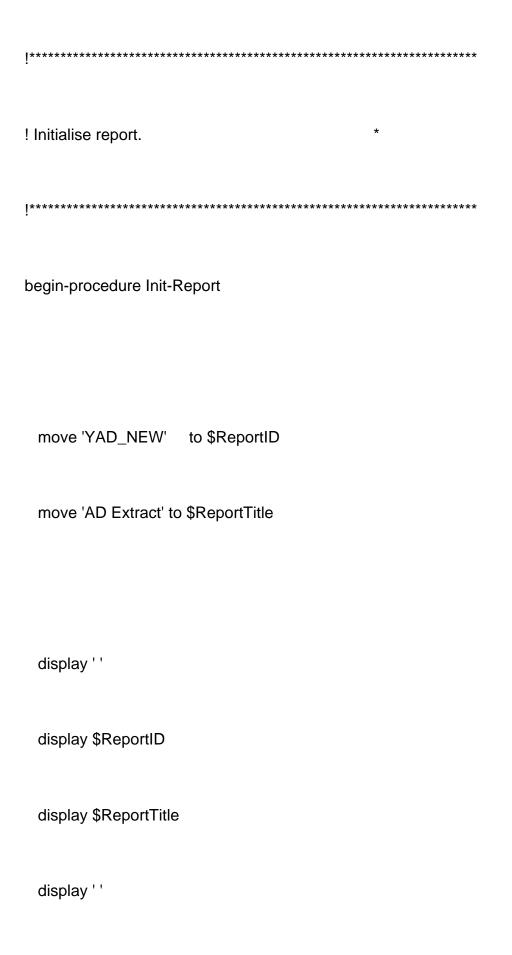
! AD New: Send data to active directory for new employees. *
!*************************************
! SQR LOG *
! *
! DESCRIPTION: This program extracts employee and non-employee current *
! data to be sent to AD. *
! *
! ASSUMPTION: Always current information when run. *
! Only active, leave and suspended employees. *
! Companies PS, TH and JST. *
! *
[*************************************

<u></u>
!*************************************
! 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
!	<del></del>
#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



```
#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'

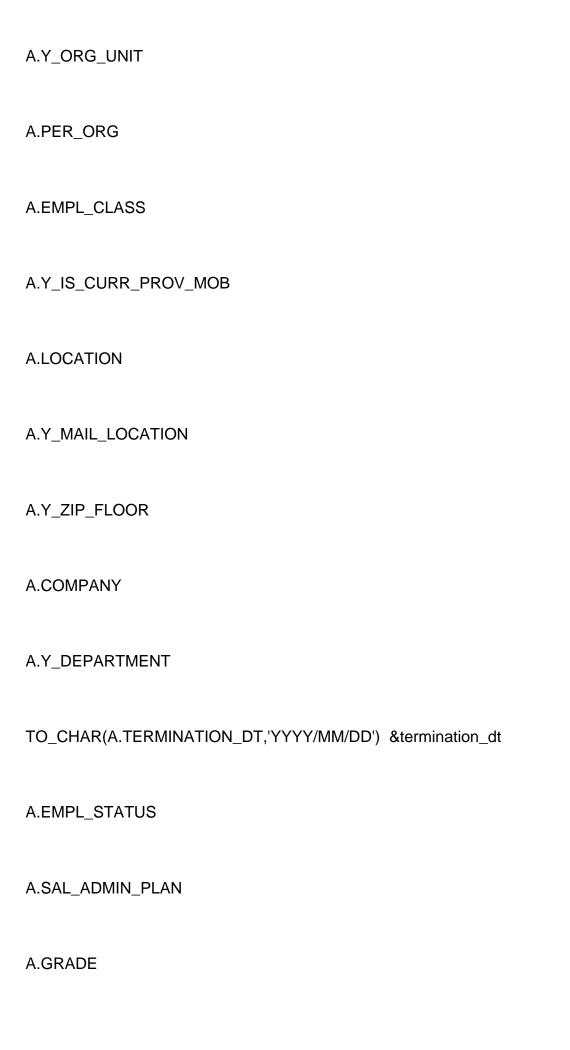
open \$Filepath as 1 for-writing record=1000:vary

do Check-Prov-Implemented (\$Prov\_Implemented)

end-procedure
<b> </b> ************************************
! 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.
<b>.</b>

********************
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

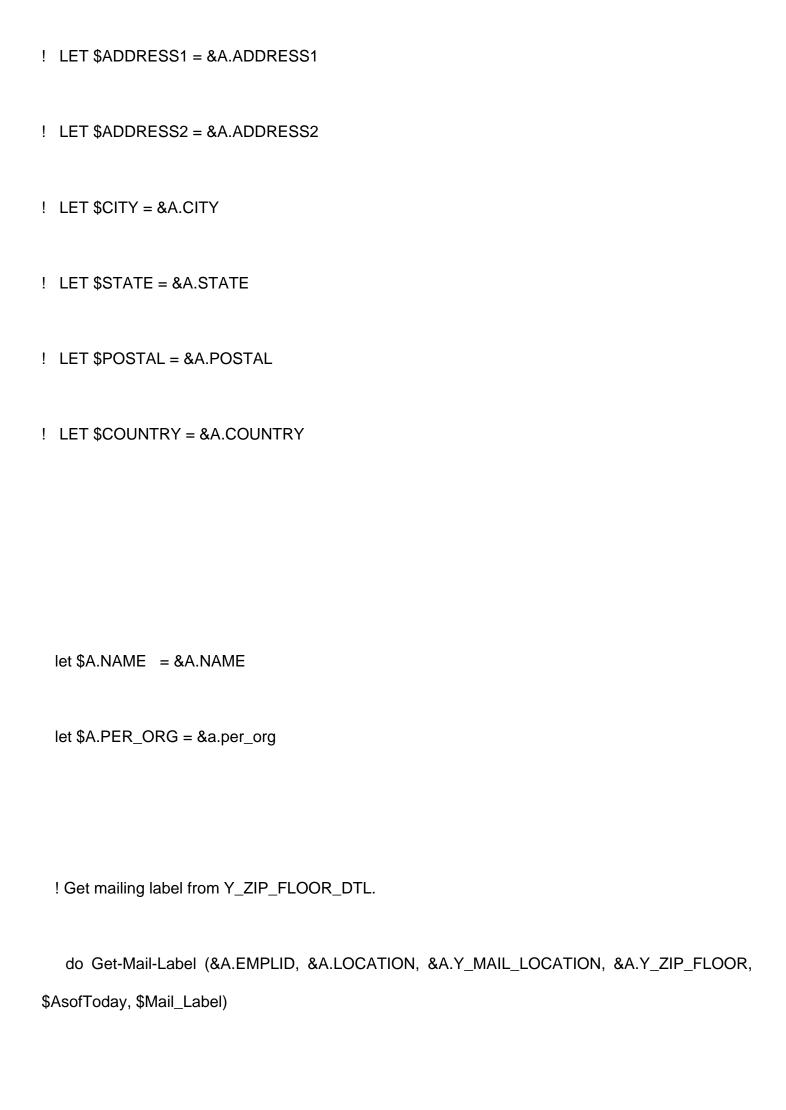


A.SUPERVISOR_ID &A.SUPERVISOR_ID
A.UNION_CD &UNION_CD
!A.ADDRESS1
!A.ADDRESS2
!A.CITY
!A.STATE
!A.POSTAL
!A.COUNTRY
let \$emplid = &A.EMPLID
let \$A.EMPLID = &A.EMPLID
LET \$A.FIRST_NAME = &A.FIRST_NAME
LET \$A.LAST_NAME = &A.LAST_NAME

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 = ' '

DO GET-ADDRESS-FOR-LOCATION

**END-IF** 



! 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
```

!	!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 \$termdt = ' '

Let \$status = ' '

Let \$salplan = ' '

Let \$grade = ' '

Let \$TIE\_LINE = ' '

LET \$OUT\_VIP = 'N'

LET \$SUPERVISOR\_ID = ' '

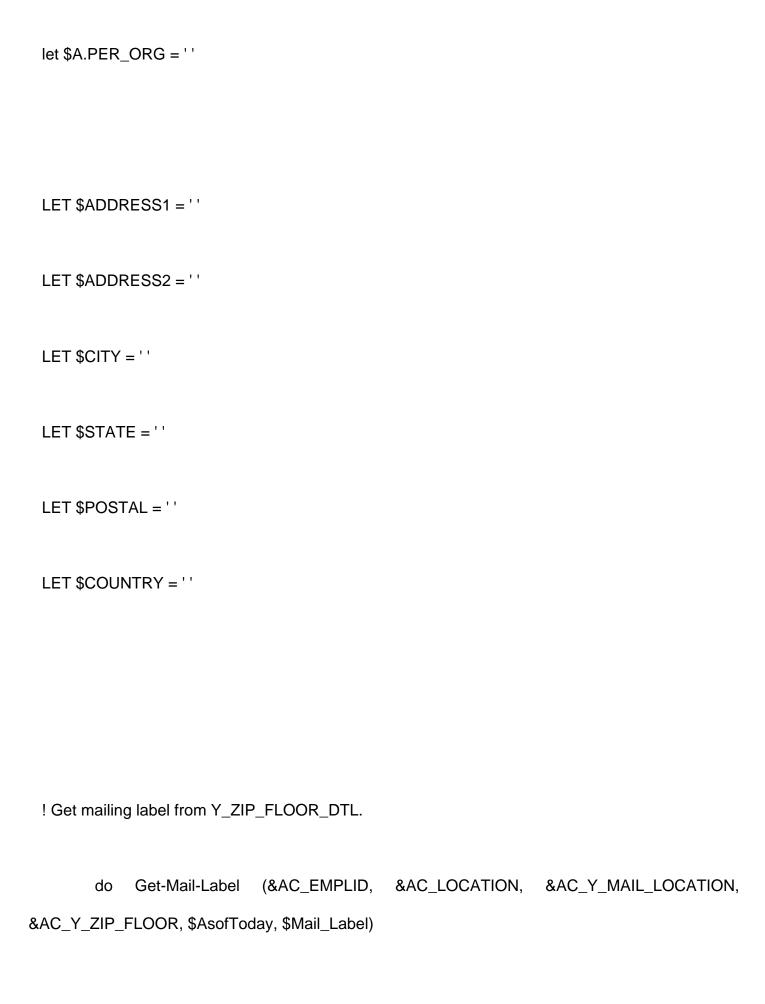
let \$emplid = &AC\_EMPLID

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



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'
```



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



WHERE REPRESENTED.UNION	CD = REPRESENTED	<b>ED LINION</b>	CD
WITEINE INELIGENTED. CINION	OD - NEI NEGENTED	LD.UNIUN	$\cup$ $\cup$

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'

**END-IF** 

**END-IF** 

**END-IF** 

**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

ı	-	דוד	LINIT		CVTC	ICION
ı	$\vdash$ 1 3	ын	1 IINF =	&CWR	<b>FXIF</b> I	りいしい

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



**END-SELECT** 

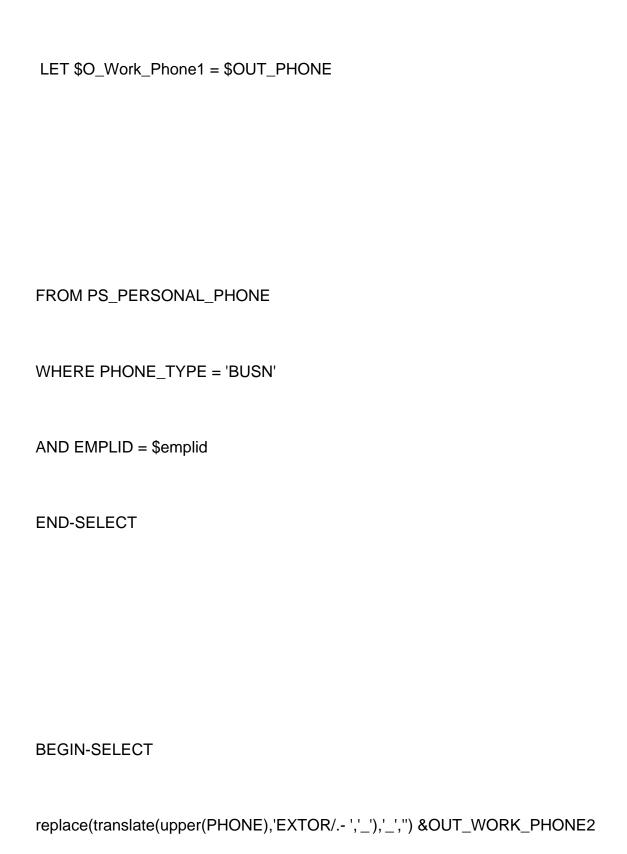
**END-PROCEDURE GET-CWR-LOCATION** 

BEGIN-PROCEDURE GET-WORK-PHONE

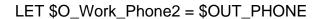
**BEGIN-SELECT** 

replace(translate(upper(PHONE), 'EXTOR/.- ','\_'),'\_\_',") &OUT\_WORK\_PHONE1

LET \$OUT\_PHONE = &OUT\_WORK\_PHONE1



LET \$OUT\_PHONE = &OUT\_WORK\_PHONE2



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

	N I	$\overline{}$	-5	_	_	$\sim$	•
_	N	1		-1	_		ı

**END-PROCEDURE GET-WORK-PHONE** 

BEGIN-PROCEDURE GET-CWR-NAME

**BEGIN-SELECT** 

CWRN.FIRST\_NAME &CWR\_FIRST\_NAME

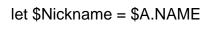
SUBSTR(CWRN.MIDDLE\_NAME,1,1) &CWR\_MIDDLE\_INITIAL

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



FROM PS\_NAMES CWRN

WHERE CWRN.EMPLID = \$emplid

AND CWRN.NAME\_TYPE = 'PRI'

AND CWRN.EFFDT = (SELECT MAX(CWRN1.EFFDT) FROM PS\_NAMES CWRN1

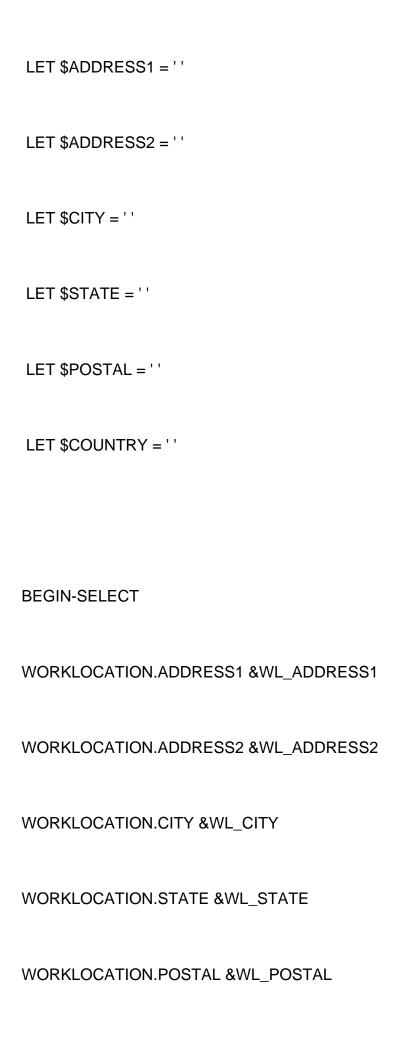
WHERE CWRN.EMPLID = CWRN1.EMPLID

AND CWRN.NAME\_TYPE = CWRN1.NAME\_TYPE)

**END-SELECT** 

**END-PROCEDURE** 

BEGIN-PROCEDURE GET-ADDRESS-FOR-LOCATION



## 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

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  $Y_{ZIP_FLOOR} = Z.Y_{ZIP_FLOOR}$ 

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,

let \$Pager1 = "

let \$Pager2 = "

let \$Fax

let \$Telex = "

:\$Work\_Phone1, :\$Work\_Phone2, :\$Cell\_Phone1, :\$Cell\_Phone2, :\$Pager1, :\$Pager2, :\$Fax, :\$Telex)

let \$Work\_Phone1 = "

let \$Work\_Phone2 = "

let \$Cell\_Phone1 = "



break

when = 'OTR'

let \$Cell\_Phone2 = &PHONE

break

when = 'BPG1'

let \$Pager1 = &PHONE

break

when = 'BPG2'

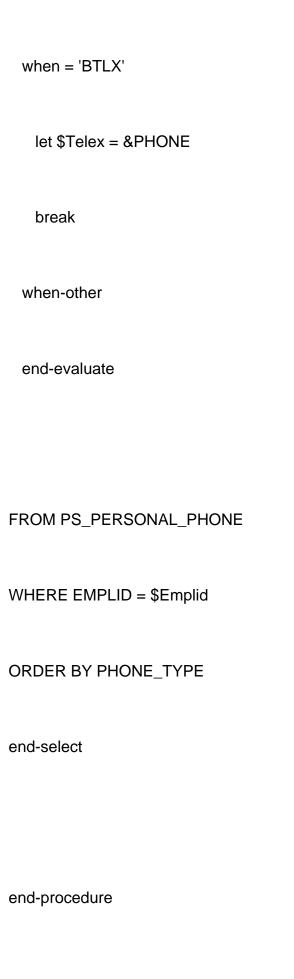
let \$Pager2 = &PHONE

break

when = 'BFAX'

let \$Fax = &PHONE

break



!*************************************
! 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.
<u> </u> ************************************
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'

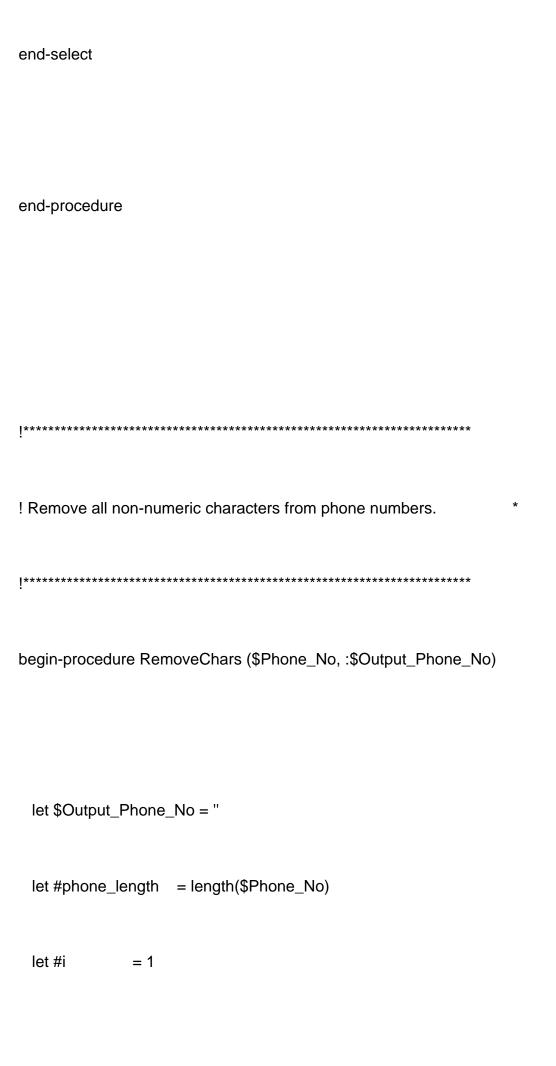
let \$Fax = &PHONE



FROM PS\_Y\_cnsltnt\_PHONE

WHERE EMPLID = \$Emplid

ORDER BY PHONE\_TYPE



while #i <= #phone\_length
let \$textchar = substr(\$Phone\_No, #i, 1)</pre>

evaluate \$textchar

when = '1'

when = '2'

when = '3'

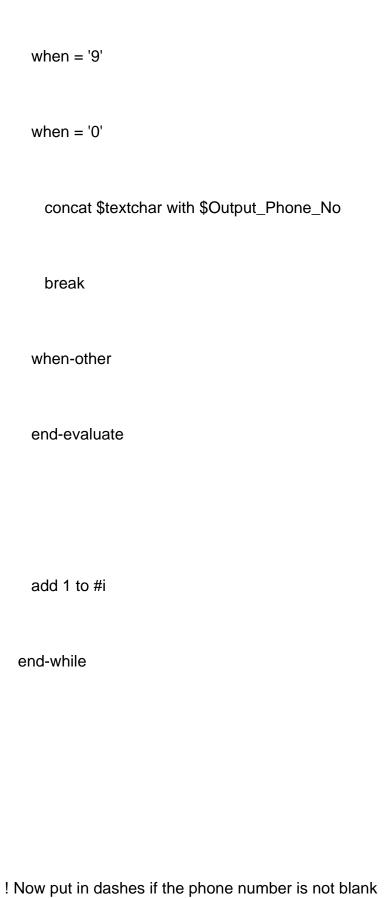
when = '4'

when = '5'

when = '6'

when = '7'

when = '8'



LET	\$0	Output_Phone_No		=
SUBSTR(\$Output_I	Phone_No,1,3)  '-'  SUBS	TR(\$Output_Phon	e_No,4,3)  '-'  SU	JBSTR(\$Output_Pho
ne_No,7,4)				
END-IF				
end-procedure				
[*************************************	*******	******	**	
!				
! Get org unit descri	ption and rollup dept des	cription .	*	
. Oot org arm accom	priori aria romap dopt doo	op.a.o		
!*********	*******	*******	**	
begin-procedure	Get-Org-Unit-Descr	(\$Company,	\$Org_Unit,	:\$Org_Unit_Desc,
:\$Department_Desc	c)			
let \$Org_Unit_Des	sc = "			

let \$Department\_Desc = "

begin-select

Y\_ORG\_UNIT\_DS

Y\_DEPARTMENT\_DS

let \$Org\_Unit\_Desc = &Y\_ORG\_UNIT\_DS

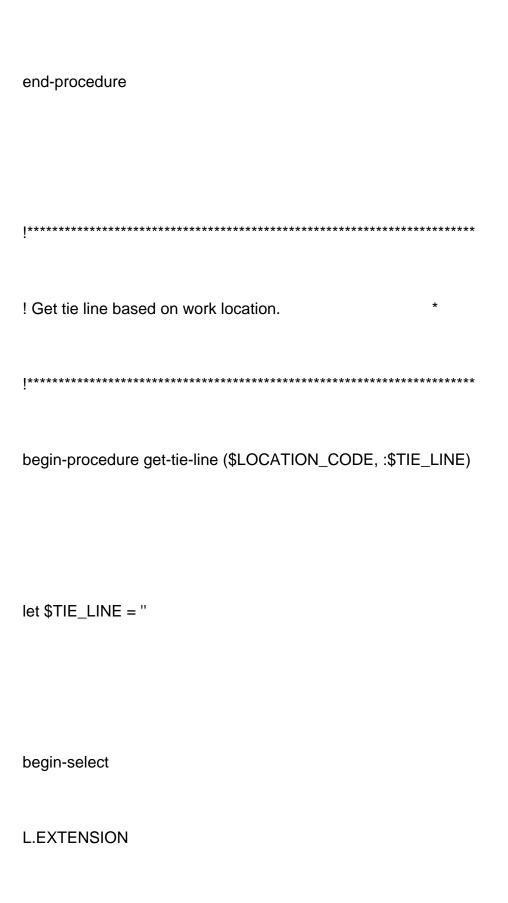
let \$Department\_Desc = &Y\_DEPARTMENT\_DS

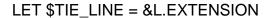
FROM PS\_Y\_ORG\_ROLLUP

WHERE COMPANY = \$Company

AND Y\_ORG\_UNIT = \$Org\_Unit

end-select





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

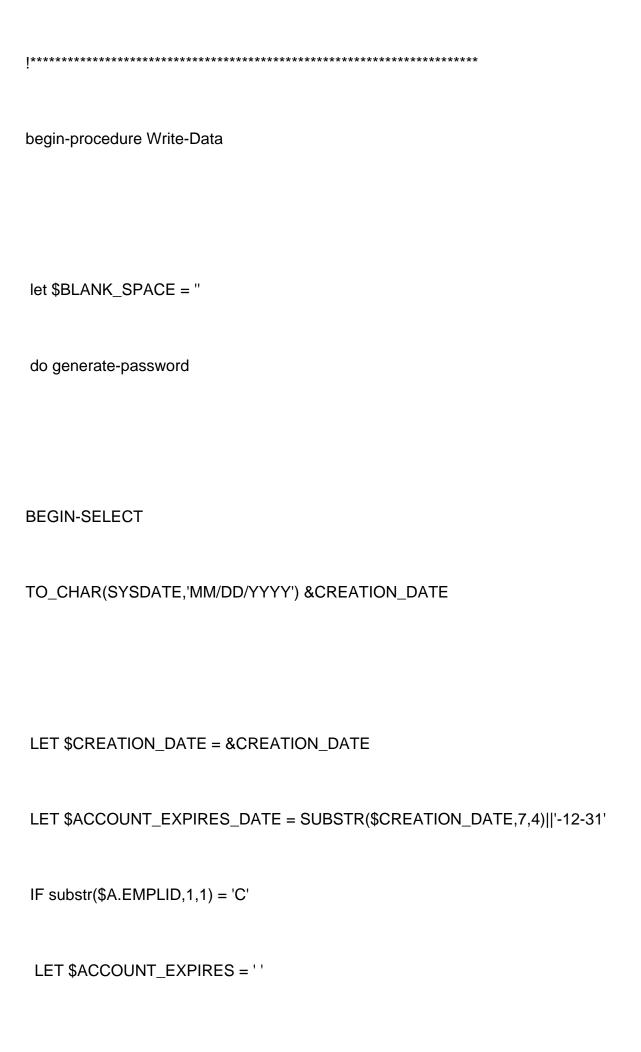
AND L.LOCATION = L2.LOCATION)

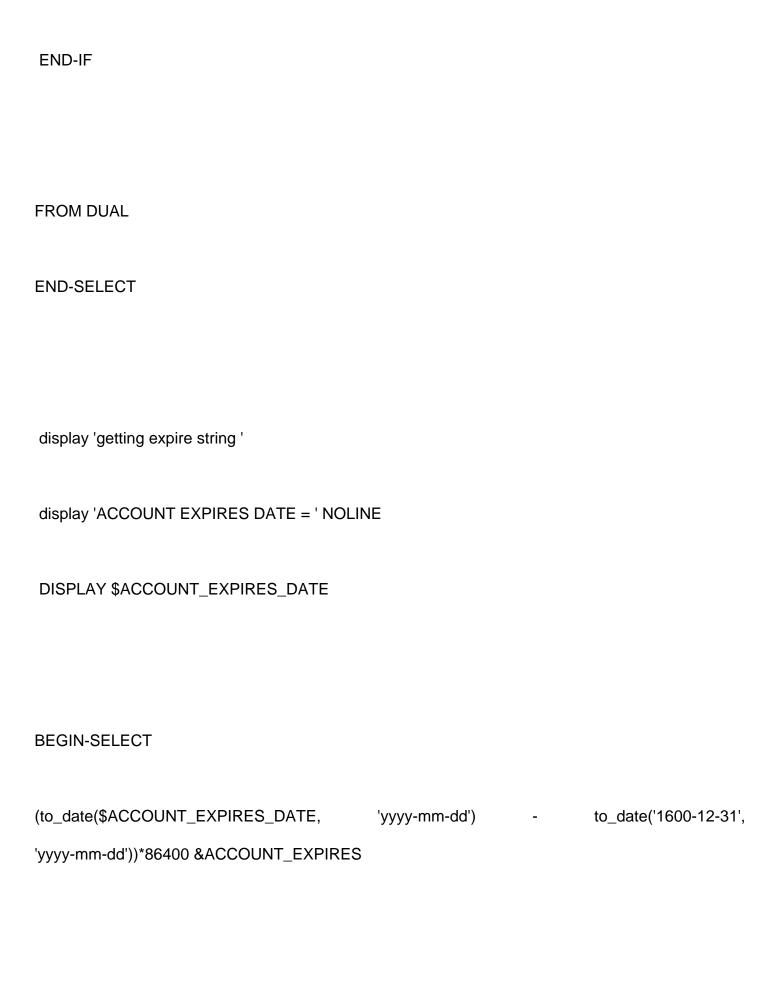
end-select

end-procedure



! end-if	
FROM PSOPRDEFN M	
WHERE M.EMPLID = \$emplid	
end-select	
end-procedure	
*************************************	





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
```

	\$A.NAME
	"" ""
\$A.FIRS	ST_NAME
	'"  "
\$A.LAS <sup>-</sup>	T_NAME
	· <b>·</b> ··[···
	\$Nickname
	111 111
	\$O_Work_Phone1
	"" ""

'|"'

\$OPRID

"" "
\$O_Cell_Phone1
"" ""
\$O_Cell_Phone2
"" ""
\$O_Pager1
"" ""
\$O_Pager2
"" ""
\$O_Fax
"" ""
\$O_Telex

\$O\_Work\_Phone2

```
'"|"'
$Org_Unit_Desc
'"|"'
$jobcodedescr
'"|"'
$Department_Desc
'"|"'
$Org_Unit
'"|"'
$A.PER_ORG
'"|"'
```

'"|"'

\$Mail\_Label

" "		
\$company		
"" ""		
\$termdt		
"" ""		
\$status		
"" "		
\$salplan		
" "		
\$grade		
" "		
\$TIE_LINE		

\$deptid

! \$BLANK\_SPACE

""|"

\$ADDRESS2

! \$BLANK\_SPACE

'"|"

\$CITY

! \$BLANK\_SPACE

'"|"'

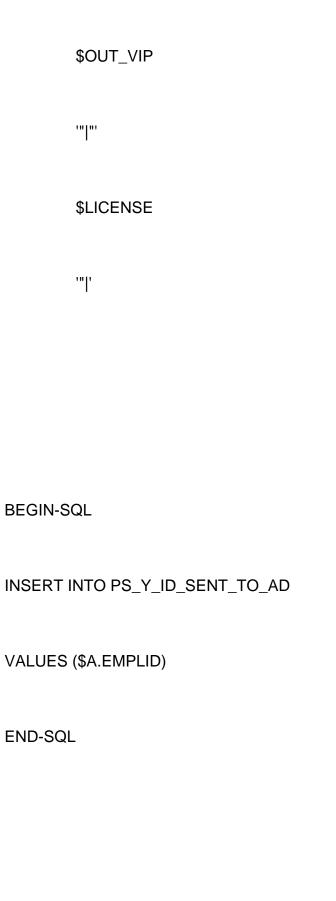
\$STATE

""|""

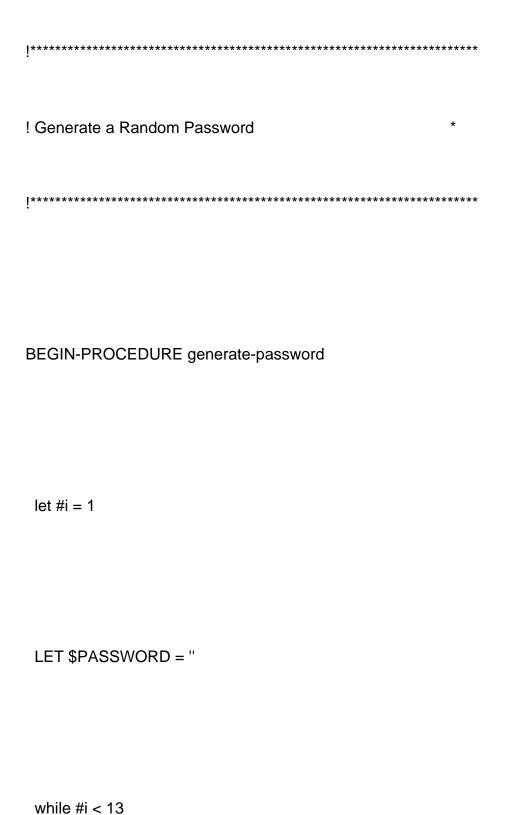
\$POSTAL

```
'"|"'
        $COUNTRY
! $BLANK_SPACE
 $PASSWORD
        $ACCOUNT_EXPIRES
        '"|"'
 $CREATION_DATE
        ""|""
        $SUPERVISOR_ID
        '"|"'
```

! \$BLANK\_SPACE



end-procedure



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 = "

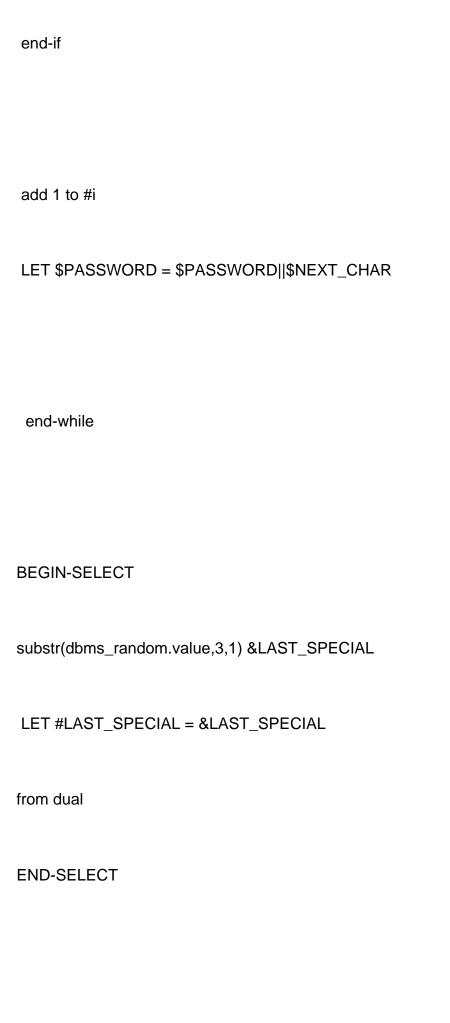
do select-random-number

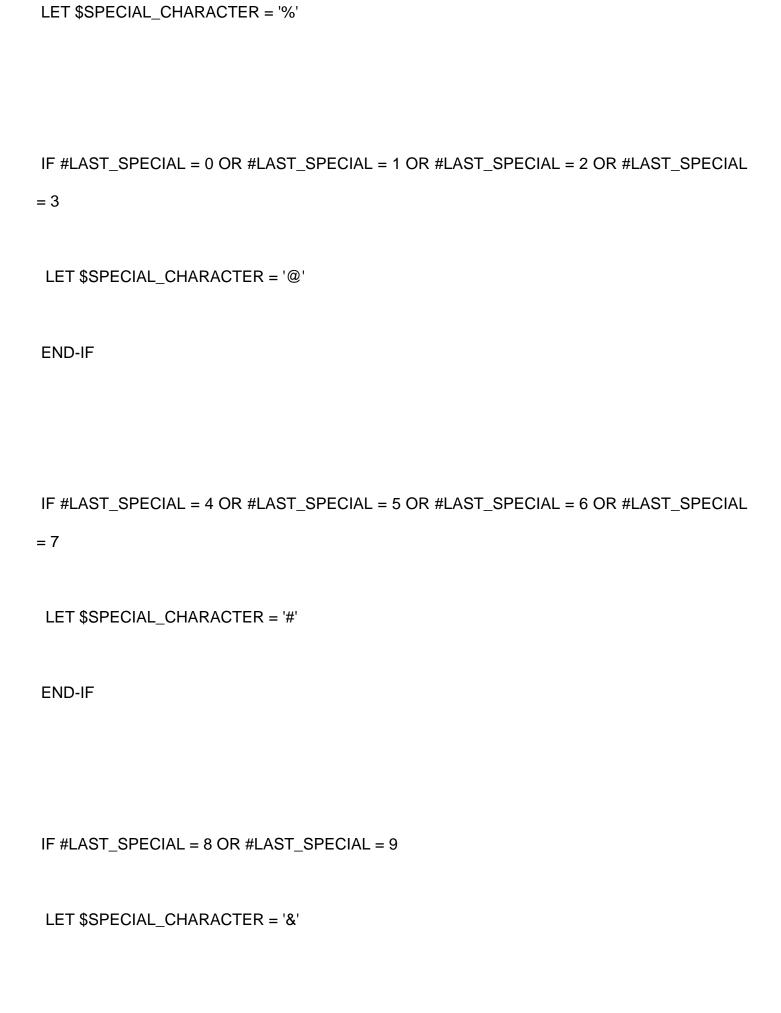
end-if

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

LET \$NEXT\_CHAR = "

do select-random-lower





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

<b>END-SELECT</b>	ΕN	D.	-S	Ε	LE	C٦	Γ
-------------------	----	----	----	---	----	----	---

IF \$PBA\_UNION\_CD = 'PBA'

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

**END-IF** 

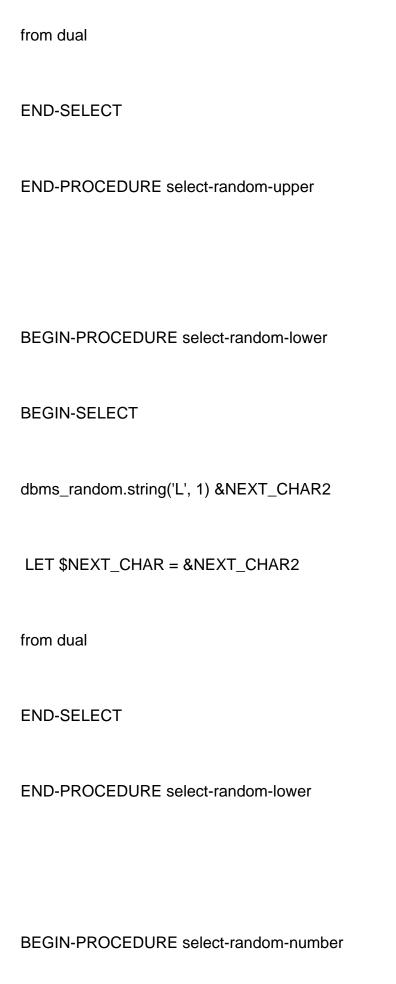
END-PROCEDURE generate-password

BEGIN-PROCEDURE select-random-upper

**BEGIN-SELECT** 

dbms\_random.string('U', 1) &NEXT\_CHAR1

LET \$NEXT\_CHAR = &NEXT\_CHAR1

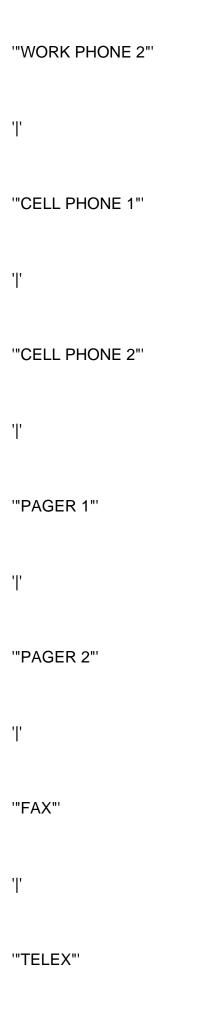


substr(dbms_random.value,3,1) &NEXT_CHAR3
LET \$NEXT_CHAR = &NEXT_CHAR3
from dual
END-SELECT
END-PROCEDURE select-random-number
[*************************************
! Write HEADER to fixed length file *
[*************************************
begin-procedure Write-Header

**BEGIN-SELECT** 

write 1 from '"EMPLID"'

'l'		
"USERID"		
'I'		
"NAME"		
']'		
"FIRST NAME"		
'l'		
"LAST NAME"		
'I'		
"NICKNAME"		
'l'		
"WORK PHONE 1"		
Т		



Ή'

Ή'

' '		
'"CO"'		
' '		
"TERMDT"		
' '		
"STATUS"		
' '		
'"SPLN"'		
' '		
'"GRD"'		
' '		
'"TL"'		

"DEPTID"

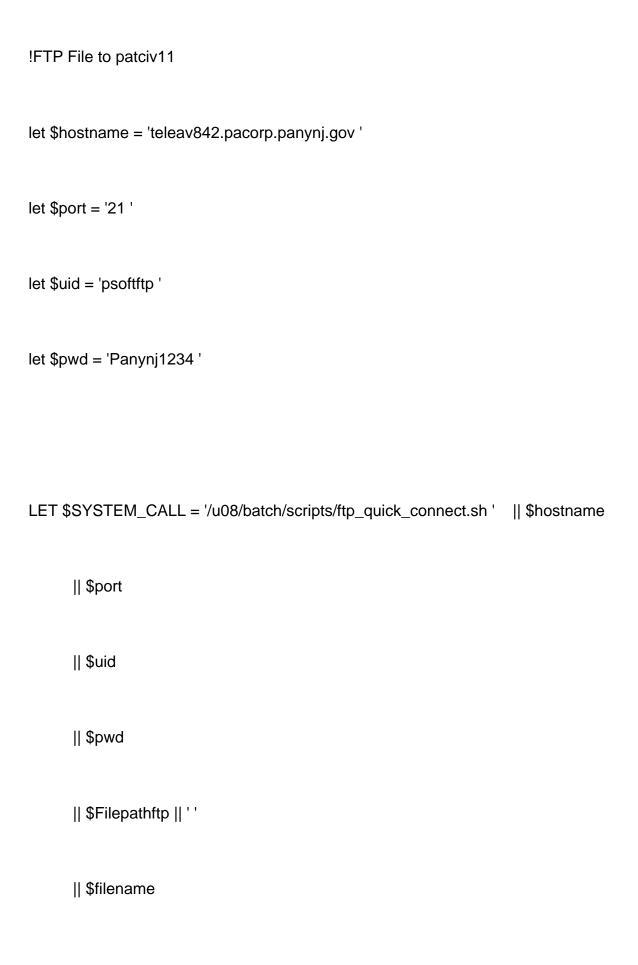
"ADDRESS 1"	
' '	
"ADDRESS 2"	
' '	
"CITY"	
' '	
"STATE"	
'l'	
"POSTAL"	
'l'	
"COUNTRY"	
'l'	

'['

"I"
"ACCOUNT EXPIRES"
"
"CREATION DATE"
']'
"SUPERVISOR ID"
']'
"VIP"
']'
"LICENSE"
']'

"PASSWORD"

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



## 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
!forma	tting
#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