

```
#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
```

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

```
!                                     *
```

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

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

!\*\*\*\*\*

\*

!\*\*\*\*\*

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

!\*\*\*\*\*

\*

!\*\*\*\*\*

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

!\*\*\*\*\*

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

```

```

!*****

```

```

      *

```

```

!*****

```

```

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

```

!*****

!      *

!*****

```

```

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

\*

!\*\*\*\*\*

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\_cnsItnt\_PHONE

WHERE EMPLID = \$Emplid

ORDER BY PHONE\_TYPE

end-select

end-procedure

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

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

```
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

!\*\*\*\*\*

! \*

!\*\*\*\*\*

begin-procedure Get-Org-Unit-Desc (\$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

!\*\*\*\*\*

\*

!\*\*\*\*\*

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
```

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

```
      *
```

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

```
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

!\*\*\*\*\*

\*

!\*\*\*\*\*

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

\*

!\*\*\*\*\*

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
```

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

```
      *
```

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

```
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

!\*\*\*\*\*

! \*

!\*\*\*\*\*

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

!-----

!-----

#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 Params

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

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