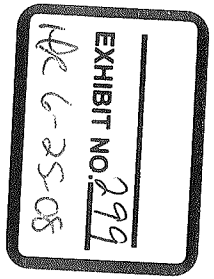
Item #: CSS-TN-1214062594	Summary: New legislation, effective Jan. 1, 2007, provides that the overtime should be itemized as corrections on the pay stub for the next regular pay period. Any correction in a subsequently issued pay stub must state the date of the pay period to which it relates [Cal. Lab. Cd. § 204(b)(2) , as amended by L. 2005, AB 2095. **Source Reference:** http://www.leginfo.ca.gov/cgi-bin/calawquery?codesection=lab&codebody= *Full text of CA Labor Code section 204 (b)* (b) (1) Notwithstanding any other provision of this section, **all wages earned for labor in excess of the normal work period shall be paid no later than the payday for the next regular payroll period.** (2) An employer is in compliance with the requirements of subdivision (a) of Section 226 relating to total hours worked by the employee, if hours worked in excess of the normal work period during the current pay period are itemized as corrections on the pay stub for the next regular pay period. **Any corrections set out in a subsequently issued pay stub shall state the inclusive dates of the pay period for which the employer is correcting its initial report of hours worked.** The employees' paycheck (PAY003) and advice (DDP003) will need to be modified to the following criteria: 1) All non-salaried employees who have pay earnings for the State of California. 2) All overtime earnings earned for current payroll to be paid during next pay period will need to be reflected as a correction for next pay period. 3) All earnings adjustments whose earnings end date is less than current pay period begin date. Modify PAY003 (Paycheck print) and DDP003 (Advice print) to meet the following requirements for all non-salaried employees: • If there are prior pay period earnings associated with current payroll for overtime and other earnings, modify paycheck/advice print programs to display these earnings as a separate line item. • Modify paycheck/advice print programs display earnings code, earnings, earnings begin date and earnings end date for prior period California earnings. • Modify paycheck/advice print programs display earnings code, earnings, pay begin date and pay end date for current payroll California earnings. • Modify paycheck/advice print programs to print a cross foot message for California Prior Pay Period earnings which has reached the maximum of 13 earnings and has summarized additional current/prior earnings in to one amount with a description of "Other". • Modify paycheck/advice to ensure sort order allows for all current payroll earnings appear first, followed by any prior pay period earnings on the paycheck for all California hourly employees.
Databases: H831TSUM	Successfully processed the modified PAY003.SQR and DDP003.SQR against these databases.
Prerequisites: **Application Dependencies** **Direct Deposit Setup** **Payroll Process** **Peopletools: Project for Temporary Work Table**	Application Dependencies: PAY003/DDP003 – Must run after a successful Pay Confirmations *Data Dependencies:* Payroll Data for California/Non California Employees



NOTE: Navigation is for version PeopleSoft ver 8.31.

Direct Deposit Data/Non California Employees

Direct Deposit Setup

1. Verify Banking information has been setup –
   Define Business Rules > Manage Human Resources (GBL) > Setup > Pay Group Table (Page: Calc Parameters)
   Define Business Rules > Manage Human Resources (GBL) > Setup > Source Bank (listed on Pay Group Table - Calc Parameters)
2. Verify Bank/Branch table has been setup with valid bank information. - Define Business Rules > Manage Human Resources (GBL) > Setup > Bank/Branch Table
3. Setup Employee Direct Deposit - Compensate Employees > Maintain Payroll Data (US) > Use > Direct Deposit

Payroll Process:

1. Create Paysheets - Compensate Employees > Manage Payroll Process (US) > Process > Paysheet Creation
2. Update Paysheets with both Current and Prior Pay Earnings - Compensate Employees > Manage Payroll Process (US) > Use > Payline
3. Calculate Payroll - Compensate Employees > Manage Payroll Process (US) > Process > Pay Calculation
4. Review and Correct Payroll Error Messages - Compensate Employees > Manage Payroll Process (US) > Inquire > Payroll Error Messages
5. Final Calculation – (same navigation as 4)
6. Confirm Payroll - Compensate Employees > Manage Payroll Process (US) > Process > Pay Confirmation
7. Print Pay Checks – PAY003.SQR - Compensate Employees > Manage Payroll Process (US) > Report 1 > Check Print
8. Print Advices – DDP003.SQR - Compensate Employees > Manage Payroll Process (US) > Report 1 > DDP Advice Print

Navigation:

Program	Navigation	Run Control Parameter(s)
PAY003.SQR	Compensate Employees > Manage Payroll Process (US) > Report 1 > Check Print	Payrun ID
DDP003.SQR	Compensate Employees > Manage Payroll Process (US) > Report 1 > DDP Advice Print	Payrun ID

Project: Created Temporary Work Table used to Sort California Earnings.
Version 7.02 Project: PRJ1214062594_702_TN - PeopleSoft Version did not have field COMP_RATECD.
Version 7.02 does not have field COMP_RATECD available.

Version 7.51, 8.01, and 8.31: PRJ1214062594_TN.

Version 8.8 and 8.9: PRJ1214062594_881_890_TN.

```
CREATE TABLE PS_WRK_PYERN_PRC (PROCESS_INSTANCE DECIMAL(10) NOT NULL,
    EARNS_TYPE CHAR(1) NOT NULL,
    ERNCD CHAR(3) NOT NULL,
    EARNS_BEGIN_DT PSDATE NULL,
    EARNS_END_DT PSDATE NULL,
    COMP_RATECD CHAR(6) NOT NULL,
    HOURLY_RT DECIMAL(18, 6) NOT NULL,
    DESCR CHAR(30) NOT NULL,
    EARNS_HRS DECIMAL(6, 2) NOT NULL,
    EARNS_AMT DECIMAL(10, 2) NOT NULL)
```

California Employees:

Oracle SQL Statement used to select potential California Wage Earning Employees:

```
SELECT
A.EMPLID,
A.EMPL_RCD,
A.EFFDT,
A.EMPL_TYPE,
A.DEPTID,
A.LOCATION,
A.TAX_LOCATION_CD,
A.REG_TEMP,
A.FULL_PART_TIME,
A.COMPANY,
A.PAYGROUP,
A.STD_HOURS,
A.COMPRATE,
A.BUSINESS_UNIT,
B.STATE,
B.RESIDENT,
B.UI_JURISDICTION
FROM PS_JOB A,
   PS_STATE_TAX_DATA B
WHERE A.EMPLID = B.EMPLID
 AND B.COMPANY = A.COMPANY
 AND A.EFFDT =
          (SELECT MAX(A_ED.EFFDT) FROM PS_JOB A_ED
          WHERE A.EMPLID = A_ED.EMPLID
           AND A.EMPL_RCD = A_ED.EMPL_RCD
           AND A_ED.EFFDT <= SYSDATE)
 AND A.EFFSEQ =
          (SELECT MAX(A_ES.EFFSEQ) FROM PS_JOB A_ES
          WHERE A.EMPLID = A_ES.EMPLID
           AND A.EMPL_RCD = A_ES.EMPL_RCD
           AND A.EFFDT = A_ES.EFFDT)
 AND B.EFFDT =
          (SELECT MAX(B_ED.EFFDT) FROM PS_STATE_TAX_DATA B_ED
          WHERE B.EMPLID = B_ED.EMPLID
```

```
            AND B.COMPANY = B_ED.COMPANY
            AND B_ED.EFFDT <= SYSDATE)
 AND A.EMPL_STATUS = 'A'
 AND A.COMPANY = 'GBI'
 AND A.PAYGROUP = 'KU2'
/*  AND B.STATE = 'CA'*/
ORDER BY B.STATE,
A.EMPL_TYPE,
A.EMPLID
```

SQL Server SQL Statement used to select potential California Wage Earning Employees:

```
SELECT
A.EMPLID,
A.EMPL_RCD,
(CONVERT(CHAR(10),A.EFFDT,121)),
A.EMPL_TYPE,
A.DEPTID,
A.LOCATION,
A.TAX_LOCATION_CD,
A.REG_TEMP,
A.FULL_PART_TIME,
A.COMPANY,
A.PAYGROUP,
A.STD_HOURS,
A.COMPRATE,
A.BUSINESS_UNIT,
B.STATE,
B.RESIDENT,
B.UI_JURISDICTION
FROM PS_JOB A,
   PS_STATE_TAX_DATA B
WHERE A.EMPLID = B.EMPLID
 AND B.COMPANY = A.COMPANY
 AND A.EFFDT =  (SELECT MAX(A_ED.EFFDT)
                 FROM PS_JOB A_ED
                   WHERE A.EMPLID = A_ED.EMPLID
                     AND A.EMPL_RCD = A_ED.EMPL_RCD
                     AND A_ED.EFFDT <= SUBSTRING(CONVERT(CHAR,GETDATE(),121), 1, 10))
 AND A.EFFSEQ =  (SELECT MAX(A_ES.EFFSEQ)
                  FROM PS_JOB A_ES
                     WHERE A.EMPLID = A_ES.EMPLID
                       AND A.EMPL_RCD = A_ES.EMPL_RCD
                       AND A.EFFDT = A_ES.EFFDT)
 AND B.EFFDT =  (SELECT MAX(B_ED.EFFDT)
                 FROM PS_STATE_TAX_DATA B_ED
                   WHERE B.EMPLID = B_ED.EMPLID
                     AND B.COMPANY = B_ED.COMPANY
                     AND B_ED.EFFDT <= SUBSTRING(CONVERT(CHAR,GETDATE(),121), 1, 10))
 AND A.EMPL_STATUS = 'A'
 AND A.COMPANY = 'GBI'
 AND A.PAYGROUP = 'KU2'
/*  AND B.STATE = 'CA'*/
ORDER BY A.EMPL_TYPE,
```

	A.EMPLID **Db2 SQL Statement used to select potential California Wage Earning Employees:** SELECT A.EMPLID, A.EMPL_RCD, A.EFFDT, A.EMPL_TYPE, A.DEPTID, A.LOCATION, A.TAX_LOCATION_CD, A.REG_TEMP, A.FULL_PART_TIME, A.COMPANY, A.PAYGROUP, A.STD_HOURS, A.COMPRATE, A.BUSINESS_UNIT, B.STATE, B.RESIDENT, B.UI_JURISDICTION FROM PS_JOB A, PS_STATE_TAX_DATA B WHERE A.EMPLID = B.EMPLID AND B.COMPANY = A.COMPANY AND A.EFFDT = (SELECT MAX(A_ED.EFFDT) FROM PS_JOB A_ED WHERE A.EMPLID = A_ED.EMPLID AND A.EMPL_RCD = A_ED.EMPL_RCD AND A_ED.EFFDT <= (CURRENT DATE)) AND A.EFFSEQ = (SELECT MAX(A_ES.EFFSEQ) FROM PS_JOB A_ES WHERE A.EMPLID = A_ES.EMPLID AND A.EMPL_RCD = A_ES.EMPL_RCD AND A.EFFDT = A_ES.EFFDT) AND B.EFFDT = (SELECT MAX(B_ED.EFFDT) FROM PS_STATE_TAX_DATA B_ED WHERE B.EMPLID = B_ED.EMPLID AND B.COMPANY = B_ED.COMPANY AND B_ED.EFFDT <= (CURRENT DATE)) AND A.EMPL_STATUS = 'A' AND A.COMPANY = 'GBI' AND A.PAYGROUP = 'KU2' ORDER BY A.EMPL_TYPE, A.EMPLID
Steps performed to Resolve Issue:	
Modified the PAY003.SQR and PAY003.SQR with the following code changes relatively across all clients impacted:	

!*** ! !*** ! Modified for Education & Government * ! HP99999 Release 8 Technical Merge * ! HP99998 E&G 7.51 AU Merge * ! !*** ! !*** ! Modified for Education & Government * ! HP99999 Release 8 Technical Merge * ! HP99998 E&G 7.51 AU Merge * ! * ! * ! Modification Type: TomorrowNow Update - changes * ! Modification Tags/TN Report ID: 1214062594 * ! * !***	1. Modification History
let #lastRow = (n)-1 let #lastCol = (n)-1 let $lastCode = '***' let $lastName1 = 'Other' let $lastCmpRtCd = '***' let #lastRate = 0 create-array name = ETDarray - size = (n) - field = Code:char:(n) - field = Name:char:(n) - field = CmpRtCd:char:(n) - ! Comp_rate_code field = Cur:number:(n) - field = YTD:number:(n)	let #lastRow = (n)-1 let #lastCol = (n)-1 let $lastCode = '***' let $lastName1 = 'Other' let $lastCmpRtCd = '***' let #lastRate = 0 let $lastEarnBegin = '' !1214062594 let $lastEarnEnd = '' !1214062594 create-array name = ETDarray - size = (n) - field = Code:char:(n) - field = Name:char:(n) - field = Earn_Begin_Dt:char:(n) !1214062594 field = Earn_End_Dt:char:(n) !1214062594 field = CmpRtCd:char:(n) - ! Comp_rate_code field = Cur:number:(n) - field = YTD:number:(n)	begin-procedure Init-Arrays 1. Create variables to process last earnings end date in the array. 2. Create an array to process Earns Begin and Earns End date for California employees who have prior period earnings.
begin-SELECT A.COMPANY,A.PAYGROUP,A.PAY_BEGIN_DT,A.PAY_END_DT,A1.BALANCE_ID A1.BALANCE_YEAR,A1.BALANCE_PERIOD,A.CHECK_DT B.OFF_CYCLE,B.PAGE_NUM,B.LINE_NUM,B.SEPCHK B.FORM_ID,B.PAYCHECK_NBR,B.CHECK_DT B.EMPLID,B.EMPL_RCD,B.NAME,B.PAYCHECK_NAME,B,ADDRESS1,B.ADDRESS2,B.ADDRESS3 B.CITY,B.STATE,B.POSTAL B.PAYCHECK_OPTION,B.PAYCHECK_ADDR_OPTN B.TOTAL_GROSS,B.TOTAL_TAXES,B.TOTAL_DEDUCTIONS,B.NET_PAY B.PAY_SHEET_SRC, B.BUSINESS_UNIT G.EFFDT Y.TOTAL_GROSS_YTD,Y.TOTAL_TAXES_YTD,Y.TOTAL_DEDNS_YTD,Y.NET_PAY_YTD add #CheckNetPay to #TotalNetPay	begin-SELECT A.COMPANY,A.PAYGROUP,A.PAY_BEGIN_DT,A.PAY_END_DT,A1.BALANCE_ID A1.BALANCE_YEAR,A1.BALANCE_PERIOD,A.CHECK_DT B.OFF_CYCLE,B.PAGE_NUM,B.LINE_NUM,B.SEPCHK B.FORM_ID,B.PAYCHECK_NBR,B.CHECK_DT B.EMPLID,B.EMPL_RCD,B.NAME,B.PAYCHECK_NAME,B,ADDRESS1,B.ADDRESS2,B.ADDRESS3 B.CITY,B.STATE,B.POSTAL B.PAYCHECK_OPTION,B.PAYCHECK_ADDR_OPTN B.TOTAL_GROSS,B.TOTAL_TAXES,B.TOTAL_DEDUCTIONS,B.NET_PAY B.PAY_SHEET_SRC, B.BUSINESS_UNIT G.EFFDT Y.TOTAL_GROSS_YTD,Y.TOTAL_TAXES_YTD,Y.TOTAL_DEDNS_YTD,Y.NET_PAY_YTD let $CA_Earns_Found = 'N' !1214062594 do Find-CA-Earns !1214062594 add #CheckNetPay to #TotalNetPay do Delete-Wrk_Pyern_Prc !1214062594	begin-procedure Get-Ee-Job-Data 1. Determine if there are any Employees who have California earnings. 2. Refers to a procedure used to clear temporary work table used to sort California Employees earnings.
!*** begin-procedure Get-Ee-Job-Data !*** begin-SELECT J.DEPTID,J.JOBCODE,J.LOCATION,J.COMPRATE P.NAME,P.ADDRESS1,P.ADDRESS2,P.CITY,P.STATE,P.POSTAL J.BUSINESS_UNIT J.SETID_JOBCODE FQ.FREQUENCY_TYPE	!*** begin-procedure Get-Ee-Job-Data !*** begin-SELECT J.DEPTID,J.JOBCODE,J.LOCATION,J.COMPRATE P.NAME,P.ADDRESS1,P.ADDRESS2,P.CITY,P.STATE,P.POSTAL J.BUSINESS_UNIT J.SETID_JOBCODE J.EMPL_TYPE !1214062594 FQ.FREQUENCY_TYPE	begin-procedure Get-Ee-Job-Data 1. Empl_Type is used to select non-salary California employees.

	!Begin-1214062594
	!**********
	begin-procedure Find-CA-Earns
	!**********
	
	begin-SELECT
	'X'
	
	  let $CA_Earns_Found = 'Y'
	
	FROM PS_PAY_EARNINGS EA
	WHERE EA.COMPANY = $A.Company
	  AND EA.PAYGROUP = $A.PayGroup
	  AND EA.PAY_END_DT = $A.Pay_End_Dt
	  AND EA.OFF_CYCLE = $B.Off_Cycle
	  AND EA.PAGE_NUM = $B.PAGE_NUM
	  AND EA.LINE_NUM = $B.LINE_NUM
	  AND EA.SEPCHK = $B.SepChk
	  AND EA.SINGLE_CHECK_USE IN ('C', 'N')
	  AND EA.STATE = 'CA'
	end-SELECT
	
	  if $CA_Earns_Found = 'Y'
	  do Find-Min-Max-Earn-Dts
	  end-if
	
	end-procedure
		begin-procedure Find-CA-Earns
		1. New procedure used to determine if there are any employees who have California Earnings.

	!**********
	begin-procedure Find-Min-Max-Earn-Dts
	!**********
	
	begin-SELECT
	MAX(MME.EARNS_END_DT) &MME_Earns_End
	MIN(MME.EARNS_BEGIN_DT) &MME_Earns_Begin
	
	FROM PS_PAY_EARNINGS MME
	WHERE MME.COMPANY = $A.Company
	  AND MME.PAYGROUP = $A.PayGroup
	  AND MME.PAY_END_DT = $A.Pay_End_Dt
	  AND MME.OFF_CYCLE = $B.Off_Cycle
	  AND MME.PAGE_NUM = $B.PAGE_NUM
	  AND MME.LINE_NUM = $B.LINE_NUM
	  AND MME.SEPCHK = $B.SepChk
	  AND MME.SINGLE_CHECK_USE IN ('C', 'N')
	  AND MME.STATE = 'CA'
	  AND MME.EARNS_END_DT <= $A.PAY_BEGIN_DT
	end-SELECT
	
	  move &MME_Earns_Begin to $lastEarnBegin
	  move &MME_Earns_End to $lastEarnEnd
	
	end-procedure
	
	!End-1214062594
		begin-procedure Find-Min-Max-Earn-Dts
		1. Select minimum and maximum earnings begin date for summarized prior pay period earnings.
		2. These are the dates shown on the paycheck when "Other" earns are comprised of all prior period earnings.

```
begin-procedure Get-Earnings-Detail
!*************************************************************

begin-SELECT
! E.HOURLY_RT
E.REG_HRS
E.REG_HRLY_EARNS
E.OT_HRS
E.OT_HRLY_EARNS
E.REG_EARNS
E.COMP_RATECD_REG
E.COMP_RATECD_OT
E.COMPRATE_USED_REG
E.COMPRATE_USED_OT
E.ADDL_NBR
E.RATE_USED
E.SHIFT_RT
```

```
!*************************************************************
begin-procedure Get-Earnings-Detail
!*************************************************************

!Begin-1214062594
  if $CA_Earns_Found = 'Y'
     and &J.EMPL_TYPE <> 'S'
     let $E_Paysheet_Src = 'AND E.PAY_SHEET_SRC <> ''F'''
     let $E_OrderBy = 'ORDER BY E.EARNS_END_DT DESC, E.EARNS_BEGIN_DT DESC'
  else
     let $E_Paysheet_Src = ''
     let $E_OrderBy = 'ORDER BY E.ADDL_NBR'
  end-if
!End-1214062594

begin-SELECT
! E.HOURLY_RT
E.REG_HRS
E.REG_HRLY_EARNS
E.OT_HRS
E.OT_HRLY_EARNS
E.REG_EARNS
E.COMP_RATECD_REG
E.COMP_RATECD_OT
E.COMPRATE_USED_REG
E.COMPRATE_USED_OT
E.ADDL_NBR
E.RATE_USED
E.SHIFT_RT
E.EARNS_BEGIN_DT                                    !1214062594
E.EARNS_END_DT                                      !1214062594

!Begin-1214062594
  if $CA_Earns_Found = 'Y'
     and &J.EMPL_TYPE <> 'S'
        if &E.EARNS_END_DT < &A.PAY_BEGIN_DT
           move &E.EARNS_BEGIN_DT     to $Earn_Begin_Dt
           move &E.EARNS_END_DT       to $Earn_End_Dt
        else
           move &A.PAY_BEGIN_DT       to $Earn_Begin_Dt
           move &A.PAY_END_DT         to $Earn_End_Dt
        end-if
  else
     let $Earn_Begin_Dt = ''
     let $Earn_End_Dt = ''
  end-if
!End-1214062594
```

begin-procedure Get-Earnings-Detail

1. If Hourly Employee has California earnings processes all earnings that are not defined by FLSA processing rules and orders the earnings by Earnings Begin and End Dates.
2. Added Earnings Begin Dt and Earnings End Dt to select statement to process California prior payroll earnings.
3. If hourly California Employee has prior period California Earnings, populates earnings date variables with Earnings Begin Dt and Earnings End Dt.
4. If hourly California Employee does not have prior period California Earnings populates earnings date variables with Pay Begin Dt and Pay End Dt.
5. For all other state earnings and Salaried California employee, populates earnings date variables with a null value.
6. Sets Earn Begin Date and Earn End Dt to null to ensure current employee value processing.
7. If Employee does not have California earnings or is a Salaried California employee, the earnings process defaults to current processing and the paycheck format.
8. Prepare California Earnings array to be sorted by earnings begin dt, earnings type, erncd, and hourly rate.

```
FROM  PS_PAY_EARNINGS E
WHERE E.COMPANY     = &A.Company
  AND E.PAYGROUP    = &A.PayGroup
  AND E.PAY_END_DT  = &A.Pay_End_Dt
  AND E.OFF_CYCLE   = &B.Off_Cycle
  AND E.PAGE_NUM      = &B.PAGE_NUM
  AND E.LINE_NUM      = &B.LINE_NUM
  AND E.SEPCHK      = &B.SepChk
  AND E.SINGLE_CHECK_USE IN ('C', 'N')

ORDER BY E.ADDL_NBR
end-SELECT

  do Get-Earning-Balances

end-procedure
```

```
FROM  PS_PAY_EARNINGS E
WHERE E.COMPANY     = &A.Company
  AND E.PAYGROUP    = &A.PayGroup
  AND E.PAY_END_DT  = &A.Pay_End_Dt
  AND E.OFF_CYCLE   = &B.Off_Cycle
  AND E.PAGE_NUM      = &B.PAGE_NUM
  AND E.LINE_NUM      = &B.LINE_NUM
  AND E.SEPCHK      = &B.SepChk
  AND E.SINGLE_CHECK_USE IN ('C', 'N')
[$E_Paysheet_Src]
[$E_Orderby]
end-SELECT

!Begin-1214062594
  let $Earn_Begin_Dt = ''
  let $Earn_End_Dt = ''

  if $CA_Earns_Found = 'Y'
    do Process-Earnings
  end-if
!End-1214062594

  do Get-Earning-Balances

end-procedure
```

```
!reset ETDarray
  move 0 to #i
  while #i <= #lastRow
    move 0 to #j
    while #j <= #lastCol
      put ' ' ' ' ' ' 0.0 into ETDarray(#i) Code(#j) Name(#j) CapRtCd(#j) Cur(#j) YT

    add 1 to #j
    end-while
    add 1 to #i
  end-while

!reset vacation totals
  move 0 to #VacStartBal
  move 0 to #VacEarned
  move 0 to #VacBought
  move 0 to #VacTaken
  move 0 to #VacSold
  move 0 to #VacAdjust
  move 0 to #VacEndBal

!reset non-crossfoot message switch
  move 'N' to $NeedXfootMsg


!reset current and YTD recalc totals
  move 0 to #HourCurTotal
  move 0 to #HourYTDTotal
  move 0 to #PreTaxDedCurTotal
  move 0 to #PreTaxDedYTDTotal
  move 0 to #PostTaxDedCurTotal
  move 0 to #PostTaxDedYTDTotal
  move 0 to #FedTaxGrossCur
  move 0 to #FedTaxGrossYTD

!reset net pay distribution totals
  move 0 to #CheckNetPay
  move 0 to #DepositNetPay


end-procedure
```

```
!reset ETDarray
  move 0 to #i
  while #i <= #lastRow
    move 0 to #j
    while #j <= #lastCol
      put                    0.0 into ETDarray(#i) Code(#j) Name(#j)      !121
        Earn_Begin_Dt(#j) Earn_End_Dt(#j) CapRtCd(#j) Cur(#j) YTD(#j)      !121
      add 1 to #j
    end-while
    add 1 to #i
  end-while

!reset vacation totals
  move 0 to #VacStartBal
  move 0 to #VacEarned
  move 0 to #VacBought
  move 0 to #VacTaken
  move 0 to #VacSold
  move 0 to #VacAdjust
  move 0 to #VacEndBal

!reset non-crossfoot message switch
  move 'N' to $NeedXfootMsg
  move 'N' to $Print_Other_Msg                    !1214062594


!reset current and YTD recalc totals
  move 0 to #HourCurTotal
  move 0 to #HourYTDTotal
  move 0 to #PreTaxDedCurTotal
  move 0 to #PreTaxDedYTDTotal
  move 0 to #PostTaxDedCurTotal
  move 0 to #PostTaxDedYTDTotal
  move 0 to #FedTaxGrossCur
  move 0 to #FedTaxGrossYTD

!reset net pay distribution totals
  move 0 to #CheckNetPay
  move 0 to #DepositNetPay

!reset California Employees values
      let $Earn_Begin_Dt = ''                     !1214062
      let $Earn_End_Dt = ''                       !1214062
      do Delete-Wrk_Pyern_Prc                     !1214062

end-procedure
```

begin-procedure Reset-Variables

1. Added Earns Begin and Earns End date to reset array variables used to process California Wages earned for current payroll processing.
2. Reset Print Other Msg Variable used to determine whether to print cross foot message for hourly California Employee's Prior Pay Period earnings which has reached the maximum of 13 earnings and has summarized additional prior earnings in to one amount with a description of "Other" .
3. Resets Earn Begin Dt and Earn End Dr to null.
4. Refers to a procedure used to clear temporary work table used to sort California Employees earnings.

```
    move #RatesRow to #i
    move 0 to #j
    while #j <= #lastCol
      get $jCode $jCmpRtCd #jRate from ETDarray(#i) Code(#j) CmpRtCd(#j) Cur(#j)



    if rtrim($jCode,' ') = ''                    !if array entry is null
      break                                      !exit the "while"
    else
      if $Code = $jCode and $CmpRtCd = $jCmpRtCd and #Rate = #jRate    !if array e


        break                                                  !exit the "
      else
        if #j = #lastCol


          move $lastCode to $Code
          move $lastName1 to $Name
          move $lastCmpRtCd to $CmpRtCd
          move #lastRate to #Rate
          break                              !exit the "while"


        end-if
      end-if
    end-if
    add 1 to #j
  end-while

  while #i <= #EarnsRow
    evaluate #i
      when = #RatesRow
        move #Rate  to #Cur
      when = #HoursRow
        move #Hours to #Cur
      when = #EarnsRow
        move #Earns to #Cur
      when-other
        move 0      to #Cur
    end-evaluate

    put $Code $Name $CmpRtCd into ETDarray(#i) Code(#j) Name(#j) CmpRtCd(#j)
```

```
    move #RatesRow to #i
    move 0 to #j
    while #j <= #lastCol
!Begin-1214062594
      get $jCode $jEarnBegin $jEarnEnd $jCmpRtCd #jRate -
        from ETDarray(#i) Code(#j) Earn_Begin_Dt(#j) Earn_End_Dt(#j) CmpRtCd(#j) Cu

!End-1214062594
      if rtrim($jCode,' ') = ''                                        !if array
        break                                                          !exit the
      else
        if $Code = $jCode and $CmpRtCd = $jCmpRtCd and #Rate = #jRate      !if array
           and $Earn_Begin_Dt = $jEarnBegin                                !121
           and $Earn_End_Dt = $jEarnEnd                                    !121
          break                                                        !exit the
        else
          if #j = #lastCol
            if $Earn_End_Dt = $A_PAY_END_DT                                !121
              move $A_PAY_END_DT to $lastEarnEnd                           !121
            end-if                                                         !121
                                        move $lastCode      to $Code
                                        move $lastName1     to $Name
                                        move $lastCmpRtCd   to $CmpRtCd
                                        move #lastRate      to #Rate
                                        move $lastEarnBegin to $Earn_Begin_Dt
                                        move $lastEarnEnd   to $Earn_End_Dt
            break
          end-if
        end-if
      end-if
      add 1 to #j
    end-while

    while #i <= #EarnsRow
      evaluate #i
        when = #RatesRow
          move #Rate  to #Cur
        when = #HoursRow
          move #Hours to #Cur
        when = #EarnsRow
          move #Earns to #Cur
        when-other
          move 0      to #Cur
      end-evaluate

      put $Code $Name $Earn_Begin_Dt $Earn_End_Dt $CmpRtCd -
        into ETDarray(#i) Code(#j) Name(#j) Earn_Begin_Dt(#j) Earn_End_Dt(#j) CmpRt
```

begin-procedure Update-ETDarray-Current-Earnings

1. Added Earns Begin and Earns End date to update earnings array variables used to process hourly California Wages earned for current payroll processing.
2. Added logic to process Hourly California wages when "Other" is a summation of both current and prior pay period earnings. The $LastEarnEnd is now default to Pay Period End Date instead of Earnings End Dt.

```
!*************************************************************************
begin-procedure Read-ETDarray
!*************************************************************************

  move 0 to #i
  while #i <= #lastRow
    get $Name $CmpRtCd #Cur #YTD from ETDarray(#i) Name(#j) CmpRtCd(#j) Cur(#j) YTD

    evaluate #i
      when = #RatesRow
        move $Name to $EarnName
        move $CmpRtCd to $CompCode
        move #Cur  to #EarnRate


      when = #HoursRow
        move #Cur  to #HourCur
        move #YTD  to #HourYTD
        add  #Cur  to #HourCurTotal
```

```
!*************************************************************************
begin-procedure Read-ETDarray
!*************************************************************************

  move 0 to #i
  while #i <= #lastRow
    get $Name $Earn_Begin_Dt $Earn_End_Dt $CmpRtCd #Cur #YTD from -
      ETDarray(#i) Name(#j) Earn_Begin_Dt(#j) Earn_End_Dt(#j) CmpRtCd(#j) Cur(#j)
    evaluate #i
      when = #RatesRow
        move $Name to $EarnName
        move $CmpRtCd to $CompCode
        move #Cur  to #EarnRate
        move $Earn_Begin_Dt to $Prv_EarnBegin
        move $Earn_End_Dt   to $Prv_EarnEnd
      when = #HoursRow
        move #Cur  to #HourCur
        move #YTD  to #HourYTD
        add  #Cur  to #HourCurTotal
```

begin-procedure Read-ETDarray

1. Added Earns Begin and Earns End date to read earnings array variables used to process California Wages earned for current payroll processing.

```
print 'Rate'                (+1,25)

print '-'                   (+1,34,16) fill
print 'Current'             (,44)
print '-'                   (,50,16)   fill
print '-'                   (,64,12)   fill
print 'YTD'                 (,71)
print '-'                   (,75,10)   fill
print 'Description'         (+1,2)
print 'Code'                (,25)
print 'Rate'                (,34)
print 'Hours'               (,45)
print 'Earnings'            (,53)
print 'Hours'               (,64)
print ' Earnings'           (,74)
```

```
columns 2 25 32 44 51 63 71  86 105 117  2 24 33  44 66 75  86 108 117
```

```
move 0 to #j
while #j <= {n}
  if #j = {n}                !totals lines
    move 'Total:'            to $EarnName

    move 0                   to #EarnRate
    move #HourCurTotal       to #HourCur
    move #HourYTDTotal       to #HourYTD
    move &B.Total_Gross      to #EarnCur
    move &Y.Total_Gross_YTD  to #EarnYTD
```

```
!Begin-1214062594
  if $CA_Earns_Found = 'Y'
   and &J.EMPL_TYPE <> 'S'
    print '-'                 (+1,14,4) fill
    print 'Current/Prior Period'  (,17)
    print '-'                 (,31,3) fill
    print '-'                 (,37,16) fill
    print 'Current'           (,46)
    print '-'                 (,52,16)  fill
    print '-'                 (,66,12)  fill
    print 'YTD'               (,73)
    print '-'                 (,77,10)  fill

    print 'Description'       (+1,2)
    print 'Begin Date'        (,14)
    print 'End Date'          (,25)
    print 'Rate'              (,37)
    print 'Hours'             (,46)
    print 'Earnings'          (,55)
    print 'Hours'             (,66)
    print ' Earnings'         (,75)
  else
    print '-'                 (+1,34,16) fill
    print 'Current'           (,44)
    print '-'                 (,50,16)   fill
    print '-'                 (,64,12)   fill
    print 'YTD'               (,71)
    print '-'                 (,75,10)   fill
    print 'Description'       (+1,2)

    print 'Rate'              (,34)
    print 'Hours'             (,45)
    print 'Earnings'          (,53)
    print 'Hours'             (,64)
    print ' Earnings'         (,74)
  end-if
!End-1214062594
```

```
!Begin-1214062594
  if $CA_Earns_Found = 'Y'
   and &J.EMPL_TYPE <> 'S'
    columns 2 14 25 33 44 52 64 72  86 105 117  2 24 33  44 66 75  86 100 117
  else
    columns 2 25 32 44 51 63 71  86 105 117  2 24 33  44 66 75  86 108 117
  end-if
!End-1214062594
```

```
move 0 to #j
while #j <= {n}
  if #j = {n}                !totals lines
    move 'Total:'            to $EarnName
    move ''                  to $Prt_EarnBegin          !1214062594
    move ''                  to $Prt_EarnEnd            !1214062594
    move 0                   to #EarnRate
    move #HourCurTotal       to #HourCur
    move #HourYTDTotal       to #HourYTD
    move &B.Total_Gross      to #EarnCur
    move &Y.Total_Gross_YTD  to #EarnYTD
```

begin-procedure Print-Statement

1. If an Hourly Employee has California earnings, modified Print Statement to Print Earnings Begin and Earnings End Date for Employees.
2. If Employee does not have California earnings defaults to original pay earnings Print Statement.
3. If a California Employee who has Prior Pay Period Earnings has reached the maximum of 13 earnings, all additional prior pay period earnings will summarized in to one amount with a description of "Other". The following message will appear on the cross foot notes section of the Pay Check. *"NOTE: "Other" can be summarization of both current/prior period earnings. Please contact the Payroll Administrator for a detailed list of other prior period earnings."*
4. If Earnings Name Exceeded 14 characters in length, modified Earnings Name to prevent to the description from overwriting Earnings Begin Date.

```
print $EarnName                 (#row1,1)


! print $CompCode                 ()  +1


next-column
if #HourCur <> 0
  print #EarnRate               () edit b999.999999mi  ! change to 12,6 if needed
end-if
next-column
print #HourCur                  () edit b999.99mi
next-column
if rtrim($EarnName,' ') <>
  print #EarnCur                () edit 9,999,990.99mi
end-if
next-column
print #HourYTD                  () edit b9,999.99mi
next-column
if rtrim($EarnName,' ') <>
  print #EarnYTD                () edit 999,999,990.99mi
end-if
```

```
!Begin-1214062594
    if $EarnName = 'Other'
      if $CA_Earns_Found = 'Y'
        and 4J.EMPL_TYPE <> 'S'
          move 'Y'          to $Print_Other_Msg
        end-if
    end-if


  let $length = length($EarnName)
  if $length > 14
     and $CA_Earns_Found = 'Y'
        and 4J.EMPL_TYPE <> 'S'
           let $EarnName = substr($EarnName,1,14)
           print $EarnName                (#row1,1)
  else
           print $EarnName                (#row1,1)
  end-if
!End-1214062594

!Begin-1214062594
  if $CA_Earns_Found = 'Y'
     and 4J.EMPL_TYPE <> 'S'
        do Format-DateTime($Prt_EarnBegin, $out, (DEFDATE), '', '')
        print $out
        next-column
        do Format-DateTime($Prt_EarnEnd, $out, (DEFDATE), '', '')
        print $out
        next-column
      if #HourCur <> 0
        print #EarnRate             () edit b999.999999mi  ! change to 12,6 if needed
      end-if
      next-column
      print #HourCur                () edit b999.99mi
      next-column
      if rtrim($EarnName,' ') <>
      print #EarnCur                () edit 9,999,990.99mi
      end-if
      next-column
      print #HourYTD                () edit b9,999.99mi
      next-column
      if rtrim($EarnName,' ') <>
        print #EarnYTD              () edit 999,999,990.99mi
      end-if
  else
      next-column
      if #HourCur <> 0
        print #EarnRate             () edit b999.999999mi  ! change to 12,6 if needed
      end-if
      next-column
      print #HourCur                () edit b999.99mi
      next-column
      if rtrim($EarnName,' ') <>
      print #EarnCur                () edit 9,999,990.99mi
      end-if
      next-column
      print #HourYTD                () edit b9,999.99mi
      next-column
      if rtrim($EarnName,' ') <>
        print #EarnYTD              () edit 999,999,990.99mi
      end-if
  end-if
!End-1214062594

!Begin-1214062594
      if $Print_Other_Msg = 'Y'
            let #row = 49
            columns 30
                  print 'NOTE: "Other" can be summarization of both current/
                  print 'period earnings.  Please contact the Payroll Admini
                  print 'for a detailed list of other prior period earnings'
      end-if
!End-1214062594
```

```
!Begin-1214062594
!*********************************************************************
begin-procedure Delete-Wrk_Pyern_Prc
!*********************************************************************

begin-SQL on-error=Ignore
DELETE FROM PS_WRK_PYERN_PRC
WHERE PROCESS_INSTANCE = #prcs_process_instance
end-SQL

end-procedure
```

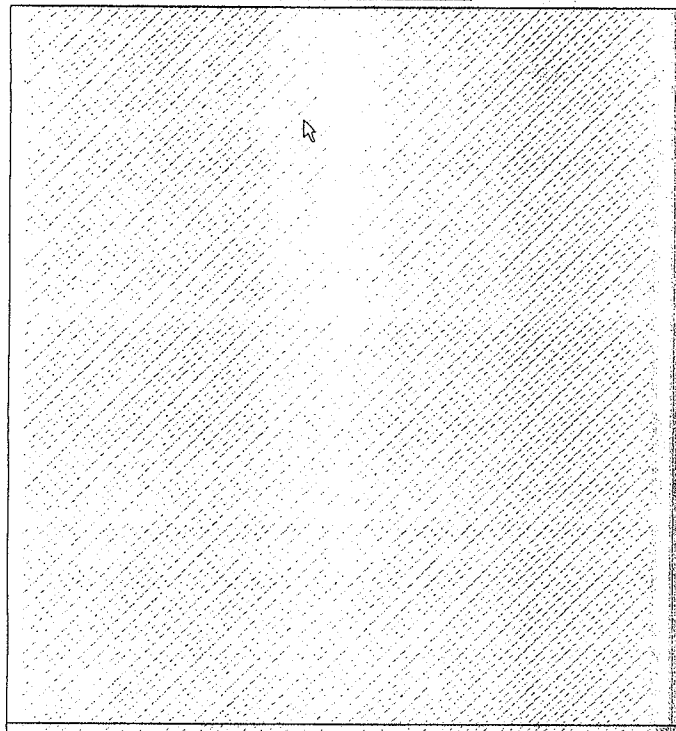
begin-procedure Delete-Wrk_Pyern_Prc

1. Procedure used to clear temporary work table used to sort California Employees earnings.

```
!*********************************************************************
begin-procedure Process-Earnings
!*********************************************************************

  move 0 to #j
  while #j <= #lastCol
    get $jCode $jName #jEarnBegin #jEarnEnd $jCmpRtCd #jRate -
      from ETDarray(0) Code(#j) Name(#j) Earn_Begin_Dt(#j) Earn_End_Dt(#j) CmpR
    if rtrim($jCode,' ') = ''                    !if array entry is null
      break                                      !exit the "while"
    else
      get #jEarns_Hrs
            from ETDarray(1) Cur(#j)
      get #jEarns_Amt
            from ETDarray(2) Cur(#j)
      evaluate $jCode
        when = $PG.ErnCd_Reg_Hrs
          move '1'   to $jEarns_Type    !Regular Hourly
          break
        when = $PG.ErnCd_Reg_Earns
          move '2'   to $jEarns_Type    !Regular Salary
          break
        when = $PG.ErnCd_OT_Hrs
          move '3'   to $jEarns_Type    !Overtime
          break
        when = '***'                                          !1214062594
          move '9'   to $jEarns_Type    !Other Earnings       !1214062594
          break                                               !1214062594
        when-other
          move '4'   to $jEarns_Type
          break
      end-evaluate
      do Process-Earns-Wrk
    end-if
    add 1 to #j
  end-while

!reset ETDarray
  move 0 to #i
  while #i <= #lastRow
    move 0 to #j
    while #j <= #lastCol
      put                    0 0 into ETDarray(#i) Code(#j) Name(#j) -
        Earn_Begin_Dt(#j) Earn_End_Dt(#j) CmpRtCd(#j) Cur(#j) YTD(#j)

      add 1 to #j
    end-while
    add 1 to #i
  end-while

  do Process-Earnings-Array

end-procedure
```

begin-procedure Process-Earnings

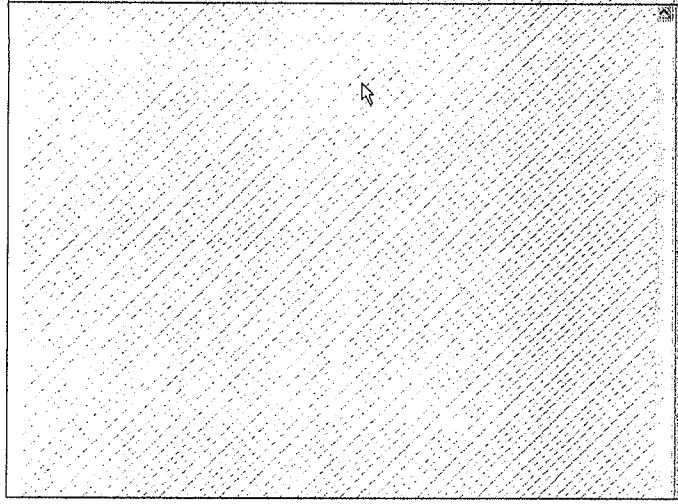
1. Procedure used to get Earnings from array to be sorted by Earnings Type. i.e. Regular Hourly, Regular Salary, Regular Overtime and other earnings.



```
!***********************************************************************
begin-procedure Process-Earns-Wrk
!***********************************************************************

  let $Earns_Wrk_Found = 'N'

  if IsNull(rtrim($jEarnBegin, ' '))
    let $jEarnBegin = &A.PAY_BEGIN_DT
  end-if

  if IsNull(rtrim($jEarnEnd, ' '))
    let $jEarnEnd = &A.PAY_END_DT
  end-if

  if IsNull(rtrim($jCmpRtCd, ' '))
    let $jCmpRtCd = ' '
  end-if

begin-SELECT DISTINCT
WRK.PROCESS_INSTANCE

  let $Earns_Wrk_Found = 'Y'

FROM PS_WRK_PYERN_PRC WRK
WHERE WRK.PROCESS_INSTANCE = #prcs_process_instance
AND WRK.EARNS_TYPE = $jEarns_Type
AND WRK.ERNCD = $jCode
AND WRK.EARNS_BEGIN_DT = $jEarnBegin
AND WRK.EARNS_END_DT = $jEarnEnd
AND WRK.COMP_RATECD = $jCmpRtCd
AND WRK.HOURLY_RT = #jRate
end-SELECT

  if $Earns_Wrk_Found = 'N'
    do Insert-Earns-Wrk
  else
    do Update-Earns-Wrk
  end-if

end-procedure
```

begin-procedure Process-Earns-Wrk

1. Sets Default values for Earnings Begin Dt, Earnings End Dt and comp rate code which may be null. Cannot insert null values in to temporary work table.
2. Procedure used to determine if California Earnings has been processed previously for an hourly employee.
3. If California earnings exist for an earnings code, will add amounts and hours and update values.
4. If California earnings do not exist will insert new earnings.



```
!***********************************************************************
begin-procedure Insert-Earns-Wrk
!***********************************************************************

begin-SQL
INSERT INTO PS_WRK_PYERN_PRC
(PROCESS_INSTANCE,
EARNS_TYPE,
ERNCD,
EARNS_BEGIN_DT,
EARNS_END_DT,
COMP_RATECD,
HOURLY_RT,
DESCR,
EARNS_HRS,
EARNS_AMT)
VALUES(#prcs_process_instance ,
$jEarns_Type,
$jCode,
$jEarnBegin,
$jEarnEnd,
$jCmpRtCd,
#jRate,
$jName,
#jEarns_Hrs,
#jEarns_Amt)
end-SQL
end-procedure
```

begin-procedure Insert-Earns-Wrk

1. Inserts California Earnings in to Temporary table used to sort earnings.

```
!*****************************************************************
begin-procedure Update-Earns-Wrk
!*****************************************************************
begin-SQL
UPDATE PS_WRK_PYERN_PRC
SET EARNS_HRS = EARNS_HRS + #jEarns_Hrs,
    EARNS_AMT = EARNS_AMT + #jEarns_Amt
WHERE PROCESS_INSTANCE = #prcs_process_instance
AND EARNS_TYPE = $jEarns_Type
AND ERNCD = $jCode
AND EARNS_BEGIN_DT = $jEarnBegin
AND EARNS_END_DT = $jEarnEnd
AND COMP_RATECD = $jCmpRtCd
AND HOURLY_RT = #jRate
end-SQL
end-procedure
```

begin-procedure Update-Earns-Wrk

1. Updates California earnings for an existing earnings code.

```
!*****************************************************************
begin-procedure Process-Earnings-Array
!*****************************************************************

  move 0  to #j
begin-SELECT
EW.EARNS_TYPE
EW.ERNCD
EW.EARNS_BEGIN_DT
EW.EARNS_END_DT
EW.COMP_RATECD
EW.HOURLY_RT
EW.DESCR
EW.EARNS_HRS
EW.EARNS_AMT

  move 0 to #i
  while #i <= #EarnsRow
    evaluate #i
      when = 0
        put &EW.ERNCD &EW.DESCR &EW.EARNS_BEGIN_DT &EW.EARNS_END_DT
            &EW.COMP_RATECD &EW.HOURLY_RT
                into ETDarray(#i) Code(#j) Name(#j) Earn_Begin_Dt(#j) Earn_End_Dt(#
                     CmpRtCd(#j)) Cur(#j)
      when = 1
        put &EW.ERNCD &EW.DESCR &EW.EARNS_BEGIN_DT &EW.EARNS_END_DT
            &EW.COMP_RATECD &EW.EARNS_HRS
                into ETDarray(#i) Code(#j) Name(#j) Earn_Begin_Dt(#j) Earn_End_Dt(#
                     CmpRtCd(#j)) Cur(#j)
      when = 2
        put &EW.ERNCD &EW.DESCR &EW.EARNS_BEGIN_DT &EW.EARNS_END_DT
            &EW.COMP_RATECD &EW.EARNS_AMT
                into ETDarray(#i) Code(#j) Name(#j) Earn_Begin_Dt(#j) Earn_End_Dt(#
                     CmpRtCd(#j)) Cur(#j)
    end-evaluate
    add 1 to #i
  end-while

  add 1 to #j

FROM PS_WRK_PYERN_PRC EW
WHERE EW.PROCESS_INSTANCE = #prcs_process_instance
ORDER BY EW.EARNS_BEGIN_DT DESC,
                        EW.EARNS_TYPE ASC,
                        EW.ERNCD ASC,
          EW.HOURLY_RT ASC
end-SELECT

end-procedure
```

begin-procedure Process-Curr-Earns-Array

1. Selects earnings array California Earnings from temporary work table by earnings begin dt, earnings type, erncd, and hourly rate.Puts Earnings in to array based upon whether the earnings code has an hourly rate, hours earned or earnings amount.