

31 August 2022

**Patient Level Cost Allocation (PLCA) for the
Standard Inpatient Data Record (SIDR) for the
MHS Data Repository (MDR)
(Version 1.13.00)**

Current Specification

Revision History

Version	Date	Originator	Para/Tbl/Fig	Description of Change
1.00.00	08/11/2010	K. Hutchinson	<ul style="list-style-type: none"> • Entire document 	<ul style="list-style-type: none"> • Initial specification.
1.01.00	11/24/2010	K. Hutchinson	<ul style="list-style-type: none"> • 	<ul style="list-style-type: none"> • Correcting computation of Ancillary costs for FY09+.
1.02.00	08/29/2011	K. Hutchinson	<ul style="list-style-type: none"> • Table 1 • Table 2 	<ul style="list-style-type: none"> • Updated for new tables • Updated PSP computation for FY10+
1.03.00	09/07/2012	K. Hutchinson	<ul style="list-style-type: none"> • Table 1 • Table 2 	<ul style="list-style-type: none"> • Updated for new FY11-13 tables • Updated PSP computation for FY11+
1.04.00	08/06/2013	K. Hutchinson	<ul style="list-style-type: none"> • Table 1 • Table 2 	<ul style="list-style-type: none"> • Updated for new FY12-14 tables • Updated PSP computation for FY12+
1.05.00	10/02/2014	K. Hutchinson	<ul style="list-style-type: none"> • Table 1 • Table 2 	<ul style="list-style-type: none"> • Updated for new FY13-15 tables • Updated PSP computation for FY13+
1.06.00	08/06/2015	Y. Alexander/K. Hutchinson	<ul style="list-style-type: none"> • Table 1 • Table 2 	<ul style="list-style-type: none"> • Updated for new FY14-16 tables • Updated PSP computation for FY14+
1.07.00	07/25/2016	Y. Alexander/K. Hutchinson	<ul style="list-style-type: none"> • Table 1 • Table 2 	<ul style="list-style-type: none"> • Updated for new FY15-17 tables • Updated PSP computation for FY15+
1.08.00	09/08/2017	Y. Alexander/K. Hutchinson	<ul style="list-style-type: none"> • Table 1 • Table 2 	<ul style="list-style-type: none"> • Updated for new FY16-18 tables • Updated PSP computation for FY16+
1.09.00	08/02/2018	Y. Alexander/K. Hutchinson	<ul style="list-style-type: none"> • Table 1 • Table 2 	<ul style="list-style-type: none"> • Updated for new FY17-19 tables • Updated PSP computation for FY17+
1.10.00	07/19/2019	Y. Alexander/K. Hutchinson	<ul style="list-style-type: none"> • Table 1 • Table 2 	<ul style="list-style-type: none"> • Updated for new FY18-20 tables • Updated PSP computation for FY18+
1.11.00	08/13/2020	Y. Alexander/K. Hutchinson	<ul style="list-style-type: none"> • Table 1 • Table 2 	<ul style="list-style-type: none"> • Updated for new FY19-21 tables • Updated PSP computation for FY19+
1.12.00	08/10/2021	Y. Alexander/K. Hutchinson	<ul style="list-style-type: none"> • Table 1 • Table 2 	<ul style="list-style-type: none"> • Updated for new FY20-22 tables • Updated PSP computation for FY20+
1.13.00	08/31/2022	Y. Alexander/K. Hutchinon	<ul style="list-style-type: none"> • Table 1 • Table 2 	<ul style="list-style-type: none"> • Updated for new FY21-23 tables • Updated PSP computation for FY21+

PATIENT LEVEL COST ALLOCATION (PLCA) FOR THE MDR

I. SOURCE

The code and reference tables are maintained and produced by Defense Health Agency (DHA) contract support using information from the Standard Inpatient Data Record (SIDR) and the Medical Expense and Performance Reporting System (MEPRS).

II. TRANSMISSION (FORMAT AND FREQUENCY)

The code and reference tables are updated on an annual basis, usually around July/August. The code is provided in this specification. The reference tables are posted to the MDR (/mdr/ref and /mdr/aref), the exact location is specified in an email to all pertinent parties.

III. PROCESS

- Step 1: the SIDR records (post grouping) are processed through the "CLEANUP" program provided in Appendix A. The output should be saved as a permanent SAS dataset as it is used in Step 2.
- Step 2: the "cleaned" SIDR is processed through the PLCA program in Appendix B using the information from Tables 1 and 2 as inputs based on the FY being processed. The dyymmdd dates are the most recent listed in the directory.
- Step 3: the SIDR records (post grouping) are processed through the "pricing" program provided in Appendix C.

Table 1. Reference Tables used in PLCA Application and Prices

FY	Description	Table Location
2007	DRG Weights DRG Lab, Rad, and PWP Weights DRG Surgical Indicator PLCA Unit Costs Prices	/mdr/aref/sidr/drgwts/fy07/dyymmdd/v24.sas7bdat /mdr/aref/sidr/pwp/fy07/dyymmdd/pwp.sas7bdat /mdr/aref/sidr/drgsurg/dyymmdd/criteria.sas7bdat /mdr/aref/sidr/costs/fy07/dyymmdd/costs.sas7bdat /mdr/aref/sidr/prices/fy07/dyymmdd/fy07.sas7bdat
2008	DRG Weights DRG Lab, Rad, and PWP Weights DRG Surgical Indicator PLCA Unit Costs Prices	/mdr/aref/sidr/drgwts/fy08/dyymmdd/v25.sas7bdat /mdr/aref/sidr/pwp/fy08/dyymmdd/pwp.sas7bdat /mdr/aref/sidr/drgsurg/dyymmdd/criteria.sas7bdat /mdr/aref/sidr/costs/fy08/dyymmdd/costs.sas7bdat /mdr/aref/sidr/prices/fy08/dyymmdd/fy08.sas7bdat
2009	MSDRG Weights MSDRG Lab, Rad, and PWP Weights MSDRG Surgical Indicator PLCA Unit Costs Prices	/mdr/aref/sidr/msdrgwts/fy09/dyymmdd/v26.sas7bdat /mdr/aref/sidr/pwp/fy09/dyymmdd/pwp.sas7bdat /mdr/aref/sidr/msdrgsurg/dyymmdd/criteria.sas7bdat /mdr/aref/sidr/costs/fy09/dyymmdd/costs.sas7bdat /mdr/aref/sidr/prices/fy09/dyymmdd/fy09.sas7bdat
2010	MSDRG Weights MSDRG Lab, Rad, and PWP Weights MSDRG Surgical Indicator PLCA Unit Costs Prices	/mdr/aref/sidr/msdrgwts/fy10/dyymmdd/v27.sas7bdat /mdr/aref/sidr/pwp/fy10/dyymmdd/pwp.sas7bdat /mdr/aref/sidr/msdrgsurg/dyymmdd/criteria.sas7bdat /mdr/aref/sidr/costs/fy10/dyymmdd/costs.sas7bdat /mdr/aref/sidr/prices/fy10/dyymmdd/fy10.sas7bdat

FY	Description	Table Location
2011	MSDRG Weights MSDRG Lab, Rad, and PWP Weights MSDRG Surgical Indicator PLCA Unit Costs Prices	/mdr/aref/sidr/msdrgwts/fy11/dyymmdd/v28.sas7bdat /mdr/aref/sidr/pwp/fy11/dyymmdd/pwp.sas7bdat /mdr/aref/sidr/msdrgsurg/dyymmdd/criteria.sas7bdat /mdr/aref/sidr/costs/fy11/dyymmdd/costs.sas7bdat /mdr/aref/sidr/prices/fy11/dyymmdd/fy11.sas7bdat
2012	MSDRG Weights MSDRG Lab, Rad, and PWP Weights MSDRG Surgical Indicator PLCA Unit Costs Prices	/mdr/aref/sidr/msdrgwts/fy12/dyymmdd/v29.sas7bdat /mdr/aref/sidr/pwp/fy12/dyymmdd/pwp.sas7bdat /mdr/aref/sidr/msdrgsurg/dyymmdd/criteria.sas7bdat /mdr/aref/sidr/costs/fy12/dyymmdd/costs.sas7bdat /mdr/aref/sidr/prices/fy12/dyymmdd/fy12.sas7bdat
2013	MSDRG Weights MSDRG Lab, Rad, and PWP Weights MSDRG Surgical Indicator PLCA Unit Costs Prices	/mdr/aref/sidr/msdrgwts/fy13/dyymmdd/v30.sas7bdat /mdr/aref/sidr/pwp/fy13/dyymmdd/pwp.sas7bdat /mdr/aref/sidr/msdrgsurg/dyymmdd/criteria.sas7bdat /mdr/aref/sidr/costs/fy13/dyymmdd/costs.sas7bdat /mdr/aref/sidr/prices/fy13/dyymmdd/fyxx.sas7bdat
2014	MSDRG Weights MSDRG Lab, Rad, and PWP Weights MSDRG Surgical Indicator PLCA Unit Costs Prices	/mdr/aref/sidr/msdrgwts/fy14/dyymmdd/v31.sas7bdat /mdr/aref/sidr/pwp/fy14/dyymmdd/pwp.sas7bdat /mdr/aref/sidr/msdrgsurg/dyymmdd/criteria.sas7bdat /mdr/aref/sidr/costs/fy14/dyymmdd/costs.sas7bdat /mdr/aref/sidr/prices/fy14/dyymmdd/fy14.sas7bdat
2015	MSDRG Weights MSDRG Lab, Rad, and PWP Weights MSDRG Surgical Indicator PLCA Unit Costs Prices	/mdr/aref/sidr/msdrgwts/fy15/dyymmdd/v32.sas7bdat /mdr/aref/sidr/pwp/fy15/dyymmdd/pwp.sas7bdat /mdr/aref/sidr/msdrgsurg/dyymmdd/criteria.sas7bdat /mdr/aref/sidr/costs/fy15/dyymmdd/costs.sas7bdat /mdr/aref/sidr/prices/fy15/dyymmdd/fy15.sas7bdat
2016	MSDRG Weights MSDRG Lab, Rad, and PWP Weights MSDRG Surgical Indicator PLCA Unit Costs Prices	/mdr/aref/sidr/msdrgwts/fyxx/dyymmdd/vxx.sas7bdat /mdr/aref/sidr/pwp/fyxx/dyymmdd/pwp.sas7bdat /mdr/aref/sidr/msdrgsurg/dyymmdd/criteria.sas7bdat /mdr/aref/sidr/costs/fyxx/dyymmdd/costs.sas7bdat /mdr/aref/sidr/prices/fyxx/dyymmdd/fyxx.sas7bdat
2017	MSDRG Weights MSDRG Lab, Rad, and PWP Weights MSDRG Surgical Indicator PLCA Unit Costs Prices	/mdr/aref/sidr/msdrgwts/fyxx/dyymmdd/vxx.sas7bdat /mdr/aref/sidr/pwp/fyxx/dyymmdd/pwp.sas7bdat /mdr/aref/sidr/msdrgsurg/dyymmdd/criteria.sas7bdat /mdr/aref/sidr/costs/fyxx/dyymmdd/costs.sas7bdat /mdr/aref/sidr/prices/fyxx/dyymmdd/fyxx.sas7bdat
2018	MSDRG Weights MSDRG Lab, Rad, and PWP Weights MSDRG Surgical Indicator PLCA Unit Costs Prices	/mdr/aref/sidr/msdrgwts/fyxx/dyymmdd/vxx.sas7bdat /mdr/aref/sidr/pwp/fyxx/dyymmdd/pwp.sas7bdat /mdr/aref/sidr/msdrgsurg/dyymmdd/criteria.sas7bdat /mdr/aref/sidr/costs/fyxx/dyymmdd/costs.sas7bdat /mdr/aref/sidr/prices/fyxx/dyymmdd/fyxx.sas7bdat
2019	MSDRG Weights MSDRG Lab, Rad, and PWP Weights MSDRG Surgical Indicator PLCA Unit Costs Prices	/mdr/aref/sidr/msdrgwts/fyxx/dyymmdd/vxx.sas7bdat /mdr/aref/sidr/pwp/fyxx/dyymmdd/pwp.sas7bdat /mdr/aref/sidr/msdrgsurg/dyymmdd/criteria.sas7bdat /mdr/aref/sidr/costs/fyxx/dyymmdd/costs.sas7bdat /mdr/aref/sidr/prices/fyxx/dyymmdd/fyxx.sas7bdat
2020	MSDRG Weights MSDRG Lab, Rad, and PWP Weights MSDRG Surgical Indicator PLCA Unit Costs Prices	/mdr/aref/sidr/msdrgwts/fyxx/dyymmdd/vxx.sas7bdat /mdr/aref/sidr/pwp/fyxx/dyymmdd/pwp.sas7bdat /mdr/aref/sidr/msdrgsurg/dyymmdd/criteria.sas7bdat /mdr/aref/sidr/costs/fyxx/dyymmdd/costs.sas7bdat /mdr/aref/sidr/prices/fyxx/dyymmdd/fyxx.sas7bdat

FY	Description	Table Location
2021+ xx =21, 22, 23 ## = 38, 39, 40	MSDRG Weights MSDRG Lab, Rad, and PWP Weights MSDRG Surgical Indicator PLCA Unit Costs Prices	/mdr/aref/sidr/msdrgwts/fyxx/dyymmdd/vxx.sas7bdat /mdr/aref/sidr/pwp/fyxx/dyymmdd/pwp.sas7bdat /mdr/aref/sidr/msdrgsurg/dyymmdd/criteria.sas7bdat /mdr/aref/sidr/costs/fyxx/dyymmdd/costs.sas7bdat /mdr/aref/sidr/prices/fyxx/dyymmdd/fyxx.sas7bdat

Table 2. Formula for PSP Computation

FY	PSP Formula
2007	If surg=1 then psp=round((0.1034) + (1.0846*calcdays) + (8.8764*pwp), .0001); Else if surg=0 then psp=round((-1.0818) + (1*calcdays) + (7.6618*pwp), .0001);
2008	If surg=1 then psp=round((-0.4662) + (1.1985*calcdays) + (8.4605*pwp), .0001); Else if surg=0 then psp=round((-1.1616) + (1*calcdays) + (6.9897*pwp), .0001);
2009	If surg=1 then psp=round((-0.1346) + (1.0035*calcdays) + (8.7303*pwp), .0001); Else if surg=0 then psp=round((-1.3609) + (1*calcdays) + (7.2547*pwp), .0001);
2010	If surg=1 then psp=round((-0.1417) + (1.2944*calcdays) + (7.0080*pwp), .0001); Else if surg=0 then psp=round((-1.1764) + (1*calcdays) + (5.5714*pwp), .0001);
2011	If surg=1 then psp=round((-0.6755) + (1.1964*calcdays) + (8.8322*pwp), .0001); Else if surg=0 then psp=round((-1.1752) + (1*calcdays) + (7.4046*pwp), .0001);
2012	If surg=1 then psp=round((-0.1305) + (1.1413*calcdays) + (8.6756*pwp), .0001); Else if surg=0 then psp=round((-0.9537) + (1*calcdays) + (7.1738*pwp), .0001);
2013	If surg=1 then psp=round((-1.2084) + (1.0846*calcdays) + (9.1388*pwp), .0001); Else if surg=0 then psp=round((-1.2347) + (1*calcdays) + (7.3032*pwp), .0001);
2014	If surg=1 then psp=round((-0.9342) + (1.1889*calcdays) + (6.7374*pwp), .0001); Else if surg=0 then psp=round((-1.1078) + (1*calcdays) + (4.6526*pwp), .0001);
2015	If surg=1 then psp=round((0.3212) + (1.3783*calcdays) + (6.6686*pwp), .0001); Else if surg=0 then psp=round((-0.6981) + (1*calcdays) + (5.3378*pwp), .0001);
2016	If surg=1 then psp=round((1.3938) + (1.4496*calcdays) + (6.0188*pwp), .0001); Else if surg=0 then psp=round((-0.0862) + (1*calcdays) + (5.0420*pwp), .0001);
2017	If surg=1 then psp=round((-0.1076) + (1.156*calcdays) + (7.3493*pwp), .0001); Else if surg=0 then psp=round((-0.9567) + (1*calcdays) + (5.983*pwp), .0001);
2018	If surg=1 then psp=round((0.3220) + (0.9536*calcdays) + (5.1046*pwp), .0001); Else if surg=0 then psp=round((-1.5613) + (1*calcdays) + (2.6482*pwp), .0001);
2019	If surg =1 then psp=round((-0.1294) + (1.1236*calcdays) + (6.8928*pwp), .0001); Else if surg =0 then psp=round((-1.3363) + (1*calcdays) + (6.0247*pwp), .0001);
2020	If surg=1 then psp=round((0.0196) + (1.3148*calcdays) + (6.9587*pwp), .0001); Else if surg=0 then psp=round((-0.9344) + (1*calcdays) + (6.0180*pwp), .0001);
2021+	If surg=1 then psp=round((0.9868) + (1.278*calcdays) + (6.3932*pwp), .0001); Else if surg=0 then psp=round((-1.027) + (1*calcdays) + (6.1805*pwp), .0001);

IV. REFRESH FREQUENCY

- FYs for which the PLCA unit costs are based on are typically not updated (for example, FY21-based unit costs for FY21 are static; FY21-based unit costs for FY22 and FY23 will be updated the following year using FY21 data).

Appendix A. "CLEAN UP" Program

```
OPTIONS LINESIZE=79 PAGESIZE=500;

* CREATE A NEW TOTAL DAYS VARIABLE WHICH IS WHERE WE TAKE CARE OF
* FIXING THOSE RECORDS WHERE DMISDAYS = 0. THIS DATASET WILL BE MERGED
* WITH THE CLEANED UP DATA AT THE END OF THE PROGRAM;

DATA FIRST; SET IN.FYXX;
TOTDAYS=DMISDAYS*1;
RECNUM=_N_;
PROC SORT DATA=FIRST; BY RECNUM;

* CREATE A SUBSET WITH ONLY THE VARIABLES NEEDED FOR CLEAN UP;
DATA SECOND; SET FIRST(KEEP=CLNADM CLN2 CLN3 CLNDISP CLN1DAYS
CLN2DAYS CLN3DAYS CLN4DAYS MTF DMISDAYS TOTDAYS RECNUM);

DISPMEPR=SUBSTR(CLNDISP,1,3);

* CREATE DAYS VARIABLES THAT ARE ASSOCIATED WITH THE FOUR POSSIBLE
* MEPRS CODES. THIS IS DONE SO THAT THE ORIGINAL DATA IS NOT CHANGED;
DATA ONE; SET SECOND;
DAYS1=CLN1DAYS;
DAYS2=CLN2DAYS;
DAYS3=CLN3DAYS;
DAYS4=CLN4DAYS;

* CREATE BED DAYS VARIABLES WHICH WILL BE USED IN THE ALGORITHM TO CLEAN UP THE DATA;
BEDS1=CLN1DAYS;
BEDS2=SUM(OF CLN1DAYS CLN2DAYS);
BEDS3=SUM(OF CLN1DAYS CLN2DAYS CLN3DAYS);
BEDS4=SUM(OF CLN1DAYS CLN2DAYS CLN3DAYS CLN4DAYS);

MEPR1=SUBSTR(CLNADM,1,3);
MEPR2=SUBSTR(CLN2,1,3);
MEPR3=SUBSTR(CLN3,1,3);
MEPR4=SUBSTR(CLNDISP,1,3);

*****
DATA ZERODAYS OKAY4 OKAY3 OKAY2 OKAY1 REDO; SET ONE;

IF DMISDAYS=0 AND BEDS4=0 THEN DO;
  IF MEPR4= ' ' THEN DO;
    TOTDAYS=1;
    DAYS1=1;
    MEPR1='AAA';
    OUTPUT ZERODAYS;
  END;

  ELSE IF MEPR4 NE ' ' THEN DO;
    TOTDAYS=1;
    DAYS1=1;
    MEPR1=MEPR4;
    MEPR4= ' ';
    OUTPUT ZERODAYS;
  END;
END;

ELSE IF (DMISDAYS NE 0 AND BEDS4 NE 0) AND BEDS4 = DMISDAYS
  THEN OUTPUT OKAY4;

ELSE IF BEDS3 = DMISDAYS THEN DO;
  DAYS4=0;
  MEPR4= ' ';
  OUTPUT OKAY3;
```

```

END;

ELSE IF BEDS2 = DMISDAYS THEN DO;
  DAYS3=0;
  MEPR3=' ';
  DAYS4=0;
  MEPR4=' ';
  OUTPUT OKAY2;
END;

ELSE IF BEDS1 = DMISDAYS THEN DO;
  DAYS2=0;
  MEPR2=' ';
  DAYS3=0;
  MEPR3=' ';
  DAYS4=0;
  MEPR4=' ';
  OUTPUT OKAY1;
END;

ELSE OUTPUT REDO;

PROC DATASETS NOLIST; DELETE ONE;

DATA FIXREDO; SET REDO;
IF DMISDAYS > BEDS4 THEN DAYS4 = CLN4DAYS + (DMISDAYS - BEDS4);
  ELSE IF DMISDAYS < BEDS4 THEN DAYS4 = DMISDAYS - BEDS3;

PROC DATASETS NOLIST; DELETE REDO;

DATA TOG; SET ZERODAYS OKAY4 OKAY3 OKAY2 OKAY1 FIXREDO;

SUMDAYS=SUM(OF DAYS1 DAYS2 DAYS3 DAYS4);

DROP CLNADM CLN2 CLN3 CLNDISP CLN1DAYS CLN2DAYS CLN3DAYS CLN4DAYS
DMISDAYS BEDS1 BEDS2 BEDS3 BEDS4;

PROC DATASETS NOLIST;
DELETE ZERODAYS OKAY4 OKAY3 OKAY2 OKAY1 FIXREDO;

* IF ANY RECORDS MAKE IT TO THE CHECK DATASET, THEY MUST BE CHECKED;
DATA CHECK; SET TOG; IF TOTDAYS NE SUMDAYS;

PROC SORT DATA=TOG; BY RECNUM;

DATA OUT.OUTPUT_DATASET_NAME;
MERGE FIRST TOG; BY RECNUM;
DROP SUMDAYS;
RUN;

```


Appendix B. PLCA Program

libname in1 CLEANED SIDR
libname in2 DRG/MSDRG WEIGHTS, FY09+ use the MSDRG
libname in3 DRG/MSDRG SURGICAL INDICATOR, FY09+ use the MSDRG
libname in4 PWP
libname in5 PLCA UNIT COSTS

OPTIONS PAGESIZE=5000 LINESIZE=130;

APPEND DENOMINATORS (PSP, LAB, RAD, ETC) AND APPLY UNIT COSTS)
IN.FY09CLEAND0419 CONTAINS VARIABLE MSDRGSURG WITH VALUES M AND S.
SINCE IT IS ALPHABETIC (S,M) RATHER THAN NUMERIC (1,0) I WILL not DROP IT HERE. SR. 7/27/10

DATA ONE; SET IN1.CLEANED SIDR;
DRGNUM=VARIABLE NAME FOR DRG/MSDRG BASED ON FY*1;
MEPR=SUBSTR(CLNDISP,1,3);

IF MTF='0052' AND CLNDISP='AFAB' THEN DELETE;

* THIS STATEMENT IS ONLY NECESSARY IF SIDR RECORDS ALREADY HAVE THE FIELDS AND ARE BEING RERUN
OR RETROFITTED;
DROP FCOTHANC FCANCLAB FCANCRAD FCCLNSAL FCDIRECT FCICU FCOTHSAL FCSUPPRT FCSURG
VCOTHANC VCANCLAB VCANCRAD VCCLNSAL VCDIRECT VCICU VCOTHSAL VCSUPPRT VCSURG
FULLCOST INCCOST;
RUN;

DATA WTS; SET IN2.V###(KEEP=DRGNUM WEIGHT); RUN;

DATA PWP; SET IN4.PWP(KEEP= VARIABLE NAME FOR DRG/MSDRG BASED ON FY PWP LAB RAD);
DRGNUM= VARIABLE NAME FOR DRG/MSDRG BASED ON FY *1;
DROP VARIABLE NAME FOR DRG/MSDRG BASED ON FY;

DATA SRG;
SET IN3.criteria(KEEP=VARIABLE NAME FOR DRG/MSDRG VARIABLE NAME FOR DRGSURG/MSDRGSURG);
DRGNUM= VARIABLE NAME FOR DRG/MSDRG *1;
SURG= VARIABLE NAME FOR DRGSURG/MSDRGSURG;
DROP VARIABLE NAME FOR DRG/MSDRG VARIABLE NAME FOR DRGSURG/MSDRGSURG; RUN;

PROC SORT DATA=ONE; BY DRGNUM; RUN;
PROC SORT DATA=WTS; BY DRGNUM; RUN;
PROC SORT DATA=PWP; BY DRGNUM; RUN;
PROC SORT DATA=SRG; BY DRGNUM; RUN;

* ADD DRG WEIGHT;
DATA TOG; MERGE ONE(IN=DAT) WTS; BY DRGNUM; IF DAT; RUN;

* ADD PWP, LAB, AND RAD;
DATA TOG2; MERGE TOG(IN=DAT) PWP; BY DRGNUM; IF DAT; RUN;

* ADD SURGICAL INDICATOR, COMPUTE CALCDAYS, AND PSP;
DATA RECS; MERGE TOG2(IN=DAT) SRG; BY DRGNUM; IF DAT;

IF DMISDAYS=0 THEN CALCDAYS=1; ELSE CALCDAYS=DMISDAYS;

* USE VALUES FROM TABLE 2;
IF SURG=1 THEN PSP=ROUND((###) + (###*CALCDAYS) + (###*PWP), .0001);
ELSE IF SURG=0 THEN PSP=ROUND((###) + (1*CALCDAYS) + (###*PWP), .0001);

IF WEIGHT=. THEN WEIGHT=0;

IF PWP=. THEN PWP=0;
IF LAB=. THEN LAB=0;
IF RAD=. THEN RAD=0;
RUN;

*****,

DATA UNITS; SET IN5.PLCA UNIT COSTS; RUN;

*****,

* COMPUTE ANCLAB, ANCRAD, OTHANC, CLNSAL, SURG, AND ICU PORTION FOR TOTAL;
PROC SORT DATA=UNITS; BY MTF MEPR;
PROC SORT DATA=RECS; BY MTF MEPR;

DATA TOG; MERGE RECS(IN=DAT) UNITS; BY MTF MEPR; IF DAT;

FCANCLAB=ROUND(ANCLAB_F*LAB,.01);
FCANCRAD=ROUND(ANCRAD_F*RAD,.01);
FCOTHANC=ROUND(ANCOTH_F*TOTRWP for FY08 back or MSDRGRWP for FY09+,.01);
FCCLNSAL=ROUND(CLNSAL_F*PSP,.01);
FCSURG=ROUND(SURG_F*WEIGHT*SURG,.01);
FCICU=ROUND(ICU_F*ICUDAYS,.01);

VCANCLAB=ROUND(ANCLAB_I*LAB,.01);
VCANCRAD=ROUND(ANCRAD_I*RAD,.01);
VCOTHANC=ROUND(ANCOTH_I*TOTRWP for FY08 back or MSDRGRWP for FY09+,.01);
VCCLNSAL=ROUND(CLNSAL_I*PSP,.01);
VCSURG=ROUND(SURG_I*WEIGHT*SURG,.01);
VCICU=ROUND(ICU_I*ICUDAYS,.01);

KEEP RECNUM MTF MEPR
FCANCLAB FCANCRAD FCOTHANC FCCLNSAL FCSURG FCICU
VCANCLAB VCANCRAD VCOTHANC VCCLNSAL VCSURG VCICU
ANCLAB_F ANCRAD_F ANCOTH_F CLNSAL_F SURG_F ICU_F
ANCLAB_I ANCRAD_I ANCOTH_I CLNSAL_I SURG_I ICU_I
SURG TOTAL RWP FROM DRG/MSDRG PSP ICUDAYS LAB RAD;

RUN;

* BE SURE TO INCLUDE SURG IN THE KEEP STATEMENT TO RUN THIS CHECK. IF THE
FULL COST UNIT COST IS MISSING OR ZERO, SO IS THE VARIABLE UNIT COST;
DATA CHECK5A; SET TOG;
IF SURG=0 THEN DELETE;
IF SURG_F=. OR SURG_F=0 THEN DELETE;
IF FCSURG=. OR FCSURG=0 OR VCSURG=. OR VCSURG=0;
PROC FREQ DATA=CHECK5A; TABLES MTF*MEPR / LIST; TITLE'SURG CHECK';

* BE SURE TO INCLUDE ICUDAYS IN THE KEEP STATEMENT TO RUN THIS CHECK. IF THE
FULL COST UNIT COST IS MISSING OR ZERO, SO IS THE VARIABLE UNIT COST;
DATA CHECK5B; SET TOG;
IF ICUDAYS=0 THEN DELETE;
IF ICU_F=. OR ICU_F=0 THEN DELETE;
IF FCICU=. OR FCICU=0 OR VCICU=. OR VCICU=0;
PROC FREQ DATA=CHECK5B; TABLES MTF*MEPR / LIST; TITLE'ICU CHECK';

* BE SURE TO INCLUDE PSP IN THE KEEP STATEMENT TO RUN THIS CHECK. IF THE
FULL COST UNIT COST IS MISSING OR ZERO, SO IS THE VARIABLE UNIT COST;
DATA CHECK5C; SET TOG;
IF PSP=0 OR PSP=. THEN DELETE;
IF CLNSAL_F=. OR CLNSAL_F=0 THEN DELETE;
IF FCCLNSAL=. OR FCCLNSAL=0 OR VCCLNSAL=. OR VCCLNSAL=0;
PROC FREQ DATA=CHECK5C; TABLES MTF*MEPR / LIST; TITLE'CLNSAL CHECK';
RUN;

* BE SURE TO INCLUDE LAB IN THE KEEP STATEMENT TO RUN THIS CHECK. IF THE
FULL COST UNIT COST IS MISSING OR ZERO, SO IS THE VARIABLE UNIT COST;

```

DATA CHECK5D; SET TOG;
IF LAB=0 OR LAB=. THEN DELETE;
IF ANCLAB_F=. OR ANCLAB_F=0 THEN DELETE;
IF FCANCLAB=. OR FCANCLAB=0 OR VCANCLAB=. OR VCANCLAB=0;
PROC FREQ DATA=CHECK5D; TABLES MTF*MEPR / LIST; TITLE'LAB CHECK';
RUN;

```

```

* BE SURE TO INCLUDE RAD IN THE KEEP STATEMENT TO RUN THIS CHECK. IF THE
  FULL COST UNIT COST IS MISSING OR ZERO, SO IS THE VARIABLE UNIT COST;
DATA CHECK5E; SET TOG;
IF RAD=0 OR RAD=. THEN DELETE;
IF ANCRAD_F=. OR ANCRAD_F=0 THEN DELETE;
IF FCANCRAD=. OR FCANCRAD=0 OR VCANCRAD=. OR VCANCRAD=0;
PROC FREQ DATA=CHECK5E; TABLES MTF*MEPR / LIST; TITLE'RAD CHECK';
RUN;

```

```

* BE SURE TO INCLUDE TOTRWP IN THE KEEP STATEMENT TO RUN THIS CHECK. IF THE
  FULL COST UNIT COST IS MISSING OR ZERO, SO IS THE VARIABLE UNIT COST;
DATA CHECK5F; SET TOG;
IF TOTRWP=0 OR TOTRWP=. THEN DELETE;
IF ANCOTH_F=. OR ANCOTH_F=0 THEN DELETE;
IF FCOTHANC=. OR FCOTHANC=0 OR VCOOTHANC=. OR VCOOTHANC=0;
PROC FREQ DATA=CHECK5F; TABLES MTF*MEPR / LIST; TITLE'OTHER ANC CHECK';
RUN;

```

*****,

```

* COMPUTE DIR AND SUP, MEPR1;
PROC SORT DATA=UNITS; BY MTF MEPR1;
PROC SORT DATA=RECS; BY MTF MEPR1;

```

```

DATA TOG1; MERGE RECS(IN=DAT) UNITS; BY MTF MEPR1; IF DAT;

```

```

DIR1F=ROUND(DIRECT_F*DAYS1,.01);
SUP1F=ROUND(SUPPRT_F*DAYS1,.01);
SAL1F=ROUND(OTHSAL_F*DAYS1,.01);

```

```

DIR1I=ROUND(DIRECT_I*DAYS1,.01);
SUP1I=ROUND(SUPPRT_I*DAYS1,.01);
SAL1I=ROUND(OTHSAL_I*DAYS1,.01);

```

```

KEEP RECNUM MTF MEPR1 DIR1F SUP1F SAL1F DIR1I SUP1I SAL1I DAYS1
  DIRECT_F DIRECT_I SUPPRT_F SUPPRT_I OTHSAL_F OTHSAL_I;
RUN;

```

```

* BE SURE TO INCLUDE DAYS1 IN THE KEEP STATEMENT TO RUN THIS CHECK;
DATA CHECK1; SET TOG1;
IF DAYS1=0 OR DAYS1=. THEN DELETE;
IF DIRECT_F=. OR DIRECT_F=0 OR SUPPRT_F=. OR SUPPRT_F=0 OR OTHSAL_F=. OR OTHSAL_F=0 THEN DELETE;
IF DIR1F=. OR DIR1F=0 OR SUP1F=. OR SUP1F=0 OR SAL1F=. OR SAL1F=0 OR
  DIR1I=. OR DIR1I=0 OR SUP1I=. OR SUP1I=0 OR SAL1I=. OR SAL1I=0;
PROC FREQ DATA=CHECK1; TABLES MTF*MEPR1 / LIST; TITLE'MEPR1 CHECK';

```

*****,

```

* COMPUTE DIR AND SUP, MEPR2;
PROC SORT DATA=UNITS; BY MTF MEPR2;
PROC SORT DATA=RECS; BY MTF MEPR2;

```

```

DATA TOG2; MERGE RECS(IN=DAT) UNITS; BY MTF MEPR2; IF DAT;

```

```

DIR2F=ROUND(DIRECT_F*DAYS2,.01);
SUP2F=ROUND(SUPPRT_F*DAYS2,.01);
SAL2F=ROUND(OTHSAL_F*DAYS2,.01);

```

```

DIR2I=ROUND(DIRECT_I*DAYS2,.01);
SUP2I=ROUND(SUPPRT_I*DAYS2,.01);

```

SAL2I=ROUND(OTHSAL_I*DAY2,.01);

KEEP RECNUM MTF MEPR2 DIR2F SUP2F SAL2F DIR2I SUP2I SAL2I DAY2
DIRECT_F DIRECT_I SUPPRT_F SUPPRT_I OTHSAL_F OTHSAL_I;
RUN;

* BE SURE TO INCLUDE DAY2 IN THE KEEP STATEMENT TO RUN THIS CHECK;
DATA CHECK2; SET TOG2;
IF DAY2=0 OR DAY2=. THEN DELETE;
IF DIRECT_F=. OR DIRECT_F=0 OR SUPPRT_F=. OR SUPPRT_F=0 OR OTHSAL_F=. OR OTHSAL_F=0 THEN DELETE;
IF DIR2F=. OR DIR2F=0 OR SUP2F=. OR SUP2F=0 OR SAL2F=. OR SAL2F=0 OR
DIR2I=. OR DIR2I=0 OR SUP2I=. OR SUP2I=0 OR SAL2I=. OR SAL2I=0;
PROC FREQ DATA=CHECK2; TABLES MTF*MEPR2 / LIST; TITLE'MEPR2 CHECK';

*****;

* COMPUTE DIR AND SUP, MEPR3;
PROC SORT DATA=UNITS; BY MTF MEPR3;
PROC SORT DATA=RECS; BY MTF MEPR3;

DATA TOG3; MERGE RECS(IN=DAT) UNITS; BY MTF MEPR3; IF DAT;

DIR3F=ROUND(DIRECT_F*DAY3,.01);
SUP3F=ROUND(SUPPRT_F*DAY3,.01);
SAL3F=ROUND(OTHSAL_F*DAY3,.01);

DIR3I=ROUND(DIRECT_I*DAY3,.01);
SUP3I=ROUND(SUPPRT_I*DAY3,.01);
SAL3I=ROUND(OTHSAL_I*DAY3,.01);

KEEP RECNUM MTF MEPR3 DIR3F SUP3F SAL3F DIR3I SUP3I SAL3I DAY3
DIRECT_F DIRECT_I SUPPRT_F SUPPRT_I OTHSAL_F OTHSAL_I;
RUN;

* BE SURE TO INCLUDE DAY3 IN THE KEEP STATEMENT TO RUN THIS CHECK;
DATA CHECK3; SET TOG3;
IF DAY3=0 OR DAY3=. THEN DELETE;
IF DIRECT_F=. OR DIRECT_F=0 OR SUPPRT_F=. OR SUPPRT_F=0 OR OTHSAL_F=. OR OTHSAL_F=0 THEN DELETE;
IF DIR3F=. OR DIR3F=0 OR SUP3F=. OR SUP3F=0 OR SAL3F=. OR SAL3F=0 OR
DIR3I=. OR DIR3I=0 OR SUP3I=. OR SUP3I=0 OR SAL3I=. OR SAL3I=0;
PROC FREQ DATA=CHECK3; TABLES MTF*MEPR3 / LIST; TITLE'MEPR3 CHECK';

*****;

* COMPUTE DIR AND SUP, MEPR4;
PROC SORT DATA=UNITS; BY MTF MEPR4;
PROC SORT DATA=RECS; BY MTF MEPR4;

DATA TOG4; MERGE RECS(IN=DAT) UNITS; BY MTF MEPR4; IF DAT;

DIR4F=ROUND(DIRECT_F*DAY4,.01);
SUP4F=ROUND(SUPPRT_F*DAY4,.01);
SAL4F=ROUND(OTHSAL_F*DAY4,.01);

DIR4I=ROUND(DIRECT_I*DAY4,.01);
SUP4I=ROUND(SUPPRT_I*DAY4,.01);
SAL4I=ROUND(OTHSAL_I*DAY4,.01);

KEEP RECNUM MTF MEPR4 DIR4F SUP4F SAL4F DIR4I SUP4I SAL4I DAY4
DIRECT_F DIRECT_I SUPPRT_F SUPPRT_I OTHSAL_F OTHSAL_I;
RUN;

* BE SURE TO INCLUDE DAY4 IN THE KEEP STATEMENT TO RUN THIS CHECK;
DATA CHECK4; SET TOG4;
IF DAY4=0 OR DAY4=. THEN DELETE;
IF DIRECT_F=. OR DIRECT_F=0 OR SUPPRT_F=. OR SUPPRT_F=0 OR OTHSAL_F=. OR OTHSAL_F=0 THEN DELETE;

```

IF DIR4F=. OR DIR4F=0 OR SUP4F=. OR SUP4F=0 OR SAL4F=. OR SAL4F=0 OR
  DIR4I=. OR DIR4I=0 OR SUP4I=. OR SUP4I=0 OR SAL4I=. OR SAL4I=0;
PROC FREQ DATA=CHECK4; TABLES MTF*MEPR4 / LIST; TITLE'MEPR4 CHECK';

*****;
PROC DATASETS; DELETE RECS UNITS;

PROC SORT DATA=TOG; BY RECNUM;
PROC SORT DATA=TOG1; BY RECNUM;
PROC SORT DATA=TOG2; BY RECNUM;
PROC SORT DATA=TOG3; BY RECNUM;
PROC SORT DATA=TOG4; BY RECNUM;

DATA TOG6; MERGE TOG TOG1 TOG2 TOG3 TOG4; BY RECNUM;

FCDIRECT=SUM(OF DIR1F DIR2F DIR3F DIR4F);
FCSUPPRT=SUM(OF SUP1F SUP2F SUP3F SUP4F);
FCOTHSAL=SUM(OF SAL1F SAL2F SAL3F SAL4F);

VCDIRECT=SUM(OF DIR1I DIR2I DIR3I DIR4I);
VCSUPPRT=SUM(OF SUP1I SUP2I SUP3I SUP4I);
VCOTHSAL=SUM(OF SAL1I SAL2I SAL3I SAL4I);

FULLCOST=SUM(OF FCOTHSAL FCCLNSAL FCDIRECT FCSUPPRT FCANCLAB FCANCRAD FCOTHANC FCSURG FCICU);
INCCOST=SUM(OF VCOTHSAL VCCLNSAL VCDIRECT VCSUPPRT VCANCLAB VCANCRAD VCOTHANC VCSURG VCICU);

PROC DATASETS LIBRARY=WORK; DELETE TOG TOG1 TOG2 TOG3 TOG4;

*****;

PROC SORT DATA=TOG6; BY RECNUM;
PROC SORT DATA=ONE; BY RECNUM;

DATA out.COSTED SIDR; MERGE ONE TOG6; BY RECNUM;

IF MTF='0052' AND CLNDISP='AFAB' THEN DO;
  FCOTHANC=0;
  VCOTHANC=0;
  FCANCLAB=0;
  VCANCLAB=0;
  FCANCRAD=0;
  VCANCRAD=0;
  FCCLNSAL=0;
  VCCLNSAL=0;
  FCDIRECT=0;
  VCDIRECT=0;
  FCICU=0;
  VCICU=0;
  FCOTHSAL=0;
  VCOTHSAL=0;
  FCSUPPRT=0;
  VCSUPPRT=0;
  FCSURG=0;
  VCSURG=0;
  FULLCOST=0;
  INCCOST=0;
END;

DROP DAYS1-DAYS4 MEPR MEPR1-MEPR4 PSP RECNUM TOTDAYS DISPMETR DRGNUM LAB RAD SURG
  DIR1F DIR2F DIR3F DIR4F DIR1I DIR2I DIR3I DIR4I
  SAL1F SAL2F SAL3F SAL4F SAL1I SAL2I SAL3I SAL4I
  SUP1F SUP2F SUP3F SUP4F SUP1I SUP2I SUP3I SUP4I
  SURG_F SURG_I ANCLAB_F ANCLAB_I ANCOTH_F ANCOTH_I
  ANCRAD_F ANCRAD_I CLNSAL_F CLNSAL_I ICU_F ICU_I;

RUN;

```

Appendix C. Pricing Program

libname in1 SIDR, FY09+ use the MSDRG
libname in2 PRICES, FY09+ use the MSDRG

DATA SIDR; SET IN1.SIDR; RUN;

DATA PRICES; SET IN2.FYXX;

PROC SORT DATA=SIDR; BY VARIABLE NAME FOR DRG/MSDRG; RUN;
PROC SORT DATA=PRICES; BY VARIABLE NAME FOR DRG/MSDRG; RUN;

DATA TOG; MERGE SIDR PRICES;
BY VARIABLE NAME FOR DRG/MSDRG;
RUN;