '0'

   LET $OUT_VIP = 'Y'

  END-IF

  IF SUBSTR($GRADE,1,2) = 'EM'

   LET $OUT_VIP = 'Y'

  END-IF

  IF $salplan = 'NRZ' AND $grade = '06'
   LET $OUT_VIP = 'Y'

  END-IF

  IF $salplan = 'NRZ' AND $grade = '07'

   LET $OUT_VIP = 'Y'

  END-IF

  IF $salplan = 'NRY' AND $grade = '06'

   LET $OUT_VIP = 'Y'

  END-IF

  IF $salplan = 'NRY' AND $grade = '07'

   LET $OUT_VIP = 'Y'

  END-IF

 END-IF
```

```
END-PROCEDURE


begin-procedure GET-CWR-ORG


   let $Org_Unit_Desc   = ''
   let $Department_Desc = ''
   LET $Org_Unit = ''


begin-select
CWROU.Y_ORG_UNIT_DS &CWR_ORG_UNIT_DS
CWROU.Y_DEPARTMENT_DS &CWR_DEPT_DS


   let $Org_Unit_Desc   = &CWR_ORG_UNIT_DS
   let $Department_Desc = &CWR_DEPT_DS

   LET $Org_Unit = &CWR_ORG_UNIT



FROM PS_Y_ORG_ROLLUP CWROU

WHERE CWROU.COMPANY    = $company

  AND CWROU.Y_ORG_UNIT = $deptid

end-select



end-procedure GET-CWR-ORG



BEGIN-PROCEDURE GET-CWR-LOCATION



BEGIN-SELECT
```

```
CWRLOC.ADDRESS1  &CWR_ADDRESS1

CWRLOC.ADDRESS2  &CWR_ADDRESS2

CWRLOC.CITY  &CWR_CITY

CWRLOC.POSTAL  &CWR_POSTAL

CWRLOC.COUNTRY  &CWR_COUNTRY

CWRLOC.EXTENSION &CWR_EXTENSION
```

```
 LET $ADDRESS1 = &CWR_ADDRESS1

 LET $ADDRESS2 = &CWR_ADDRESS2

 LET $CITY = &CWR_CITY

 LET $POSTAL = &CWR_POSTAL

 LET $COUNTRY = &CWR_COUNTRY
 LET $TIE_LINE = &CWR_EXTENSION


FROM PS_LOCATION_TBL CWRLOC

WHERE CWRLOC.SETID = 'PASET'

AND CWRLOC.LOCATION = $CWR_LOCATION

AND CWRLOC.EFFDT = (SELECT MAX(CWRLOCED.EFFDT)

FROM PS_LOCATION_TBL CWRLOCED

WHERE CWRLOC.SETID = CWRLOCED.SETID

AND CWRLOC.LOCATION = CWRLOCED.LOCATION)


END-SELECT


BEGIN-SELECT
```

```
CWRML.Y_MAIL_LABEL &CWRML.Y_MAIL_LABEL

 LET $Mail_Label = &CWRML.Y_MAIL_LABEL

FROM  PS_Y_ZIP_FLOOR_DTL CWRML
WHERE CWRML.Y_MAIL_LOCATION = $CWR_MAIL_LOCATION
  AND CWRML.Y_ZIP_FLOOR     = $CWR_Y_ZIP_FLOOR
  AND CWRML.EFFDT           = (SELECT MAX(CWRML1.EFFDT)
                            FROM PS_Y_ZIP_FLOOR_DTL CWRML1
                           WHERE CWRML.LOCATION        = CWRML1.LOCATION
                             AND CWRML.Y_MAIL_LOCATION = CWRML1.Y_MAIL_LOCATION
                             AND CWRML.Y_ZIP_FLOOR     = CWRML1.Y_ZIP_FLOOR
```

```
                              AND CWRML1.EFFDT          <= $Effdt)

END-SELECT

END-PROCEDURE GET-CWR-LOCATION

BEGIN-PROCEDURE GET-WORK-PHONE

BEGIN-SELECT

replace(translate(upper(PHONE),'EXTOR/.- ','_'),'_','') &OUT_WORK_PHONE1

 LET $OUT_PHONE = &OUT_WORK_PHONE1
 LET $O_Work_Phone1 = $OUT_PHONE

FROM PS_PERSONAL_PHONE

WHERE PHONE_TYPE = 'BUSN'

AND EMPLID = $emplid

END-SELECT

BEGIN-SELECT

replace(translate(upper(PHONE),'EXTOR/.- ','_'),'_','') &OUT_WORK_PHONE2

 LET $OUT_PHONE = &OUT_WORK_PHONE2
 LET $O_Work_Phone2 = $OUT_PHONE

FROM PS_PERSONAL_PHONE

WHERE PHONE_TYPE = 'BUS2'

AND EMPLID = $emplid

END-SELECT

BEGIN-SELECT

REPLACE(TRANSLATE(PHONE,'/.-','_'),'_','') &OUT_CELL_PHONE1

 LET $OUT_CELL_PHONE1 = &OUT_CELL_PHONE1

 LET $O_Cell_Phone1 = &OUT_CELL_PHONE1
```

```
FROM PS_PERSONAL_PHONE

WHERE PHONE_TYPE IN('BCEL')

AND EMPLID = $emplid

END-SELECT



BEGIN-SELECT

REPLACE(TRANSLATE(PHONE,'/.-','_'),'_',") &OUT_CELL_PHONE2



 LET $OUT_CELL_PHONE2 = &OUT_CELL_PHONE2

 LET $O_Cell_Phone2 = &OUT_CELL_PHONE2



FROM PS_PERSONAL_PHONE
```

```
WHERE PHONE_TYPE IN('PCEL','CELL')

AND EMPLID = $emplid

END-SELECT



BEGIN-SELECT

REPLACE(TRANSLATE(PHONE,'/.-','_'),'_','') &OUT_FAX



 LET $OUT_FAX = &OUT_FAX

 LET $O_Fax = $OUT_FAX



FROM PS_PERSONAL_PHONE

WHERE PHONE_TYPE IN('BFAX')

AND EMPLID = $emplid
```

```
END-SELECT

END-PROCEDURE GET-WORK-PHONE

BEGIN-PROCEDURE GET-CWR-NAME

BEGIN-SELECT

CWRN.FIRST_NAME    &CWR_FIRST_NAME

SUBSTR(CWRN.MIDDLE_NAME,1,1)  &CWR_MIDDLE_INITIAL

CWRN.LAST_NAME    &CWR_LAST_NAME

   LET $A.FIRST_NAME = &CWR_FIRST_NAME

   LET $A.LAST_NAME = &CWR_LAST_NAME

   let $A.NAME   = &CWR_LAST_NAME || ', ' || &CWR_FIRST_NAME
```

```
  let $Nickname = $A.NAME


FROM PS_NAMES CWRN
WHERE CWRN.EMPLID = $emplid
AND CWRN.NAME_TYPE = 'PRI'
AND CWRN.EFFDT = (SELECT MAX(CWRN1.EFFDT) FROM PS_NAMES CWRN1
WHERE CWRN.EMPLID = CWRN1.EMPLID
AND CWRN.NAME_TYPE = CWRN1.NAME_TYPE)
END-SELECT
END-PROCEDURE


BEGIN-PROCEDURE GET-ADDRESS-FOR-LOCATION
```

```
 LET $ADDRESS1 = ' '
 LET $ADDRESS2 = ' '
 LET $CITY = ' '
 LET $STATE = ' '
 LET $POSTAL = ' '
 LET $COUNTRY = ' '

BEGIN-SELECT
WORKLOCATION.ADDRESS1 &WL_ADDRESS1
WORKLOCATION.ADDRESS2 &WL_ADDRESS2
WORKLOCATION.CITY &WL_CITY
WORKLOCATION.STATE &WL_STATE
WORKLOCATION.POSTAL &WL_POSTAL
WORKLOCATION.COUNTRY &WL_COUNTRY


 LET $ADDRESS1 = &WL_ADDRESS1
 LET $ADDRESS2 = &WL_ADDRESS2
 LET $CITY = &WL_CITY
 LET $STATE = &WL_STATE
 LET $POSTAL = &WL_POSTAL
 LET $COUNTRY = &WL_COUNTRY


FROM PS_LOCATION_TBL WORKLOCATION
WHERE WORKLOCATION.SETID = 'PASET'
AND WORKLOCATION.LOCATION = $MAIL_LOCATION
END-SELECT
```

```
END-PROCEDURE

!*************************************************************************************
! Return mailing label from Y_ZIP_FLOOR_DTL for a user specified        *
! employee id, location, mail location, zip floor and effective date.   *
!*************************************************************************************
begin-procedure Get-Mail-Label ($Emplid, $Location, $Mail_Location, $Zip_Floor, $Effdt,
:$Mail_Label)

   let $Mail_Label = ''

begin-select
```

```
Z.Y_MAIL_LABEL

  let $Mail_Label = &Z.Y_MAIL_LABEL

FROM  PS_Y_ZIP_FLOOR_DTL Z
WHERE !Z.LOCATION        = $Location
 Z.Y_MAIL_LOCATION = $Mail_Location
 AND Z.Y_ZIP_FLOOR     = $Zip_Floor
 AND Z.EFFDT           = (SELECT MAX(EFFDT)
                    FROM PS_Y_ZIP_FLOOR_DTL
                   WHERE LOCATION        = Z.LOCATION
                    AND Y_MAIL_LOCATION = Z.Y_MAIL_LOCATION
                    AND Y_ZIP_FLOOR     = Z.Y_ZIP_FLOOR
                    AND EFFDT          <= $Effdt)

end-select



end-procedure



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

! Return preferred first name if it exists for an employee id.          *

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

begin-procedure Get-Preferred-Name ($Emplid, $Name_Type, $Name_Part, :$Nickname)


  let $Nickname = ''


begin-select
```

```
A.FIRST_NAME

   let $Nickname = &A.FIRST_NAME

FROM PS_NAMES A

WHERE A.EMPLID = $Emplid

  AND A.NAME_TYPE = $Name_Type

  AND A.EFFDT = (

   SELECT MAX(B.EFFDT)

   FROM PS_NAMES B

   WHERE A.EMPLID = B.EMPLID

   AND A.NAME_TYPE = B.NAME_TYPE

   AND B.EFFDT <= SYSDATE)
```

```
end-select

end-procedure



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

! Return the following set of phone numbers from EMPLOYMENT_PHON for a  *

! user specified employee id and employment record number:             *

!     Work phone 1 & 2                                                 *

!     Cell phone 1 & 2                                                 *

!     Page 1 & 2                                                       *

!     Fax                                                              *

!     Telex                                                            *

! Restrict to where the telephone switch is one of 1,2,4 or 5.         *

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

```
begin-procedure Get-Phone-Details ($Emplid, #Empl_Rcd,

                        :$Work_Phone1, :$Work_Phone2, :$Cell_Phone1, :$Cell_Phone2, :$Pager1,
:$Pager2, :$Fax, :$Telex)

  let $Work_Phone1  = ''
  let $Work_Phone2  = ''
  let $Cell_Phone1  = ''
  let $Cell_Phone2  = ''
  let $Pager1       = ''
  let $Pager2       = ''
  let $Fax          = ''
  let $Telex        = ''
```

```
begin-select

PHONE_TYPE

PHONE



  evaluate &PHONE_TYPE

  when = 'BUSN'

    let $Work_Phone1 = &PHONE

    break

  when = 'BUS2'

    let $Work_Phone2 = &PHONE

    break

  when = 'BCEL'

    let $Cell_Phone1 = &PHONE
    break
  when = 'OTR'
    let $Cell_Phone2 = &PHONE
    break
  when = 'BPG1'
    let $Pager1 = &PHONE
    break
  when = 'BPG2'
    let $Pager2 = &PHONE
    break
  when = 'BFAX'
    let $Fax = &PHONE
    break
  when = 'BTLX'
    let $Telex = &PHONE
    break
  when-other
  end-evaluate

FROM PS_PERSONAL_PHONE
WHERE EMPLID = $Emplid
ORDER BY PHONE_TYPE
end-select

end-procedure
```

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

! Return the following set of phone numbers from EMPLOYMENT_PHON for a  *

! user specified employee id and employment record number:              *

!     Work phone 1 & 2                                    *

!     Cell phone 1 & 2                                    *

!     Page 1 & 2                                          *

!     Fax                                                 *

!     Telex                                               *

! Restrict to where the telephone switch is one of 1,2,4 or 5.          *

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

begin-procedure Get-Phone-Details-Consultant ($Emplid, #Empl_Rcd,

                                  :$Work_Phone1, :$Work_Phone2, :$Cell_Phone1, :$Cell_Phone2, :$Pager1,
:$Pager2, :$Fax, :$Telex)
```

```
  let $Work_Phone1  = ''

  let $Work_Phone2  = ''

  let $Cell_Phone1  = ''

  let $Cell_Phone2  = ''

  let $Pager1       = ''

  let $Pager2       = ''

  let $Fax          = ''

  let $Telex        = ''




begin-select

PHONE_TYPE
```

PHONE

```
 evaluate &PHONE_TYPE

 when = 'BUSN'

   let $Work_Phone1 = &PHONE

   break

 when = 'BUS2'

   let $Work_Phone2 = &PHONE

   break

 when = 'BCEL'

   let $Cell_Phone1 = &PHONE

   break

 when = 'CELL'
```

```
  let $Cell_Phone1 = &PHONE

  break

when = 'OTR'

  let $Cell_Phone2 = &PHONE

  break

when = 'BPG1'

  let $Pager1 = &PHONE

  break

when = 'BPG2'

  let $Pager2 = &PHONE

  break

when = 'BFAX'

  let $Fax = &PHONE
```

```
    break
  when = 'FAX'
    let $Fax = &PHONE
    break
  when = 'BTLX'
    let $Telex = &PHONE
    break
  when-other
  end-evaluate

FROM PS_Y_cnsltnt_PHONE
WHERE EMPLID = $Emplid
ORDER BY PHONE_TYPE
```

```
end-select



end-procedure





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

! Remove all non-numeric characters from phone numbers.                *

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

begin-procedure RemoveChars ($Phone_No, :$Output_Phone_No)



   let $Output_Phone_No = ''

   let #phone_length    = length($Phone_No)

   let #i               = 1
```

```
while #i <= #phone_length

   let $textchar = substr($Phone_No, #i, 1)



   evaluate $textchar

   when = '1'

   when = '2'

   when = '3'

   when = '4'

   when = '5'

   when = '6'

   when = '7'

   when = '8'
```

```
      when = '9'
      when = '0'
         concat $textchar with $Output_Phone_No
         break
      when-other
      end-evaluate

      add 1 to #i
   end-while

! Now put in dashes if the phone number is not blank
```

```
 IF $Output_Phone_No <> ''

  LET $Output_Phone_No = SUBSTR($Output_Phone_No,1,3)||'-'||SUBSTR($Output_Phone_No,4,3)||'-'||SUBSTR($Output_Phone_No,7,4)

 END-IF



end-procedure



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

! Get org unit description and rollup dept description .               *

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

begin-procedure Get-Org-Unit-Descr ($Company, $Org_Unit, :$Org_Unit_Desc, :$Department_Desc)



   let $Org_Unit_Desc   = ''
   let $Department_Desc = ''

begin-select
Y_ORG_UNIT_DS
Y_DEPARTMENT_DS

   let $Org_Unit_Desc   = &Y_ORG_UNIT_DS
   let $Department_Desc = &Y_DEPARTMENT_DS

FROM PS_Y_ORG_ROLLUP
WHERE COMPANY    = $Company
  AND Y_ORG_UNIT = $Org_Unit
end-select
```

```
end-procedure

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

! Get tie line based on work location.                              *

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

begin-procedure get-tie-line ($LOCATION_CODE, :$TIE_LINE)

let $TIE_LINE = ''

begin-select

L.EXTENSION
```

```
 LET $TIE_LINE = &L.EXTENSION

FROM PS_LOCATION_TBL L

WHERE L.SETID = 'PASET'

AND L.LOCATION = $LOCATION_CODE

AND L.EFFDT = (SELECT MAX(L2.EFFDT) FROM PS_LOCATION_TBL L2

WHERE L.SETID = L2.SETID

AND L.LOCATION = L2.LOCATION)

end-select

end-procedure
```

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

! Get operid          *

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

begin-procedure get-oprid



let $OPRID = ''



begin-select

M.OPRID



 LET $OPRID = &M.OPRID

! if $OPRID <> ''

!         do Write-Data
!     end-if

FROM PSOPRDEFN M

WHERE M.EMPLID = $emplid

end-select

end-procedure

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

! Write data to fixed length file                              *
```

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

begin-procedure Write-Data



 let $BLANK_SPACE = ''

 do generate-password



BEGIN-SELECT

TO_CHAR(SYSDATE,'MM/DD/YYYY') &CREATION_DATE



 LET $CREATION_DATE = &CREATION_DATE

 LET $ACCOUNT_EXPIRES_DATE = SUBSTR($CREATION_DATE,7,4)||'-12-31'

 IF substr($A.EMPLID,1,1) = 'C'

  LET $ACCOUNT_EXPIRES = ' '
 END-IF

FROM DUAL

END-SELECT

 display 'getting expire string '

 display 'ACCOUNT EXPIRES DATE = ' NOLINE

 DISPLAY $ACCOUNT_EXPIRES_DATE

BEGIN-SELECT

(to_date($ACCOUNT_EXPIRES_DATE, 'yyyy-mm-dd') - to_date('1600-12-31',
'yyyy-mm-dd'))*86400 &ACCOUNT_EXPIRES
```

```
 LET $ACCOUNT_EXPIRES = &ACCOUNT_EXPIRES

! DISPLAY 'ACCOUNT EXPIRES1 = ' NOLINE

! DISPLAY $ACCOUNT_EXPIRES

 LET $ACCOUNT_EXPIRES = SUBSTR($ACCOUNT_EXPIRES,1,11)

! DISPLAY 'ACCOUNT EXPIRES2 = ' NOLINE

! DISPLAY $ACCOUNT_EXPIRES

 LET $ACCOUNT_EXPIRES = $ACCOUNT_EXPIRES||'0000000'

! DISPLAY 'ACCOUNT EXPIRES3 = ' NOLINE

! DISPLAY $ACCOUNT_EXPIRES



from dual

END-SELECT
```

```
IF substr($A.EMPLID,1,1) = 'C'

 LET $ACCOUNT_EXPIRES = $ACCOUNT_EXPIRES

ELSE

 LET $ACCOUNT_EXPIRES = ' '

END-IF



DISPLAY 'SENT TO AD: ' NOLINE

DISPLAY $A.EMPLID NOLINE

DISPLAY '    ' NOLINE

DISPLAY $OPRID



  write 1 from ''

    $A.EMPLID
```

```
          '|'''

          $OPRID

          '''|'''

          $A.NAME

          '''|'''

$A.FIRST_NAME

          '''|'''

$A.LAST_NAME

          '''|'''

          $Nickname

          '''|'''

          $O_Work_Phone1

          '''|'''
```

```
$O_Work_Phone2

'"|"'

$O_Cell_Phone1

'"|"'

$O_Cell_Phone2

'"|"'

$O_Pager1

'"|"'

$O_Pager2

'"|"'

$O_Fax

'"|"'

$O_Telex
```

```
'"|"'

$Mail_Label

'"|"'

$Org_Unit_Desc

'"|"'

$jobcodedescr

'"|"'

$Department_Desc

'"|"'

$Org_Unit

'"|"'

$A.PER_ORG

'"|"'
```

```
$deptid

'''|'''

$company

'''|'''

$termdt

'''|'''

$status

'''|'''

$salplan

'''|'''

$grade

'''|'''

$TIE_LINE
```

```
       '"|"'

 $ADDRESS1

! $BLANK_SPACE

       '"|"'

       $ADDRESS2

! $BLANK_SPACE

       '"|"'

       $CITY

! $BLANK_SPACE

       '"|"'

 $STATE

 '"|"'

       $POSTAL
```

```
!  $BLANK_SPACE

        '"|"'

        $COUNTRY

!  $BLANK_SPACE

        '"|"'

 $PASSWORD

        '"|"'

        $ACCOUNT_EXPIRES

        '"|"'

 $CREATION_DATE

        '"|"'

        $SUPERVISOR_ID

        '"|"'
```

```
    $OUT_VIP

    '"|"'

    $LICENSE

    '"|'



BEGIN-SQL

INSERT INTO PS_Y_ID_SENT_TO_AD

VALUES ($A.EMPLID)

END-SQL



end-procedure
```

```
!************************************************************************
! Generate a Random Password                                            *
!************************************************************************

BEGIN-PROCEDURE generate-password

  let #i = 1

  LET $PASSWORD = ''

  while #i < 13
```

```
if #i = 1 or #i = 4 or #i = 7 or #i = 10

 LET $NEXT_CHAR = ''

 do select-random-upper

end-if



if #i = 2 or #i = 5 or #i = 8 or #i = 11

 LET $NEXT_CHAR = ''

 do select-random-number

end-if



if #i = 3 or #i = 6 or #i = 9 or #i = 12

 LET $NEXT_CHAR = ''

 do select-random-lower
 end-if

 add 1 to #i

 LET $PASSWORD = $PASSWORD||$NEXT_CHAR

  end-while

BEGIN-SELECT

substr(dbms_random.value,3,1) &LAST_SPECIAL

 LET #LAST_SPECIAL = &LAST_SPECIAL

from dual

END-SELECT
```

```
 LET $SPECIAL_CHARACTER = '%'

 IF #LAST_SPECIAL = 0 OR #LAST_SPECIAL = 1 OR #LAST_SPECIAL = 2 OR #LAST_SPECIAL
= 3
  LET $SPECIAL_CHARACTER = '@'
 END-IF

 IF #LAST_SPECIAL = 4 OR #LAST_SPECIAL = 5 OR #LAST_SPECIAL = 6 OR #LAST_SPECIAL
= 7
  LET $SPECIAL_CHARACTER = '#'
 END-IF

 IF #LAST_SPECIAL = 8 OR #LAST_SPECIAL = 9
  LET $SPECIAL_CHARACTER = '&'
 END-IF

   LET $PASSWORD = $PASSWORD||$SPECIAL_CHARACTER

 LET $PBA_UNION_CD = ' '

BEGIN-SELECT

CHECKPBA.UNION_CD &PBA_UNION_CD

 LET $PBA_UNION_CD = &PBA_UNION_CD

FROM PS_YQ_QUERY_FUT CHECKPBA

WHERE CHECKPBA.EMPLID = $A.EMPLID
```

```
END-SELECT

 IF $PBA_UNION_CD = 'PBA'

  LET $PASSWORD = 'Welcome'||$A.EMPLID||'#'

 END-IF

END-PROCEDURE generate-password

BEGIN-PROCEDURE select-random-upper

BEGIN-SELECT

dbms_random.string('U', 1) &NEXT_CHAR1

 LET $NEXT_CHAR = &NEXT_CHAR1
```

```
from dual
END-SELECT
END-PROCEDURE select-random-upper


BEGIN-PROCEDURE select-random-lower
BEGIN-SELECT
dbms_random.string('L', 1) &NEXT_CHAR2
 LET $NEXT_CHAR = &NEXT_CHAR2
from dual
END-SELECT
END-PROCEDURE select-random-lower


BEGIN-PROCEDURE select-random-number
```

```
BEGIN-SELECT

substr(dbms_random.value,3,1) &NEXT_CHAR3

 LET $NEXT_CHAR = &NEXT_CHAR3

from dual

END-SELECT

END-PROCEDURE select-random-number



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

! Write HEADER to fixed length file                                    *

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

begin-procedure Write-Header



   write 1 from '"EMPLID"'
```

'|'

'"USERID"'

'|'

'"NAME"'

'|'

'"FIRST NAME"'

'|'

'"LAST NAME"'

'|'

'"NICKNAME"'

'|'

'"WORK PHONE 1"'

'|'

'"WORK PHONE 2"'

'|'

'"CELL PHONE 1"'

'|'

'"CELL PHONE 2"'

'|'

'"PAGER 1"'

'|'

'"PAGER 2"'

'|'

'"FAX"'

'|'

'"TELEX"'

```
'|'

'"MAIL LABEL"'

'|'

'"ORG UNIT DESCR"'

'|'

'"JOB DESCR"'

'|'

'"DEPT DESCR"'

'|'

'"ORG UNIT"'

'|'

'"PER STATUS"'

'|'
```

'"DEPTID"'

'|'

'"CO"'

'|'

'"TERMDT"'

'|'

'"STATUS"'

'|'

'"SPLN"'

'|'

'"GRD"'

'|'

'"TL"'

```
'|'

'"ADDRESS 1"'

'|'

'"ADDRESS 2"'

'|'

'"CITY"'

'|'

'"STATE"'

'|'

'"POSTAL"'

'|'

'"COUNTRY"'

'|'
```

'"PASSWORD"'

'|'

'"ACCOUNT EXPIRES"'

'|'

'"CREATION DATE"'

'|'

'"SUPERVISOR ID"'

'|'

'"VIP"'

'|'

'"LICENSE"'

'|'

```
end-procedure



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

! Wrap up SQR - output number of rows processed and close file.        *

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

begin-procedure Wrap-Up


  display ' '

  display #rows_processed 99999 noline

  display ' Total Rows Processed'


  close 1
```

```
!FTP File to patciv11

let $hostname = 'teleav842.pacorp.panynj.gov '

let $port = '21 '

let $uid = 'psoftftp '

let $pwd = 'Panynj1234 '


LET $SYSTEM_CALL = '/u08/batch/scripts/ftp_quick_connect.sh '    || $hostname

        || $port

        || $uid

        || $pwd

        || $Filepathftp || ' '

        || $filename
```

```
  show $SYSTEM_CALL

  CALL SYSTEM USING $SYSTEM_CALL #RETVAL WAIT



end-procedure



!------------------------------------------------------------------------------

! SQCs

!------------------------------------------------------------------------------



#Include 'curdttim.sqc'         !GET-CURRENT-DATETIME procedure

#Include 'datetime.sqc'         !INIT-DATETIME and other procedures for date and time

                                !formatting

#Include 'number.sqc'           !INIT-NUMBER and other procedures for formatting
```

```
                    !numbers

#Include 'rotname1.sqc'     !ROTATE_NAME procedure to format Name -> First MI Last

#Include 'stdapi.sqc'       !STDAPI-INIT procedure to update process scheduler API

#Include 'prcsapi.sqc'      !Get Run Control Parms

#Include 'stderror.sqc'     !Unknown-SQL-Error-Message

#Include 'datemath.sqc'     !dtu-subtract-days
```