## Version History

<table>
<thead>
<tr>
<th>Version #</th>
<th>Date</th>
<th>Revised By</th>
<th>Reason for Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>9/17/14</td>
<td>Mary Meyer</td>
<td>Initial Draft</td>
</tr>
<tr>
<td>1.1</td>
<td>10/9/14</td>
<td>Mary Meyer</td>
<td>Removed USER008 and USER012; added programs PPACTUTL, PPAPPUTL, PPEC108, PPEC703, PPEDBDET, PPEI505, PPFAU018, PPPCCUTL, PPP100</td>
</tr>
<tr>
<td>1.2</td>
<td>10/16/14</td>
<td>Mary Meyer</td>
<td>Added RITM0049965</td>
</tr>
<tr>
<td>1.3</td>
<td>10/17/14</td>
<td>Mary Meyer</td>
<td>Changed PPPI730 to PPI730; added description</td>
</tr>
</tbody>
</table>
Table of Contents

Version History ........................................................................................................................................................................ 1

1  Introduction ........................................................................................................................................................................... 3
2  Background ........................................................................................................................................................................... 3
3  Overview of System Modifications ................................................................................................................................... 5
4  Design Considerations ....................................................................................................................................................... 5
    4.1  Assumptions and Dependencies ................................................................................................................................... 5
6  Mainframe Design .............................................................................................................................................................. 8
    6.1.1  Cobol Programs ................................................................................................................................................... 8
7  Unit Testing Requirements ............................................................................................................................................... 9
1 Introduction

Service Now Ticket - RITM0047174

RITM0047174 requests that programs containing SQL statements ‘Select *’ be changed to list the column names and host variables and the insert statements using ‘from’ also be changed to list the column names and values.

Service Now Ticket – RITM0048039

RITM0048039 requests that the following changes be made, if not already present, at the same time the changes for RITM0047174 are being made:

1) Move the program name to ‘DB2MSG-PGM-ID’ in the Initialization section before any SQL statements are executed.
2) Populate DB2MSG-TAG just before each and every executable SQL request.

Service Now Ticket - RITM0049965

RITM0049965 requests that in EDB Maintenance, PPP050 not abend if input file is empty. Instead, an error message should be displayed.

2 Background

- Service Now Ticket - RITM0047174

The DBA has requested that for efficiency and reduced overhead, SQL statements containing ‘Select *’ be replaced by statements that list the column and host variable names. DBA provided a list of ‘select * from DB2P created in Mar 2011:

```
db2pSelAsteris
ck.txt
```

To get a more current program list, run: SPGRRR.JCL.CNTL(SELASTR). It takes a few hours to run.

This release does not contain any programs using the group ‘insert’ logic. Below is a list of programs requiring changes that will be addressed in a future release. However, for any programs where the ‘select *’ is being changed, check for ‘insert’ statements and correct them as well.

```
SQL shortcut
inventory.pdf
```

An example of an ‘insert’ statement that should be changed can be found in ‘PAY.BASE.COBL (PPACTUTW):

```
EXEC SQL
    INSERT INTO PPPVACT2_ACT
    VALUES (:ACT-ROW-DATA)
```
After typing on command line: `sql ins pb0da.PPPVACT2_ACT`, the tool will replace it with this code:

```
EXEC SQL
    INSERT INTO PPPVACT2_ACT
    (EMPLOYEE_ID, APPT_NUM, DIST_NUM, ACTION_CODE, ACTION_EFFDATE)
    VALUES (:DCLPPPVACT2-ACT.EMPLOYEE-ID, :DCLPPPVACT2-ACT.APPT-NUM, :DCLPPPVACT2-ACT.DIST-NUM, ...)
END-EXEC
```

**Service Now Ticket - RITM0048039**

RITM0048039 requests changes to improve error handling and facilitate debugging.

1) Move the program name to ‘DB2MSG-PGM-ID’in the initialization section before any SQL statements are executed.

2) Populate DB2MSG-TAG just before each and every executable SQL request. **Do not put a period at the end of the statement.**

Executable SQL statements are:
- select
- open cursor
- close cursor
- fetch
- update
- insert
- delete

(Statements for declare, include do not required msg tags).
Service Now Ticket - RITM0049965

A separate TSD has been created for this. It is attached to the Service Now Ticket and also included in the project Folder (R2160) on Sharepoint.

3 Overview of System Modifications

For RITM0047174, changes will be made to 11 programs to replace ‘select *’ statements with the list of column and host variable names. If the statement is a ‘declare cursor’, the ‘fetch’ statement will be changed to list the host variables. The programs included in this Release do not contain any ‘insert’ statements requiring changes.

For RITM0048039, two programs included in this Release require changes.

- **PPP630**
  
  In 992000-SQL-COUNT-PER SECTION, before the SELECT COUNT(*) INTO :EMP-READ-CTR statement, add “MOVE 'SELECT PER COUNT' TO DB2MSG-TAG.”

- **PPP635**
  
  - Move ‘PPP635’ to DB2MSG-PGM-ID in M1100-INITIALIZE.
  
  - In F4120-FORMAT-WFA-EMPLOYEE-OUT SECTION, before “SELECT * INTO :DCLPPPVZJGT-JGT”, add “MOVE ‘F4120-FORMAT-WFA-EMPLOYEE-OUT’ TO DB2MSG-TAG”.

  - In F3100-GET-JGT-DEPT-OPTION SECTION, before “SELECT * INTO :DCLPPPVZPRM-PRM”, add “MOVE ‘F3100-GET-JGT-DEPT-OPTION’ TO DB2MSG-TAG”.

- **PPFAU018**
  
  Populate DB2MSG-TAG before “OPEN FND_CURSOR”, “SELECT COUNT (*)”, “FETCH FND_CURSOR” and ‘CLOSE FND_CURSOR”

Various processes will require testing.

4 Design Considerations

4.1 Assumptions and Dependencies

- For this release, eleven programs were selected. The remaining programs will be changed in future Technical Releases.

- The list of columns needed to replace “*” in a SELECT (or in re-coding an INSERT) must include all of the columns in the view being used and there will be no attempt made to remove columns that are not referred to by the programs.

- Code review is required.
Testing will include executing the SQL statements that were changed for RITM0047174.

- No testing is required for changes made for RITM0048039.
- RITM0047174 and RITM0048039 will remain open until the entire inventory is completed.
- No QA or UAT

5. Testing Considerations

This Section pertains to RITM0047174

- **DBA tool**

  DBA provided a tool that looks up all the table’s or view’s column names from the DB2 catalog and inserts them in the cobol program.

  The whole model SQL statement is written. You will need to edit it depending on requirements.

  The advantage is that all the column names are copied in the right order for you.

---

```
012000 012000  MOVE ACTION-CODE (WS-IXI)
012300 012300  SKIP1
012400 012400  MOVE ACTION-EFFDATE (WS-IX)
012500 012500  PERFORM CONVERT-DB2-DATE.
012600 012600  MOVE WSX-STD-DATE TO XACN-ACTION-DATE (WS-IX2, WS-IX3).
012700 012700  EJECT
012800 012800  DATE-CONVERSION-Routines Section.
012900 012900  SKIP3
013000 013000  COPY CPPOXOOC2.
013100 013100  END OF PPACAUTL ** ** ** ** ** **
```

---

```
013001  EXEC SQL
013002  SQL FOR TABLE PPPPER
013003  EMPLOYEE_ID
013004  ,EMP_NAME
013005  ,NAMESUFFIX
013006  ,BIRTH_DATE
013007  ,HOME_PHONE
```

This is just the beginning of what was written by ‘SQL’.
• **CHECK COMPILe LISTINGS CAREFULLY**

Look at all messages from compile listing if return code is > 0 to ensure that they are not a result of changes you made.

**Conditions that will cause return code 4 or 8:**

- **“Deleted Columns”**

Some views still contain column names that are no longer used. For example, in PAY.BASE.INCLUDE(PPPZVZDEP), DEP-INS-DEENROLL has been commented out and replaced by filler in the COBOL declaration:

```cobol
*****10 DEP-INS-DEENROLL  PIC X(01).
10 DEP-FILLER                PIC X(01).
```

However, since it still exists in the DB2 view (PB0DA.PPPVZDEP_DEP), the tool will insert it as a column name.

This will become evident when you do a compile. You will get a warning that the column does not exist. You will then need to delete it from the list of names generated by the tool. To date, PPPVZDEP_DEP and PPPVZBEN_BEN (and other views associated with PPPDEP and PPPBEN) have been identified as containing “deleted” columns, but there could be others.

- **Ambiguous References**

There could be multiple ‘select *’ statements that have the same column name from different tables or views. This is not an issue for ‘select *’, but it will be a problem when all the host variable names are listed. The compiler will give an error that the referenced field name is ambiguous.

For example, PPEC705 contains these statements:

```sql
FETCH PAY_ROW INTO :PAY-ROW-DATA and FETCH DEP_ROW INTO :DUPE-DEP-ROW-DATA
```

The DBA tool will insert the host variable names like this:

```cobol
FETCH PAY_ROW INTO
  :EMPLOYEE-ID
  :CITIZEN-CODE
  :VISA-TYPE ....
```

and

```cobol
FETCH DEP_ROW INTO
  :EMPLOYEE-ID
  :DEP-NUM
  :DEP-ADC-CODE ....
```

The compiler will flag this as ambiguous. You will need to qualify the host variable names as follows:
FETCH PAY_ROW INTO
   :PAY-ROW-DATA.EMPLOYEE-ID
   :CITIZEN-CODE
   :VISA-TYPE ....

and

FETCH DEP_ROW INTO
   :DUPE-DEP-ROW-DATA.EMPLOYEE-ID
   :DEP-NUM
   :DEP-ADC-CODE ....

NOTE: Other references to the field in the code will already be qualified, for example, ‘MOVE EMPLOYEE-ID OF DUPE-DEP-ROW-DATA’, so you do not need to make any further changes.

6 Mainframe Design

6.1.1 Cobol Programs

For the following programs, make changes to list column names and host variables in place of ‘select *’ statements as described in this document:

PPACTUTL retrieves PPPACT data for an employee into an array.

PPAPPUTL loads all appointment data from the EDB into linkage interface CPLNKAPP.

PPDETUTL retrieves PPPDET data in the CPWSXDET format.

PPEC108 performs consistency edits and maintenance on Direct Deposit (Surepay) enrollment.

PPEC703 performs consistency edits and maintenance on layoff transaction sets.

PPEC705 - performs consistency edits and maintenance on employee/dependent medical insurance enrollment data.

PPEC706 - performs consistency edits and maintenance on employee/dependent dental insurance enrollment data.

PPEC707 - performs consistency edits and maintenance on employee/dependent vision insurance enrollment data.

PPEC708 performs consistency edits and maintenance on employee/dependent legal insurance enrollment data.

PPEDBDET is a program in the EDB Fetch Complex which identifies the DB2 table for a given data element.

PPEC705 performs consistency edits and maintenance for BRSC (Benefits Rates Selection Code) data.

PPPCCUTL retrieves PPPCC data for an employee.

PPFAU018 returns fund group details for a given FAU and fund group.

PPP100 produces the error reports for the EDB maintenance process.
PPP180 produces a report of data element changes (before and after values) from an Employee Change File (ECF).

PPP400 performs Gross-to-Net processing within the compute creates a PAR file from the EDB and the preliminary PAR file with gross-to-net activity.

PPP630 creates folder labels for new hires and gives an audit list.

PPP635 produces the Affirmative Action Extract file.

7 Unit Testing Requirements

This Section pertains to RITM0047174

For each of these programs, use debugger, display or some other method to verify that each of the changed statements is executed. You may have to set up test data to meet certain conditions in order to achieve that.

- Use CICS EDEP to test:
  PPEC705 - Change medical insurance effective and/or end coverage date.
  PPEC706 - Change dental insurance effective and/or end coverage date.
  PPEC707 - Change vision insurance effective and/or end coverage date.
  PPEC708 - Change legal insurance effective and/or end coverage date.

- Use CICS ECHK to test PPEC108
- Use CICS ELOF to test PPEC703.
- Use CICS EALN to test PPACTUTL
- Run Batch EDB Maintenance to test:
  PPDETUTL
  PPP180
  PPEI505
  PPP100
  PPPCCUTL
  PPEDBDET

- Run Compute to test:
  PPP400
  PPFAU018

- Other processing:
  PPP630 - Execute PPP630
  PPP635 - Execute PPP635
  PPAPPPUTL – Execute PPI730 which creates the STIP file for UCRP.