<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTRODUCTION</td>
<td>2</td>
</tr>
<tr>
<td>DDL MEMBERS FOR DB2 CHANGES</td>
<td>3</td>
</tr>
<tr>
<td>INCLUDES</td>
<td>15</td>
</tr>
<tr>
<td>PROGRAMS</td>
<td>17</td>
</tr>
<tr>
<td>COPYMEMBERS</td>
<td>49</td>
</tr>
<tr>
<td>BIND MEMBERS</td>
<td>51</td>
</tr>
<tr>
<td>TABLE UPDATES</td>
<td>52</td>
</tr>
<tr>
<td>FORMS</td>
<td>54</td>
</tr>
<tr>
<td>JCL</td>
<td>55</td>
</tr>
<tr>
<td>ATTACHMENTS</td>
<td>56</td>
</tr>
</tbody>
</table>
Introduction

Service Request 17275 requested a Web based Merit application.

Service Request 17275 contained guidelines for the development of a Web based Merit application. Although it contained some specific guidelines on screen content, it did not fully address screen functionality, and it did not address the issues surrounding EDB data extraction and update at all. Modification and refinement continued as more specific functionality and technical solutions were evaluated and defined during weekly meetings of functional and technical staff working on this project. These refinements were documented in issues lists developed by those groups. In addition, a functional users Merit Testing document was developed. It defines expected edit results, and features such as pull-down and pop-up windows. It serves as a “design” document for the web portion of this application.

Web related objects PPS objects were first issued in Release 1408 Web EDB Inquiry. Some of those objects were further modified in Release 1415 PAN Review. While this application does not share the net-data web server platform, some objects issued in those releases, such as DB2 stored procedures and the PPS Main Menu, will be shared by this application.

This document defines the required changes on the MVS side, i.e. it defines the DB2 changes, batch programs, COBOL stored procedures, copymembers, etc. Some of the objects defined in this document have already been issued in separate releases. Their references have been left here to provide a single location for the overall design.

The modifications to add the Merit Selection column to the PPPBUT table were issued in Release 1405.

The modifications to create the Merit Control department table PPPMCP were issued in Release 1413.

The modifications to add PPPFND edits for Merit Costing Group Definitions were issued in Release 1413.

The modifications to add a Performance Evaluation Code and Date to the Appointment table PPPAPP were issued in Release 1419.
DDL Members for DB2 Changes

Merit Cycle Database (PPPMRT):
A new database will be created for the various tables used by the Web Merit process.

Merit Cycle Administration Tables

The Merit Cycle Administration Tables define the Merit Cycle criteria. The PPPMCC defines the main eligibility and selection criteria for the EDB extract. The extract also uses the PPPMAT table to include and exclude selected Title Codes. The PPPLEC, PPPMRR and PPPSPR define the recommended and required rate increases, per defined evaluation codes.

Merit Cycle Criteria Table (PPPMCC):
A new table will be added to the PPPMRT database that will contain the eligibility and selection criteria for each Merit Cycle. This will be the main table for the entire Merit process, and most other tables will be related to it by Cycle ID.

The standard DDL members defining the tablespace, table, index and view for the PPPMCC table will be created.
- TSMCC00C
- TBMCC00C
- IXMCC00C
- PPPVZMCC

| MCC_CYCLE_ID | CHAR (18) NOT NULL, |
| MCC_CYCLE_DATE | DATE NOT NULL, |
| MCC_CBUC | CHAR(2) NOT NULL WITH DEFAULT, |
| MCC_SUB_LOCATION | CHAR(2) NOT NULL WITH DEFAULT, |
| MCC_PROG_TYPE | CHAR(3) NOT NULL WITH DEFAULT, |
| MCC_STATUS_CD | CHAR(1) NOT NULL WITH DEFAULT, |
| MCC_DESCRIPTION | CHAR(30) NOT NULL WITH DEFAULT, |
| MCC_STEP_RANGE | CHAR(1) NOT NULL WITH DEFAULT, |
| MCC_COLLAPSE | CHAR(1) NOT NULL WITH DEFAULT, |
| MCC_NEXT_SALREV_DT | DATE NOT NULL, |
| MCC_SALREV_TYPE1 | CHAR(1) NOT NULL WITH DEFAULT, |
| MCC_SALREV_TYPE2 | CHAR(1) NOT NULL WITH DEFAULT, |
| MCC_SALREV_TYPE3 | CHAR(1) NOT NULL WITH DEFAULT, |
| MCC_SALREV_TYPE4 | CHAR(1) NOT NULL WITH DEFAULT, |
| MCC_SALREV_TYPE5 | CHAR(1) NOT NULL WITH DEFAULT, |
| MCC_APP_TYPE1 | CHAR(1) NOT NULL WITH DEFAULT, |
| MCC_APP_TYPE2 | CHAR(1) NOT NULL WITH DEFAULT, |
| MCC_APP_TYPE3 | CHAR(1) NOT NULL WITH DEFAULT, |
| MCC_APP_TYPE4 | CHAR(1) NOT NULL WITH DEFAULT, |
| MCC_APP_TYPE5 | CHAR(1) NOT NULL WITH DEFAULT, |
| MCC_APP_TYPE6 | CHAR(1) NOT NULL WITH DEFAULT, |
| MCC_APP_TYPE7 | CHAR(1) NOT NULL WITH DEFAULT, |
| MCC_APP_TYPE8 | CHAR(1) NOT NULL WITH DEFAULT, |
| MCC_OLD_REV_DT | DATE NOT NULL, |
| MCC_HIRE_DATE | DATE NOT NULL, |
| MCC_PROB_DATE | DATE NOT NULL, |
| MCC_APPT_REP_CODE | CHAR(3) NOT NULL WITH DEFAULT, |
| MCC_AT_INCL_EXCL | CHAR(1) NOT NULL WITH DEFAULT, |
| MCC_BW_EFF_DATE | DATE NOT NULL, |
| MCC_MO_EFF_DATE | DATE NOT NULL, |
Local Evaluation Codes Table (PPPLEC):
A new table will be added to the PPPMRT database that will contain the local evaluation codes mapped to application defined standard codes.

The standard DDL members defining the tablespace, table, index and view for the PPPLEC table will be created.
- TSLEC00C
- TBLEC00C
- IXLEC00C
- PPPVZLEC

Range Performance Rating Table (PPPMRR):
A new table will be added to the PPPMRT database that will contain the minimum and maximum recommended or required range increases for each standard evaluation code.

The standard DDL members defining the tablespace, table, index and view for the PPPMRR table will be created.
- TSMRR00C
- TBMRR00C
- IXMRR00C
- PPPVZMRR

Step Performance Rating Table (PPPSPR):
A new table will be added to the PPPMRT database that will contain the minimum and maximum recommended or required step increases for each standard evaluation code.

The standard DDL members defining the tablespace, table, index and view for the PPPSPR table will be created.

- TSSPR00C
- TBSPR00C
- IXSPR00C
- PPPVZSPR

```sql
SPR_CYCLE_ID CHAR(18) NOT NULL,
SPR_STD_RATING CHAR(1) NOT NULL,
SPR_MIN_STEP DEC(4,2) NOT NULL WITH DEFAULT,
SPR_MAX_STEP DEC(4,2) NOT NULL WITH DEFAULT,
SPR_REQ_REC CHAR(1) NOT NULL WITH DEFAULT,
PRIMARY KEY (SPR_CYCLE_ID, SPR_STD_RATING),
FOREIGN KEY (SPR_CYCLE_ID) REFERENCES PPPMCC ON DELETE CASCADE
```

**Title Code List Table (PPPMAT):**

A new table will be added to the PPPMRT database that will contain the additional Title Code selection criteria for each Merit Cycle. It will contain single Title Codes or Title Code ranges Per the PPPMCC criteria, the Title Codes will be included or excluded.

The standard DDL members defining the tablespace, table, index and view for the PPPMCC table will be created.

- TSMATC00C
- TBMAT00C
- IXMAT00C
- PPPVZMAT

```sql
MAT_CYCLE_ID CHAR(18) NOT NULL,
MAT_RANGE_BEGIN CHAR(4) NOT NULL,
MAT_RANGE_END CHAR(4) NOT NULL,
PRIMARY KEY (MAT_CYCLE_ID, MAT_RANGE_BEGIN, MAT_RANGE_END),
FOREIGN KEY (MAT_CYCLE_ID) REFERENCES PPPMCC ON DELETE CASCADE
```

**Merit Cycle Roster Tables**

The Merit Cycle Roster Tables include tables that are loaded from the EDB to establish a beginning roster for a Merit Cycle, and some work tables used to control the process along with ARSM rules. The PPPMEE, PPPMAP and PPPMED contain EDB data, and are not updated by the Web Merit application. The PPPMLA is the main tool for updating an employee’s performance evaluation and rate data. It is the source for the eventual EDB update. The PPPMDP table is used by the ARSM process to define a user’s access by department code, and joined with the PPPMLA to select a user specific roster.

**Merit Employee Table (PPPMEE):**

A new table will be added to the PPPMRT database that will contain EDB Employee data for rosters selected for each Merit Cycle.
The standard DDL members defining the tablespace, table, index and view for the PPPMEE table will be created.

- TSMEE00C
- TBMEE00C
- IXMEE00C
- PPPVZMEE

```
MEE_CYCLE_ID    CHAR(18)  NOT NULL,
MEE_EMPLOYEE_ID CHAR(9)   NOT NULL,
MEE_NAMESUFFIX  CHAR(4)   NOT NULL WITH DEFAULT,
MEE_EMP_NAME    CHAR(26)  NOT NULL WITH DEFAULT,
MEE_FIRST_NAME  CHAR(30)  NOT NULL WITH DEFAULT,
MEE_MIDDLE_NAME CHAR(30)  NOT NULL WITH DEFAULT,
MEE_LAST_NAME   CHAR(30)  NOT NULL WITH DEFAULT,
MEE_BIRTH_DATE  DATE      NOT NULL,
MEE_HIRE_DATE   DATE      NOT NULL,
MEE_HOME_DEPT   CHAR(6)   NOT NULL WITH DEFAULT,
MEE_HOME_DEPT_NAME CHAR(30)  NOT NULL WITH DEFAULT,
MEE_EMP_STATUS  CHAR(1)   NOT NULL WITH DEFAULT,
MEE_PRIMARY_TITLE  CHAR(4)   NOT NULL WITH DEFAULT,
MEE_PRI_PAY_SCHED CHAR(2)   NOT NULL WITH DEFAULT,
MEE_EMP_REL_CODE CHAR(1)   NOT NULL WITH DEFAULT,
MEE_EMP_REL_UNIT CHAR(1)   NOT NULL WITH DEFAULT,
MEE_NEXT_SALREV_CD CHAR(1)  NOT NULL WITH DEFAULT,
MEE_NEXT_SALREV_DT DATE      NOT NULL,
MEE_PROB_END_DATE DATE      NOT NULL,
```

Primary Key
```
(MEE_CYCLE_ID, MEE_EMPLOYEE_ID),
```

Foreign Key
```
(MEE_CYCLE_ID) REFERENCES PPPMCC ON DELETE CASCADE
```

Merit Appointment Table (PPPMAP):
A new table will be added to the PPPMRT database that will contain the EDB appointment data for rosters selected for each Merit Cycle.

The standard DDL members defining the tablespace, table, index and view for the PPPMAP table will be created.

- TSMAPC00C
- TBMAP00C
- IXMAP00C
- PPPVZMAP

```
MAP_CYCLE_ID    CHAR(18)  NOT NULL,
MAP_EMPLOYEE_ID CHAR(9)   NOT NULL,
MAP_DEPT        CHAR(6)   NOT NULL,
MAP_TITLE_CODE  CHAR(04)  NOT NULL,
MAP_SUB_LOCATION CHAR(02)  NOT NULL,
MAP_APPT_REP_CODE CHAR(01) NOT NULL,
MAP_APPT_TYPE   CHAR(01)  NOT NULL,
MAP_GRADE       CHAR(02)  NOT NULL,
MAP_RATE_CODE   CHAR(01)  NOT NULL,
MAP_PAY_SCHEDULE CHAR(02)  NOT NULL,
```
Merit Distribution Table (PPPMED):
A new table will be added to the PPPMRT database that will contain the EDB appointment and distribution data for rosters selected for each Merit Cycle. The PPPMED will play a primary role in creating EDB updates from Logical Appointments.

The standard DDL members defining the tablespace, table, index and view for the PPPMED table will be created.

• TSMED00C
• TBMED00C
• IXMED00C
• PPPVZMED

```sql
MERIT DISTRIBUTION TABLE (PPPMED):

CREATE TABLE PPPMED (  
  MED_CYCLE_ID CHAR(18) NOT NULL,  
  MED_EMPLOYEE_ID CHAR(9) NOT NULL,  
  MED_HOME_DEPT CHAR(6) NOT NULL,  
  MED_APPT_DEPT CHAR(6) NOT NULL,  
  MED_APPT_TITLE CHAR(4) NOT NULL,  
  MED_APPT_SUB_LOCN CHAR(2) NOT NULL,  
  MED_APPT_REP_CODE CHAR(1) NOT NULL,  
  MED_APPT_TYPE CHAR(1) NOT NULL,  
  MED_SALARY_GRADE CHAR(2) NOT NULL,  
  MED_APPT_RATE_CODE CHAR(1) NOT NULL,  
  MED_PAY_SCHEDULE CHAR(2) NOT NULL,  
  MED_DIST_RATE DEC(9,4) NOT NULL,  
  MED_DIST_STEP CHAR(4) NOT NULL,  
  MED_DIST_NUM DEC(2,0) NOT NULL,  
  MED_APPT_TITLE_NM CHAR(30) NOT NULL WITH DEFAULT,  
  MED_PAY_BEGIN_DATE DATE NOT NULL,  
  MED_PAY_END_DATE DATE NOT NULL,  
  MED_FULL_ACCT_UNIT CHAR(30) NOT NULL WITH DEFAULT,  
  MED_DIST_PERCENT DEC(5,4) NOT NULL WITH DEFAULT,  
  MED_DIST_DOS CHAR(3) NOT NULL WITH DEFAULT,  
  PRIMARY KEY  
  (MED_CYCLE_ID,  
   MED_EMPLOYEE_ID,  
   MED_HOME_DEPT,  
   MED_APPT_DEPT,  
   MED_APPT_TITLE,  
   MED_APPT_SUB_LOCN,  
   MED_APPT_REP_CODE,  
   MED_APPT_TYPE,  
   MED_SALARY_GRADE,  
   MED_APPT_RATE_CODE,  
   MED_PAY_SCHEDULE,  
   MED_DIST_RATE),  
  FOREIGN KEY  
  (MED_CYCLE_ID) REFERENCES PPPMCC ON DELETE CASCADE
)
```
(MED_CYCLE_ID,
MED_EMPLOYEE_ID,
MED_HOME_DEPT,
MED_APPT_DEPT,
MED_APPT_TITLE,
MED_APPT_SUB_LOCN,
MED_APPT_REP_CODE,
MED_APPT_TYPE,
MED_SALARY_GRADE,
MED_APPT_RATE_CODE,
MED_PAY_SCHEDULE,
MED_DIST_RATE,
MED_DIST_STEP,
MED_DIST_NUM),
FOREIGN KEY
(MED_CYCLE_ID) REFERENCES PPMCC ON DELETE CASCADE

Merit Logical Appointment Table (PPPMLA):
A new table will be added to the PPPMRT database that will contain Logical Appointment data for each Merit Cycle. A single Logical Appointment is created from all PPPMED entries sharing the key data of the PPPMLA table.

The standard DDL members defining the tablespace, table, index and view for the PPPMLA table will be created.
- TSMLA00C
- TBMLA00C
- IXMLA00C
- PPVZMLA

MLA_CYCLE_ID CHAR(18) NOT NULL,
MLA_EMPLOYEE_ID CHAR(9) NOT NULL,
MLA_DEPT CHAR(6) NOT NULL,
MLA_TITLE_CODE CHAR(4) NOT NULL,
MLA_APPT_SUB_LOCN CHAR(2) NOT NULL,
MLA_APPT_REP_CODE CHAR(1) NOT NULL,
MLA_APPT_TYPE CHAR(1) NOT NULL,
MLA_GRADE CHAR(2) NOT NULL,
MLA_RATE_CODE CHAR(1) NOT NULL,
MLA_PAY_SCHEDULE CHAR(2) NOT NULL,
MLA_DIST_PAYRATE DEC(9,4) NOT NULL,
MLA_DIST_STEP CHAR(4) NOT NULL,
MLA_DIST_UNIT_CODE CHAR(1) NOT NULL WITH DEFAULT,
MLA_DIST_PERCENT DEC(5,4) NOT NULL WITH DEFAULT,
MLA_ANNUAL_SALARY DEC(9,2) NOT NULL WITH DEFAULT,
MLA_NEWANN_SALARY DEC(9,2) NOT NULL WITH DEFAULT,
MLA_NEWNXT_SAL_DT DATE NOT NULL,
MLA_NEW_PAYRATE DEC(9,4) NOT NULL WITH DEFAULT,
MLA_NEW_STEP CHAR(4) NOT NULL WITH DEFAULT,
MLA_PCT_INCREASE DEC(4,2) NOT NULL WITH DEFAULT,
MLA_EVAL_DATE DATE NOT NULL,
MLA_PERFORM_RATING CHAR(3) NOT NULL WITH DEFAULT,
MLA_EMP_NAME CHAR(26) NOT NULL WITH DEFAULT,
MLA_DEPT_NAME CHAR(30) NOT NULL WITH DEFAULT,
MLA_APPT_TITLE_NM CHAR(30) NOT NULL WITH DEFAULT,
**Merit Messages Table (PPPMMS):**
A new table will be added to the PPPMRT database that will contain the messages created during the roster update process.

The standard DDL members defining the tablespace, table, index and view for the PPPMMS table will be created.

- TSMMS00C
- TBMMS00C
- IXMMS00C
- PPPVZMMS

```sql
MMS_CYCLE_ID        CHAR(18)  NOT NULL,
MMS_EMPLOYEE_ID     CHAR(9)   NOT NULL,
MMS_DEPT            CHAR(6)   NOT NULL,
MMS_TITLE_CODE      CHAR(4)   NOT NULL,
MMS_APPT_SUB_LOCN   CHAR(2)   NOT NULL,
MMS_APPT_REP_CODE   CHAR(1)   NOT NULL,
MMS_APPT_TYPE       CHAR(1)   NOT NULL,
MMS_GRADE           CHAR(2)   NOT NULL,
MMS_RATE_CODE       CHAR(1)   NOT NULL,
MMS_PAY_SCHEDULE    CHAR(2)   NOT NULL,
MMS_DIST_PAYRATE    DEC(9,4)  NOT NULL,
MMS_DIST_STEP       CHAR(4)   NOT NULL,
MMS_MSG_NUMBER      CHAR(5)   NOT NULL,
MMS_CHANGED_BY      CHAR(8)   NOT NULL WITH DEFAULT,
MMS_CHANGED_AT      TIMESTAMP NOT NULL,
MMS_COMMENTS        CHAR(200) NOT NULL WITH DEFAULT,
```

**PRIMARY KEY**
(MMS_CYCLE_ID, MMS_EMPLOYEE_ID, MMS_DEPT, MMS_TITLE_CODE, MMS_APPT_SUB_LOCN, MMS_APPT_REP_CODE, MMS_APPT_TYPE, MMS_GRADE, MMS_RATE_CODE, MMS_PAY_SCHEDULE, MMS_DIST_PAYRATE, MMS_DIST_STEP)
Merit Department Table (PPPMDP):
A new table will be added to the PPPMRT database that will contain the list of departments accessible by an individual user based on ARSM rules and CTL Merit Control Department Table structure. It does not reference the PPPMCC table. It is identified primarily by the Session ID of the Web Merit session.

The standard DDL members defining the tablespace, table, index and view for the PPPMDP table will be created.
- TSMDP00C
- TBMDP00C
- IXMDP00C
- PPPVZMDP

Control Tables

Merit Control Department Table (PPPMCP):
A new table will be added to the PPPCTL database that will contain the list of departments accessible by the Web Merit process. Each department will be defined with its "controlling" parent department. The path up and down through departments and control departments will define the Control Point structure for Web Merit.

The standard DDL members defining the tablespace, table, index and view for the PPPMCP table will be created.
- TSMCP00C
- TBMC00C
- IXCMP00C
- PPPVZMCP
**Bargaining Unit Table (PPPBUT):**
The Bargaining Unit Table will be modified to add a flag indicating that a bargaining unit can be processed via the Web Merit. The bargaining unit codes with the flag set to Y will be displayed for selection on the main Merit Administration screen which updates the PPPMCC table.

The standard DDL members defining the table and view for the PPPMCP table will be modified.

- TBBUT00C
- PPPVZBUT

```
BUT_MERIT_SELECT     CHAR(01) NOT NULL WITH DEFAULT,
```

**History Bargaining Unit Table (PPPBUTH):**
The Bargaining Unit Table will be modified to carry the flag indicating that a bargaining unit can be processed via the Web Merit.

The standard DDL members defining the table and view for the PPPMCP table will be modified.

- TBBUTH00C
- PPPVNBUTH

```
BUT_MERIT_SELECT     CHAR(01) NOT NULL WITH DEFAULT,
```
Stored Procedures DDL

Create Procedure DDL will be created for the new stored procedures. The DDL member names will start with SP, but the procedure names themselves will be as follows.

PPBUTSPR (new)

CREATE PROCEDURE PPBUTSPR
  (OUT   RETCODE CHAR(2),
   OUT   CBUCS   VARCHAR(520))
LANGUAGE COBOL
DYNAMIC RESULT SET 0
EXTERNAL NAME PPBUTSPR
PARAMETER STYLE GENERAL
COLLID *COLLID*
STAY RESIDENT YES
NO WLM ENVIRONMENT
;

PPCSTSPR (new)

CREATE PROCEDURE PPCSTSPR
  (IN    CYCLEID    CHAR(18),
   IN    SESSIONID  CHAR(50),
   IN    LEVEL      CHAR(03),
   IN    ARSMRULE   CHAR(8),
   OUT   RETCODE    CHAR(02),
   OUT   COSTS   VARCHAR(4360))
LANGUAGE COBOL
DYNAMIC RESULT SET 0
EXTERNAL NAME PPCSTSPR
PARAMETER STYLE GENERAL
COLLID *COLLID*
STAY RESIDENT YES
NO WLM ENVIRONMENT
;

PPCTTSP2 (new)

CREATE PROCEDURE PPCTTSP2
  (OUT   RETCODE CHAR(2),
   OUT   CYCSTAT VARCHAR(612))
LANGUAGE COBOL
DYNAMIC RESULT SET 0
EXTERNAL NAME PPCTTSP2
PARAMETER STYLE GENERAL
COLLID *COLLID*
STAY RESIDENT YES
NO WLM ENVIRONMENT
;

PPMDPSPR (new)

CREATE PROCEDURE PPMDPSPR
(IN SESSION CHAR(50),
 IN ARSMRULE CHAR(8),
 IN DEPTNO CHAR(6),
 IN ACTION CHAR(6),
 OUT LEVEL CHAR(3),
 OUT RETCODE CHAR(2))

LANGUAGE COBOL
DYNAMIC RESULT SET 0
EXTERNAL NAME PPMDPSPR
PARAMETER STYLE GENERAL
COLLID *COLLID*
STAY RESIDENT YES
NO WLM ENVIRONMENT
;

PPTCTSP2 (new)

CREATE PROCEDURE PPTCTSP2
(IN TITLE CHAR(4),
 IN SUBL CHAR(2),
 IN REPC CHAR(1),
 IN DATE CHAR(10),
 OUT RATES VARCHAR(1360))

LANGUAGE COBOL
DYNAMIC RESULT SET 0
EXTERNAL NAME PPTCTSP2
PARAMETER STYLE GENERAL
COLLID *COLLID*
STAY RESIDENT YES
NO WLM ENVIRONMENT
;

PPTCTSP5 (new)

CREATE PROCEDURE PPTCTSP5
(IN TITLE CHAR(4),
 IN SUBL CHAR(2),
 IN REPC CHAR(1),
 IN DATE CHAR(10),
 OUT RATES CHAR(87),
 OUT RATES2 CHAR(44))

LANGUAGE COBOL
DYNAMIC RESULT SET 0
EXTERNAL NAME PPTCTSP5
PARAMETER STYLE GENERAL
COLLID *COLLID*
STAY RESIDENT YES
NO WLM ENVIRONMENT
;

UCPPPMRT (new)

CREATE PROCEDURE UCPPPMRT
(IN AUTHID CHAR(8),
 IN ACCESS CHAR(6),
IN    CYCLESTAT  CHAR(1),
IN    SESSION    CHAR(50),
OUT   EMPLOYEEID CHAR(9),
OUT   PERMISSION CHAR(1),
OUT   RETCODE    CHAR(2))
LANGUAGE COBOL
DYNAMIC RESULT SET 0
EXTERNAL NAME UCPPPMRT
PARAMETER STYLE GENERAL
COLLID *COLLID*
STAY RESIDENT YES
NO WLM ENVIRONMENT
;
Includes

PPPVZBUT:  
The standard Include member defining the working storage and View for a table row in the PPPBUT table will be modified to include new column BUT_MERIT_SELECT.

PPPVBUTH:  
The standard Include member defining the working storage and View for a table row in the History PPPBUTH table will be modified to include new column BUT_MERIT_SELECT.

PPPVZLEC (new):  
The standard Include member defining the working storage and View for a table row in the PPPLEC table will be created.

PPPVZMAP (new):  
The standard Include member defining the working storage and View for a table row in the PPPMAP table will be created.

PPPVZMAT (new):  
The standard Include member defining the working storage and View for a table row in the PPPMAT table will be created.

PPPVZMCC (new):  
The standard Include member defining the working storage and View for a table row in the PPPMCC table will be created.

PPPVZMCP (new):  
The standard Include member defining the working storage and View for a table row in the PPPMCP table will be created.

PPPVZMDP (new):  
The standard Include member defining the working storage and View for a table row in the PPPMDP table will be created.

PPPVZMED (new):  
The standard Include member defining the working storage and View for a table row in the PPPMED table will be created.

PPPVZMEE (new):  
The standard Include member defining the working storage and View for a table row in the PPPMEE table will be created.

PPPVZMLA (new):  
The standard Include member defining the working storage and View for a table row in the PPPMLA table will be created.

PPPVZMMS (new):  
The standard Include member defining the working storage and View for a table row in the PPPMMS table will be created.

PPPVZMRR (new):  

The standard Include member defining the working storage and View for a table row in the PPPMRR table will be created.

**PPPVZSPR (new):**
The standard Include member defining the working storage and View for a table row in the PPPSPR table will be created.
Programs

**PPBUTHUP**

PPBUTHUP is part of the History update process for the PPPBUTH table. It will be modified to process the BUT_MERIT_SELECT column.

**PPCTT44 (new)**

PPCTT44 will be the transaction handler for the Merit Control Department Table 44. It will be passed all Table 44 transactions and move the data into the structure defined by copymember CPCTMCPI for processing by edit and update modules.

PPCTT44 will call the Merit Control Department transaction edit program PPCTMCPE to edit the transaction data.

Depending on the results of the edit and the run mode, i.e. update or not, PPCTT44 will call the Merit Control Department table update program PPCTMCPUI.

**PPCTFNDE**

PPCTFNDE edits the Fund Grouping transaction data. This design assumes that the PPPFND update has been converted to the new PPP004 direct DB2 update process.

Three new Fund Group Definitions will be added to the PPPFND table for controlling costing aggregation by fund source for the Web Merit online Costing Summary. The Fund Group Code for these entries will be a four digit number which will be used for sub-scripting purposes in stored procedure PPCSTSPR.

An edit will be added specifically for Fund Group Definitions MERITADM MERITCTL and MERITDPT. If the Fund group Code is not numeric the transaction will be rejected. See Attachment F.

**PPCTMCPE (new)**

PPCTMCPE will edit the Merit Control Department transaction data for the following conditions.

The Merit Department cannot be blank, and must be on the Home Department Table. The transaction will be rejected.

The Merit Control Department cannot be blank, and must be on the Home Department Table. The transaction will be rejected.

Note: These edits are only performed at the time the Merit Control Department Table is updated. The Merit Control Department Table is not linked referentially to the Home Department Table. Therefore subsequent update activity against the Home department Table, i.e., deletions, can result in PPPMCP entries that no longer match an entry on the PPPHME.

Standard A/D/C logic will also be performed. A record cannot be added when it already exists. A record cannot be changed or deleted when it does not exist.

**PPCTMCPUI (new)**

PPCTMCPUI will update the Merit Control Department Table during update mode for transaction data which passes the edits. It will perform Inserts, Updates and Deletions.

Last Action and Last Action Date will be updated as well.

**PPCTR44 (new)**

PPCTR44 will generate a report listing of the Merit Control Department Table. It will be called whenever the table is successfully updated, or when a specific request is made for a Table 44 report.
The report will display the Department Number and Control Department Number from each row on the Merit Control Department Table. For both it will also obtain and display the Department Name from the Home Department Table, if it exists there.

The default report order will be in Merit Control Department and Department order. An alternate, Department order, will be defined on the Table of Tables, and will be selectable on report request transactions.

**PPCTC07 (new)**
A consistency edit module will be created to report on inconsistencies between the Merit Control Department Table and the Home Department Table, and inconsistencies within the Merit Control Department Table. Warning messages will be issued for the following conditions:

- A Home Department that does not have an entry on the Merit Control Department table.
- A Merit Department that does not exist on the Home Department Table (PPPHME). This appears redundant given the transaction edit, but the consistency edit program can also be triggered by Home department updates, e.g. deletions.
- A Merit Control Department that does not exist on the Home Department Table (PPPHME). Again, PPPHME activity could have caused this.
- A Merit Control Department that does not exist as a Merit Department (this indicates a loose limb on the control tree)
- A Merit Department has itself as a Merit Control Department This indicates the top of a control group. There can be multiple occurrences of this condition, and it is a valid condition.
- Merit Departments, which when walked up through their control, and then to its control, etc, do not reach a top control entry within a locally defined number of levels.

**PPP675 (new)**
PPP675 will be a new program which will create Web Merit roster data based on the criteria defined for a Merit Cycle ID read in via a Run Specification Record (see new form UPAY911).

The Web Merit System will process data in its own DB2 tables which have to be populated from the existing PPS EDB tables. Some of the data going to the new tables needs to reformatted or calculated from more than one data field. PPP675 will perform the extraction from the EDB tables and insertion into the Merit tables.

There are four tables populated by this program.
1. PPPMEE  - Merit Employee Table
2. PPPMLA  - Merit Logical Appointment Table
3. PPPMAP  - Merit Appointment Table
4. PPPMED  - Merit Distribution Table

In addition the Merit Cycle Table (PPPMCC) is accessed to obtain the selection criteria for the extract.

**1. PPPMEE - Merit Employee Table**

<table>
<thead>
<tr>
<th>Merit Employee Table</th>
<th>Description of Field Name</th>
<th>Data Extracted From the Following EDB Table Field</th>
<th>Len</th>
<th>Fmt</th>
<th>Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEE_CYCLE_ID</td>
<td>Merit Cycle ID</td>
<td>Foreign Key with Cascade Delete PPPMCC.MCC_CYCLE_ID</td>
<td>18</td>
<td>Char</td>
<td>Yes</td>
</tr>
<tr>
<td>MEE_EMPLOYEE_ID</td>
<td>Emp ID</td>
<td>PPPPER.EMPLOYEE_ID</td>
<td>9</td>
<td>Char</td>
<td>Yes</td>
</tr>
<tr>
<td>MEE_NAMESUFFIX</td>
<td>Emp Name Suffix</td>
<td>PPPPER.NAMESUFFIX</td>
<td>4</td>
<td>Char</td>
<td>No</td>
</tr>
<tr>
<td>MEE_EMP_NAME</td>
<td>Emp Name</td>
<td>PPPPER.EMP_NAME</td>
<td>26</td>
<td>Char</td>
<td>No</td>
</tr>
<tr>
<td>MEE_FIRST_NAME</td>
<td>Emp First Name</td>
<td>PPPPER.FIRST_NAME</td>
<td>30</td>
<td>Char</td>
<td>No</td>
</tr>
<tr>
<td>MEE_MIDDLE_NAME</td>
<td>Emp Middle Name</td>
<td>PPPPER.MIDDLE_NAME</td>
<td>30</td>
<td>Char</td>
<td>No</td>
</tr>
<tr>
<td>MEE_LAST_NAME</td>
<td>Emp Last Name</td>
<td>PPPPER.LAST_NAME</td>
<td>30</td>
<td>Char</td>
<td>No</td>
</tr>
<tr>
<td>Merit Employee Table</td>
<td>Description of Field Name</td>
<td>Data Extracted From the Following EDB Table.Field</td>
<td>Len</td>
<td>Fmt</td>
<td>Key</td>
</tr>
<tr>
<td>----------------------</td>
<td>---------------------------</td>
<td>--------------------------------------------------</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
</tr>
<tr>
<td>PPPMEE Field Name</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MEE_BIRTH_DATE</td>
<td>Date of Birth</td>
<td>PPPPER.BIRTH_DATE</td>
<td>10</td>
<td>Date</td>
<td>No</td>
</tr>
<tr>
<td>MEE_HIRE_DATE</td>
<td>Recent Hire Date</td>
<td>PPPPER.HIRE_DATE</td>
<td>10</td>
<td>Date</td>
<td>No</td>
</tr>
<tr>
<td>MEE_HOME_DEPT</td>
<td>Home Dept Code</td>
<td>PPPPER.HOME_DEPT</td>
<td>6</td>
<td>Char</td>
<td>No</td>
</tr>
<tr>
<td>MEE_HOME_DEPT_NAME</td>
<td>Home Dept Name</td>
<td>PPPHME.HME.DEPT_NAME</td>
<td>30</td>
<td>Char</td>
<td>No</td>
</tr>
<tr>
<td>MEE_EMP_STATUS</td>
<td>Employment Status</td>
<td>PPPPER.EMP_STATUS</td>
<td>1</td>
<td>Char</td>
<td>No</td>
</tr>
<tr>
<td>MEE_PRIMARY_TITLE</td>
<td>Primary Title Code</td>
<td>PPPPER.PRIMARY_TITLE</td>
<td>4</td>
<td>Char</td>
<td>No</td>
</tr>
<tr>
<td>MEE_PRI_PAY_SCHED</td>
<td>Primary Pay Schedule</td>
<td>PPPPCM_PRI_PAY_SCHED</td>
<td>2</td>
<td>Char</td>
<td>No</td>
</tr>
<tr>
<td>MEE_EMP_REL_CODE</td>
<td>Emp Relations Code</td>
<td>PPPPER.EMP_REL_CODE</td>
<td>1</td>
<td>Char</td>
<td>No</td>
</tr>
<tr>
<td>MEE_EMP_REL_UNIT</td>
<td>Emp Relations Unit</td>
<td>PPPPER.EMP_REL_UNIT</td>
<td>2</td>
<td>Char</td>
<td>No</td>
</tr>
<tr>
<td>MEE_EMP_SPEC_HAND</td>
<td>Emp Spec Handling</td>
<td>PPPPER.EMP_SPEC_HAND</td>
<td>1</td>
<td>Char</td>
<td>No</td>
</tr>
<tr>
<td>MEE_NEXT_SALREV_CD</td>
<td>Next Sal Review Code</td>
<td>PPPPER.NEXT_SALREV_DATE</td>
<td>1</td>
<td>Char</td>
<td>No</td>
</tr>
<tr>
<td>MEE_NEXT_SALREV_DT</td>
<td>Next Sal Review Date</td>
<td>PPPPER.NEXT_SALREV_DATE</td>
<td>10</td>
<td>Date</td>
<td>No</td>
</tr>
<tr>
<td>MEE_PROB_END_DATE</td>
<td>Probation End Date</td>
<td>PPPPER.PROB_END_DATE</td>
<td>10</td>
<td>Date</td>
<td>No</td>
</tr>
</tbody>
</table>

2. PPPMLA - Merit Logical Appointment Table

<table>
<thead>
<tr>
<th>Merit Logical Appt Table</th>
<th>Description of Field Name</th>
<th>Data Extracted From the Following EDB Table.Field</th>
<th>Len</th>
<th>Fmt</th>
<th>Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>PPPMLA Field Name</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MLA_CYCLE_ID</td>
<td>Merit Cycle ID</td>
<td>Foreign Key with Cascade Delete</td>
<td>18</td>
<td>Char</td>
<td>Yes</td>
</tr>
<tr>
<td>MLA_EMPLOYEE_ID</td>
<td>Employee ID</td>
<td>PPPPER.EMPLOYEE_ID</td>
<td>9</td>
<td>Char</td>
<td>Yes</td>
</tr>
<tr>
<td>MLA_DEPT</td>
<td>Home Dept Code</td>
<td>PPPPER.HOME_DEPT</td>
<td>6</td>
<td>Char</td>
<td>Yes</td>
</tr>
<tr>
<td>MLA_Title_CODE</td>
<td>Appt Title Code</td>
<td>PPPAPP.TITLE_CODE</td>
<td>4</td>
<td>Char</td>
<td>Yes</td>
</tr>
<tr>
<td>MLA_APPT_TYPE</td>
<td>Appt Type Code</td>
<td>PPPAPP.APPT_TYPE</td>
<td>1</td>
<td>Char</td>
<td>Yes</td>
</tr>
<tr>
<td>MLA_GRADE</td>
<td>Salary Grade</td>
<td>PPPAPP.GRADE</td>
<td>2</td>
<td>Char</td>
<td>Yes</td>
</tr>
<tr>
<td>MLA_RATE_CODE</td>
<td>Appt Rate Code</td>
<td>PPPAPP.RATE_CODE</td>
<td>1</td>
<td>Char</td>
<td>Yes</td>
</tr>
<tr>
<td>MLA_PAY_SCHEDULE</td>
<td>Appt Pay Sched Code</td>
<td>PPPAPP.PAY_SCHEDULE</td>
<td>2</td>
<td>Char</td>
<td>Yes</td>
</tr>
<tr>
<td>MLA_DIST_PAYRATE</td>
<td>Distribution Pay Rate</td>
<td>PPPDIS.DIST_PAYRATE</td>
<td></td>
<td>Num</td>
<td>Yes</td>
</tr>
<tr>
<td>MLA_DIST_STEP</td>
<td>Salary Step</td>
<td>PPPDIS.DIST_STEP</td>
<td>3</td>
<td>Char</td>
<td>Yes</td>
</tr>
<tr>
<td>MLA_SUB_LOCATION</td>
<td>Sub Location</td>
<td>PPPAPP.APPT_SUB_LOCATIO N</td>
<td>2</td>
<td>Char</td>
<td>Yes</td>
</tr>
<tr>
<td>MLA_APPT_REP_CODE</td>
<td>Appt Rep Code</td>
<td>PPPAPP.APPT_REP_CODE</td>
<td>1</td>
<td>Char</td>
<td>Yes</td>
</tr>
<tr>
<td>MLA_DIST_UNIT_CODE</td>
<td>Distribution Unit Code</td>
<td>PPPDIS.DIST_UNIT_CODE</td>
<td>1</td>
<td>Char</td>
<td>No</td>
</tr>
<tr>
<td>MLA_DIST_PERCENT</td>
<td>Dist Percent Time</td>
<td>PPPDIS.DIST_PERCENT</td>
<td></td>
<td>Num</td>
<td>No</td>
</tr>
<tr>
<td>MLA_ANNUAL_SALARY</td>
<td>Annualized Salary</td>
<td>Computed Using other fields</td>
<td></td>
<td>Num</td>
<td>No</td>
</tr>
<tr>
<td>MLA_NEWANN_SALARY</td>
<td>New Annualized Salary</td>
<td>Computed Using other fields. It is MLA_ANNUAL_SALARY at the first time.</td>
<td></td>
<td>Num</td>
<td>No</td>
</tr>
<tr>
<td>MLA_NEWNXT_SAL_DT</td>
<td>New Next Salary Review Date</td>
<td>PPPPER.NEXT_SALREV_DATE</td>
<td>10</td>
<td>Date</td>
<td>No</td>
</tr>
<tr>
<td>MLA_NEW_PAYRATE</td>
<td>New Pay Rate</td>
<td>MLA_DIST_PAYRATE is the default at the first time.</td>
<td></td>
<td>Num</td>
<td>No</td>
</tr>
<tr>
<td>Merit Logical Appt Table</td>
<td>Description of Field Name</td>
<td>Data Extracted From the Following EDB Table.Field</td>
<td>Len</td>
<td>Fmt</td>
<td>Key</td>
</tr>
<tr>
<td>--------------------------</td>
<td>----------------------------</td>
<td>-----------------------------------------------</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
</tr>
<tr>
<td>MLA_NEW_STEP</td>
<td>New Step</td>
<td>MLA-DIST-STEP is the default at the first time.</td>
<td>3</td>
<td>Char</td>
<td>No</td>
</tr>
<tr>
<td>MLA_PCT_INCREASE</td>
<td>% Inc of Pay Rate</td>
<td>Default is 0% at the first time.</td>
<td>99.99</td>
<td>Num</td>
<td>No</td>
</tr>
<tr>
<td>MLA_EVAL_DATE</td>
<td>Evaluation Date</td>
<td>Default is Low DB2 Date</td>
<td>10</td>
<td>Date</td>
<td>No</td>
</tr>
<tr>
<td>MLA_PERFORM_RATING</td>
<td>Perf Rating</td>
<td>Default is Blank</td>
<td>3</td>
<td>Char</td>
<td>No</td>
</tr>
<tr>
<td>MLA_EMP_NAME</td>
<td>Employee Name</td>
<td>PPPPER.EMP_NAME</td>
<td>26</td>
<td>Char</td>
<td>No</td>
</tr>
<tr>
<td>MLA_DEPT_NAME</td>
<td>Department Name</td>
<td>PPHME.HME.DEPT_NAME</td>
<td>30</td>
<td>Char</td>
<td>No</td>
</tr>
<tr>
<td>MLA_APPT_TITLE_NAME</td>
<td>Appt Title Code Name</td>
<td>PPPTCI.TCI_TITLE_NM_ABBREV</td>
<td>30</td>
<td>Char</td>
<td>No</td>
</tr>
<tr>
<td>MLA_CHANGED_BY</td>
<td>User ID</td>
<td>Session User ID</td>
<td>8</td>
<td>Char</td>
<td>No</td>
</tr>
<tr>
<td>MLA_CHANGED_AT</td>
<td>Date and Time</td>
<td>CURRENT_TIMESTAMP</td>
<td>26</td>
<td>Time Stmp</td>
<td>No</td>
</tr>
<tr>
<td>MLA_COMMENTS_FLAG</td>
<td>Comments Flag</td>
<td>D, if Row is Deleted</td>
<td>1</td>
<td>Char</td>
<td>No</td>
</tr>
<tr>
<td>MLA_MAX_RATE_FLAG</td>
<td>New Pay Rate at TCT maximum Flag</td>
<td>Y, if a rate increase was limited to TCT maximum step/rate. Default is Blank.</td>
<td>1</td>
<td>Char</td>
<td>No</td>
</tr>
<tr>
<td>MLA_SELF_UPD_FLAG</td>
<td>User Updated OwnMerit Flag</td>
<td>Y, if user updated Employee ID associated with ARSM PERSONAL Rule Default is Blank</td>
<td>1</td>
<td>Char</td>
<td>No</td>
</tr>
</tbody>
</table>

The details of the computed fields are as below:

<table>
<thead>
<tr>
<th>Merit Emp Table PPPMLA Field Name</th>
<th>How the calculation is done</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLA_ANNUAL_SALARY (Annualized Salary)</td>
<td>MLA_ANNUAL_SALARY is calculated as described in the following table and it is rounded to the nearest whole dollar amount.</td>
</tr>
<tr>
<td>This is calculated from MLA_DIST_PAYRATE, MLA_RATE_CODE, MLA_PAY_SCHEDULE, STD_HRS_PER_YEAR and MLA_DIST_PERCENT.</td>
<td>MLA_RATE_CODE</td>
</tr>
<tr>
<td>Note: STD-HRS-PER-YEAR is obtained from the System Parameter table (PPPPRM) for parameter number 041.</td>
<td>A or B</td>
</tr>
<tr>
<td>MLA_NEWANN_SALARY (New Annualized Salary)</td>
<td>MLA_NEWANN_SALARY is calculated as described in the following table and it is rounded to the nearest whole dollar amount.</td>
</tr>
<tr>
<td>This is calculated from MLA_NEW_PAYRATE, MLA_RATE_CODE, MLA_PAY_SCHEDULE, STD_HRS_PER_YEAR and MLA_DIST_PERCENT.</td>
<td>MLA_RATE_CODE</td>
</tr>
<tr>
<td>Note: STD-HRS-PER-YEAR is obtained from the System Parameter table (PPPPRM) for parameter number 041.</td>
<td>A or B</td>
</tr>
<tr>
<td></td>
<td>SM</td>
</tr>
<tr>
<td></td>
<td>BW</td>
</tr>
<tr>
<td></td>
<td>H</td>
</tr>
</tbody>
</table>
### 3. PPPMAP - New Merit Appointment Table

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description of Field Name</th>
<th>Data Extracted From the Following EDB Table.Field</th>
<th>Len</th>
<th>Fmt</th>
<th>Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAP_CYCLE_ID</td>
<td>Merit Cycle ID</td>
<td>Foreign Key with Cascade Delete PPCMCC.MCC_CYCLE_ID.</td>
<td>18</td>
<td>Char</td>
<td>Yes</td>
</tr>
<tr>
<td>MAP_EMPLOYEE_ID</td>
<td>Employee ID</td>
<td>PPPAPP.EMPLOYEE_ID</td>
<td>9</td>
<td>Char</td>
<td>Yes</td>
</tr>
<tr>
<td>MAP_DEPT</td>
<td>Appt Dept Code</td>
<td>PPPAPP.APPT_DEPT</td>
<td>6</td>
<td>Char</td>
<td>Yes</td>
</tr>
<tr>
<td>MAP&gt;Title_CODE</td>
<td>Appt Title Code</td>
<td>PPPAPP.TITLE_CODE</td>
<td>4</td>
<td>Char</td>
<td>Yes</td>
</tr>
<tr>
<td>MAP_SUB_LOCATION</td>
<td>Sub Location</td>
<td>PPPAPP.APPT_SUB_LOCATION</td>
<td>2</td>
<td>Char</td>
<td>No</td>
</tr>
<tr>
<td>MAP_APPT_REP_CODE</td>
<td>Appt Rep Code</td>
<td>PPPAPP.APPT_REP_CODE</td>
<td>1</td>
<td>Char</td>
<td>No</td>
</tr>
<tr>
<td>MAP_APPT_TYPE</td>
<td>Appt Type Code</td>
<td>PPPAPP.APPT_TYPE</td>
<td>1</td>
<td>Char</td>
<td>Yes</td>
</tr>
<tr>
<td>MAP_GRADE</td>
<td>Salary Grade</td>
<td>PPPAPP.GRADE</td>
<td>2</td>
<td>Char</td>
<td>Yes</td>
</tr>
<tr>
<td>MAP_RATE_CODE</td>
<td>Appt Rate Code</td>
<td>PPPAPP.RATE_CODE</td>
<td>1</td>
<td>Char</td>
<td>Yes</td>
</tr>
<tr>
<td>MAP_PAY_SCHEDULE</td>
<td>Appt Pay Schedule Code</td>
<td>PPPAPP.PAY_SCHEDULE</td>
<td>2</td>
<td>Char</td>
<td>Yes</td>
</tr>
<tr>
<td>MAP_APPT_NUM</td>
<td>Appt Number</td>
<td>PPPAPP.APPT_NUM</td>
<td>2</td>
<td>Num</td>
<td>Yes</td>
</tr>
<tr>
<td>MAP_APPT_DURATION</td>
<td>Duration of Appointment</td>
<td>PPPAPP.APPT_DURATION</td>
<td>1</td>
<td>Char</td>
<td>No</td>
</tr>
<tr>
<td>MAP_DEPT_NAME</td>
<td>Appt Dept Name</td>
<td>PPPHME.HME_DEPT_NAME</td>
<td>30</td>
<td>Char</td>
<td>No</td>
</tr>
<tr>
<td>MAP_APPT_BEGIN_DT</td>
<td>Appt Begin Date</td>
<td>PPPAPP.APPT_BEGIN_DATE</td>
<td>10</td>
<td>Date</td>
<td>No</td>
</tr>
<tr>
<td>MAP_APPT_END_DT</td>
<td>Appt End Date</td>
<td>PPPAPP.APPT_END_DATE</td>
<td>10</td>
<td>Date</td>
<td>No</td>
</tr>
<tr>
<td>MAP_TITLE_UNIT_CD</td>
<td>Title Unit Code</td>
<td>PPPAPP.TITLE_UNIT_CODE (Match to MCC_CYCLE_CBUC)</td>
<td>2</td>
<td>Char</td>
<td>No</td>
</tr>
</tbody>
</table>

### 4. PPPMED - New Merit Distribution Table

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description of Field Name</th>
<th>Data Extracted From the Following EDB Table.Field</th>
<th>Len</th>
<th>Fmt</th>
<th>Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>MED_CYCLE_ID</td>
<td>Merit Cycle ID</td>
<td>Foreign Key with Cascade Delete PPCMCC.MCC_CYCLE_ID.</td>
<td>18</td>
<td>Char</td>
<td>Yes</td>
</tr>
<tr>
<td>MED_EMPLOYEE_ID</td>
<td>Employee ID</td>
<td>PPDIS.EMPLOYEE_ID</td>
<td>9</td>
<td>Char</td>
<td>Yes</td>
</tr>
<tr>
<td>MED_HOME_DEPT</td>
<td>Home Dept Code</td>
<td>PPRPER.HOME_DEPT</td>
<td>6</td>
<td>Char</td>
<td>Yes</td>
</tr>
<tr>
<td>MED_APPT_DEPT</td>
<td>Appt Dept Code</td>
<td>PPPAPP.APPT_DEPT</td>
<td>6</td>
<td>Char</td>
<td>Yes</td>
</tr>
<tr>
<td>MED_APPT_TITLE</td>
<td>Appt Title Code</td>
<td>PPPAPP.TITLE_CODE</td>
<td>4</td>
<td>Char</td>
<td>Yes</td>
</tr>
<tr>
<td>MED_APPT_SUB_LOCN</td>
<td>Sub Location</td>
<td>PPPAPP.APPT_SUB_LOCATION</td>
<td>2</td>
<td>Char</td>
<td>Yes</td>
</tr>
<tr>
<td>MED_APPT_REP_CODE</td>
<td>Appt Rep Code</td>
<td>PPPAPP.APPT_REP_CODE</td>
<td>1</td>
<td>Char</td>
<td>Yes</td>
</tr>
<tr>
<td>MED_APPT_TYPE</td>
<td>Appt Type Code</td>
<td>PPPAPP.APPT_TYPE</td>
<td>1</td>
<td>Char</td>
<td>Yes</td>
</tr>
<tr>
<td>MED_SALARY_GRADE</td>
<td>Salary Grade</td>
<td>PPPAPP.GRADE</td>
<td>2</td>
<td>Char</td>
<td>Yes</td>
</tr>
<tr>
<td>MED_APPT_RATE_CODE</td>
<td>Appt Rate Code</td>
<td>PPPAPP.RATE_CODE</td>
<td>1</td>
<td>Char</td>
<td>Yes</td>
</tr>
<tr>
<td>MED_PAY_SCHEDULE</td>
<td>Appt Pay Schedule Code</td>
<td>PPPAPP.PAY_SCHEDULE</td>
<td>2</td>
<td>Char</td>
<td>Yes</td>
</tr>
</tbody>
</table>
### Merit Emp Table PPPMED

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description of Field Name</th>
<th>Data Extracted From the Following EDB Table.Field</th>
<th>Len</th>
<th>Fmt</th>
<th>Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>MED_DIST_RATE</td>
<td>Distribution Pay Rate</td>
<td>PPPDIS.DIST_PAYRATE</td>
<td>S999</td>
<td>Num</td>
<td>Yes</td>
</tr>
<tr>
<td>MED_DIST_STEP</td>
<td>Salary Step</td>
<td>PPPDIS.DIST_STEP</td>
<td>3</td>
<td>Char</td>
<td>Yes</td>
</tr>
<tr>
<td>MED_DIST_NUM</td>
<td>Dist Number</td>
<td>PPPDIS.DIST_NUM</td>
<td>2</td>
<td>Num</td>
<td>Yes</td>
</tr>
<tr>
<td>MED_APPT_TITLE_NM</td>
<td>Appt Title Code Name</td>
<td>PPPTCI.TCI_TITLE_NM_ABBRV</td>
<td>30</td>
<td>Char</td>
<td>No</td>
</tr>
<tr>
<td>MED_PAY_BEGIN_DATE</td>
<td>Dist Begin Date</td>
<td>PPPDIS.PAY_BEGIN_DATE</td>
<td>10</td>
<td>Date</td>
<td>No</td>
</tr>
<tr>
<td>MED_PAY_END_DATE</td>
<td>Dist End Date</td>
<td>PPPDIS.PAY_END_DATE</td>
<td>10</td>
<td>Date</td>
<td>No</td>
</tr>
<tr>
<td>MED_FULL_ACCT_UNIT</td>
<td>Full Acct Unit</td>
<td>PPPDIS.FULL_ACCT_UNIT</td>
<td>30</td>
<td>Char</td>
<td>No</td>
</tr>
<tr>
<td>MED_DIST_PERCENT</td>
<td>Dist Percent Time</td>
<td>PPPDIS.DIST_PERCENT</td>
<td>S9.9</td>
<td>Num</td>
<td>No</td>
</tr>
<tr>
<td>MED_DIST_DOS</td>
<td>Dist DOS Code</td>
<td>PPPDIS.DIST_DOS</td>
<td>3</td>
<td>Char</td>
<td>No</td>
</tr>
</tbody>
</table>

### Merit Cycle Tables

#### PPPMCC - Web Merit Cycle Control Table

The data for this table will be populated from the user inputs on the Merit Administration Screen.

<table>
<thead>
<tr>
<th>Merit Cycle Control Table PPPMCC Fields</th>
<th>Description of Field Name</th>
<th>Le</th>
<th>Fmt</th>
<th>Data Matched with following fields of EDB</th>
<th>Sel</th>
<th>key</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCC_CYCLE_ID</td>
<td>Merit Cycle ID</td>
<td>18</td>
<td>Char</td>
<td>MEE_CYCLE_ID, MLA_CYCLE_ID, MAP_CYCLE_ID, MED_CYCLE_ID</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>MCC_CYCLE_DATE</td>
<td>Merit Cycle Date</td>
<td>10</td>
<td>Date</td>
<td>PPPDIS.PAY_END_DATE, PPPAPP.APPT_END_DATE</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>MCC_CBUC</td>
<td>Title Unit Code</td>
<td>2</td>
<td>Char</td>
<td>PPPAPP.TITLE_UNIT_CODE</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>MCC_SUB_LOCATION</td>
<td>Local Sub-Location</td>
<td>2</td>
<td>Char</td>
<td>PPPAPP.APPT_SUB_LOCATION</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>MCC_PROG_TYPE</td>
<td>Program Type (User Input)</td>
<td>3</td>
<td>Char</td>
<td>PSS - Professional &amp; Support Staff, MSP - Managers and Senior Professionals, SMG - Senior Management Group, BGU - Bargaining Unit</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>MCC_STATUS_CD</td>
<td>Merit Cycle Status</td>
<td>1</td>
<td>Char</td>
<td>User Input. Extract process will start if this is P or F. After processing, it will change them as below: P -&gt; H and F -&gt; S</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>MCC_DESCRIPTION</td>
<td>Cycle Description</td>
<td>30</td>
<td>Char</td>
<td>User Input</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>MCC_STEP_RANGE</td>
<td>Merit Cycle Type</td>
<td>1</td>
<td>Char</td>
<td>User Input</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>MCC_COLLAPSE</td>
<td>Appt/Dept Collapsing Basis in Logical Appt</td>
<td>1</td>
<td>Char</td>
<td>H -&gt; Home, A -&gt; Appt Dept</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>MCC_NEXT_SALREV_DT</td>
<td>Next Salary Rev Date</td>
<td>10</td>
<td>Date</td>
<td>PPPPER.NEXT_SALREV_DATE</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>
### Merit Cycle Control Table: PPPMCC Fields

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description of Field Name</th>
<th>Le</th>
<th>Fmt</th>
<th>Data Matched with following fields of EDB</th>
<th>Sel</th>
<th>key</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCC_SALREV_TYPE1</td>
<td>Next Salary Review Type</td>
<td>1</td>
<td>Char</td>
<td>PPPPER.NEXT_SALARY_REV 1. 6-Month Increase 2. Merit Increase 3. Special performance award 4. Trainee Increase 5. No salary review – At Maximum</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>MCC_SALREV_TYPE2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MCC_SALREV_TYPE3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MCC_SALREV_TYPE4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MCC_SALREV_TYPE5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MCC_SALREV_TYPE6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MCC_SALREV_TYPE7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MCC_SALREV_TYPE8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MCC_APPT_TYPE2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MCC_APPT_TYPE3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MCC_APPT_TYPE4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MCC_APPT_TYPE5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MCC_APPT_TYPE6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MCC_APPT_TYPE7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MCC_APPT_TYPE8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MCC_OLD_REV_DT</td>
<td>Old Review Date</td>
<td>10</td>
<td>Date</td>
<td>Not used for selection</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>MCC_HIRE_DATE</td>
<td>Employee Hire Date</td>
<td>10</td>
<td>Date</td>
<td>Used for Employee Level Selection Criteria</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>MCC_PROB_DATE</td>
<td>Employee Probation Date</td>
<td>10</td>
<td>Date</td>
<td>Used for Employee Level Selection Criteria</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>MCC_APPT_REP_CODE</td>
<td>Appt Rep Code</td>
<td>3</td>
<td>Char</td>
<td>PPPAPP.APPT-REP-CODE COV=Covered UNC=Uncovered, Supervisor ALL = COV and UNC</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>MCC_AT_INCL_EXCL</td>
<td>Additional Titles Range is Limited to or not present</td>
<td>1</td>
<td>Char</td>
<td>User Input. Valid Values are L - Additional Titles Limited to E - Additional Titles Exclude Blank - No Additional Titles</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>MCC_BW_EFF_DATE</td>
<td>BW Effective Date</td>
<td>10</td>
<td>Date</td>
<td>User Entry</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>MCC_MO_EFF_DATE</td>
<td>MO Effective Date</td>
<td>10</td>
<td>Date</td>
<td>User Entry</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>MCC_CONTROL_PCT</td>
<td>Control Percent</td>
<td>4</td>
<td>Num</td>
<td>99V99</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>MCC_MO_REMAIN_FY</td>
<td>Months Remaining in the Fiscal Yr</td>
<td>2</td>
<td>Num</td>
<td>Derived from MCC_CYCLE_DATE</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>MCC_NEW_SALREV_DT</td>
<td>New Next Sal Rev Date</td>
<td>10</td>
<td>Date</td>
<td>User Entry</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>MCC_CHANGED_BY</td>
<td>User ID of Update</td>
<td>8</td>
<td>Char</td>
<td>Current User ID who is updating</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>MCC_CHANGED_AT</td>
<td>Date and Time of Update</td>
<td>26</td>
<td>Char</td>
<td>Current Timestamp while adding or updating.</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

### PPPMAT - Merit Cycle Additional Titles Table

The data for this table will be populated from the user inputs on the Merit Administration Screen. Additional Titles table(PPPMAT) will be used by the extract process to know which title code ranges should be included or excluded from the selection.
<table>
<thead>
<tr>
<th>Merit Cycle Additional Titles Table</th>
<th>PPPMAT Field Name</th>
<th>Description of Field Name</th>
<th>Len</th>
<th>Fmt</th>
<th>Source of Data</th>
<th>Def</th>
<th>Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAT_CYCLE_ID</td>
<td>Merit Cycle ID</td>
<td>18 Char Foreign Key with Cascade Delete</td>
<td>PPPMCC.MCC_CYCLE_ID</td>
<td>No</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAT_RANGE_BEGIN</td>
<td>Begin Title Code Range</td>
<td>4 Char User Input.</td>
<td>No</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAT_RANGE_END</td>
<td>End Title Code Range</td>
<td>4 Char User Input.</td>
<td>No</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**FILE Definitions**

**CARDFILE**
The DD name **CARDFIL** will be defined for the input Specification Record File.

WIA-RUN-SPEC-RECORD.
- WIA-PROG-ID   X(11)
- WIA-CYCLE-ID X(18)
- FILLER       X(51)

WIA-PROG-ID should have a valid value of ‘PPP675-SPEC’ for the program to proceed further to process the Cycle ID specified in WIA-CYCLE-ID.
The total record Length is 80.

**REPORT-FILE**
The DD name **PRNTFIL** will be defined for the output Report File.

PRINT-REC.
- CARRIAGE-CTL-CHAR X
- PRINT-LN          X(132)
The total record Length is 133.

This is the normal reporting file and the sample output is shown in Attachment A.

**Includes**

<table>
<thead>
<tr>
<th>Table</th>
<th>Include</th>
</tr>
</thead>
<tbody>
<tr>
<td>EDB Employee Table</td>
<td>PPPVZPER</td>
</tr>
<tr>
<td>EDB Appointment Table</td>
<td>PPPVZAPP</td>
</tr>
<tr>
<td>EDB Distribution Table</td>
<td>PPPVZDIS</td>
</tr>
<tr>
<td>CTL Bargaining Unit Code Table</td>
<td>PPPVZBUT</td>
</tr>
<tr>
<td>CTL Home Dept Table</td>
<td>PPPVZHME</td>
</tr>
<tr>
<td>CTL Parameter Table</td>
<td>PPPVZPRM</td>
</tr>
<tr>
<td>CTL Title Code Table</td>
<td>PPPVZTCI</td>
</tr>
<tr>
<td>Merit Cycle Control Table</td>
<td>PPPVZMCC</td>
</tr>
<tr>
<td>Merit Cycle Additional Titles Table</td>
<td>PPPVZMAT</td>
</tr>
<tr>
<td>Merit Employee Table</td>
<td>PPPVZMEE</td>
</tr>
<tr>
<td>Merit Appointment Table</td>
<td>PPPVZMAP</td>
</tr>
<tr>
<td>Merit Distribution Table</td>
<td>PPPVZMED</td>
</tr>
</tbody>
</table>
Cursor Declaration

A cursor PER_ROW will be declared for selecting Employee Details from the PPPPER table for an Employee whose Next Salary Review Date is <= Merit Next Salary Review Date and who has at least one Appointment in the PPPAPP table which has Appointment End Date >= Merit Cycle Date.

(SELECT
EMPLOYEE_ID, EMP_NAME, NAMESUFFIX,
BIRTH_DATE, HIRE_DATE, HOME_DEPT,
EMP_STATUS, NEXT_SALARY_REV, NEXT_SALREV_DATE,
EMP_REL_CODE, EMP_Rel_UNIT, EMP_SPEC_HAND,
EMP_REP_CODE, PRIMARY_TITLE, PROB_END_DATE,
FIRST_NAME, MIDDLE_NAME, LAST_NAME
FROM PPPVZPER_PER PER
WHERE NEXT_SALREV_DATE <= :MCC-NEXT-SALREV-DT
AND 1 <= (SELECT COUNT(*) FROM PPPVZAPP_APP
WHERE EMPLOYEE_ID = PER.EMPLOYEE_ID
AND ((PAY_SCHEDULE = 'BW' AND
APPT_END_DATE >= :MCC-BW-EFF-DATE)
OR
(APPT_END_DATE >= :MCC-MO-EFF-DATE))
ORDER BY EMPLOYEE_ID)

A cursor APPT_ROW will be declared for selecting Appointment Details for the Employee ID returned from PER_ROW from the table PPPAPP whose Next Salary Review Date is <= Merit Next Salary Review Date and has a Pay End Date equal to or greater than the MO or BW Effective Date, whichever is appropriate for the Appointment Pay Schedule.

(SELECT
APPT_NUM, RATE_CODE, PAY_SCHEDULE,
APPT_BEGIN_DATE, APPT_END_DATE, APPT_DURATION,
TITLE_CODE, APPT_TYPE, PERSONNEL_PGM,
TITLE_UNIT_CODE, APPT_REP_CODE, APPT_DEPT,
GRADE, APPT_SUB_LOCATION
FROM PPPVZAPP_APP APP
WHERE EMPLOYEE_ID = :WA-EMPLOYEE-ID
AND ((PAY_SCHEDULE = 'BW' AND
APPT_END_DATE >= :MCC-BW-EFF-DATE)
OR
(APPT_END_DATE >= :MCC-MO-EFF-DATE))
ORDER BY APPT_NUM ASC)

A Cursor DIS_ROW will be declared for selecting Distribution Details for the Appointment returned from APPT_ROW from the table PPPDIS which has Pay End Date >= Merit Cycle Date.

(SELECT
DIST_NUM, PAY_BEGIN_DATE, PAY_END_DATE,
DIST_PERCENT, DIST_PAYRATE, DIST_UNIT_CODE,
DIST_STEP, FULL_ACCT_UNIT
FROM PPPVZDIS_DIS DIS
WHERE EMPLOYEE_ID  =  :WA-EMPLOYEE-ID
    AND APPT_NUM     =  :WA-APPT-NUM
    AND PAY_END_DATE >= :WA-DIST-EFF-DATE
    AND 'D' <> (SELECT DOS_PAY_CATEGORY
                FROM PPPVZDOS_DOS
                WHERE DOS_EARNINGS_TYPE = DIS.DIST_DOS)
ORDER BY DIST_NUM ASC)

A Cursor BUT_ROW will be declared for selecting Bargaining Unit Codes defined for the Merit System on the table PPPBUT.

(SELECT BUT_BUC
FROM PPPVZBUT_BUT
WHERE BUT_MERIT_SELECT = 'Y'
ORDER BY BUT_BUC ASC)

A Cursor MAT_ROW will be declared for selecting all Additional Title Codes defined for the Merit Cycle.

(SELECT MAT_RANGE_BEGIN,MAT_RANGE_END
FROM PPPVZMAT_MAT
WHERE MAT_CYCLE_ID = :MCC-CYCLE-ID
ORDER BY MAT_RANGE_BEGIN)

MED_APPT_SUB_LOCN
MED_APPT_REP_CODE
A Cursor MEDLH_ROW will be declared for selecting Distinct Logical Appointments from the PPPMED table for an Employee, Cycle ID, Home Department, Appt Title, Appt Sub Location, Appt Rep Code, Appt Title Name, Appt Type, Salary Grade, Appt Rate Code, Pay Schedule, Dist Rate and Dist Step.

(SELECT DISTINCT
MED_CYCLE_ID,   MED_EMPLOYEE_ID,
MED_HOME_DEPT,  MED_APPT_TITLE,
MED_APPT_SUB_LOCN, MED_APPT_REP_CODE
MED_APPT_TITLE_NM,  MED_APPT_TYPE,
MED_SALARY_GRADE, MED_APPT_RATE_CODE, MED_PAY_SCHEDULE,
MED_DIST_RATE, MED_DIST_STEP,  SUM(MED_DIST_PERCENT)
FROM PPPVZMED_MED
WHERE MED_CYCLE_ID    = :MCC-CYCLE-ID
    AND MED_EMPLOYEE_ID = :WA-EMPLOYEE-ID
GROUP BY   MED_CYCLE_ID,   MED_EMPLOYEE_ID,
           MED_HOME_DEPT,  MED_APPT_TITLE,
           MED_APPT_SUB_LOCN, MED_APPT_REP_CODE,
           MED_APPT_TITLE_NM,  MED_APPT_TYPE,
           MED_SALARY_GRADE, MED_APPT_RATE_CODE,
           MED_PAY_SCHEDULE, MED_DIST_RATE,
           MED_DIST_STEP
ORDER BY MED_HOME_DEPT ASC)

A Cursor MEDLA_ROW will be declared for selecting Distinct Logical Appointments from the PPPMED table for an Employee, Cycle ID, Appointment Department, Appt Title, Appt Type, Salary Grade, Appt Rate Code, Pay Schedule, Dist Rate and Dist Step.
(SELECT DISTINCT
MED_CYCLE_ID, MED_EMPLOYEE_ID, MED_APPT_DEPT,
MED_APPT_TITLE, MED_APPT_SUB_LOCN, MED_APPT_REP_CODE
MED_APPT_TITLE_NM, MED_APPT_TYPE,
MED_SALARY_GRADE, MED_APPT_RATE_CODE, MED_PAY_SCHEDULE,
MED_DIST_RATE, MED_DIST_STEP, SUM(MED_DIST_PERCENT)
FROM PPPVZMED_MED
WHERE MED_CYCLE_ID = :MCC-CYCLE-ID
AND MED_EMPLOYEE_ID = :WA-EMPLOYEE-ID
GROUP BY MED_CYCLE_ID, MED_EMPLOYEE_,
MED_APPT_DEPT, MED_APPT_TITLE,
MED_APPT_SUB_LOCN, MED_APPT_REP_CODE,
MED_APPT_TITLE_NM, MED_APPT_TYPE,
MED_SALARY_GRADE, MED_APPT_RATE_CODE,
MED_PAY_SCHEDULE, MED_DIST_RATE,
MED_DIST_STEP
ORDER BY MED_APPT_DEPT ASC)

Processing Logic

Initiate the Processing:

To initiate the processing of the Merit Cycle Extract, the Merit Cycle Status (MCC_STATUS_CD) has to have the following values:
P → Run/Release Preliminary Extract or F → Run/Release Final Extract

After successful processing of the Merit Cycle Extract, the Merit Cycle Status has to be changed to:
P → H (Open for HR) or F → S (Open for HR, Control Point)

Once the Merit Cycle Status is identified as P or F then the following steps are followed to extract the records from the EDB database.

Steps for Extracting Data:

Step 1:

Write the headers for the output.

Read the First Input Record from the Specification File.
If the input Specification record is valid, then Load the Valid CBUC values for the campus into an array from PPPBUT table and get the Standard Hours Per Year from PPPPRM table for parameter +041.

Repeat Performing Step 2 until no more Valid Specification Record.


Step 2:

Using the Cycle ID Specified in the Specification File, Get all details of this Cycle ID from PPPMCC Table.
If the Cycle Status is ‘P’ or ‘F’, then perform the following steps:
   i. Update the Cycle Status as P → H or F → S.
ii. Print the Header Details for the Cycle ID. Also initialize the Grand Totals for the Cycle and Totals for the First Employee to Zero.

iii. Delete all previously extracted records from the tables PPPMEE, PPPMAP, PPPMED and PPPMLA for this Cycle ID.

iv. Load the Additional Titles from Table PPPMAT into an Array for this Cycle ID.

v. Get the First Employee who has a Salary Review Date <= Next Salary Review Date of the Cycle AND who has at least one Appointment with the Appt End Date >= Cycle Date.

vi. Repeat Performing Step 3 until no more Employee Found who meets the above Criteria.

vii. If the total number of Employees Written into PPPMEE, WA-TOT-CYC-EMP > 0, Write the Grand Totals for the whole Merit Cycle.

Read the Next Input Record from the Specification File.

Step 3:

Do the Employee Level Screening Steps E-1 thru E-5 mentioned below so that Employee’s Status, Employee’s Salary Review Type, Hire Date and Probation Date lie within the Selection Criteria. If the Employee is selected, then perform the following steps:

i. Get the First Appointment for this Employee which has the Appt End Date >= Cycle Date and has at least one Distribution with Pay End Date >= Cycle Date.

ii. Repeat Performing Step 4 until no more Appointment found with the above Criteria.

iii. If WA-TOT-CYC-EMP-APPT > 0, then we need to write the Employee Details into PPPMEE table. We also have to get the Logical Appointments for this Employee and write them into PPPMLA table.

a. Populate all the fields necessary for the PPPMEE table from the current EDB extracted fields.

b. Insert this record into PPPMEE table.

c. After successful insertion of this record, increment WA-TOT-CYC-EMP which is the count of number of Employees written for the Merit Cycle.

d. Depending upon the Collapsing Option(Home/Appt), get the distinct rows from the PPPMED table for the Employee and Cycle. For each Logical row, repeat performing the following Steps:

   1. Populate all the fields necessary for the PPPMLA table from the current EDB extracted fields.

   2. Insert this record into PPPMLA table.

   3. After successful insertion of this record, increment WA-TOT-CYC-EMP-APPT-LGL which is the count of number of Logical Appointments written for the Employee.


e. Write the Totals for the Employee and also add the totals to the Grand Totals of the Merit Cycle.

Get the Next Employee who has a Salary Review Date <= Next Salary Review Date of the Cycle AND who has at least one Appointment with the Appt End Date >= Cycle Date.

Step 4:

Do the Appointment Level Screening Steps A-1 thru A-8 mentioned above so that Appointment’s TUC, Representation Code, Appt Type, Sub Location and Additional Titles match the Selection Criteria. If this Appointment is selected, then perform the following steps:

i. Initialize the working storage field WA-TOT-APPT-DIS to zero.

ii. Get the First Distribution for this Employee’s Appointment which has Pay End Date >= Cycle Date.
iii. **Repeat** Performing **Step 5 until no more Distribution** found with the above Criteria.

iv. If WA-TOT-APPT-DIS > 0, then we need to write the Appt details into PPPMAP table.
   a. Add the WA-TOT-APPT-DIS to WA-TOT-CYC-EMP-APPT-DIS, which is the total Distributions written for the Employee.
   b. Populate all the fields necessary for the PPPMAP table from the current EDB extracted fields.
   c. Insert this record into PPPMAP table.
   d. After successful insertion of this record, increment WA-TOT-CYC-EMP-APPT which is the count of number of Appointments written for the Employee.

Get the Next Appointment for this Employee which has the Appt End Date >= MCC MO or BW Effective.

**Step 5:**

Now we have a valid Employee-Appointment-Distribution link which is eligible to be extracted from EDB to the Merit Tables. So as a first step we need to proceed further to write the PPPMED, Merit Distribution Table.

i. Populate all the fields necessary for the PPPMED table from the current EDB extracted fields.

ii. Insert this record into PPPMED table.

iii. After successful insertion of this record, increment WA-TOT-APPT-DIS which is the count of number of Distributions written for an Appointment.

Get the Next Distribution for this Employee’s Appointment which has Pay End Date >= Cycle Date.

**Selection Process Steps:**

**Employee Level Steps:**

**Step E-1:**

**Employee Status**

If PPPPER.EMP-STATUS =( ‘A’ or ‘N’ or ‘P’)
Then
   Allow the Employee to the Next Step.

**Step E-2:**

**Next Salary Review Date**

If MCC_NEXT_SALREV_DT >= PPPPER.NEXT_SALREV_DATE
Then
   Allow the Employee to the Next Step.

**Step E-3:**

**Salary Review Types**

If ( (PPPPER.NEXT_SALARY_REV = 1) AND (MCC_SALREV_TYPE1 = ‘Y’) )  OR
 ( (PPPPER.NEXT_SALARY_REV = 2) AND (MCC_SALREV_TYPE2 = ‘Y’) )  OR
 ( (PPPPER.NEXT_SALARY_REV = 3) AND (MCC_SALREV_TYPE3 = ‘Y’) )  OR
( (PPPPER.NEXT_SALARY_REV = 4) AND (MCC_SALREV_TYPE4 = ‘Y’) ) OR
( (PPPPER.NEXT_SALARY_REV = 5) AND (MCC_SALREV_TYPE5 = ‘Y’) ) OR
Then
   Allow the Employee to the Next Step.

**Step E-4:**

**Hire Date -- Default Value is ‘01/01/0001’ if no value is entered in the Web Screen**

If (MCC_HIRE_DATE > ‘01/01/0001’) AND (MCC_HIRE_DATE >= PPPPER.HIRE_DATE)
Then
   Allow the Employee to the Next Step.

**Step E-5:**

**Probation Date -- Default Value is ‘01/01/0001’ if no value is entered in the Web Screen**

If (MCC_PROB_DATE > ‘01/01/0001’) AND (MCC_PROB_DATE >= PPPPER.PROB_END_DATE)
Then
   Allow the Employee to the Next Step.

**Appointment Level Steps:**

**Step A-1:**

**Appointment – End Date**

If PPPAPP.APPT_END_DATE >= MCC_BW_EFF_DATE or MCC_MO_EFF_DATE, as appropriate for Pay Schedule
Then
   Allow all records to the Next Step.

**Step A-2:**

**Appointment – Personal Program**

If MCC_PROG_TYPE = ‘PSS’
Then
   Allow Records with PPPAPP.PERSONNEL_PGM = ‘1’
Else
If MCC_PROG_TYPE = ‘MSP’
Then
   Allow Records with PPPAPP.PERSONNEL_PGM = ‘2’ AND PPPAPP.TITLE_CODE <> ‘0200’ THRU ‘0799’
Else
If MCC_PROG_TYPE = ‘SMG’
Then
   Allow Records with PPPAPP.PERSONNEL_PGM = ‘2’ AND PPPAPP.TITLE_CODE = ‘0001’ THRU ‘0199’
Else
If MCC_PROG_TYPE = ‘CBU’
Then
   Allow all records to the Next Step.

**Step A-3:**
Appointment – Appointment Representation Code

If MCC_APPT_REP_CODE = 'ALL' and MCC_PROG_TYPE = 'PSS' and MCC-CBUC = '***'
Then
   Allow Records with
   a. (APPT_REP_CODE = 'C' OR 'S' OR 'U') and MCC-CBUC = '99') OR
   b. (APPT_REP_CODE = 'U' OR 'S') and MCC-CBUC = All Valid BUC from PPPBUT
Skip Step A-4 and go to Step A-5
Else
   If MCC_APPT_REP_CODE = 'COV'
   Then
      Allow Records with PPPAPP.APPT_REP_CODE = 'C'
   Else
   If MCC_APPT_REP_CODE = 'UNC'
   Then
      Allow Records with PPPAPP.APPT_REP_CODE = 'U' OR 'S'
   Else
   If MCC_APPT_REP_CODE = 'ALL'
   Then
      Allow Records with PPPAPP.APPT_REP_CODE = 'C' OR 'S' OR 'U'.

Step A-4:

Title Unit Code – CBUC

If MCC_CBUC = '--'  \(\Rightarrow\) It means CBUC is N/A
Then
   Allow all records to the Next Step
Else
   If MCC_CBUC = '@@' -- (Value 'ALL' is stored as '@@')
   Then
      Allow Records with PPPAPP.TITLE_UNIT_CODE = All Valid CBUC for the Campus to Next Step
   Else
      Allow Records with PPPAPP.TITLE_UNIT_CODE = MCC_CBUC to the Next Step.

All valid CBUC for the Campus is arrived by getting the values of PPPBUT.BUT_BUC with the condition of the newly added field PPPBUT.BUT_MERIT_SELECT = 'Y'.

Step A-5:

Appointment – Appointment Rate Lookup

If MCC-STEP-RANGE = 'S' (Step Based)  Then   Allow Records with PPPAPP.APPT_TYPE = 'S'.
If MCC-STEP-RANGE = 'R' (Range Based)  Then   Allow Records with PPPAPP.APPT_TYPE = 'G' or 'M' or 'X' with PPPAPP.GRADE not blank.

Step A-6:

Appointment – Appointment Type

If MCC_APPT_TYPE1 = 'Y' (Contract)  Then   Allow Records with PPPAPP.APPT_TYPE = '1'.
If MCC_APPT_TYPE2 = 'Y' (Regular/Career)  Then   Allow Records with PPPAPP.APPT_TYPE = '2'.
If MCC_APPT_TYPE3 = 'Y' (Limited) Then Allow Records with PPPAPP.APPT_TYPE = '3'.
If MCC_APPT_TYPE4 = 'Y' (Casual/Restricted) Then Allow Records with PPPAPP.APPT_TYPE = '4'.
If MCC_APPT_TYPE5 = 'Y' (Academic) Then Allow Records with PPPAPP.APPT_TYPE = '5'.
If MCC_APPT_TYPE6 = 'Y' (Per Diem) Then Allow Records with PPPAPP.APPT_TYPE = '6'.
If MCC_APPT_TYPE7 = 'Y' (Partial Year/Career) Then Allow Records with PPPAPP.APPT_TYPE = '7'.
If MCC_APPT_TYPE8 = 'Y' (Floater) Then Allow Records with PPPAPP.APPT_TYPE = '8'.

Step A-7:
Appointment – Sub Location --- (ALL or any One of the Valid Values for the Campus)
If MCC_SUB_LOCATION = '@@' -- (Value ‘ALL’ is stored as ‘@@’)
Then
Allow All Records to Next Step
Else
Allow Records with PPPAPP.APPT_SUB_LOCATION = MCC_SUB_LOCATION to the Next Step.

Step A-8:
Additional Title Code Range(s) – ‘Limited to’ or ‘Exclude’
If MCC_AT_INCL_EXCL = ‘L’
Then
Allow all the rows with their Title Code lies between the possibly more than one Additional Title Range(s) entered by the user into the PPPMAT table for the Current Cycle ID.
Else
If MCC_AT_INCL_EXCL = ‘E’
Then
Allow all the rows with a Title Code which does not lie between the possibly more than one Additional Title Range(s) entered by the user into the PPPMAT table for the Current Cycle ID
Else
Accept all Title Codes.

Distribution Level Steps:

Step D-1:
Appointment – End Date
If PPPDIS.PAY_END_DATE >= MCC_BW_EFF_DATE or MCC_MO_EFF_DATE as appropriate per Pay Schedule
Then
Allow all records to the Next Step.

PPP676 (new)
PPP676 will be a new program which will create EDB update transaction based on the roster changes coded for a Merit Cycle ID read in via a Run Specification Record (see new form UPAY912). It will also create costing transactions and merit match file input.

FILE Definitions
CARDFIL
The DD name CARDFIL will be defined for the Input Specification Record File.

WIA-RUN-SPEC-RECORD.
WIA-PROG-ID X(11).
WIA-CYCLE-ID X(18).
WIA-MERIT-DATA-FILE-FLAG X(1).
FILLER X(50).

WIA-PROG-ID should have a valid value of ‘PPP676-SPEC ’ for the program to proceed further to process the Cycle ID specified in WIA-CYCLE-ID.
The total record Length is 80.

EDBTRANS
The DD name EDBTRANS will be defined for the output EDB Update Transactions File.

WOA-EDB-TRANS-RECORD.
WOA-TRAN-CD X(2).
FILLER X.
WOA-TRAN-EMP-ID X(9).
WOA-TRAN-EFF-DATE X(6).
WOA-TRAN-DATA X(84).

The total record Length is 102 Characters.

A1, A2, X1, Appt and Distribution transaction record layout.

01 TRANS-AREA.
  05 NO-TRANS PIC 9(7) COMP-3.
  05 TRANSACTIONS OCCURS 99 TIMES.
  07 TRANSACTION.
   09 TRAN-CD.
      11 TRAN-CD-HI PIC X.
      11 TRAN-CD-LO PIC X.
   09 FILLER PIC X.
   09 TRAN-EMP-ID PIC X(9).
   09 TRAN-EFF-DATE PIC X(6).
   09 TRAN-DATA PIC X(84).
   09 TRAN-A1-DATA REDEFINES TRAN-DATA.
      11 TRAN-GEN-NO PIC X(4).
      11 TRAN-ACTION-CD PIC XX.
      11 FILLER PIC X(78).
   09 TRAN-A2-DATA REDEFINES TRAN-DATA.
      11 FILLER PIC X.
      11 TRAN-MERIT-PCTX REDEFINES TRAN-MERIT-PCT PIC X(4).
      11 FILLER PIC X.
      11 TRAN-REVIEW-DATE.
         13 TRAN-REVIEW-MO PIC XX.
         13 TRAN-REVIEW-YR PIC XX.
      11 TRAN-REVIEW-TYPE PIC X.
      11 FILLER PIC X(30).
      11 TRAN-ACT-STAT PIC XX.
11 FILLER PIC X(41).
09 TRAN-10-DATA REDEFINES TRAN-DATA.
11 FILLER PIC XX.
11 TRAN-APPT-ADC-CD PIC X.
11 TRAN-PERSNL-PROGRM PIC X.
11 TRAN-TYPE PIC X.
11 FILLER PIC X(3).
11 TRAN-BASIS PIC 9(2).
11 TRAN-PD-OVER PIC 9(2).
11 FILLER PIC X(4).
11 TRAN-BEGN-DATE.
  13 TRAN-BEGN-MO PIC 9(2).
  13 TRAN-BEGN-DA PIC 9(2).
  13 TRAN-BEGN-YR PIC 9(2).
11 TRAN-END-DATE.
  13 TRAN-END-MO PIC 9(2).
  13 TRAN-END-DA PIC 9(2).
  13 TRAN-END-YR PIC 9(2).
11 TRAN-DUR-EMPLMT PIC X.
11 TRAN-TITLE-CODE PIC X(5).
11 TRAN-GRAGE PIC X(2).
11 FILLER PIC X(3).
11 TRAN-PCT-FULL-TIME PIC 9V99.
11 TRAN-FXD-VAR PIC X.
11 TRAN-ANNL-HR-RATEX PIC X(8).
11 TRAN-ANNL-HR-RATE9 REDEFINES TRAN-ANNL-HR-RATEX PIC 9(6)V99.
11 TRAN-RATE-CODE PIC X.
11 TRAN-PAY-SCHED PIC X(2).
11 TRAN-TIME-CODE PIC X.
11 TRAN-LEAVE PIC X.
  11 FILLER PIC X(6).
11 TRAN-FLSA-IND PIC X.
11 TRAN-PEV-CODE PIC X.
11 TRAN-PEV-DATE.
  13 TRAN-PEV-MO PIC 9(2).
  13 TRAN-PEV-YR PIC 9(2).
11 FILLER PIC X(16).
09 TRAN-11-DATA REDEFINES TRAN-DATA.
11 FILLER PIC XX.
11 TRAN-DIST-ADC-CD PIC X.
11 TRAN-DIST-ACTION PIC X(02).
  11 FILLER PIC X(04).
11 TRAN-PERQ PIC X(03).
11 TRAN-DUC PIC X(01).
11 TRAN-WSP PIC X(01).
11 FILLER PIC X(05).
11 TRAN-STEP PIC X(3).
11 TRAN-OFF-ABOVE-IND PIC X.
11 TRAN-FTE PIC 9V99.
11 TRAN-DIST-PCT PIC 9V9999.
11 TRAN-PAY-BEGIN-DATE PIC X(6).
11 TRAN-PAY-END-DATE.
  13 TRAN-PAY-END-MO PIC XX.
  13 TRAN-PAY-END-DA PIC XX.
13 TRAN-PAY-END-YR PIC XX.
11 TRAN-PAY-RATE PIC X(7).
11 TRAN-DOS PIC XXX.
11 FILLER PIC X(01).
11 TRAN-FAU PIC X(30).
01 APPT-CHG-AREA.
  05 APPT-CHG-DETAILS OCCURS 9 TIMES.
   10 APPT-CHG-SW PIC X(01).
   10 APPT-CHG-PEV-CD PIC X(01).
   10 APPT-CHG-PEV-DT PIC X(06).

COSTING
The DD name COSTING will be defined for the output Costing Input File.

WOB–COSTING–RECORD.
COPY CPWSXCSF.

The total record Length is 160 Characters.

PRNTFIL
The DD name PRNTFIL will be defined for the output audit Report File.

PRINT–REC.
   CARRIAGE–CTL–CHAR X
   PRINT–LN X(132)

The total record Length is 133.

MERITFL
The DD name MERITFL will be defined for the Merit Data File.

WOD–MERIT–DATA–RECORD.
COPY CPWSXMDF.

The total record Length is 70 Characters.

INCLUDES
The following Includes will be used.

<table>
<thead>
<tr>
<th>Table</th>
<th>Include</th>
</tr>
</thead>
<tbody>
<tr>
<td>Merit Logical Appt Table</td>
<td>PPPVZMLA</td>
</tr>
<tr>
<td>Merit Appointment Table</td>
<td>PPPVZMAP</td>
</tr>
<tr>
<td>Merit Distribution Table</td>
<td>PPPVZMED</td>
</tr>
<tr>
<td>Merit Comments Table</td>
<td>PPPVZMMS</td>
</tr>
<tr>
<td>Merit Cycle Control Table</td>
<td>PPPVZMCC</td>
</tr>
<tr>
<td>EDB Appointment Table</td>
<td>PPPVZAPP</td>
</tr>
<tr>
<td>EDB Employee Table</td>
<td>PPPVZPER</td>
</tr>
<tr>
<td>EDB Distribution Table</td>
<td>PPPVZDIS</td>
</tr>
<tr>
<td>System Parameter Record Table</td>
<td>PPPVZPRM</td>
</tr>
</tbody>
</table>
Cursor Declarations

A Cursor MLA_ROW will be defined to select PPPMLA rows which have been modified and have the requested Cycle ID. A record will be identified as modified when either MLA_DIST_PAYRATE is not equal to MLA_NEW_PAYRATE or MLA_DIST_STEP is not equal to MLA_NEW_STEP.

```
(SELECT
MLA_CYCLE_ID, MLA_EMPLOYEE_ID, MLA_DEPT,
MLA_TITLE_CODE, MLA_APPT_SUB_LOCN, MLA_APPT_REP_CODE,
MLA_APPT_TYPE, MLA_GRADE, MLA_RATE_CODE,
MLA_PAY_SCHEDULE, MLA_DIST_PAYRATE, MLA_DIST_STEP,
MLA_DIST_UNIT_CODE, MLA_DIST_PERCENT, MLA_ANNUAL_SALARY,
MLA_NEW_PP, MLA_NEW_PP_DATE, MLA_NEW_PAYRATE,
MLA_NEW_STEP, MLA_PCT_INCREASE, MLA_EVAL_DATE,
MLA_PERFORM_RATING, MLA_EMP_NAME, MLA_DEPT_NAME,
MLA_APPT_TITLE_NM, MLA_CHANGED_BY, MLA_CHANGED_AT
FROM PPPVZMLA_MLA
WHERE MLA_CYCLE_ID = :MCC-CYCLE-ID
AND (MLA_NEW_PAYRATE >= MLA_DIST_PAYRATE
AND MLA_COMMENTS_FLAG <> 'D')
ORDER BY MLA_EMPLOYEE_ID ASC)
```

Declare a Cursor MEDH_ROW for selecting all the Distributions linked with the current Logical Appointment from the PPPMED table for an Employee, Cycle ID, **Home Department**, Appt Title, Appt Sub Location, Appt Rep Code, Appt Type, Salary Grade, Appt Rate Code, Pay Schedule, Dist Rate and Dist Step. This Cursor will be used if the Collapsing Option on the PPPPMCC table for Cycle ID is ‘H’ (Home Dept).

```
(SELECT MED_DIST_NUM
FROM PPPVZMED_MED
WHERE MED_CYCLE_ID = :MLA-CYCLE-ID
AND MED_EMPLOYEE_ID = :MLA-EMPLOYEE-ID
AND MED_HOME_DEPT = :MLA_DEPT
AND MED_APPT_TITLE = :MLA_TITLE_CODE
AND MED_APPT_SUB_LOCN = :MLA_APPT_SUB_LOCN
AND MED_APPT_REP_CODE = :MLA_APPT_REP_CODE
AND MED_APPT_TYPE = :MLA_APPT_TYPE
AND MED_SALARYGRADE = :MLA_GRADE
AND MED_APPT_RATE_CODE = :MLA_RATE_CODE
AND MED_PAY_SCHEDULE = :MLA_PAY_SCHEDULE
AND MED_DIST_RATE = :MLA_DIST_PAYRATE
AND MED_DIST_STEP = :MLA_DIST_STEP
ORDER BY MED_DIST_NUM ASC)
```

Declare a Cursor MEDA_ROW for selecting all the Distributions linked with the current Logical Appointment from the PPPMED table for an Employee, Cycle ID, **Appointment Department**, Appt Title, Appt Sub Location, Appt Rep Code, Appt Type, Salary Grade, Appt Rate Code, Pay Schedule, Dist Rate and Dist Step. This Cursor will be used if the Collapsing Option on the PPPPMCC table for Cycle ID is ‘A’ (Appointment Department).

```
(SELECT MED_DIST_NUM
FROM PPPVZMED_MED
WHERE MED_CYCLE_ID = :MLA-CYCLE-ID
AND MED_EMPLOYEE_ID = :MLA-EMPLOYEE-ID
AND MED_APPT_DEPT = :MLA_DEPT
```
AND MED_APPT_TITLE = :MLA_TITLE_CODE
AND MED_APPT_SUB_LOCN = :MLA-APPT-SUB-LOCN
AND MED_APPT_REP_CODE = :MLA_APPT_REP_CODE
AND MED_APPT_TYPE = :MLA_APPT_TYPE
AND MED_SALARY_GRADE = :MLA_GRADE
AND MED_APPT_RATE_CODE = :MLA_RATE_CODE
AND MED_PAY_SCHEDULE = :MLA_PAY_SCHEDULE
AND MED_DIST_RATE = :MLA_DIST_PAYRATE
AND MED_DIST_STEP = :MLA_DIST_STEP
ORDER BY MED_DIST_NUM ASC)

Processing Logic

The Run Specification File will be opened for input: the EDB Transactions File, Costing File and Report File will be opened for output.

The EDB transaction Batch Header Record will be created and written to the EDB Transactions File.

PPP676 will be able to process multiple Run Specification Records. All output will be written to the same files. Repeat Step 1 until End-Of-Spec-File or Program-Id-Invalid

Step 1:

1. Read the Run Spec File and Check the Program Id. Exit Step 1 if the program id is not PPP676.
2. Get the Cycle-ID from the Spec Card. Select PPPMCC to get the details about this Cycle-ID.
   If Rec-not-found, write a warning and Exit Step 1.
   To initiate the processing of the Merit Cycle’s EDB Update, the Value of Merit Cycle Status Code Should be ‘R’(Ready for EDB Update). After successful processing of the Merit Cycle’s EDB Update, the Merit Cycle Status has to be changed to ‘C’(Cycle Closed).
   If MCC_STATUS_CD <> ‘R’
      Exit Step 1
   Else
      Update MCC_STATUS_CD = ‘C’ in PPPMCC
      End-if
3. Open the Cursor MLA_ROW.
4. Fetch the Cursor MLA_ROW -- Fetch the 1st MLA_ROW
5. Repeat Until End-of-Cursor-MLA-ROW
   Move MLA-EMPLOYEE-ID to WA-EMPLOYEE-ID
   If No Row for this Employee in PPPPCM OR PPPPER.EMP-STATUS EQUAL 'S' OR PPPPER. LAST-DAY-ON-PAY < MCC-CYCLE-DATE
      Write the error that Employee is separated or Quit
      Fetch the Cursor MLA_ROW until End-of-Cursor-MLA Rowe OR MLA_EMPLOYEE_ID > WA-EMPLOYEE-ID --- Skip the Employee
   Else
      Perform Step 2
      Fetch the Cursor MLA_ROW
      End-if.
6. Close the Cursor MLA_ROW.

Step 2:

1. If MCC-COLLAPSE = ‘H’
   Open the Cursor MEDH_ROW
Fetch the MEDH_ROW Cursor -- Fetch the First Record
Else
  Open the Cursor MEDA_ROW
  Fetch the MEDA_ROW Cursor -- Fetch the First Record
End-if

2. If SQL-RETURN-CODE = 0, Perform Step 3 --Dist Found for Logical Appt
3. Repeat Until End-of-Cursor of MEDH-ROW or MEDA_ROW (using MCC-COLLAPSE).
   Perform Step 5
   If MCC-COLLAPSE = ‘H’
     Fetch the MEDH_ROW Cursor
   Else
     Fetch the MEDA_ROW Cursor
   End-if.
4. If NO-TRANS > 2, Perform Step 4 --Logical Appt has transactions
5. If MCC-COLLAPSE = ‘H’
   Close the Cursor MEDH_ROW
Else
   Close the Cursor MEDA_ROW
End-if

Step 3:

Move Zeroes to WS-NEW-APPT-NUM & WS-NEW-DIST-NUM.
Move ‘N’ to WS-APPT-CREATE-SW
Initialize TRANS-AREA.

Populate the A1 Record Details for the current Employee as below:
a. Move ‘A1’ to TRAN-CD
b. Move MLA-EMPLOYEE-ID to TRAN-EMP-ID
c. Move MCC-**-EFF-DATE’s MMDDYY to TRAN-EFF-DATE
d. Move PAF-GEN-NUM of PPPPER to TRAN-GEN-NO
Note that this has to be the 1st Transaction Record for an Employee.

Populate the A2 Record Details for the Current Employee as below:
a. Move MCC-NEW-SALREV-DT’s MMYY to TRAN-REVIEW-DATE
b. Move ‘A2’ to TRAN-CD
c. Move MLA-EMPLOYEE-ID to TRAN-EMP-ID
d. Move MCC-**-EFF-DATE’s MMDDYY to TRAN-EFF-DATE
Note that TRAN-ACT-STAT is not used and so we do not need to populate this.
Note that TRAN-REVIEW-TYPE is not changed and so we do not populate this.
Note that this has to be the 2nd Transaction Record for an Employee.

Move 2 to NO-TRANS.

Step 4:

If WS-APPT-CREATE-SW = ‘Y’ -- New Appointment has to be created.
   Perform Step 6 to Create an Add Transaction for WS-NEW-APPT-NUM.

Perform Writing Transaction of the Current Employee to DBM-TRANSACTIONS File Until all Transactions (NO-TRANS) are written.
Also make sure while writing that the saved Performance Data for an Appointment has to be moved back when the transaction is an Appt.
**Step 5:**

1. From the MED-DIST-NUM and MLA-EMPLOYEE-ID, get all the details about the EDB Distribution from PPPDIS, EDB Appointment from PPPAPP and Employee Details from PPPPER and PPPPCM.

2. Populate the Costing Input File as follows:
   a. Move MLA-EMPLOYEE-ID to XCSF-EMPLOYEE-ID.
   b. Move EMP_NAME to XCSF-EMPLOYEE-NAME.
   c. Move NAMESUFFIX to XCSF-EMPLOYEE-SUFFIX.
   d. Move '27' to XCSF-ACTION-CODE.
   e. If MCC-**-EFF-DATE is less than PAY-BEGIN-DATE
      Move PAY-BEGIN-DATE(YYMMDD) to XCSF-EFFECTIVE-DATE
      Else
      Move MCC-**-EFF-DATE(YYMM'01') to XCSF-EFFECTIVE-DATE
      End-if.
   f. Move TITLE-CODE to XCSF-TITLE-CODE.
   g. Move FULL-ACCT-UNIT to XCSF-FAU.
   h. Move RATE-CODE to XCSF-RATE-CODE.
   i. Move DIST-PERCENT to XCSF-PCT-R.
   j. Move DIST-DOS to XCSF-DOS.
   k. Move DIST-PAYRATE(use RATE-CODE) to XCSF-OLD-RATEX.
   l. Move MLA-NEW-PAYRATE(like above) to XCSF-NEW-RATEX.
   m. Move DIST-FTE to XCSF-FTE.
   n. Move EMP-CBUC to XCSF-0161-CBUC.
   o. Move EMP-REL-UNIT to XCSF-0255-EMP-UNIT-CODE.
   p. Move EMP-SPEC-HAND to XCSF-0256-EMP-SPCL-HNDLG.
   q. Move EMP-DIST-UNIT-CODE to XCSF-0257-EMP-DIST-UNIT.
   r. Move TITLE-UNIT-CODE to XCSF-0295-TUC.
   s. Move APPT-REP-CODE to XCSF-0295-EMP-COV.
   t. Move APPT-DURATION to XCSF-DUR-EMPLMT.
   u. Move PAY-END-DATE(YYMMDD) to XCSF-PAY-END-DATE.
   v. Move ZERO to XCSF-ERROR-CODE.
   w. If MERIT-DATA-FILE is requested then populate Merit Data File as follows.
      I. Move XCSF-EMPLOYEE-ID to XMDF-EMPLOYEE-ID
      II. Move XCSF-EMPLOYEE-NAME to XMDF-EMPLOYEE-NAME
      III. Move XCSF-TITLE-CODE to XMDF-TITLE-CODE
      IV. Move APPT-SUB-LOCATION to XMDF-SUB-LOCATION
      V. Move XCSF-RATE-CODE to XMDF-RATE-CODE
      VI. Move XCSF-OLD-RATEX to XMDF-OLD-RATEX
      VII. Move XCSF-NEW-RATEX to XMDF-NEW-RATEX
      VIII. Calculate XMDF-OLD-HOURLY-RATE and XMDF-NEW-HOURLY-RATE
            Using MLA-RATE-CODE and MLA-PAY-SCHEDULE.

3. Write the Costing Input File from XCSF-RECORD Populated above if XMDF-NEW-RATEX does not match XMDF-OLD-RATEX.
   If MERIT-DATA-FILE is requested then write the Merit Data File as well.

4. If future Distribution, it is only necessary to change the Pay Rate and the Step. A new Distribution is not needed. If the MCC-BW-EFF-DATE or MCC-MO-EFF-DATE (as appropriate) is less than PAY-BEGIN-DATE of the distribution then do 5, else go to 6.
5. Now populate the Change Transaction for the Old Distribution as described below:

Add +1 to NO-TRANS.

a. Move ‘C’ to TRAN-DIST-ADC-CD (NO-TRANS)
b. Move ‘27’ to TRAN-DIST-ACTION (NO-TRANS).
c. Move MLA-NEW-STEP to TRAN-STEP (NO-TRANS).
d. Move MLA-NEW-PAYRATE to TRAN-PAY-RATE (NO-TRANS).
e. Move MED-DIST-NUM to TRAN-CD (NO-TRANS).
f. Move MLA-EMPLOYEE-ID to TRAN-EMP-ID (NO-TRANS).
g. Move MCC-**-EFF-DATE’s MMDDYY to TRAN-EFF-DATE (NO-TRANS).

Perform Step 7 to Create change transaction for the Existing Appointment.

Now Exit Step 5.

6. Now populate the Change Transaction for the Old Distribution as described below:

Add +1 to NO-TRANS.

a. Move MCC-CYCLE-DATE - 1 (MMDDYY) to TRAN-PAY-END-DATE (NO-TRANS)
b. Move ‘C’ to TRAN-DIST-ADC-CD (NO-TRANS)
c. Move ‘27’ to TRAN-DIST-ACTION (NO-TRANS).
d. Move MED-DIST-NUM to TRAN-CD (NO-TRANS).
e. Move MLA-EMPLOYEE-ID to TRAN-EMP-ID (NO-TRANS).
f. Move MCC-**-EFF-DATE’s MMDDYY to TRAN-EFF-DATE (NO-TRANS).

7. Now look for an empty slot for the new distribution.

If WS-APPT-CREATE-SW = “N”

Go thru PPPDIS to get 1st available DIST-NUM > WS-NEW-DIST-NUM

If DIST-NUM Found -- Not all the Distributions are used

Move DIST-NUM to WS-NEW-DIST-NUM

Perform Step 7 to Create change transaction for the Existing Appointment

Else -- All the Distributions are used

Select Count(*) from PPPAPP for MLA-EMPLOYEE-ID

If Count(*) < 9 -- Not all the Appointments are used

Go thru PPPAPP to get the 1st available APPT-NUM

Move ‘Y’ to WS-APPT-CREATE-SW

Move APPT-NUM to WS-NEW-APPT-NUM

Move (WS-NEW-APPT-NUM +1) to WS-NEW-DIST-NUM

Else -- All the Appointments are used

Write Error and Exit Step 5

End-if

End-if

Else

If WS-NEW-DIST-NUM = (WS-NEW-APPT-NUM +8) -- All Distributions are Used

Perform Step 6 to Create an Add Transaction for WS-NEW-APPT-NUM

Go thru PPPAPP to get 1st available APPT-NUM which is > WS-NEW-APPT-NUM

If APPT-NUM Found -- Not all the Appointments are used

Move APPT-NUM to WS-NEW-APPT-NUM

Move (WS-NEW-APPT-NUM +1) to WS-NEW-DIST-NUM

Else -- All the Appointments are used

Write Error and Exit Step 5

End-if

Else

Add +1 to WS-NEW-DIST-NUM

End-if

End-if

Else

8. Now populate the Add Transaction for the New Distribution as described below:
Add +1 to NO-TRANS.

a. Move PAY-SEND-DATE Old Dist format. If PAY-SEND-DATE is Indefinite(12/31/9999) then Move 999999.
b. Move MCC-**.EFF-DATE(MMDDYY) to TRAN-PAY-BEGIN-DATE (NO-TRANS)
c. Move '27' to TRAN-DIST-ACTION (NO-TRANS).
d. Move DIST-PERQ to TRAN-PERQ (NO-TRANS)
e. TRAN-DUC is not populated(??)
f. Move WORK-STUDY-PGM to TRAN-WSP (NO-TRANS)
g. Move MLA-NEW-STEP to TRAN-STE (NO-TRANS)
h. Move DIST-OFF-ABOVE to TRAN-OFF-ABOVE-IND (NO-TRANS)
i. Move DIST-FTE to TRAN-FTE (NO-TRANS)
j. Move DIST-PERCENT to TRAN-DIST-PCT (NO-TRANS)
k. Move MLA-NEW-PAYRATE to TRAN-PAY-RATE (NO-TRANS)
l. Move DIST-DOS to TRAN-DOS (NO-TRANS)
m. Move FULL-ACCT-UNIT to TRAN-FAU (NO-TRANS)

Step 6:

1. Now populate the Add Transaction for the New Appointment as described below.

   WS-NEW-APPT-NUM has the new Appointment Number that has to be created.
   a. Add +1 to NO-TRANS.
   b. Move 'A' to TRAN-APPT-ADC-CD (NO-TRANS)
   c. Move PERSONNEL-PGM to TRAN-PERSNL-PROGRM (NO-TRANS)
   d. Move APPT-TYPE to TRAN-TYPE (NO-TRANS)
   e. Move ACADEMIC-BASIS to TRAN-BASIS (NO-TRANS)
   f. Move APPT-PAY-ORDER to TRAN-PD-OVER (NO-TRANS)
   g. Move APPT-BEGIN-DATE(MMDDYY) to TRAN-BEGN-DATE (NO-TRANS)
   h. Move APPT-END-DATE(MMDDYY) to TRAN-END-DATE (NO-TRANS)
   i. Move APPT-DURATION to TRAN-DUR-EMPLMT (NO-TRANS)
   j. Move TITLE-CODE to TRAN-TITLE-CODE (NO-TRANS)
   k. Move GRADE to TRAN-GRADE (NO-TRANS)
   l. Move PERCENT-FULLTIME to TRAN-PCT-FULL-TIME (NO-TRANS)
   m. Move FIXED-VAR-CODE to TRAN-FXD-VAR (NO-TRANS)
   n. Move MLA-NEWANN-SALARY to TRAN-ANNL-HR-RATEX / 9(NO-TRANS)
   o. Move RATE-CODE to TRAN-RATE-CODE (NO-TRANS)
   p. Move PAY-SCHEDULE to TRAN-PAY-SCHED (NO-TRANS)
   q. Move TIME-REPT-CODE to TRAN-TIME-CODE (NO-TRANS)
   r. Move LEAVE-ACRUCODE to TRAN-LEAVE (NO-TRANS)
   s. Move WS-NEW-APPT-NUM to TRAN-CD (NO-TRANS)
   t. Move MLA-EMPLOYEE-ID to TRAN-EMP-ID (NO-TRANS)
   u. Move MCC-**.EFF-DATE’s MMDDYY to TRAN-CD (NO-TRANS)
   v. Move MLA-EMPLOYEE-ID to TRAN-EMP-ID (NO-TRANS)
   w. Move MCC-**.EFF-DATE’s MMDDYY to TRAN-EFF-DATE (NO-TRANS).

2. Now populate the X1 Transaction for the above Appointment to update its Performance Evaluation Code, Performance Evaluation Date and Appointment FLSA Indicator.

   Calculate WS-APPT-OFFSET = 300*(WS-NEW-APPT-NUM – 1)
   Add +1 to NO-TRANS.
   a. Move ‘X1’ to TRAN-CD (NO-TRANS)
   b. Move MLA-EMPLOYEE-ID to TRAN-EMP-ID (NO-TRANS)
   c. Move MCC-**.EFF-DATE’s MMDDYY to TRAN-EFF-DATE (NO-TRANS)
   d. Move MLA-PERFORM-RATING to TRAN-PEV-CD-A (NO-TRANS)
e. Move MLA-EVAL-DATE’s MMDDYY to TRAN-PEV-DT-A (NO-TRANS)
f. Move APPT-FLSA-IND to TRAN-FLSA-A (NO-TRANS)
g. Move 2026+WS-APPT-OFFSET to ELEM-NO. 
   Calculate the Check Digit using code in the standard copybook CPPDXP03 
   Concatenate(ELEM-NO, CALC-CHECK_DIGIT) to TRAN-PEV-CD-ELEM-A (NO-TRANS)
h. Move 2027+WS-APPT-OFFSET to ELEM-NO. 
   Calculate the Check Digit using code in the standard copybook CPPDXP03 
   Concatenate(ELEM-NO, CALC-CHECK_DIGIT) to TRAN-PEV-DT-ELEM-A (NO-TRANS)
i. Move 2005+WS-APPT-OFFSET to ELEM-NO 
   Calculate the Check Digit using code in the standard copybook CPPDXP03 
   Concatenate(ELEM-NO, CALC-CHECK_DIGIT) to TRAN-FLSA-ELEM-A(NO-TRANS)

Step 7:

1. Now populate the Change Transaction for the Existing Appointment to update its Performance Evaluation Code and Performance Evaluation Date as described below.
Add +1 to NO-TRANS.
   a. Move ‘C’ to TRAN-APPT-ADC-CD(NO-TRANS)
   b. Move PPPAPP.APPT-NUM to TRAN-CD (NO-TRANS)
   c. Move MLA-EMPLOYEE-ID to TRAN-EMP-ID (NO-TRANS)
   d. Move MCC-**.EFF-DATE’s MMDDYY to TRAN-EFF-DATE (NO-TRANS).

Since there would be a possibility that more than one Logical Appointment might update the Appointment’s Perf Evaluation Details, we need to choose the best now, save it and move it back before writing the Appointment Change Transaction.

PPP677(new)
PPP677 will be a new program which will read a Run Specification Record (see new form UPAY913) and call modules as requested to create reports and files relating to Web Merit reconciliation. The Merit Cycle ID selected for processing will also be provided on the Run Specification Record. PPP677 will be capable of processing multiple Run Specification Records in a single run.

A standard warnings and messages report will be established as PPP6770. It will be written to by PPP677 and by the called modules as well.

The only program PPP677 will initially be able to call, as requested on the Run Specification Record, is:
• PPMRTRP1

PPMRTRP1(new)
PPHPARP1 will be a new program, called from PPP677, which will read all rows on the PPPMLA table for a selected Cycle ID and write out a file in tab-delimited format.

A main cursor will be defined which will select all PPPMLA rows for a selected Cycle ID, ordered in table key order. The cursor will be opened and the first row fetched. All of the rows will be fetched, and processed along with the PPPMED cursor below. At the end of the table data the cursor will be closed.

A PPPMED cursor will be defined to select all PPPMED rows which share the PPPMLA key. The cursor will be opened and the first row fetched. All of the rows will be fetched, and processed. At the end of the table data the cursor will be closed.

For each PPPMLA/PPPMED combination a tab delimited record will be written. The record content is defined by new copymember CPWSTMRT.
MOVE MLA–CYCLE–ID TO TMRT–CYCLE–ID.
MOVE MLA-EMP-NAME       TO TMRT-EMP-NAME.
MOVE MLA-EMPLOYEE-ID    TO TMRT-EMPLOYEE-ID.
MOVE MLA-TITLE-CODE     TO TMRT-TITLE-CODE.
MOVE MLA-APPT-TITLE-NM  TO TMRT-TITLE-CODE-NAME.
MOVE MLA-GRADE          TO TMRT-GRADE.
MOVE MLA-DIST-STEP      TO TMRT-DIST-STEP.
MOVE MLA-NEW-STEP       TO TMRT-DIST-NEW-STEP.
MOVE MLA-DIST-PAYRATE   TO TMRT-DIST-PAYRATE.
MOVE MLA-NEW-PAYRATE    TO TMRT-DIST-NEW-PAYRATE.
MOVE MLA-PCT-INCREASE   TO TMRT-DIST-PERCENT-INC.
MOVE MLA-PERFORM-RATING TO TMRT-PERF-EVAL-CODE.
MOVE MLA-EVAL-DATE      TO TMRT-PERF-EVAL-DATE.
MOVE MLA-COMMENTS-FLAG  TO TMRT-COMMENTS-FLAG.
MOVE MLA-MAX-RATE-FLAG  TO TMRT-MAX-RATE-FLAG.
MOVE MLA-SELF-UPD-FLAG  TO TMRT-SELF-UPD-FLAG.
MOVE MLA-CHANGED-BY     TO TMRT-CHANGED-BY.
MOVE MED-DIST-NUM       TO TMRT-DIST-NUM.
MOVE MED-HOME-DEPT      TO TMRT-HOME-DEPT.
MOVE MED-APPT-DEPT      TO TMRT-APPT-DEPT.
MOVE MED-DIST-PERCENT   TO TMRT-DIST-PERCENT.
MOVE MED-FULL-ACCT-UNIT TO TMRT-DIST-FAU.

The appointment number will be created from the distribution number.
MOVE MED-DIST-NUM       TO WS-APPT-NUM-99.
MOVE ZERO               TO WS-APPT-NUM-2.
MOVE WS-APPT-NUM-99     TO TMRT-APPT-NUM.

The TUC description will be obtained from the PPPMAP row.
MOVE MAP-TITLE-UNIT-CD   TO TMRT-APPT-TUC.

The annual increase will be calculated and moved to the record.
MOVE WS-ANNUAL-INCREASE TO TMRT-DIST-DOLLAR-INC.

Under-the-line program PPFAU007 will be called to obtain the FAU description.
MOVE F007-TITLE         TO TMRT-DIST-FAU-DESC.

PPMRTRP1 will be callable multiple times, however the data will be written to a single output file.

The method for downloading the tab-delimited file to other platforms will be defined locally.

PPXBUUTL
PPXBUUTL recreates the CPWSXBUT format from the DB2 data. It will be modified to process the new BUT_MERIT_SELECT column.
Stored Procedure Programs

**PPBUTSPR**
A COBOL stored procedure will be written to provide a bargaining unit list for selection by a Web Merit Cycle. The returned list of codes and translation will be displayed on the Merit Administration screen to provide for bargaining unit selection for a Merit Cycle. See stored procedure DDL for in and out parameters.

No parameters will be passed in. PPBUTSPR will select the Bargaining Unit Codes (BUT_BUC) from the PPPBUT table which have a Y in the Merit Selection Flag (BUT_MERIT_SELECT). The Code Translation table will then be accessed for the code translation, if any. “UNKNOWN” will be the default translation. An array will be returned as an output parameter.

```cobol
05 WS-CBUCS.  
   10 FILLER OCCURS 20.  
   15 WS-CBUC PIC X(02).  
   15 WS-CBUC-TRANSLATION PIC X(24).
```

**PPCSTSPR (new)**
A COBOL stored procedure will be written to provide Merit Cycle Costing Summary data for a selected roster. The data will be displayed on a pop-up window triggered from the Roster screen of the Web Merit application. See stored procedure DDL for in and out parameters.

A Cycle ID will be passed in to identify the PPPMLA table rows for possible selection.
A Session ID will be passed in to identify the PPPMDP table for possible departments to be used in a DB2 Join to further define PPPMLA rows for selection. The PPPMDP entries will be controlled via PPMdpSPR.
An MDP Level will be passed in to identify which level of PPPMDP entries should be used.
An Access Intent will be passed in to identify whether MDP entries should only be selected for the passed level, or for all levels equal or greater than that level, i.e. a Control Point roster reflects a Control Point structure.
An ARSM Rule of the user will be passed in to identify which funding aggregation Group Definition on the PPPFND table should be used to build the costing summary.

A cursor will be defined to select PPPMLA rows which have the passed Cycle ID, and whose MLA Department Code (Home Department or Appointment Department, as defined on the PPPMCC table for the cycle) match a list of Department Codes developed from the PPPMDP table.

If the passed Access Intent asks for Get One level, then the Department Codes will be selected from the PPPMPDP rows which match on the passed Session ID and MDP Level. If the passed Access Intent asks for Get All levels, then the Department Codes will be selected from the PPPMDP rows which match on the passed Session ID and MDP Levels equal to or greater than the passed level.

A second cursor will be defined to select PPPMED rows which match the PPPMLA key, and which have a Pay Begin Date prior to or equal to the end of the fiscal year. The fiscal year end will be defined from the PPPMCC Cycle Date. The cursor will be opened, processed and closed for each PPPMLA row selected.

For each PPPMLA/PPPMED row returned, costing data will be calculated and placed in an array. The first level define sub-total groups for the costing fund summary. Sub-script 101 will be used for the Grand Total. The second level will define individual fund ranges within the sub-total group. Sub-script 100 will be used for the sub-total.

```cobol
01 WS-SUB-GROUPS OCCURS 101.  
10 WS-FUND-GROUPS OCCURS 100.  
   15 WS-GROUP-TXT PIC X(30).  
   15 WS-ALLOCATION PIC S9(09)V99.  
```
15 WS-OVER-UNDER-PCT       PIC S9(02)V9999.

Under-the-line program PPFAU018 will be called, passing the PPPMED FAU and the passed ARSM Rule. The ARSM Rule will define which Fund Group Definition to use from the PPPFND table. A Fund Group Code should be returned. The Fund Group Code for Costing Summary processing should consist of a four digit number (see program PPCTFNDE). The first two digits define one sub-script, and the last two digits define a second subscript. These control where the calculated costs are placed in the costing array. A text field for the fund range is also returned by PPFAU018 and will be used as the fund range description in the costing summary. See Attachment F.

The current salary base will be calculated from the PPPMED percentage and PPPMLA current annual salary.

\[
\text{COMPUTE WS-COST ROUNDED} = \text{MED-DIST-PERCENT} \times \text{MLA-ANNUAL-SALARY.}
\]

The new salary base will be calculated from the PPPMED percentage and PPPMLA new annual salary (or current salary if a new salary has not yet been updated).

\[
\begin{align*}
\text{IF MLA-NEWANN-SALARY} & > \text{ZERO} \\
\text{COMPUTE WS-COST ROUNDED} & = \text{MED-DIST-PERCENT} \times \text{MLA-NEWANN-SALARY} \\
\text{ELSE} & \\
\text{COMPUTE WS-COST ROUNDED} & = \text{MED-DIST-PERCENT} \times \text{MLA-ANNUAL-SALARY}
\end{align*}
\]

The fiscal year cost will be calculated from the PPPMED percentage and the increase from the PPPMLA current and new annual salary. It will be set to zero if a new salary has not yet been updated.

\[
\begin{align*}
\text{IF MLA-NEWANN-SALARY} & > \text{ZERO} \\
\text{COMPUTE WS-COST ROUNDED} & = \text{MED-DIST-PERCENT} \times (\text{MLA-NEWANN-SALARY} - \text{MLA-ANNUAL-SALARY}) \\
\text{COMPUTE WS-MONTHLY-COST ROUNDED} & = \text{WS-COST} / 12 \\
\text{COMPUTE WS-COST} & = \text{WS-MONTHLY-COST} \times \text{WS-FISCAL-YEAR-MONTHS} \\
\text{ELSE} & \\
\text{MOVE ZERO TO WS-COST}
\end{align*}
\]

Once all the cursor data has been processed, the costing array will be processed. For any row that has a current base salary greater than zero, the other fields of the row will be calculated: the merit allocation, full year cost, the amount the full year cost exceeds (or not) the allocation and the percentage increase in cost. The control percent will be obtained from the PPPMCC row matching the passed Cycle ID.

\[
\begin{align*}
\text{COMPUTE WS-ALLOCATION} & = (\text{WS-IX1 WS-IX2}) \times (\text{WS-IX1 WS-IX2}) \\
\text{WS-CUR-SAL-BASE} & \times \text{WS-CONTROL-PCT} \\
\text{COMPUTE WS-ANNUAL-COST} & = (\text{WS-IX1 WS-IX2}) \times (\text{WS-IX1 WS-IX2}) \\
\text{WS-NEW-SAL-BASE} & - \text{WS-CUR-SAL-BASE} \\
\text{COMPUTE WS-OVER-UNDER-DOL} & = \text{WS-ALLOCATION} - \text{WS-ANNUAL-COST}
\end{align*}
\]
Compute $WS-OVER-UNDER-PCT = WS-ANNUAL-COST / WS-CUR-SAL-BASE$

The final array will be moved to a display defined array and returned as a parameter.

10 WS-MRT-RETURN-COSTS OCCURS 40.
15 WS-MRT-GROUP-TXT PIC X(30).

**PPCTTSP2 (new)**

A COBOL stored procedure will be written to provide Merit Cycle Status Code translations in a predefined order. The order will reflect the normal progression of status codes from cycle setup through completion of the EDB update. See stored procedure DDL for in and out parameters.

No parameters will be passed in. PPCTTSPR will contain a hard coded array of Merit Cycle Status Codes in the predefined order: U, P, H, D, F, O, N, A, R, C, X. See the PPCTT CYSTAT transactions for the translations. “UNKNOWN” will be the default translation. An array will be returned as an output parameter.

05 WS-STATUS-ARRAY REDEFINES WS-STATUS OCCURS 12.
  10 WS-STATUS-CODE PIC X(01).
  10 WS-STATUS-TRANSLATION PIC X(50).

**PPMDPSPR (new)**

A COBOL stored procedure will be written to provide department lists for roster selection based on user ARSM rules initially established by UCPPPMMRT. See stored procedure DDL for in and out parameters.

The initial entries on the PPPMDP table for a user will be established by UCPPPMMRT. Rows will be inserted at the zero level identifying all the UC0ASC rules and their related Department Codes that the user has access to. PPMDPSPR will perform four access functions based on those initial entries, and subsequent PPPMDP entries developed by this program.

A Session ID, ARSM Rule, Department Code and Access Intent will be passed in. The following Access Intents will be processed.

**Get One**
The passed Department Code will be inserted in the PPPMDP table at the next highest MDP Level. The new MDP Level will be returned as a parameter. The Web Merit roster selection will then use that highest level value, the Session ID and Cycle ID to select a roster from the PPPMLA table.

**Get All**
The passed Department Code will be inserted in the PPPMDP table at the next highest MDP Level. This new MDP Level will be returned as a parameter. PPMDPSPR will then build more entries based on the structure in the PPPMCP table. All the Department Codes on the PPMCP table that have the passed Department Code as their control Department will be inserted into the PPPMDP table at the next highest level. Then the new entries will be processed to find their controlled departments. And so on until no more controlled departments are found, or the expected control structure level is exceeded. The Web Merit roster selection will then use that returned MDP Level
(from the initial insert) value, the Session ID and Cycle ID to select a roster from the PPPMLA table. In this case it will select MPD entries equal to or greater than the returned MDP Level.

Down
The Down access will not develop a roster. It will develop a new list of departments for display for further actions. The passed Department Code will be inserted in the PPPMDP table at the next highest MDP Level. The new MDP Level will be returned as a parameter. PPMMDPSPR will then build more entries based on the structure in the PPPMCP table. All the Department Codes on the PPMCP table that have the passed Department Code as their Control Department will be inserted into the PPPMDP table at the same level. Unlike Get All, just one level will be processed. The Web Merit process will then use the returned MDP Level and Session ID to display the new level of departments for further action.

Up
The Up access will not develop a roster. It will develop a new list of departments for display for further actions. The highest current level of the PPPMDP table, if greater than zero, will be deleted. The new highest MDP Level will be returned as a parameter. The Web Merit process will then use the returned MDP Level and Session ID to display the new level of departments for further action.

PPTCTSP2 (new)
A COBOL stored procedure will be written to provide step based rates from the Title Code Table. See stored procedure DDL for in and out parameters.
A Title Code, Sub-Location, Coverage Code and the Cycle Date will passed as input parameters. PPTCTSP2 will call PPTCTUTL to obtain a full rate set. The data will be expected in XTCL-TRR-RATES. An array of rates particular to the input parameters will be selected out of the returned data in CPLNKTCL. A rate array will be returned as an output parameter.

05 WS-TSB-PAY-REP-INFO.
   10 WS-TRR-RATES OCCURS 40.
      15 WS-TRR-AN-RATE PIC 99999.
      15 WS-TRR-HR-RATE PIC 999.9999.
      15 WS-TRR-ALT-RATE PIC 99999.99.

PPTCTSP5 (new)
A COBOL stored procedure will be written to provide range based rates from the Title Code Table. See stored procedure DDL for in and out parameters.
A Title Code, Sub-Location, Coverage Code and the Cycle Date will passed as input parameters. PPTCTSP5 will call PPTCTUTL to obtain a full rate set. The data will be expected in either XTCL-TRR-RATES or XTCL-TGB-DETAIL-DATA. An array of rates particular to the input parameters will be selected out of the returned data in CPLNKTCL. A rate array will be returned as an output parameter.

05 WS-TSB-PAY-REP-INFO.
   10 WS-TRR-RATES OCCURS 3.
      15 WS-TRR-AN-RATE PIC 9(05).
      15 WS-TRR-HR-RATE PIC 9(03).9999.
05 WS-TGB-PAY-REP-INFO.
   10 WS-TGB-GRADE-TYPE PIC X(02).
   10 WS-TGB-GRADE PIC X(02).
   10 WS-TGB-EFFECTIVE-DATE PIC X(10).
The WS-TRR_RATES will contain rates for Rate Lookup Code M, and will come from the PPPTRR table. The WS-SGT-MIN, MID and MAX rates will contain rates for Rate Lookup Code G, and will come from the PPPSGT table.

**UCPPPMRT (new)**

A COBOL stored procedure will be written to provide ARSM processing for Web Merit. It will be called to confirm that a user ID has valid access to various screens during various Merit Cycle statuses. It will develop an initial list of user roles available for a user ID, and create an initial department access list for those roles on the PPPMDP table. See stored procedure DDL for in and out parameters.

UCPPPMRT will process four access types.

**Merit Administration Screen Access**

The user ID and Access Intent, i.e. to access Administration Screens, will be passed in. UCPPPMRT will access the ARSM UC0ASC table to see if the user ID or a Group ID associated with the user ID has an entry for a MERITADM Rule. If so, ARSM Permission will be set Y. If no such Rule exists, then ARSM Permission will be set to N. The ARSM Permission will be returned as a parameter. The Web Merit application will allow or block access to the Merit Administration screens as appropriate.

**Merit Roster Screen Access**

The user ID and Access Intent, i.e. to access Roster Screens, will be passed in. UCPPPMRT will access the ARSM UC0ASC table to see if the user ID or a Group ID associated with the user ID has an entry for a MERITADM, MERITCTL and/or MERITDPT Rule, and a PERSONAL Rule. If so, ARSM Permission will be set Y. If no such Rule exists, then ARSM Permission will be set to N. The ARSM Permission will be returned as a parameter. The Web Merit application will allow or block access to the Merit roster screens, as appropriate.

**Merit Cycle Roster Access**

The user ID, Access Intent, i.e. to access Roster Screens for a specific Cycle ID, and the Cycle Status will be passed in. UCPPPMRT will determine from the Cycle Status what user types are allowed to a Cycle with the current status. See Attachment E. UCPPPMRT will access the ARSM UC0ASC table to see if the user ID or a Group ID associated with the user ID have an entry for an appropriate Rule. If so, ARSM Permission will be set Y. If no such Rule exists, then ARSM Permission will be set to N. The ARSM Permission will be returned as a parameter. The Web Merit application will allow or block access to the selected Cycle ID’s roster screens, as appropriate.

If access is allowed, UCPPPMRT will insert into the PPPMDP table at the zero level an entry for each appropriate found Rule. The PPPMDP entry will contain the Rule and Department Code from the UC0ASC table. The Web Merit application will then display the PPPMDP entry or entries for user selection. The Rule and Department Code selected by the user, and the intended access, will then be passed to PPMDSPR for appropriate action.

UCPPPMRT will access the ARSM UC0ASC table to obtain the associated data for the PERSONAL Rule for the passed user ID. The Employee ID will be returned for later use during roster update.
Copymembers

CPLNKMRT(new):
A copymember will be created to serve as linkage between new program PPP677 and called new module PPMRTRP1.

*01 CPLNKMRT-INTERFACE.
  03 KMRT-MERIT-CYCLE-ID PIC X(18).
  03 KMRT-PROGRAM-STATUS-FLAG PIC X(02).
     88 KMRT-PROGRAM-FATAL-ERROR VALUE '??'.
  03 KMRT-HIGHEST-PGM-SEVERITY PIC 9(01).

CPWSTMRT(new):
A copymember will be created which will define the tab-delimited file written by new program PPMRTRP1, where VALUE X'05' defines a tab.

*01 TMRT-TAB-RECORD.
  05 TMRT-CYCLE-ID PIC X(18).
  05 FILLER PIC X(01) VALUE X'05'.
  05 TMRT-APPT-TUC PIC X(02).
  05 FILLER PIC X(01) VALUE X'05'.
  05 TMRT-EMP-NAME PIC X(26).
  05 FILLER PIC X(01) VALUE X'05'.
  05 TMRT-EMPLOYEE-ID PIC X(09).
  05 FILLER PIC X(01) VALUE X'05'.
  05 TMRT-HOME-DEPT PIC X(06).
  05 FILLER PIC X(01) VALUE X'05'.
  05 TMRT-APPT-DEPT PIC X(06).
  05 FILLER PIC X(01) VALUE X'05'.
  05 TMRT-APPT-NUM PIC 9(02).
  05 FILLER PIC X(01) VALUE X'05'.
  05 TMRT-TITLE-CODE PIC X(04).
  05 FILLER PIC X(01) VALUE X'05'.
  05 TMRT-TITLE-CODE-NAME PIC X(30).
  05 FILLER PIC X(01) VALUE X'05'.
  05 TMRT-GRADE PIC X(02).
  05 FILLER PIC X(01) VALUE X'05'.
  05 TMRT-DIST-NUM PIC 9(02).
  05 FILLER PIC X(01) VALUE X'05'.
  05 TMRT-DIST-FAU PIC X(30).
  05 FILLER PIC X(01) VALUE X'05'.
  05 TMRT-DIST-FAU-DESC PIC X(30).
  05 FILLER PIC X(01) VALUE X'05'.
  05 TMRT-DIST-PERCENT PIC 9.9999.
  05 FILLER PIC X(01) VALUE X'05'.
  05 TMRT-DIST-STEP PIC X(03).
  05 FILLER PIC X(01) VALUE X'05'.
  05 TMRT-DIST-NEW-STEP PIC X(03).
  05 FILLER PIC X(01) VALUE X'05'.
  05 TMRT-DIST-PAYRATE PIC 99999.9999.
  05 FILLER PIC X(01) VALUE X'05'.
CPWSX BUT:
The copymember defining a VSAM BUT record will be modified to add the Merit Selection flag. The field will not be included in the VSAM update process, but is needed for other uses of CPWSX BUT.

10 X BUT-MERIT-SELECT PICTURE X(01).

CPWSBUTH:
The copymember defining a PPPBUTH record will be modified to add the Merit Selection flag.

10 BUT-MERIT-SELECT PIC X(1).
Bind Members

**PPBUTSPR (new)**
A new package bind member will be created for stored procedure PPBUTSPR.

**PPCSTSPR (new)**
A new package bind member will be created for stored procedure PPCSTSPR.

**PPCTTSP2 (new)**
A new package bind member will be created for stored procedure PPCTTSP2.

**PPMDPSPR (new)**
A new package bind member will be created for stored procedure PPMDPSPR.

**PPP675 (new)**
A new plan bind member will be created for PPP675. It will include members PPP675, PPTCTUTL, PPCTTUTL, PPRMUT2 and PPMSSG2.

**PPP676 (new)**
A new plan bind member will be created for PPP676. It will include members PPP676, PPCTTUTL and PPMSSG2.

**PPP677 (new)**
A new plan bind member will be created for PPP677. It will include members PPMRTRP1 PPFAU007, PPTCTUTL, PPRMUT2, PPCTTUTL and PPMSSG2. PPP677 itself will not contain active SQL code.

**UCPPPMRT (new)**
A new package bind member will be created for stored procedure UCPPPMRT.
Table Updates

System Messages Table (PPPMSG):

New messages will be created for the modified PPPBUT and PPFNDF processing, and for the new PPPMCP updates.
A080187101 4 55BUT: SELECTION FOR MERIT FLAG MUST BE BLANK, Y OR N
A08014810104055MERIT DEPARTMENT CODE IS MISSING
A08014820104055MERIT CONTROL DEPARTMENT CODE CANNOT BE BLANK
A08014830104055MERIT DEPARTMENT CODE IS NOT ON HOME DEPARTMENT TABLE
A08014840104055MERIT CONTROL DEPARTMENT IS NOT ON HOME DEPARTMENT TABLE
A08014850113033HOME DEPARTMENT CODE IS NOT ON MERIT CONTROL DEPARTMENT TABLE
A08014860113033MERIT CONTROL DEPARTMENT CODE IS NOT ON HOME DEPARTMENT TABLE
A08014870113033MERIT CONTROL DEPARTMENT IS NOT ON HOME DEPARTMENT TABLE
A08014880111011MERIT DEPARTMENT = CONTROL DEPARTMENT: TOP CONTROL DEPARTMENT
A08014890113033MERIT CONTROL DEPARTMENT DOES NOT EXIST AS MERIT DEPARTMENT
A08014910103033NO TOP CONTROL DEPARTMENTS ON THE MERIT DEPARTMENT TABLE
A08014920104055SYSTEM PARM FOR MERIT CONTROL LEVELS MISSING OR NOT >= +1
A08014930103033MERIT CONTROL DEPARTMENT TREE DOES NOT END IN TOP CONTROL DEPARTMENT
A08016680103033FND: GROUP CODE FOR MERIT COSTING IS NOT A VALID NUMERIC VALUE

New messages will be created for program PPP675:
A08675010103030PPPBUT TABLE DOES NOT HAVE MERIT CBUC ENTRIES
A08675020105080STD HRS/YEAR NOT FOUND FROM PPPPRM TABLE FOR PARM NUM +041
A08675030115050MERIT CYCLE STATUS CODE MUST BE 'P' OR 'F' TO EXTRACT EDB DATA
A08675040115080PROGRAM ID IN THE SPECIFICATION CARD IS INVALID
A08675050115050MERIT CYCLE ID NOT FOUND IN PPPMCC TABLE
A08675060115080ERROR RETURNED WHILE CALLING PPTCTUTL PROGRAM

New messages will be created for program PPP676:
A08676010105090LOCATION CODE NOT AVAILABLE FROM CCR
A08676020115050MERIT CYCLE STATUS CODE MUST BE 'R' TO UPDATE EDB DATA
A08676030115050PROGRAM ID IN THE SPECIFICATION CARD IS INVALID
A08676040115033NO MINIMUM EDB DATA FOUND
A08676050113033EMPLOYEE LAST DAY ON PAY < SPECIFIED MERIT EFFECTIVE DATE
A08676060113033EMPLOYEE STATUS IS SEPARATED
A08676070113033EMPLOYEE NOT FOUND ON THE EDB
A08676080113033DIST PAY END DATE < SPECIFIED MERIT EFFECTIVE DATE
A08676090113033DISTRIBUTION NOT FOUND ON THE EDB
A08676100113033APPT END DATE < SPECIFIED MERIT EFFECTIVE DATE
A08676110113033APPOINTMENT NOT FOUND ON THE EDB
A08676120113033NO EMPTY APPOINTMENT SLOT FOUND IN PPPAPP TABLE
A08676130113033STD HRS/YEAR NOT FOUND FROM PPPPRM TABLE FOR PARM NUM +041

New messages will be created for program PPP677:
A08677010111010RUN SPECIFICATION RECORD:
A08677020105080RUN SPECIFICATION RECORD IS MISSING OR INVALID
A08677030105080SPECIFICATION RECORD DOES NOT REQUEST ANY REPORTING
A08677040105080CALL TO PPMRTRP1 FAILED
A08677050111010NUMBER OF MERIT TAB DELIMITED RECORDS WRITTEN
A08677110103050REQUESTED CYCLE ID DOES NOT HAVE ANY LOGICAL APPOINTMENT DATA
A08677120105080CALL TO PPFAU007/PPINAFP FAILED
A08677130105080PPPMED DISTRIBUTION DATA MISSING FOR LOGICAL APPOINTMENT
A08677140105080STD HRS/YEAR NOT FOUND FROM PPPPRM TABLE FOR PARM NUM +041
A08677150105080CALL TO PPTCTUTL FAILED

**Code Translation Table (PPPCTT):**

Transactions will be added to translate the Merit Cycle Status Codes.

- **A38 MRTCYSTATU** 50Under Construction
- **A38 MRTCYSTATP** 50Preliminary Merit Extract in progress
- **A38 MRTCYSTATH** 50Prelim Extract Open for HR, Control Point Review
- **A38 MRTCYSTATD** 50Open for Departmental Review
- **A38 MRTCYSTATF** 50Final Merit Extract in progress
- **A38 MRTCYSTATS** 50Final Extract Open for HR, Control Point Review
- **A38 MRTCYSTATO** 50Open for Departmental Input
- **A38 MRTCYSTATIN** 50No Departmental Input, Open to Control Point
- **A38 MRTCYSTATA** 50Open for HR Final Input
- **A38 MRTCYSTATR** 50Ready for EDB Update
- **A38 MRTCYSTATC** 50Cycle Closed
- **A38 MRTCYSTATX** 50Merit Cycle Cancelled

Transactions will be added to define the standard Merit Cycle Performance Evaluation Codes.

- **A38 MRTPRFORMX** 45XNo Performance Evaluation Conducted
- **A38 MRTPRFORM1** 451Fails to Meet, Unsatisfactory
- **A38 MRTPRFORM2** 452Partially Meets, Improvement Needed
- **A38 MRTPRFORM3** 453Meets Expectation, Satisfactory
- **A38 MRTPRFORM4** 454Exceeds Expectations, More than Satisfactory
- **A38 MRTPRFORM5** 455Superior, Outstanding

Transactions will be added to translate the Merit Cycle Sub-Location Codes.

- **A38 MRTSUBLOCGN** 33General Campus
- **A38 MRTSUBLOCMD** 33Medical Center
- **A38 MRTSUBLOCOP** 33Office of the President
- **A38 MRTSUBLOCAG** 33Agriculture and Natural Resources

**Bargaining Unit Table (PPPBUT):**

Bargaining Unit Table transactions will not be issued with the release. Each campus will need to define which bargaining units should be processed via Web Merit. Translations will need to be created to set the BUT_MERIT_SELECT flag to Y.

**Merit Control Department Table (PPPMCP):**

Merit Control Department Table transactions will not be issued with the release. Each campus will need to define its local control structure and create appropriate transactions. The structure will be used to define control groups for Control Point rosters.

**Fund Grouping Table (PPPFND):**

Fund Grouping Table transactions will not be issued with the release. Each campus will need to define its local cost summary structure and create appropriate transactions. The structure will be used by stored procedure PPCSTSPR to organize roster costing data.

**System Parameter Table (PPPPRM):**

System Parameter 106 will be used to define the expected number of levels in the Merit Control Department control structure. The value will be locally defined so a transaction will not be issued with the release. The parameter value will be used to control editing of Merit Control Department table updates, and movement down the control tree during roster processing by the Web Merit application.
Table of Tables (PPPTOT)
A row will be added to the PPPTOT for the new Merit Control Department (Table 44).

Forms

UPAY911 (new):
A new form will be created for the Run Specification records for PPP675. It will allow selection of which Merit Cycle ID (s) should be extracted from the EDB and loaded into Wed Merit accessible tables.

The Run Specification Record will contain the Program ID and which Cycle ID (s) are being extracted and loaded. Multiple Run Specification Records will be processed by PPP675.

<table>
<thead>
<tr>
<th>Columns</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CC 1-11</td>
<td>PPP675-SPEC</td>
<td>Program ID</td>
</tr>
<tr>
<td>CC 12-29</td>
<td>value</td>
<td>Merit Cycle ID on PPPMCC table</td>
</tr>
</tbody>
</table>

UPAY912 (new):
A new form will be created for the Run Specification records for PPP676. It will allow selection of which Merit Cycle ID (s) should be read from Web Merit tables and EDB update transactions and costing transactions created. A Merit file for retroactive processing can also be requested as part of the update process.

The Run Specification Record will contain the Program ID, Merit Cycle ID and whether a Merit file should be produced.

<table>
<thead>
<tr>
<th>Columns</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CC 1-11</td>
<td>PPP676-SPEC</td>
<td>Program ID</td>
</tr>
<tr>
<td>CC 12-29</td>
<td>value</td>
<td>Merit Cycle ID on PPPMCC table</td>
</tr>
<tr>
<td>CC 30</td>
<td>Y</td>
<td>Create file for Merit retroactive processing</td>
</tr>
</tbody>
</table>

UPAY913 (new):
A new form will be created for the Run Specification records for PPP677. It will allow selection of which sub-modules are called by PPP677. Initially only PPMRTRP1 will be callable.

The Run Specification Record will contain the Program ID, Merit Cycle ID and which reports are being selected for creation. Initially only the tab-delimited file will be selectable.

<table>
<thead>
<tr>
<th>Columns</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CC 1-11</td>
<td>PPP677-SPEC</td>
<td>Program ID</td>
</tr>
<tr>
<td>CC 12-29</td>
<td>value</td>
<td>Merit Cycle ID on PPPMCC table</td>
</tr>
<tr>
<td>CC 30</td>
<td>Y</td>
<td>Request PPMRTRP1 processing for tab-delimited file</td>
</tr>
</tbody>
</table>

UPAY910 (new)
A new form will be created for coding Merit Control Department Table transactions. See Attachment for sample.

PPP004 - List Requests (UPAY907B)
The new table 44 Merit Control Department Table will be added to the form.
JCL

PPP004
Sample JCL will be created to add the report DD PPP0444 for the PPPMCP table listing to PPP004.

PPP675(new)
Sample JCL will be created for new program PPP675.

PPP676(new)
Sample JCL will be created for new program PPP676.

PPP677(new)
Sample JCL will be created for new program PPP677.

Load and Unload Merit Tables
Sample JCL for loading and unloading the PPPMRT database will be created.

Load and Unload CTL Tables
Sample JCL for adding the PPPMCP table to the CTL load and unload JCL will be created.
## Attachments

<table>
<thead>
<tr>
<th>Attachment</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attachment A</td>
<td>PPP675 Run Specification Record</td>
</tr>
<tr>
<td>Attachment B</td>
<td>PPP676 Run Specification Record</td>
</tr>
<tr>
<td>Attachment C</td>
<td>PPP677 Run Specification Record</td>
</tr>
<tr>
<td>Attachment D</td>
<td>Merit Control Department Table</td>
</tr>
<tr>
<td>Attachment E</td>
<td>ARSM Rules and Cycle Status</td>
</tr>
<tr>
<td>Attachment F</td>
<td>Costing Summary and the PPPFND Table</td>
</tr>
<tr>
<td>Attachment G</td>
<td>Web Merit ARSM Logic</td>
</tr>
</tbody>
</table>
Payroll/Personnel
PPP675 PROGRAM RUN SPECIFICATION
UPAY911(mm/yy)

<table>
<thead>
<tr>
<th>IDENTIFIER</th>
<th>MERIT CYCLE ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>11</td>
</tr>
<tr>
<td>12</td>
<td>29</td>
</tr>
</tbody>
</table>

PP675-SPEC

**EDB Merit Extract Selection**

Enter a Merit Cycle ID which has been established by the Web Merit application. The Cycle Status must be either:
- P: Preliminary Merit Extract in progress
- F: Final Merit Extract in progress

Multiple Program Run Specification records can be processed by PPP675.

<table>
<thead>
<tr>
<th>Prepared By</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Authorized By</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

RETN: Accounting: Until Action Taken
**Payroll/Personnel**  
**PPP676 PROGRAM RUN SPECIFICATION**  
**UPAY912(mm/yy)**

<table>
<thead>
<tr>
<th>IDENTIFIER</th>
<th>MERIT CYCLE ID</th>
<th>MERIT</th>
</tr>
</thead>
</table>
|            | 1              | 12    | 29 30  
|            | 11             |       |     |

**PPP676-SPEC**

**Merit Cycle ID**

Enter a Merit Cycle ID which has been established by the Web Merit application and is ready for EDB Update. The Cycle Status must be:

- **R:** Ready for EDB Update

**Merit Data**

Enter a “Y” if a Merit file should be created for retroactive Merit processing.

Multiple Program Run Specification records can be processed by PPP676, but the output will be in a single file.

**Prepared By**

<table>
<thead>
<tr>
<th>Prepared By</th>
<th>Date</th>
</tr>
</thead>
</table>

**Authorized By**

<table>
<thead>
<tr>
<th>Authorized By</th>
<th>Date</th>
</tr>
</thead>
</table>

RETN: Accounting: Until Action Taken
Payroll/Personnel
PPP677 PROGRAM RUN SPECIFICATION
UPAY913(mm/yy)

<table>
<thead>
<tr>
<th>IDENTIFIER</th>
<th>MERIT CYCLE ID</th>
<th>TAB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>T</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>F</td>
</tr>
<tr>
<td></td>
<td>I</td>
<td>L</td>
</tr>
</tbody>
</table>

PPP677-SPEC

Merit Cycle ID

Enter a Merit Cycle ID which has been established by the Web Merit application.

Tab-Delimited File Creation

Enter a “Y” if a tab-delimited file should be created from the Merit Cycle’s Logical Appointment and Distribution Tables.

Multiple Program Run Specification records can be processed by PPP677, but the output will be in a single file.

Prepared By

Date

Authorized By

Date

RETN: Accounting: Until Action Taken
<table>
<thead>
<tr>
<th>A/D/C</th>
<th>Table 2-3</th>
<th>Department 4</th>
<th>Department 9</th>
<th>Control Department 10</th>
<th>Control Department 15</th>
</tr>
</thead>
<tbody>
<tr>
<td>44</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>44</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>44</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>44</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>44</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>44</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>44</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>44</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>44</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>44</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>44</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>44</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>44</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>44</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Prepared By

Date

Authorized By

Date

RETN: Until Action Taken
ARSM Rules Table
The ARSM Rules table defines access rules for ARSM access control. Three new rules, MERITADM, MERITCTL and MERITDPT, have been defined for the Web Merit application. The rules apply to Merit Administration, Merit Control Point and Merit Departmental user access.

ARSM Group Table
The ARSM Group table provides a way to link individual user ID’s with Group ID’s for which ARSM security has already been established. New groups may need to be defined for Web Merit access. Groups already established for EDB Update access may be too broad for a more restricted access to the Merit departmental update process. In addition, new groups will be needed for the Merit Administration and Merit Control Point users.

ARSM Association Table
The ARSM Association table links Group or individual user ID’s with rules, and those rules with association data. In the case of the Merit Rules, the association data identifies a Department Number. For the Departmental user, the entries identify specific departments’ roster data that the user will be able to access. For the Administration and Control Point user, the entries identify the top department in a control structure. The user will have access to that department, and all the “controlled” departments below it. These entries must be established locally for all intended users of the Web Merit application. The Department Selection screen presented after roster selection allows Administration and Control Point users to move up and down the control structure, or select all departments down the control structure. The Departmental user can only select the specific department displayed.

The following are the Cycle Status Codes and the ARSM Rules which control access to the Merit Administration and Merit Roster screens for each Cycle Status.

<table>
<thead>
<tr>
<th>Cycle Status Code</th>
<th>Rule required for administration access</th>
<th>Rule required for roster access</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>U</td>
<td>MERITADM</td>
<td>MERITADM</td>
<td>Full Merit Administration update allowed by appropriate users. Roster would not exist unless later cycle status reset to U.</td>
</tr>
<tr>
<td>P</td>
<td>MERITADM</td>
<td>None allowed</td>
<td>Merit Administration update only for Cycle Status.</td>
</tr>
<tr>
<td>H</td>
<td>MERITADM</td>
<td>MERITADM MERITCTL</td>
<td>Merit Administration update only for Cycle Status. Roster view only.</td>
</tr>
<tr>
<td>D</td>
<td>MERITADM MERITCTL MERITDPT</td>
<td>None allowed</td>
<td>Merit Administration update only for Cycle Status. Roster view only.</td>
</tr>
<tr>
<td>F</td>
<td>MERITADM</td>
<td>None allowed</td>
<td>Merit Administration update only for Cycle Status.</td>
</tr>
<tr>
<td>S</td>
<td>MERITADM</td>
<td>MERITADM MERITCTL</td>
<td>Merit Administration update only for Cycle Status. Roster update allowed by appropriate users.</td>
</tr>
<tr>
<td>O</td>
<td>MERITADM</td>
<td>MERITADM MERITCTL</td>
<td>Merit Administration update only for Cycle Status. Roster update allowed by appropriate users.</td>
</tr>
<tr>
<td>departmental input</td>
<td>MERITDPT</td>
<td>MERITADM</td>
<td>MERITCTL</td>
</tr>
<tr>
<td>--------------------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
</tr>
<tr>
<td>N</td>
<td>MERITADM</td>
<td>MERITADM</td>
<td>MERITCTL</td>
</tr>
<tr>
<td>No departmental input; open to control point</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>MERITADM</td>
<td>MERITADM</td>
<td>None allowed</td>
</tr>
<tr>
<td>Open for HR final input</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>MERITADM</td>
<td>None allowed</td>
<td></td>
</tr>
<tr>
<td>Ready for EDB update</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>MERITADM</td>
<td>MERITADM</td>
<td>MERITCTL</td>
</tr>
<tr>
<td>Cycle Closed</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X</td>
<td>MERITADM</td>
<td>None allowed</td>
<td></td>
</tr>
<tr>
<td>Merit cycle cancelled</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Fund Grouping for Merit Costing

Following are sample transactions necessary to update the Fund Grouping Table (PPPFNDF). Please ignore unreal fund range examples I have used. See UPAY866 for the transaction definition.

A41MERITCTL 3 07427 07427 0102 TUITION & FEES
A41MERITCTL 3 19900 19999 0101 GENERAL
A41MERITCTL 3 20000 20999 0201 PRIVATE GIFTS CONTRACTS & GRANT
A41MERITCTL 3 21000 33999 0202 US GOVERNMENT CONTRACT AND GR
A41MERITCTL 3 63000 63999 0301 HOSPITAL

There are several things to note.

The Grouping Definition Code here is MERITCTL. This Fund Grouping would only be used by users performing as a Control Point user. There will also have to be entries for MERITADM and MERITDPT as well. Each can be the same, or different, but they all have to be set up.

The Fund Low Range and High Range are edited. They must both be present, must be numeric, and the High Range must be equal to or greater than the Low Range. In addition, overlapping ranges are not allowed, with the second and subsequent offenders being rejected. However, there is no editing to require that all possible funding is represented. Non-defined costs will be placed into a default row of the costing array.

For example, if one wanted General and Non-General fund grouping there would need to be entries:
A41MERITCTL 3 00000 19899 0201 NON-GENERAL
A41MERITCTL 3 19900 19999 0101 GENERAL
A41MERITCTL 3 20000 99999 0201 NON-GENERAL

However, if acceptable, one could get away with:
A41MERITCTL 3 19900 19999 0101 GENERAL

Everything that did not match would end up in a default row described as OTHER.

Generally speaking, the Group Code can be alphanumeric. An edit will be added to the transaction edit program for the Merit related groupings to ensure that this code is numeric. The code value is used as subscripts for the costing array. The first two digits define a common sub-total group. The second two digits define its position within the sub-total group. Thus, though the first two transactions are in order by Low Range (as they will be on the table), the 02 in the last two digits of 0102 Tuition & Fees will place it after the 0101 General entry in the costing array.

The group Text field is 30 characters long, and will be used to describe the entries on the Costing Summary.

When an online Costing Summary is requested for a selected roster, the old and new rates and calculated costs will be displayed in an array defined by funding. Distribution FAU’s will be passed to a utility program which will access the PPPFNDF table. It will return the Group Code and text. The Group Code will be split into subscripts for accessing the costing array. FAU’s with a common Group Code will be aggregated in the same array position, and thus aggregated on a single Costing Summary line.
Web Merit
ARSM Logic

Logon to Web Merit

Valid User?

RACF validation of AUTHID and password

Main Menu Selection

Valid User?

Admin

Function Selection

Valid User?

Roster

ARSM validation of Personal, Admin, CTL and/or Dept Rules

Cycle Selection

ARSM validation of User Rule(s) for Cycle Status

Create/Update Cycle

Cycle Selection

ARSM validation of User Rule(s) for Cycle Status

Rule/Department Selection

Rule?

CTL/Admin

Access Selection

Get One?

Y

N

Dept Rule

Y

N

Develops Up/Down department list

Select single department

Develops roster department list

Rule?

N

Y

Get All?

N

Up/Down?

N

Y

Roster Displayed

ARSM Logic

Call UCPPPMRT AUTHID
Admin Access
Returns Permission

ARSM validation for Admin Rule

Call UCPPPMRT AUTHID
Roster Access
Returns Permission

Valid User?

Call UCPPPMRT AUTHID
Session ID
Cycle Status
Cycle Access

Call PPMDSPR Session ID
Department
Up/Down Access
Returns MDP Level

Call UCPPPMRT AUTHID
PERSONAL Rule ID

Valid User?

N

Y

N

Y

Valid User for Status?

ARSM validation of Personal, Admin, CTL and/or Dept Rules

ARSM validation of User Rule(s) for Cycle Status

Develops Up/Down department list

Develops roster department list

Rule/Department Selection

Call PPMDSPR Session ID
Department
One/All Access
Returns MDP Level

Call UCPPPMRT AUTHID
Session ID
Return Permission

Rule/Department Selection

Call UCPPPMRT AUTHID
Session ID
Return Permission

Rule/Department Selection

Call UCPPPMRT AUTHID
Session ID
Return Permission

Rule/Department Selection

Call UCPPPMRT AUTHID
Session ID
Return Permission

Rule/Department Selection

Call UCPPPMRT AUTHID
Session ID
Return Permission
Roster Displayed

Roster Data Entry

Submit?

Validation of PERSONAL Rule ID vs. updated ID's

Updating own ID?

Flag set on MLA row

N

Y

Y

N