

Final

Detailed Design
RELEASE 1218

Date Conversion Cleanup

November 20, 1998

Information Systems & Computing
Office of the President
University of California

Introduction

Original Service Request 12852 asks that modifications be made to the Payroll/Personnel System (PPS) to allow correct processing of dates falling between the 20th century (1900 through 1999) and the 21st century (2000 through 2999).

The century on any given date that requires a century value for comparison and/or calculation is set using the standard LE/370 date routines. Currently, the default century window in LE/370 is an 80-year window. Since we are currently in the year 98, two-digit years '18' through '99' are considered to be 1918 through 1999; two-digit years '00' through '17' are considered to be 2000 through 2017. Note that the window in LE/370 is set to a 99 year window for birthdate related data elements. Two digit years '98' through '99' are considered to be 1898 through 1899; two-digit years '00' through '98' are considered to be 1900 through 1998.

Phase I of the Date Conversion Project (Release 1025) completed modifications to all of the appropriate programs in the EDB File Maintenance Process. However, only those programs in the Compute Process that were affected by either copymember changes and/or Linkage changes were modified in Phase I. Appropriate programs in the other processes were also modified in Phase I.

Phase II of the Date Conversion Project (release 1087) completed modifications to the rest of the programs in the Compute Process not modified in Phase I. Programs in the other processes were also appropriately modified.

Phase III of the Date Conversion Project (release 1107) completed the modifications to the rest of the programs in the Payroll/Personnel System not modified in Phase I or Phase II.

Error Report 1589 states that a problem occurs in program PPFICRET. For example, the problem occurs when an employee's Appointment End Date is 02/29/2000. PPFICRET subtracts one from the year 2000 a leap year, and returns a result of year 1999. However, the year 1999 is not a leap year, and the last day of February should be 28. Without verifying that the date 02/29/1999 is valid, the appropriate Date Conversion routine is performed to convert 02/29/1999 to a julian date. Since, the date 02/29/1999 is not valid, the Date Conversion routine returned a zero julian date.

As part of Error Report 1589, the Date Conversion cleanup will modify the programs that recalculate the input dates used for date comparisons. Where the months, days, and/or years are calculated in the programs, the Date Conversion routine will be performed to verify that the number of days are within the range of the month and year. In addition, programs that contain a hard-coded table of Month-End dates in working storage, and referenced in the programs will be removed, and replaced with a call to the Date Conversion routine to obtain the last day for the appropriate month.

Program Modifications

Currently, the following programs contain logic which change the days, months and/or years in the original input dates. The sources of the original input dates are taken from the employee's EDB and/or input specification record. It is presumed that the input dates have been edited for valid dates.

PPP020

- ***UPDATE-CONTROL-REC-1000***

Currently, in this paragraph, the ***SCR Current Date*** is moved to a Work-Date. If the month in Work-Date is equal to a value of 12, a value of 1 is moved to the month and the year is incremented by 1 in Work-Date. Otherwise, the month in Work-Date is incremented by 1.

If the month in Work-Date is not equal to a value of 12 (December), and after the month in Work-Date is incremented by 1, the Date Conversion routine will be called to verify that the day value is within the range for the month in Work-Date. If the Date Conversion routine sets the date flag to indicate the date is invalid, it is presumed that the day value for the month in Work-Date is out of range; the appropriate Date Conversion routine will be called to get the last day for the month in Work-Date. The day returned will be moved to the day in Work-Date.

PPP130

- ***9520-BUILD-SCR-WORK***

Currently, in this paragraph, the ***SCR Process Date*** is moved to the Inactive Date field. For each iteration, if the number of Purge Months (from specification record) is greater than 12, the year in Inactive Date is subtracted by 1 and the number of Purge Months is subtracted by 12. If the number of Purge Months is greater than the month in Inactive-Date, the year in Inactive-Date is subtracted by 1 and the month in Inactive Date is incremented by 12. Finally, the number of Purge Months is subtracted from the month in Inactive-Date. However, the calculated Inactive Date is never referenced in the program.

Since the calculated Inactive Date is never referenced in the program, the logic that is associated with calculating the Inactive Date will be removed.

Final

PPP300

- ***DATE-MINUS-ONE-1820***

Currently, in this paragraph, the month in Work-Date is subtracted by 1. If the calculated month is equal to zero, a value of 12 is moved to the month in Work-Date and the Year in Work-Date is subtracted by one. There is hard-coded logic that determines the last day for the month in Work-Date.

The hard-coded logic that determines the last day for the month in Work-Date will be removed.

After the final calculation of the Work-Date, the Date Conversion routine will be called to get the last day for the month in Work-Date. The day returned will be moved to the day in Work-Date.

PPP390

- ***0500-INITIALIZATION***

Currently, in this paragraph, the Pay Period End Date is moved to Work-Date. If the month in Work-Date is less than 4, a value of 12 is added to the month in Work-Date and the year in Work-Date is subtracted by 1. Finally, a value of 3 is subtracted from the month in Work-Date, and Work-Date is moved to the PAY-END-DT-LESS-3-MO date field.

It is noted that the after the calculated Work-Date is moved to PAY-END-DT-LESS-3-MO, it is never referenced again. Therefore, the logic which is associated with calculating 3 months prior to the Pay Period End Date will be removed.

In addition, in this same paragraph, the Pay Period End Date is moved to Work-Date. If the day in Work-Date is greater than 16, a value of 16 is subtracted from the day in Work-Date. Otherwise the Day-Offset is calculated by subtracting 16 from the day in Work-Date. The month in Work-Date is subtracted by 1. If the month in Work-Date is equal to zero, a value of 12 is moved to the month in Work-date and 1 is subtracted from the year in Work-Date. There is hard-coded logic that determines the last day for the month in Work-Date. Finally, the Work-Date is moved to the PYCYCLE-END-DT-LESS-16DAYS date field.

The hard-coded logic that determines the last day for the month in Work-date will be removed.

It is noted that the after Work-Date is moved to PYCYCLE-END-DT-LESS-16DAYS, it is never referenced again. Therefore, the logic which is associated with calculating 16 days prior to the Pay Period End Date will be removed.

Final

PPP620

- ***VERIFY-SPEC-REC-1600***

Currently, in this paragraph, the date from the Specification record is edited. If the year is numeric, the month is greater than zero and less than 13, and the day is greater than zero and less than 32, the input date from the Specification record is valid. However, not all months have a maximum of 31 days .

The input date from the Specification record will be passed to the Date Conversion routine to edit the date. If the Date Conversion routine sets the date flag to indicate the date is invalid, existing message 62-103 will be issued.

- ***COMPUTE-LATEST-HIRE-DATE***

Currently, in this paragraph, if the month in LATEST-HIRE –DATE is greater than 10, the month in LATEST-HIRE-DATE is subtracted by 10. Otherwise, the year in LATEST-HIRE-DATE is subtracted by 1 and the month in LATEST-HIRE-DATE is incremented by 2.

After the final calculation of the LATEST-HIRE-DATE, the Date Conversion routine will be called to verify that the day value is within the range of the month in LATEST-HIRE-DATE. If the Date Conversion routine sets the date flag to indicate the date is invalid, it is presumed that the day value for the month is out of range; the appropriate Date Conversion routine will be called to get the last day of the month. The day returned will be moved to the day in LATEST-HIRE-DATE.

PPP680

- ***COMPUTE-SPEC-DATE-MAP-0201***

Currently, in this paragraph, if the month in the input date (from Specification record) is greater than 1, the month from the input date is subtracted by 1 and moved to the month in 1-MONTH-PRIOR-MAP date, and the year from the input date is moved to the year in 1-MONTH-PRIOR-MAP date. Otherwise a value of 12 (December) is moved to the 1-MONTH-PRIOR-MAP date, and the year from the input date is subtracted by 1 and moved to the 1-MONTH-PRIOR-MAP date. In addition, there is hard-coded logic that determines the last day for the month in 1-MONTH-PRIOR-MAP date.

The hard-coded logic that determines the last day for the appropriate month will be removed.

After the final calculation of the 1-MONTH-PRIOR-MAP date, the date Conversion routine will be called to get the last day for the month in the 1-MONTH-PRIOR-MAP

Final

date. The day returned from the Date Conversion routine will be moved to the day in the 1-MONTH-PRIOR-MAP date.

In addition, if the month in the input date (from Specification record) is greater than 1, the month from the input date is incremented by 5 and moved to the month in PAY-END-MO-1-MAP, and the year from the input date is moved to PAY-END-YR-1-MAP. Otherwise 12 (December) is moved to PAY-END-MO-1-MAP and the year from the input date is moved to PAY-END-YR-1-MAP. There is hard-coded logic that determines the last day for the month in PAY-END-MO-1-MAP.

The hard-coded logic that determines the last day for the month in PAY-END-YR-1-MAP date will be removed.

It is noted that after the calculation of the PAY-END-DATE-1-MAP date, the calculated date is never referenced in the program. Therefore, the logic that is associated with calculating the PAY-END-DATE-1-MAP date will be removed.

- ***COMPUTE-SPEC-DATE-APS-0202***

Currently, in this paragraph, if the month in the input date (from Specification record) is greater than 1, the month from the input date is subtracted by 1 and moved to the month in 1-MONTH-PRIOR-APS date, and the year from the input date is moved to the year in 1-MONTH-PRIOR-APS date. Otherwise a value of 12 (December) is moved to the 1-MONTH-PRIOR-APS date, and the year from the input date is subtracted by 1 and moved to the 1-MONTH-PRIOR-APS date. In addition, there is hard-coded logic that determines the last day for the month in 1-MONTH-PRIOR-APS date.

The hard-coded logic that determines the last day for the appropriate month will be removed.

After the final calculation of the 1-MONTH-PRIOR-APS date, the date Conversion routine will be called to get the last day for the month in the 1-MONTH-PRIOR-APS date. The day returned from the Date Conversion routine will be moved to the day in the 1-MONTH-PRIOR-APS date.

In addition, if the month in the input date (from Specification record) is greater than 1, the month from the input date is incremented by 5 and moved to the month in PAY-END-MO-1-APS, and the year from the input date is moved to PAY-END-YR-1-APS. Otherwise 12 (December) is moved to PAY-END-MO-1-APS and the year from the input date is moved to PAY-END-YR-1-APS. There is hard-coded logic that determines the last day for the month in PAY-END-MO-1-APS.

The hard-coded logic that determines the last day for the month in PAY-END-YR-1-APS date will be removed.

Final

It is noted that after the calculation of the PAY-END-DATE-1-APS date, the calculated date is never referenced in the program. Therefore, the logic that is associated with calculating the PAY-END-DATE-1-APS date will be removed.

- ***COMPUTE-SPEC-DATE-EXEC-0203***

Currently, in this paragraph, if the month in the input date (from Specification record) is greater than 1, the month from the input date is subtracted by 1 and moved to the month in 1-MONTH-PRIOR-EXEC date, and the year from the input date is moved to the year in 1-MONTH-PRIOR-EXEC date. Otherwise a value of 12 (December) is moved to the 1-MONTH-PRIOR-EXEC date, and the year from the input date is subtracted by 1 and moved to the 1-MONTH-PRIOR-EXEC date. In addition, there is hard-coded logic that determines the last day for the month in 1-MONTH-PRIOR-EXEC date.

The hard-coded logic that determines the last day for the appropriate month will be removed.

After the final calculation of the 1-MONTH-PRIOR-EXEC date, the date Conversion routine will be called to get the last day for the month in the 1-MONTH-PRIOR-EXEC date. The day returned from the Date Conversion routine will be moved to the day in the 1-MONTH-PRIOR-EXEC date.

In addition, if the month in the input date (from Specification record) is greater than 1, the month from the input date is incremented by 5 and moved to the month in PAY-END-MO-1-EXEC, and the year from the input date is moved to PAY-END-YR-1-EXEC. Otherwise 12 (December) is moved to PAY-END-MO-1-EXEC and the year from the input date is moved to PAY-END-YR-1-EXEC. There is hard-coded logic that determines the last day for the month in PAY-END-MO-1-EXEC.

The hard-coded logic that determines the last day for the month in PAY-END-YR-1-EXEC date will be removed.

It is noted that after the calculation of the PAY-END-DATE-1-EXEC date, the calculated date is never referenced in the program. Therefore, the logic that is associated with calculating the PAY-END-DATE-1-EXEC date will be removed.

PPP711

- ***B0000-INITIALIZE***

Currently, in this paragraph, the Severance Pay Process (SPP) Date is moved to a Work-Date. The value in (SPP Cycle –1) is subtracted from the month in Work-Date. If the month in Work-Date is less than 1, a value of 12 is moved to the month in Work-Date and 1 is subtracted from the year in Work-Date. There is hard-coded logic that determines the last day for the month in Work-Date. If the month in Work-Date is equal

Final

to 6 or 9, a day value of 30 is moved to the day in Work-Date. Otherwise a day value of 31 is moved to the day in Work-Date.

The hard-coded logic that determines the last day for the month in Work-Date will be removed.

The Date Conversion routine will be called to get the last day for the month in Work-Date. The day returned will be moved to the day in Work-Date.

PPP910

- ***J02000-CREATE-EDB-TRANS***

Currently, in this paragraph, the Input transaction Effective Date is moved to a Work-Date. The day in Work-Date is subtracted by 1. If the day in Work-Date is less than 1, the month in Work-Date is subtracted by 1, the year in Work-Date is subtracted by 1, 12 is moved to the month in Work-Date, and 31 is moved to the day in Work-Date. Otherwise, the last day for the month in Work-Date is taken from the hard-coded table containing the last day for the appropriate month in Work-Date

The hard-coded table containing the last day for each appropriate month will be removed.

The Date Conversion routine will be called to get the last day for the month in Work-Date. The day returned will be moved to the day in Work-Date.

PPPBELI

- ***31110-SET-12MO-DAY***

The employee's Appointment Begin Date is moved to Work-12-Month-End-Date. Paragraph 31100-CALC-12MO-END-DATE is performed. If the month and day in Work-12-Month-End-Date is equal to 1 and 1 respectively, month value of 12 and day value of 1 are moved to the month and day in Work-12-Month-End-Date respectively. Otherwise the year in Work-12-Month-End-Date is subtracted by 1, and paragraph 31110-SET-12MO-DAY is performed. Currently in this paragraph, if the day in Work-12-Month-End-Date is equal to 1, the month in Work-12-Month-End-Date is subtracted by 1, and the last day for the calculated month in Work-12-Month-End-date is taken from the hard-coded table containing the last day for the appropriate month in Work-Date.

The hard-coded table containing the last day for each appropriate month will be removed.

Final

The Date Conversion routine will be called to get the last day for the month in Work-12-Month-End-Date. The day returned will be moved to the day in Work-12-Month-End-Date.

- ***31210-SET-03MO-DAY***

The employee's Appointment Begin Date is moved to Work-03-Month-End-Date. Paragraph 31200-CALC-03MO-END-DATE is performed. If the month and day in Work-03-Month-End-Date is equal to 1 and 1 respectively, month value of 12 and day value of 1 are moved to the month and day in Work-03-Month-End-Date respectively. Otherwise, if the month in Work-03-Month-End-Date is greater than 9, the year in Work-03-Month-End-Date is incremented by 1, the month in Work-03-Month-End-Date is subtracted by 9 and paragraph 31110-SET-12MO-DAY is performed. If the month in Work-03-Month-End-Date is not greater than 9, the month in Work-03-Month-End-Date is incremented by 3 and paragraph 31210-SET-03MO-DAY is performed. Currently in this paragraph, if the day in Work-03-Month-End-Date is equal to 1, the month in Work-03-Month-End-Date is subtracted by 1, and the last day for the calculated month in Work-03-Month-End-Date is taken from the hard-coded table containing the last day for the appropriate month in Work-03-Month-End-Date. However, if the day in Work-03-Month-End-Date is not equal to a value of 1, the day in Work-03-Month-End-Date is subtracted by 1.

The hard-coded table containing the last day for each appropriate month will be removed.

The Date Conversion routine will be called to get the last day for the month in Work-03-Month-End-Date. The day returned will be moved to the day in Work-03-Month-End-Date. In addition, after subtracting the day in Work-03-Month-End-Date by 1, the Date Conversion routine will be called to verify that the day value is within the range of the month. If the Date Conversion routine sets the date flag to indicate the date is invalid, it is presumed that the day value for the month is out of range; the appropriate Date Conversion routine will be called to get the last day for the month in Work-03-Month-End-Date. The day returned will be moved to the day in Work-03-Month-End-Date.

- ***34100-CHECK-ADJACENT-DATES***

Currently, in this paragraph, the day in Work-Appointment-Begin-Date is incremented by 1. The calculated day in Work-Appointment-Begin-Date is compared with the day in the hard-code table containing the maximum day for each appropriate month. If the day in Work-Appointment-Begin-Date is greater than the maximum day in the hard-coded table for the appropriate month, the month in Work-Appointment-Begin-Date is incremented by 1 and the day in Work-Appointment-Begin-Date is set to a value of 1.

The hard-coded table containing the last day for each appropriate month will be removed.

Final

The Date Conversion routine will be called to verify that the day is within the range of the month in Work-Appointment-Begin-Date. If the Date Conversion routine sets the date flag to indicate the date is invalid, it is presumed that the day for the month in Work-Appointment-Begin-Date is out of range; the month in Work-Appointment-Begin-Date will be incremented by 1 and the day in Work-Appointment-Begin-Date will be set to a value of 1.

PPFICRET

- ***42110-CONTINUING-EMPLOYEE***

Currently, in this paragraph, the employee's Appointment End Date is moved to a Work-Appointment-End-Date. The year in the Work-Appointment-End-Date is subtracted by 1. Finally, the Date Conversion routine is called to convert the Work-Appointment-End-Date to a lillian date.

Since only the year in the Work-Appointment-End-Date is incremented, and not the month and/or day, code will be added to edit the month in the Work-Appointment-End-Date for a value of 2 (February) and the day in the Work-Appointment-End-Date for a value of 29. If both conditions are true, the day in the Work-Appointment-End-Date will be changed to a value of 28; leap year cannot occur two years in a row.

- ***42120-HIRE-REHIRE***

Currently, in this paragraph, the employee's Appointment Begin Date is moved to a Work-Appointment-Begin-Date. The year in the Work-Appointment-Begin-Date is incremented by 1. Finally, the Date Conversion routine is called to convert the Work-Appointment-Begin-Date to a lillian date.

Since only the year in the Work-Appointment-Begin-Date is incremented, and not the month and/or day in the Work-Appointment-Begin-Date, code will be added to check the month in the Work-Appointment-Begin-Date for a value of 2 (February) and the day in the Work-Appointment-Begin-date for a value of 29. If both conditions are true, the day in the Work-Appointment-Begin-Date will be changed to a value of 28, since leap year cannot occur two years in a row.

PPGRSAUP

- ***3000-INITIALIZATION***

Program PPGRSAUP is called from program PPP390.

Currently, in this paragraph, the PCR-End-Date is moved to Work-Date. If the month in Work-Date is less than 4, the month in Work-Date is incremented by 12 and the year in Work-Date is subtracted by 1. Finally, the month in Work-Date is subtracted by 3.

Final

The Date Conversion routine will be called to verify that the day value is within the range of the appropriate month. If the Date Conversion routine sets the date flag to indicate the date is invalid, it is presumed that the day value for the month is out of range; the appropriate Date Conversion routine will be called to get the last day for the month in Work-Date. The day returned will be moved to Work-Date.

- ***9100-DATE-MINUS-ONE***

Before this paragraph is performed, an employee's Leave Begin Date is moved to Work-Date. Currently, in this paragraph, if the day in Work-Date is greater than 1, the day in Work-Date is subtracted by 1, and the rest of the logic is bypassed. Otherwise, the month in Work-Date is subtracted by 1. If the month in Work-Date is equal to zero, a value of 12 is moved to the month in Work-Date, and the year in Work-Date is subtracted by 1.

PPEM109

- ***0100-INITIALIZE***

Currently, in this paragraph, the ***SCR Current Date*** is moved to a Work-Date. If the month in Work-Date is equal to a value of 12, a value of 1 is moved to the month and the year is incremented by 1 in Work-Date. Otherwise, the month in Work-Date is incremented by 1.

If the month in Work-Date is not equal to a value of 12 (December), and after the month in Work-Date is incremented by 1, the Date Conversion routine will be called to verify that the day value is within the range for the month in Work-Date. If the Date Conversion routine sets the date flag to indicate the date is invalid, it is presumed that the day value for the month in Work-Date is out of range; the appropriate Date Conversion routine will be called to get the last day for the month in Work-Date. The day returned will be moved to the day in Work-Date.

PPEM117

- ***0100-INITIALIZE***

Currently, in this paragraph, the ***SCR Current Date*** is moved to a Work-Date. If the month in Work-Date is equal to a value of 12, a value of 1 is moved to the month and the year is incremented by 1 in Work-Date. Otherwise, the month in Work-Date is incremented by 1.

If the month in Work-Date is not equal to a value of 12 (December), and after the month in Work-Date is incremented by 1, the Date Conversion routine will be called to verify that the day value is within the range for the month in Work-Date. If the Date Conversion routine sets the date flag to indicate the date is invalid, it is presumed that the day value for the month in Work-Date is out of range; the appropriate Date

Final

Conversion routine will be called to get the last day for the month in Work-Date. The day returned will be moved to the day in Work-Date.

System Control Table Changes

- System Messages Table

Currently, Program calls module PPMSGUT2 which returns the appropriate enhanced severity level for each error message. However, only several of the error messages, associated with program PPP620, have not been converted to the enhanced severity levels.

All appropriate error messages referenced in program PPP620 will be converted to the enhanced severity levels.