Maryland Medical Care Database (MCDB) Documentation

Prepared for
The Maryland Health Care Commission
(MHCC)

June 2008

Table of Contents

page

1. Introduction 1
2. Medical Services Component 2
   Overview 2
   Summary of Preprocessing and Application of MCDB Screens 3
   Data Quality and Completeness 4
   Contributors of Medical Services Data 4
   Distribution of Services by Plan Type 4
   Database Structure 5
   Encryptions of Patient Identifier 5
   Imputation of Secondary ("Other") Insurance Coverage 5
   Berenson-Eggers Type of Service Codes (BETOS) 5
   Assignment and Imputation of Relative Value Units (RVUs) 6
   "Odd Services" 9
   Handling of Provider Specialty Codes 9
   Standardization of Bill Type Coding and Reconciliation of Inconsistencies in Reporting of Financial Variables 11
   Date of Enrollment and Date of Disenrollment 12
   Financial Variable Edits (Outliers) 13
   Medical Services Analysis File 13
3. Medicare Component 15
   Overview 15
   Creation of 2006 Medicare File and 10 Percent Medicare Analysis File 15

Appendix A – Security and Data Confidentiality

Appendix B – SAS Programs Used in 2005 MCDB Preprocessing Activities

Appendix C – Payer-Specific Quality Review Statements for 2005

Appendix D – Payer-Specific Data Completeness Summary Reports

Appendix E – Medical Services File Contents and Codebook

Appendix F – Detailed Specifications and SAS Code Used to Impute RVUs

Appendix G – Practitioner Report Files Contents and SAS Code

1. Introduction

The 2006 Maryland Medical Care Database (MCDB) contains medical services and prescription drugs reported by Maryland insurance companies and health maintenance organizations (HMOs), called *payers* in this document. Payers with premium revenues of $1 million or more during calendar year 2006 are required to submit the following data files:

- Medical Services file *(Encounter Component*). It contains medical services claims adjudicated between January 1, 2006, and April 30, 2007.

- Prescription Drug File *(Pharmacy Component).* It contains prescription drugs claims adjudicated between January 1, 2006, and April 30, 2007.

- Provider Directories. The data (mainly provider specialties) from these directories are merged on to the medical services.

In addition, Medicare outpatient claims data for calendar year 2006 are acquired from Center for Medicare and Medicaid Services (CMS). This documentation provides information on the medical services and Medcare claims components. The prescription drug component of MCDB and its documentation have been submitted to MHCC sepatately.

Table 1 lists the 2006 deliverable files for the Medical Service and Medicare components.

Table 1: 2006 MCDB Data Files:
Medical Services and Medicare Components

MCDB Component	File (SAS Version 9.1)	File Definition	Number of Records	Number of Data Elements
Medical Services	PRV06_V4	Private Payer Medical Services, Jan. 2006 – Apr. 2007	85,931,243	101
	ANAL06_F_V2	Practitioner Report Analysis File, 2006 Medical Services	47,915,217	81
	ANAL07_JAN_APR	Practitioner Report Analysis File, Jan-Apr. 2007 Medical Services	14,299,740	74
Medicare Claims	MCR06_V2	Medicare Provder/Supllier, 2006	30,255,855	75

MCDB database development activities have been conducted in the structured sequences described in the *2006 MCDB Data Base Development Plan*. Furthermore, the MCDB database development activities are conducted under the security plan outlined in the *2006 MCDB Data Base Design Plan* and Appendix A of this report which address security considerations with regard to data collection, including procedures to protect data confidentiality, standards for receiving and sending data, data storage, disaster recovery, encryption of the patient identifier, and the physical security of facilities and equipment used in data collection. The security plan as designed and implemented conforms to Maryland's "Computerized Record System Security Requirements" and "HIPAA Information Technology Security Standards."

2. Medical Services Component

Overview

The Medical Services component of 2006 MCDB, including 85,931,243 services provided to 3,665,974 recipients, were submitted from June through November 2007 by 24 payers. Among these payers, 8 payers offered services under HMO plans and 13 payers offered services under the Comprehensive Standard Health Benefit Plan (CSHBP) designed for small businesses and self-employed individuals in Maryland (see Table 2.) Eighty-six percent of the medical services provided in 2006 were non-HMO and HMO Fee-for-Service, the remaining fourteen percent of the services were capitated specialty care.

Table 2: 2006 Payer-Submitted Data, Plans Offered, Counts of Services and Recipients

Payer	Payer Identification Number	Offers HMO Plans	Offers CSHBP[1] Plans	Number of Services	Number of Recipients
Aetna Life and Health Insurance Co.	P020		✔	6,735,681	264,215
Aetna U.S. Healthcare	P030	✔	✔	4,807,625	195,644
American Republic Insurance Co.	P070			24,070	906
CareFirst BlueChoice, Inc.	P130	✔	✔	18,774,307	1,013,694
CareFirst of Maryland, Inc.	P131		✔	23,547,560	841,273
CIGNA Healthcare Mid-Atlantic, Inc.	P160	✔		3,928,967	162,430
Assurant/Fortis Insurance Co.	P280			72,393	6,041
Golden Rule Insurance Co.	P320			146,313	11,552
Graphic Arts Benefit Corporation	P325		✔	123,837	4,666
Great-West Life and Annuity Ins. Co.	P330			462,464	21,986
Guardian Life Insurance Company of America	P350		✔	346,828	18,854
Unicare Life and Health Insurance Co.	P471			155,774	16,624
Kaiser Foundation Health Plan of Mid Atlantic	P480	✔	✔	5,087,116	255,628
MAMSI Life and Health Insurance Co.	P500		✔	2,127,195	76,753
Maryland Fidelity Insurance Co.	P510		✔	46,186	8,008
MD-Individual Practice Association, Inc.	P520	✔		3,857,438	129,618
Mega Life & Health Insurance Co.	P530		✔	185,563	17,378
Optimum Choice Inc.	P620	✔	✔	6,874,787	269,142
Coventry Healthcare of Delaware, Inc.	P680	✔	✔	1,300,757	64,713
State Farm Mutual Automobile Ins. Co.	P760			65,462	1,981
United Healthcare Insurance Co.	P820		✔	6,985,303	272,271
Trustmark Insurance Co.	P830			5,877	299
Union Labor Life Insurance Co.	P850			106,508	3874
United Healthcare of the Mid-Atlantic, Inc.	P870	✔		163,232	8,425
Total				**85,931,243**	**3,665,975**

[1] Comprehensive Standard Health Benefit Plan

Summary of Preprocessing and Application of MCDB Screens

The encounter data submitted by 24 payers contained 85,931,243 medical service records. A little over one percent of the submitted services did not pass the basic global screens for inclusion in the Medical Services database. *However, MHCC requested that SSS flagged but not removed these records from the database.* Table 3 shows number of services in the 2006 submission by type of screens.

Annual global screens applied to the 2006 submitted data are as follows:

- Recipient must be a Maryland resident.
- Routine dental services are excluded.
- Service must be outpatient.
- Payment must be under $7,000 if service is not classified as high-cost.
- Service must have been provided on or after January 1, 2006.

Table 3: 2006 Encounter MCDB Services Meeting Global Screens

Payer	Non-MD Patient Zip Code	Routine Dental Services	In-Patient Services	Financial Outlier	Out-of-Rage Date of Service	Services which Global Screens are Not Applicable	Total Services Reported
P020		4,956		636		6,730,089	6,735,681
P030		13,426		449		4,793,750	4,807,625
P070		16		57		23,997	24,070
P130	17,059	202		755		18,756,291	18,774,307
P131	21,867	387		617		23,524,689	23,547,560
P160	192	39	2,772	1,626		3,924,338	3,928,967
P280	103	286		1		72,003	72,393
P320				1		146,312	146,313
P325	130	2,962		60	16	120,669	123,837
P330	2,134			105		460,225	462,464
P350	569	82		13		346,164	346,828
P471	102	2	27	143		274	155,774
P480	107,463	30		67		4,979,556	5,087,116
P500		66	93,690	368		2,033,071	2,127,195
P510	8	20	22	9		46,127	46,186
P520	224	167	173,230	558		3,683,259	3,857,438
P530	34,154	70		399		150,940	185,563
P620	455	137	330,105	1,103		6,542,987	6,874,787
P680	725	13		69		1,299,950	1,300,757
P760	4,981			9		60,472	65,462
P820	2,879	870	4	4,117		6,977,433	6,985,303
P830				8		5,869	5,877
P850	1,471	12,369		75		92,593	106,508
P870	412	6		46		162,768	163,232
Total	**194,516**	**36,100**	**599,850**	**11,291**	**16**	**84,933,826**	**85,931,243**

The SAS programs used in 2006 MCDB preprocessing activities are presented in Appendix B.

Data Quality and Completeness

Payer-specific Quality Review Statements for 2006 are contained in Appendix C. Each report details the payer's adherence to COMAR submission requirements regarding the overall file format, quality of each of the required data fields, and linkage of the data to the provider directories. These reports for 2006 have been submitted to MHCC for transmittal to payers to assist them in preparing the 2007 submissions.

Also, the key statistics for number of services, payments, number of recipients and Relative Value Units (RVUs) has been created for each payer and compared to the same statistics from the 2005 Data Completeness Summary Reports (see Appendix D). These reports have been also submitted to MHCC for transmittal to payers.

Contributors of Medical Services Data

Aetna, CareFirst and MAMSI (now part of United Healthcare) remain the main contributors to the MCDB medical services data pool, together accounting for 86 percent of all the data submitted. Aetna (P020 and P030) contributed 13 percent of services; CareFirst DC (P130) and CareFirst Maryland (P131) contributed 50 percent; and MAMSI (P500), Optimum Choice (P620) and Maryland Individual Practice (P520), United Healthcare Corporation (P820), United Healthcare of the Mid-Atlantic, Inc. (P870) contributed 23 percent.

New York Life Insurance Co. (P600), Principal Life Insurance Co. (P690), ReliaStar Life Insurance Co. (P730), Transamerica Life Co. (P770), and American Medical Security Life Insurance Co. (P890) received full waiver from MHCC and did not submit medical services records for 2006.

Distribution of Services by Plan Type

The 2006 distribution of services and recipients by plan type are contained in Table 4. Services for recipients aged 0 through 64 are classified into a plan based primarily on the payer-submitted delivery system (DELIVTYP). Nearly half of the 2006 services (46 percent) are non-HMO FFS where as 22 percent are HMO FFS, and 14 percent are capitated services. A small fraction (0.3 percent) is Medicare Supplemental plan. The remainder of services (18 percent), classified as "Other FFS", include services provided to recipients aged 65 or older, and services without a valid coverage type code (COVTYPE).

Table 4: Summary of 2006 Encounter MCDB Services and Recipients by Payer Type

Payer Type	Number of Services	Percent of Services	Number of Recipients
Private HMO FFS	18,638.399	21.69	1,120,987
Private Non-HMO	39,407,105	45.86	1,713,389
Medicare HMO FFS	223,713	0.26	13,016
Other FFS	15,793,527	18.38	386,403
Capitated	11,868,499	13.81	956,317

Database Structure

The structure of the 2006 Medical Services file (prv05_v4.sas7bdat) differs from the structure of the 2005 Medical Services file. It contains 101 data fields as opposed to the 107 data fields in the 2005 Medical Service file. The changes in data elements in the 2006 MCDB file are described below:

- Since the month, day, and year parts of dates can be extracted from other data elements, the following data fields were dropped from the 2006 Medicare Service file: BEGDAY, BEGMO, BEGYR, BIRTHMO, BIRTHYR, CLAIMDD, CLAIMMO, CLAIMYR, ENDAY, ENDMO, ENDYR, and F_BADDT.

- MHCC requested that SSS keep only one data element for patient's age on December 31 of the reporting year; thus, AGE1206 and F_AGE were dropped from the 2006 Medicxal Service file.

- Since the services being flagged as meeting the "Global Screens" criterions were included in the 2006 Medical Service file, the following global screen flag variables were kept in the file: D_DAT_F, D_DNT_F, D_17K_F, D_INP_F and D_ZIP_F.

- Three data elements were added: F_AGE_NR, F_ANESTH and ID_CHANGE.

Appendix E contains the contents and codebook of the 2006 Medical Services file.

Encryptions of Patient Identifier

Even though payers are supposed to be providing encrypted patient identifiers as part of the initial data submission, SSS performs additional encryption to preserve confidentiality of such a sensitive field. For this year, SSS continues to use the same methodology for encryption that was used in 2005. The methodology, which uses a form of cryptographic substitution, allows encryption of IDs while individual payer files are finalized. Both methodologies were designed to ensure congruence between the identifiers for individual recipients of medical services and prescription drugs.

Imputation of Secondary ("Other") Insurance Coverage

The algorithm for imputation of the Patient Covered by Other Insurance flag did not change for 2006. For services where payers do not report whether coverage is primary or secondary, secondary coverage is imputed if total payments are less than 35 percent of the allowed amount. For 2006, data for this field was imputed for approximately two percent of the all records in the Medicalcare Database. All the services with secondary insurance coverage are excluded from the Practitioner Report analysis.

Berenson-Eggers Type of Service Codes (BETOS)

Using the data provided by payers and information published by CMS, SSS assigned to each medical service record detailed and aggregate BETOS codes (BETOSDT and BETOSAG). In addition, each record contains enhanced BETOS codes (NEWBETDT and NEWBETAG) designed for use in Practitioner Report and SHEA analysis. For 2006, the source of the assigned

detailed and aggregate BETOS codes was the 2007 version of BETOS public use file available at http://www.cms.hhs.gov/providers/pufdownload/betpuf07.zip.

At MHCC's request, the separate Childhood Immunization aggregated BETOS (BETOSAG = 8 and NEWBETAG = 8) category was dropped from the classification system starting in 2003.

Assignment and Imputation of Relative Value Units (RVUs)

Assignment of Total and Component RVUs

Starting with the 2001 MCDB, RVU components are assigned or imputed in addition to Total RVUs. The following components are included in the MCDB (Medical Services and Medicare files): (1) Malpractice Component, (2) Physician Component and (3) Work Component. The National Physician Fee Schedule Relative Value File (Calendar Year 2005) was the source of most Year 2005 RVUs (the CMS lab fee schedule was used to set RVUs for clinical lab tests). Starting with the 2002 MCDB, the practice expense RVUs are based entirely on the resource-based system as required by Section 121 of the Social Security Act Amendments of 1994; 2002 is the last year of a 4-year transition from the existing charge-based system to the new resource-based system. The four-year transition started in January 1, 1999, and is a requirement of the Balanced Budget Act of 1997. Table 4 shows the proportions of each base system for the transition years.

Table 5: Transition Years in RVU Assignments

Year	Charge-Based %	Resource-Based %
1999	75	25
2000	50	50
2001	25	75
2002	0	100

As in previous years, services were classified as facility- or non-facility-based before the assignment of RVUs. A listing of Place of Service codes that would result in a service being classified as facility-based is contained in Table 5. All other services were classified as non-facility based.

In addition, an effort was made to identify services that were (1) technical component (AMA modifier "TC"), (2) ambulatory surgery (AMA modifier "SG"), or (3) professional component (AMA modifier "26"), but not identified as such by the payer-provided AMA modifiers (MOD1 and MOD2). For radiology bills in particular, several indicators including Place of Service were used to categorize bills as either professional component (modifier "26") or total component (modifier blank) bills.

Exceptions to the general RVU assignment methodology were as follows:

- For clinical lab-related claims, the RVU was assigned as the Medicare lab fee divided by the Year 2006 Medicare Conversion Factor (37.8975).

- For CPT therapy codes 97001-97546 only non-facility RVUs were assigned, according to Medicare payment rules for these services.

- For services with high payment per RVU and with reported multiple UNITS on the record, payment was divided by UNITS before RVU assignment took place.

Table 6: Place of Service Codes Classified as Facility-Based for RVU Assignment.

Place of Service	CMS Place of Service Code
Inpatient Hospital	21
Outpatient Hospital	22
Hospital Portion of Emergency Room	23
Ambulatory Surgical Center	24
Skilled Nursing Facility	31
Hospice	34
Inpatient Psychiatric Facility	51
Partial Hospitalization Psychiatric Facility	52

Imputation of Total RVUs

Total RVUs were imputed for services with valid CPT/HCPCS procedure codes that are not in the 2006 Fee Schedule or Medicare Lab Fee schedule. The imputation process assigned RVUs to about 36 percent of all services. RVUs could not be assigned or imputed for only six percent of all services.

In broad outline, the algorithm for imputing RVUs was based on private insurer's average payments. For procedure codes without RVUs, average payment per service was deflated, by the average private conversion factor calculated for the codes that did have a valid RVU, to give an appropriately-scaled RVU value. This deflation (or inverse conversion factor) was done separately for each detailed BETOS category. For example, if all codes with RVUs in a BETOS category had an average conversion factor of $40 per RVU, then any code in that category without RVUs had an imputed RVU calculated as the mean payment per service for that code divided by $40. Using this approach, the estimated private conversion factor within each BETOS category should be the same whether the analyst includes or excludes the codes with imputed RVUs.

Imputation conversion factors were calculated from payment information at the detailed BETOS procedure classification level, using the pool of all available service records from the MCDB with valid CPT/HCPCS codes. The summary algorithm for calculation of the conversion factors is as follows:

- Sum RVUs and payment by detailed BETOS procedure category.
- Calculate the Conversion Factor (mean payment per RVU).

For services needing RVU imputation, the algorithm is following:

- Calculate mean payment for the specific procedure/AMA modifier group.
- Divide the mean payment for the group by the calculated conversion factor.

Before calculating any of the average payments or conversion factors, certain bills were removed from the subset of services used for RVU imputation. These are bills that were likely to reflect payments for things other than the procedure code on the bill. Examples of such bills were identifiable minor services (such as assistance at surgery); anesthesia billed under the surgical CPT, facility fees, and bills with unusually low or high payment amounts. Note that once RVU Conversion factors were calculated, the full RVU value was attached to each of these partial payment/minor service bills.

Detailed specifications for RVU imputation and the accompanying SAS code are contained in Appendix F. The services removed from RVU imputation calculations are shown in Table 6.

Table 6: Services Removed From RVU Imputation Calculations.

Procedure-Based Deletions	Financial Information-Based Deletions
• Dental/Orthodontic • Anesthesia • Facility Bills • Minor Surgery/Split Surgery • Assistance at Surgery • Bilateral/Multiple Procedures • Durable Medical Equipment Procedures • Drug Procedures	• Payment <= 1 • Billed Amount, Allowed Amount, Reimbursed Amount and Patient Liability = 0 • Allowed Amount = 0
Outlier Deletions	**Bill-Type Based Deletions**
• Payment < 1/3 the payment mean calculated for the procedure • Payment > 6 times the payment mean calculated for the procedure • RVU < 0.25*RVU*2006 Medicare Conversion Factor • RVU > 8* RVU*2006 Medicare Conversion Factor	• All credit adjustment service records • For services with multiple final bills, delete all but the service with the highest payment (minor claims not otherwise identified).

Imputation of Component RVUs

The methodology for RVU imputation only computes Total RVUs; it does not compute Total RVU components: Work, Practice Expense, and Malpractice Expense RVUs. The same is true for lab procedures. Total RVUs were estimated using the 2006 Clinical Diagnostic Laboratory Fee Schedule. An additional step was added to impute all RVU components under the two cases described above. Using all services with all RVU components assigned, Total RVU and its components were aggregated at the BETOS level. Then each component was divided by Total RVU to get, on average, for each BETOS, what fraction of the total each RVU component represents. For example, for BETOS XXX: Work is 42 percent, Practice Expense is 50 percent, and Malpractice Expense is 8 percent. These BETOS-level average percentages were then applied to the total RVUs on the codes with imputed RVU. For the above example, if a procedure code had an imputed Total RVU and belonged to that BETOS category XXX, Work RVU was computed as 0.42 times the imputed Total RVU. If a BETOS category was not assigned, average percentages of the Total RVU at the facility/non-facility level were used. For lab tests based on the clinical diagnostic lab fee schedule, it was assumed that there was no work or malpractice expense. In those cases, Practice Expense RVU was then equal to Total RVU.

“Odd Services”

Some medical services that pass the global screens for inclusion in the Medical Services component of the MCDB and the analytical files (described below) do not meet the quality control standards developed for inclusion in the SHEA and Practitioner reports analytical work; however, they need to be retained on the databases. These “Odd Services” are flagged for easy removal during creation of specific tables and reports for the SHEA and the Practitioner Report as they are ineligible for inclusion in the analytical activities.

The frequency of “Odd Services” in the 2006 Encounter MCDB is presented in Table 7. The most frequent reason for flagging a service as “odd” is that it could not be classified into one of the five analytical Plan types (16 percent of services fall into this category). None of the 2006 services fell into either the “Member of Debit/Credit Pair” or the “Odd/Unusual Procedure Codes” categories. The need for flagging of debit/credit pairs was obviated by the requirement, being implemented in 2003, that payers submit only final adjudicated bills.

Table 7. Definition of “Odd Services” Codes (F_ODDSVC) and Occurrence in 2005 MCDB

F_ODDSVC	Flag Definition	Number of Services	Percent of Services
01	Member of Debit/Credit Pair	0	0.00
02	Odd/Unusual Procedure Codes (e.g. RX, Dental)	0	0.00
03	Payment <= 1, Allowed Amount <= 0	1,871,064	2.18
04	Identified Anesthesia Service Outside Of Range	238,842	0.28
05	Identified Facility Bill	1,728,431	2.01
06	Minor Service Associated with Major Service	312,411	0.36
07	Billing Adjustment Record	0	0.00
08	Ambiguous Payer Type	13,598,359	15.82
09	Not First Record for Unique Service	1,610,827	1.87
10	Actuaries Trim, Payment, Low End Outlier	265,373	0.31
11	Actuaries Trim, Payment, High End Outlier	29,987	0.03
12	Actuaries Trim, Payment/RVUs, Low End Outlier	102,753	0.12
13	Actuaries Trim, Payment/RVUs, High End Outlier	180,642	0.21
14	Possible Secondary Payer on Claim	165,454	1.19
15	Assigned/Imputed RVUs Missing or 0	3,600,357	4.19
99	More than One Condition Code Set	6,668,928	7.76
Blank	No Condition Codes Set	55,557,815	64.65

Handling of Provider Specialty Codes

Overview

The major source for assignment of provider specialties for the 2006 MCDB Medical Services was the 2006 Provider Directory file. Provider specialties that could not be assigned based on

information in the 2006 Provider Roster were assigned or imputed using the procedure code in the encounter record.

Specialty Code Assignment and Imputation

For each provider there is an encounter file containing one observation for each encounter such as office visit, claim, etc. For this task the items of interest on the encounter file is the Provider ID, Federal Tax ID, and procedure code (CPT code). A second file for each provider is a Provider Directory which should contains one record for each provider and contain Provider ID, Federal Tax ID, and up to three specialty codes.

For each of these files a relatively small number of records with invalid data are acceptable, such as invalid Provider ID, Federal Tax ID, records without specialty, and records in encounter file with Provider Number not provided on the Provider Directory. Most of the processing described below is to handle these exceptions – the normal process of most of the work to resolve the last cases.

- **Pre-processing of Provider Directory:**

  We converted each of the provider directories that had up to three specialty codes to records with a single specialty code. For example if three specialty codes were reported we created three records (one for each of the reported specialty code) with all of the records having all other variable the same. We limited the output to this pre-processing to Provider ID, Federal Tax ID, and specialty code.

- **Pass One:**

  In this step we matched the encounter file to the Provider Directory twice, one by Provider ID and Federal Tax ID, and once by Provider ID only. In both matches we only kept the output of the match if the encounter matched to one non-miscellaneous specialty code. Non-miscellaneous specialty codes are those except for the following specialty codes: 034, 037, 038, 062, 085 and 093. In the case the match returned a miscellaneous and non-miscellaneous specialty codes, we kept the non-miscellaneous specialty code. We then compared the specialty code from the Provider ID and Federal Tax ID match and the Provider ID only match. If only one returned a specialty code, we kept that one. If both matches resulted in a specialty code we determined which to keep as follows (in this order):
  a. If one code is non-miscellaneous and one is miscellaneous we keep the non-miscellaneous.
  b. If both are non-miscellaneous we keep the one from the Provider ID and Federal Tax ID match.

  Approximately 86.6 percent of the encounter records processed in Pass One was assigned a specialty code. The encounter records that are not assigned a specialty code (about 13.4 percent) are processed in Pass Two and Pass Three.

- **Pass Two:**

  In this step we determine the specialty code when the results of Pass One results in more than one non-miscellaneous specialty code. For this step we use the procedure code provided on the encounter file. This is a semi-hot-deck process using the 86.6 percent of the encounters assigned in Pass One. From the Pass One results we tabulate the frequency of specialty by

procedure code (all providers combined together). Using this frequency we determine which of the specialty code in the case of multiple specialty codes match by selecting which specialty is assigned more frequently for the procedure code reported on the encounter file. For Example:

Encounter A with:
Provider: 001 Federal Tax ID: 001 Procedure: 3002
Encounter B with:
Provider: 001 Federal Tax ID: 001 Procedure: 4007

From the Provider Directory:
Provider: 001 Federal Tax ID: 001 Specialty: 020
Provider: 001 Federal Tax ID: 001 Specialty: 060

From the Pass One result frequency:

Procedure: 3002	Specialty: 020	Frequency:	1,000
Procedure: 3002	Specialty: 060	Frequency:	500
Procedure: 4007	Specialty: 020	Frequency:	100
Procedure: 4007	Specialty: 060	Frequency:	500

For Encounter A we select specialty 020 because this is more common for procedure 3002 than specialty 060. The reverse is true for Encounter B and specialty 060 is assigned.

Pass Two assigned specialty codes to approximately 5.8 percent of all the encounter records (or about 43.5 percent of the encounter records without specialty assignment in Pass one.)

- **Pass Three:**

For the encounter records not being assigned a specialty code in Pass One or Two (about 7.6%) the specialty code is assigned using the procedure code in the encounter record. This is a hot-deck process and the specialty codes are assigned in accordance with procedure codes in the same proportions of those assigned in Pass One. For example in Pass One 40% of the encounters with procedure 3089 had a specialty of 040, then 40% of the records in Pass Three with procedure 3089 would be assigned a specialty of 040. For a very small number of encounters the specialty code in Pass three encounters did not have any others with the same specialty assigned in Pass One. In this case these are assigned proportionally without regard to procedure code.

Pass three assigned specialty codes to approximately 7.6 percent of all the encounter records (or about 56.5 percent of the encounter records without specialty assignment in Pass Two.)

Standardization of Bill Type Coding and Reconciliation of Inconsistencies in Reporting of Financial Variables

Since completion of the transition to the requirement for reporting of final adjudicated bills in 2003, payers were no longer allowed the option of submitting debit or credit portions of bills for

services that had not been reconciled. The definitions of acceptable bills provided to payers in the *2005 COMAR Data Submission Manual* were as follows:

- Bill Type 1 (Final Bill): Total amount of all debits and credits paid for a claim by the insurance company to the provider.

- Bill Type 8 (Capitated Services): Set of predefined services provided by the provider to the plan's enrollees under contract with an insurance company or managed care plan in exchange for a fixed ad guaranteed monthly payment for each enrollee assigned to the provider.

Date of Enrollment and Date of Disenrollment

For 2005, payers (with the exception of Union Labor Life Insurance Co.) were required to provide two additional data fields: Date of Enrollment and Date of Disenrollment. In addition to recoding these dates based on instructions from payers, SSS also developed a software program to validate these two dates. The software categorized the dates into groups as follows:

1 = Enrollment Date = Missing
2 = Enrollment Date = Disenrollment Date
3 = Service Begin Date BEFORE Enrollment Date
4 = Service Begin Date AFTER Disenrollment Date
5 = Enrollment Date <= Service Begin and Disenrollment Date > 12-31-2005
6 = Enrollment Date <= Service Begin and Disenrollment Date < 01-01-2005
7 = Enrollment Date <= Service Begin and Disenrollment Date = Missing
8 = Enrollment Date <= Service Begin <= Disenrollment Date

Records with flags 5, 7 and 8 were considered good data. At the service level, SSS found that 95% of over 76 million services reviewed had dates. However, when looking at these dates at recipient level, SSS found data discrepancies which indicated that recoding of these dates was needed. SSS reviewed data for more than three million recipients, of which 89 percent had single set of dates and 11 percent had multiple sets of dates. Of the recipients with multiple sets of dates, 88 percent had two sets of dates and 12 percent had three or more set of dates. SSS performed additional analysis of the group of recipients with two sets of dates and flagged the data using the following categories:

1 = Same enrollment date but different disenrollment dates
2 = Continuous enrollment period
3 = Continuous enrollment periods with 0 to 5 days gap.
4 = Missing disenrollment date in one set of dates
5 = Dates that need to be reviewed.

After reviewing sample records and discussing with MHCC, for each recipient with more than one sets of dates, SSS data management staff recoded the enrollment and disenrollment dates to come up with the longest period of enrollment. Then, the enrollment and disenrollment dates were checked and adjusted if the enrollment period was shorter than the period of services provided. For example: if the enrollment period

was from 02-02-2005 to 09-30-2005 but the period of the services provided is from 01-15-2005 (Service Sate Date) to 09-15-2005 (Service End Date), SSS recoded the enrollment date to 01-15-2005 and left the disenroll date to 09-30-2005. This adjusted set of dates is used for analytical work.

Financial Variable Edits (Outliers)

In the past years, SSS identified outliers in the financial variables by applying a dollar amount ceiling and excluding services where payment exceeded that ceiling amount. We also used identified odd services based high/low payment or RVU. The following odd services flag were designed to flag outliers for HMO FFS and Non-HMO services.

- Actuaries Trim on Payment, Low (COND10): Payment less than 1/3 of the mean payment.
- Actuaries Trim on Payment, High (COND11): Payment greater than six times of the mean payment.
- Actuaries Trim on Imputed RVU, Low (COND12): Imputed RVU> 0 and payment is less than 1/4 of (Medicare CF x RVU_I).
- Actuaries Trim on Imputed RVU, High (COND13): Imputed RVU> 0 and payment is greater than 8 times of (Medicare CF x RVU_I).

In 2006, in addition to flagging outliers as odd services, SSS also adopted the methodology of identifying financial outliers being used by the Medicare Payment Advisory Commission (MedPAC). All services in the report analysis file with more than 3 times the mean payment per unit or less than 1/3 of the mean payment per unit were flagged as outliers. The mean payment per unit was calculated for each procedure by provider, plan (HMO or Non-HMO), CPT and modifier. All UHC-MAMSI payers (P500, P520, and P620) and the two Carefirst payers (P130 and P131) were grouped together for the purpose of computing outliers. Appendix D contains statistics of data *before* and *after* outlier payment were removed.

Medical Services Analysis File

Overview

Starting in 2005, in accordance with the MHCC's instruction, a Medical Services analytical file was created from the full Medical Services database (MCDB). This file is the source file for the tables published in the Practitioner Report and is known as the *Practitioner Report Analysis File*.

MCDB Global Service Eligibility Screens

In previous years, records of services that met the MCDB global service eligibility screens were excluded from the full MCDB file. In 2006, MHCC requested that these service records be included in the full MCDB file with the global service eligibility screen flags turned but excluded from Medical Services analytical file. The following are the global service eligibility screen:

- Non-Maryland Patient Zip Code
- Routine Dental Services
- Inpatient Services

- Potential Financial Outlier
- Out-of-Range Service Date

Analytical File Global Screens

Only services meeting predetermined global analysis screens are passed from the full MCDB file to the *Practitioner Report Analysis* file. These screens are initially assigned to all services in the MCDB file. The global analysis screens are listed as follows:

Common screens for non-capitated and capitated services:

- Recipient must be between 0 and 64 years of age on December 31, 2005
- Service must be provided in 2005.
- Patient ID must be non-missing.

Global Analysis Screens for non-capitated services:

- Service must be covered by provider.
- Provider must be primary payer.
- Service may not be for durable medical equipment.
- Procedure code must be HCPCS, CPT, or homegrown.
- Procedure must not be radiology or ambulatory surgery.

Global Analysis Screens for capitated Services:

- Service may not be for durable medical equipment.
- Procedure code must be HCPCS, CPT or homegrown.
- Procedure must not be radiology or ambulatory surgery.

Also, additional screens may be applied independently for the numerous SHEA calculations and Practitioner Report tables prepared annually from the analysis datasets. For 2006, the screen that wea constant across all analytical calculations based on the 2006 analysis datasets were removal of services flagged as “odd” (see discussion above). Screens used for individual tables and/or calculations can be found in the relevant published report.

Use of the Analytical Files

The *Practitioner Report Analysis* file will be used to create all tables and figures for the 2005 Practitioner Report. The programs used to create the final 2006 Practitioner Report tables and figures will be provided separately upon the completion of the analysis.

3. Medicare Component

Overview

The Medicare Component is an extract of the annual Medicare Physician/Supplier claims file obtained from the CMS.

Preprocessing for CMS data submissions is more straightforward than preprocessing for the Medical Services data, because CMS conducts extensive error checking and quality control before submittal to MHCC.

The following preprocessing steps were applied to the 2006 CMS Physician/Supplier file:

- Media/documentation examination, entering submission information into the annual submission database, and transferring data (SAS format) from DVD to the project server.

- Extraction of selected variables from the files

- Application of selected screening criteria (see below)

After preprocessing, the following screens were applied to create the final raw input file used for creation of the 2005 Medicare Component and associated 10 Percent file:

- Line item processing indicator code must be one of the following values:

  A = allowed
  R = reprocessed adjustments based on subsequent reprocessing of a claim
  S = secondary payer

- Delete line item record if line item allowed charges are zero or less.
- Recipient must be a Maryland resident.
- Service provided between January 1, 2006 and December 31, 2006
- Exclude routine dental services.

Preprocessing yielded 30,255,855 Medicare service records.

Creation of 2006 Medicare File and 10 Percent Medicare Analysis File

The 2006 MCDB Medicare File (mcr06_v2.sas7bdat) contains 75 variables. In most cases, variables parallel those in the Encounter MCDB in both naming conventions and content rules. This includes variables developed specifically for the Practitioner Report analysis (see Table 10).

Table 10: Practitioner Report Analysis Variables in 2006 MCDB Medicare Database

Variable Name	Description
CNTY_P2	County to use for Practitioner Report Analysis
F_DEMOG	Demographic Information Record-Donor for Analysis File? (1/0)
F_ODDSVC	Odd Service Flag
NEWBETAG	Aggregated BETOS, Reclassified
NEWBETDT	Detailed BETOS, Reclassified
NEWMOD	Imputed Modifier/Last Imputation
PAY_CALC	Method for calculating theoretical Medicare payment
PAY_MCR	Theoretical Medicare payment
PLAN	Plan Type

Details for the assignment of BETOS categories and RVUs can be found in the relevant sections in the description of the development of the Medical Services database. Since the CMS Physician/Supplier files contain provider specialty codes, there is no further imputation of specialty for the Medicare file and the CMS-supplied provider specialty code is contained in the variable SPEC1. Detailed descriptions of all Medicare file variables can be found in the *2006 MCDB Data Base Users Manual*.

The contents, codebook, and SAS programs used in Medicare file development are contained in Appendix H.

Appendix A

Security and Data Confidentiality

Security and Data Confidentiality

This section of the 2006 MCDB Database Documentation addresses security considerations with regard to data collection, including procedures to protect data confidentiality, standards for receiving and sending data, data storage, disaster recovery, encryption of the patient identifier, and the physical security of facilities and equipment used in data collection. SSS's approach conforms to Maryland's "Computerized Record System Security Requirements" and "HIPAA Information Technology Security Standards."

SSS shares the Commission's concern for quality and security of the data collection procedures. The use of medical data in a research environment must be safeguarded using both automated and managerial controls. Health care providers, third-party payers, and the Government (including those that contract with the Government) have a clear responsibility to protect the confidentiality of the patient medical information. Disclosure of confidential information is damaging regardless of whether it is read from paper medical record or viewed on a computer.

Balanced with the patient's right to privacy is the Government's need and responsibility to have some level of access to medical data to set rates, track changes in provider charges, and study the change of the health care system over time. This balance between legitimate research and patient privacy must be respected, and reasonable controls must be implemented by institutions that acknowledge and respect these two different needs.

When new data files arrive at SSS, we routinely strip off identifying information (e.g., Social Security Number, name, address, and phone number). For example, when we upload CMS Medicare data, we immediately extract only the fields used in this project that are similar to those in the MCDB. When information is needed from these control files for a particular analysis, programmers are required to link to these files only temporarily and not to save these fields on the resulting analytic files. SSS also understands that confidentiality and disclosure avoidance extend beyond simply encrypting identifiers or dropping identifying information to other, more subtle areas, such as cell sizes in output reports. This is of special concern for rare diseases or areas of the State with small population base. For example, if the insurance premiums per capita are displayed by county, and it is known that only one insurance company provides insurance in that county, it is possible for readers to identify the insurance company.

Standards for Receiving and Sending Data

Data are received on a variety of different media types. Media are delivered directly to the project director or database managers by an SSS courier. Media are immediately cataloged and placed in the Maryland Project Data Center (MPDC), which is secured by cipher locks and located in a secure, locked office suite..

Data Storage

By using the Windows XP Active Directory, access to sensitive data is controlled through permissions based on user authority. User IDs are created for each member of the MPDC staff, and IDs are not shared among staff. Password polices are set in an Active Directory-based group policy, which includes complex password restrictions, size limits, and reuse prohibitions. Staff are compelled to change passwords every 45 days by the group policy.

Appendix B

SAS Programs Used in 2005 MCDB Preprocessing Activities

```
proc datasets kill;
run;

%jobstart;

*================================================================================*
Program:    I:\MCdb\datacol\p020\prog\P020_enc_read.sas
Project:    Maryland, Encounter Preprocessing
Purpose:    To read the fixed format encounter data
--------------------------------------------------------------------------------
input:      P020EncA.txt
output:     P020_Enc_raw.sas7bdat
--------------------------------------------------------------------------------
global macros to adjust:
For each payer
   1. paynum
   2. subnum
   3. file
--------------------------------------------------------------------------------
Programmer: Sireesha Thati / Sanee Maphungphong
Date      : 05/10/2006
Note    : We have 2 new variables this year----Date of Enrollment,
                                Date of Disenrollment
--------------------------------------------------------------------------------
Programmer: Sanee Maphungphong
Date        : 08/09/2007
Note      : No documentation provided.
*================================================================================*;
options center obs = max pageno = 1 pagesize = 86 linesize = 123 error = 2;
options nomprint;

%let payernum = P020;
%let subnum   = 001;
%let file     = a;
%let rptYr    = 2006;
%let sbmssn   = Submission &subnum;
%let drive    = I;
%let infile1  = &drive:\Mcdb\datacol\&payernum.\rawdata\&payernum.EncA.txt;

libname library "I:\mcdb\datacol\formats\permanent\";
libname in    "&drive:\Mcdb\datacol\&payernum.\rawdata";
libname out   "J:\Mcdb_raw";

filename in1 "&infile1";

%let hmolst  = 'P030','P130','P160','P480','P520','P620',,'P680','P870';
%let mhmolst = 'P480';

title1 "Maryland Medical Care Database (MCDB)";
title2 "Year &rptYr Encounter Data";
title3 "&Payernum. (&Sbmssn) - Enc_Raw";
title4 " ";

footnote2 "program: &drive:\Mcdb\datacol\&payernum.\programs\&payernum._Enc_read.sas";

options nosource;
*=============================================*

              INCLUDE PROGRAMS

*=============================================*;
%include "&drive:\MCdb\DataCol\Include_Program\MZDCONV.SAS";
options source;

*=================================================*

                 READ RAW DATA

*=================================================*
;
data out.&payernum._Enc_raw;
```

```
P020_enc_read.sas - To read 1 fixed format data file.
P130_enc_read.sas - To read 2 fixed format data files.
P070_enc_read.sas - To read 1 variable format data file- but enroll date and disenroll date
                    are in fixed format.
P350_enc_read.sas - To read 1 variable format data file - even the enroll and disenroll date
                    are also variable.
P330_enc_read.sas - To read 2 fixed format data files - one is in one long stream format.
```

```
proc datasets kill;
run;

%jobstart;
*================================================================================*
Program:    I:\MCdb\DataCol\pxxx\prog\Pxxx_Enc_read.sas
Project:    Maryland, Encounter Preprocessing
Purpose:    Read in the variable format Encounter data
--------------------------------------------------------------------------------
input:      PxxxEncA.txt
output:     Pxxx_Enc_raw.sas7bdat
--------------------------------------------------------------------------------
global macros to adjust:
For each payer
   1. paynum
   2. subnum
   3. file
--------------------------------------------------------------------------------
Programmer:  Laurie Hamliton / Po-Lun Chou
date:        04/07/2005

Modified by: Po-Lun Chou
Date:        July 12, 2005

Modified by: Po-Lun Chou
Date:        September 27, 2005
Purpose:     Re-assigned ENCNT_ID values.
Note:        Payer P070 has variable format Encounter data file and we created
             ENCNT_ID once we read in the fixed portion of each records in the
             raw data files, so it caused the problem that we have same ENCNT_ID
             multiple times for all of the recrods from one person (i.e., the
             records which have same PIDBDG).  We fixed this mistake here to
             avoid getting incorrect SPEC later in Enc3 program

Modified by: Sireesha Thati
Date:        05/10/2006
Purpose:     To read the variable format data for 2005 Encounter data.

--------------------------------------------------------------------------------
Programmer: Sanee Maphungphong
Date       : 07/17/2007
Note      : No changes to file layout for 2006 data. Just need to update the file
            names as needed.

*================================================================================*;

options center obs = max pageno = 1 pagesize = 86 linesize = 123 error = 2;
options nomprint;

%let payernum = P070;
%let subnum   = 001;
%let file     = a;
%let rptYr    = 2006;
%let sbmssn   = Submission &subnum;
%let drive    = I;
%let infile1  = &drive:\Mcdb\datacol\&payernum.\rawdata\&payernum.EncA.txt;

libname library "&drive:\mcdb\datacol\formats\permanent";
libname in      "&drive:\mcdb\datacol\&payernum\rawdata";
libname out     "J:\mcdb_raw";

filename in1 "&infile1";

title1 "Maryland Medical Care Database (MCDB)";
title2 "Year &rptYr Encounter Data";
title3 "&Payernum. (&Sbmssn) - Enc_Raw";
title4 " ";

footnote2 "program: &drive:\mcdb\datacol\&payernum.\programs\&payernum._Enc_read.sas";

options nosource;
```

```
proc datasets kill;
run;

%jobstart;

*==================================================================================*
Program:    I:\MCdb\datacol\p130\prog\P130_enc_read.sas
Project:    Maryland, Encounter Preprocessing
Purpose:    To read the encounter data with 2 raw data files
----------------------------------------------------------------------------------
input:      P130EncA.txt, P130EncB.txt
output:     P130_Enc_raw.sas7bdat
----------------------------------------------------------------------------------
global macros to adjust:
For each payer
   1. paynum
   2. subnum
   3. file
----------------------------------------------------------------------------------
Programmer: Sireesha Thati / Sanee Maphungphong
Date        : 05/10/2006
Note      : We have 2 new variables this year----Date of Enrollment,
            Date of Disenrollment
----------------------------------------------------------------------------------
Programmer: Sanee Maphungphong
Date        : 07/19/2007
Note      : Made changes as follows:
            Set rptYr to 2006
            Set hmolst to 'P030','P130','P160','P480','P520','P620',,'P680','P870'
            Set label format in include statement to "labels_06_raw.sas"

*==================================================================================*;
options center obs = max pageno = 1 pagesize = 96 linesize = 123 error = 2;
options nomprint;

%let payernum = P130;
%let subnum   = 001;
*%let file     = a;
%let rptYr    = 2006;
%let sbmssn   = Submission &subnum;
%let drive    = I;
%let infile1  = &drive:\Mcdb\datacol\&payernum.\rawdata\&payernum.EncA.txt;
%let infile2  = &drive:\Mcdb\datacol\&payernum.\rawdata\&payernum.EncB.txt;

libname library "I:\mcdb\datacol\formats\permanent\";
libname in      "&drive:\Mcdb\datacol\&payernum.\rawdata";
libname out     "J:\Mcdb_raw";

%let hmolst  = 'P030','P130','P160','P480','P520','P620',,'P680','P870';
%let mhmolst = 'P480';


title1 "Maryland Medical Care Database (MCDB)";
title2 "Year &rptYr Encounter Data";
title3 "&Payernum. (&Sbmssn) - Enc_Raw";
title4 " ";

footnote2 "program: &drive:\Mcdb\datacol\&payernum.\programs\&payernum._Enc_read.sas";

*==============================================*

              INCLUDE PROGRAMS

*==============================================*;
options nosource2;
%include "&drive:\MCdb\DataCol\Include_Program\MZDCONV.SAS";
options source2;

%macro finalenc (file_id =, Enc =, in1 =, num =);

%let file = &file_id;
```

```
proc datasets kill;
run;

%jobstart;

*================================================================================*
Program:    I:\MCdb\datacol\p330\prog\P330_enc_read.sas
Project:    Maryland, Encounter Preprocessing
Purpose:    To read the encounter data with 2 raw data files
--------------------------------------------------------------------------------
input:      P330EncA.txt, P330EncB.txt
output:     P330_Enc_raw.sas7bdat
--------------------------------------------------------------------------------
NOTE ON SOURCE FILES: PAYER SUBMITS TWO FILES: ONE FROM DENVER AND ONE FROM
                      ST. LOUIS
                      - FILE FROM DENVER IS SUBMITTED VIA CARTRIDGE
                      - FILE FROM ST. LOUIS IS SUBMITTED VIA CD-ROM

BEFORE RUNNING THIS PROGRAM, MAKE CERTAIN THAT THE FILES ARE RENAME FROM THE ORIGINAL
NAMES SO THAT FILE FROM DENVER IS "P330ENCA.TXT" AND FILE FROM ST. LOUIS IS "P330ENCB.TXT"
--------------------------------------------------------------------------------
global macros to adjust:
For each payer
   1. paynum
   2. subnum
   3. file
--------------------------------------------------------------------------------
Programmer: Sireesha Thati / Sanee Maphungphong
Date      : 05/10/2006
Note      : We have 2 new variables this year----Date of Enrollment,
            Date of Disenrollment

Programmer: Kasey chen
Date      : 07/23/2007
Note    : No change for year 2007

*================================================================================*
* 09-17-2007 -- Process second submission.
Programmer: Kasey Chen
Date       : 09/17/2007
Note:     : 1. Read 2nd submission of EncA.
            2. The data file 'EncA' contains one long stream record.
               Revised to read 'filler' in two digits instead of '+2'.
            3. Revised the length of 'units_raw' from '3' to '4'
               to accommodating the negative sign (-) added after converting

               right bracket [}] in the zone decimal conversion code.

Modified by : Kasey Chen
Date         : 09/21/2007
Note        : Read 2nd submission of EncA and first submission EncB from St. Loius.

Modified by : Kasey Chen
Date         : 10/01/2007
Note        : Read 3rd submission of EncA and first submission EncB from St. Loius.

Modified by : Kasey Chen
Date         : 10/19/2007
Note        : Read 4th submission of EncA and first submission EncB from St. Loius.
*==================================================================================*;


options center obs = max pageno = 1 pagesize = 96 linesize = 123 error = 2;
options nomprint;

%let payernum = P330;
%let subnum   = 004;
%let file1     = a;
%let file2     = b;

%let rptYr    = 2006;
```

```
proc datasets kill;
run;
quit;

%jobstart;


*==============================================================================*
Program:    I:\MCdb\DataCol\P350\programs\P350_Enc_read.sas
Project:    Maryland, Encounter Preprocessing
Purpose:    Read in the variable format Encounter data
--------------------------------------------------------------------------------
input:      P350EncA.txt
output:     P350_Enc_raw.sas7bdat
--------------------------------------------------------------------------------
global macros to adjust:
For each payer
   1. paynum
   2. subnum
   3. file
--------------------------------------------------------------------------------
Programmer:  Laurie Hamliton / Po-Lun Chou
date:        04/07/2005

Modified by: Po-Lun Chou
Date:        July 12, 2005

Modified by: Po-Lun Chou
Date:        September 27, 2005
Purpose:     Re-assigned ENCNT_ID values.
Note:        Payer P350 has variable format Encounter data file and we created
             ENCNT_ID once we read in the fixed portion of each records in the
             raw data files, so it caused the problem that we have same ENCNT_ID
             multiple times for all of the recrods from one person (i.e., the
             records which have same PIDBDG).  We fixed this mistake here to
             avoid getting incorrect SPEC later in Enc3 program

Modified by: Sireesha Thati
Date:        05/10/2006
Purpose:     To read the variable format data for 2005 Encounter data.

Modified by: Kasey Chen
Date:        07/31/2007
Purpose:     To read the variable format data for 2006 Encounter data.
             Added 'or (compress(ln_line, '20'x) = " ") to
             drop the record if the first 200 char are all nulls

Modified by: Kasey Chen
Date:        09/04/2007
Purpose:     Read 2nd submission.
*==============================================================================*;

options center obs = max pageno = 1 pagesize = 86 linesize = 123 error = 2;
options nomprint;

%let payernum = P350;
%let subnum   = 002;
%let file     = a;
%let rptYr    = 2006;
%let sbmssn   = Submission &subnum;
%let drive    = I;
%let infile1  = &drive:\Mcdb\datacol\&payernum.\rawdata\&payernum.EncA.txt;

libname library "&drive:\mcdb\datacol\formats\permanent";
libname in      "&drive:\mcdb\datacol\&payernum\rawdata";
libname out     "J:\mcdb_raw";

filename in1 "&infile1";

%let hmolst = 'P030','P160','P480','P520','P620','P660','P680','P870';
%let mhmolst = 'P480';
```

```
proc datasets kill;
run;

%jobstart;

*================================================================================*
Program:    I:\MCdb\datacol\P020\programs\P020_enc_check.sas
Project:    Maryland, Encounter Preprocessing
Purpose:    To check the data submitted by payers
--------------------------------------------------------------------------------
input:      P020_Enc_Raw.sas7bdat
output:     N/A
--------------------------------------------------------------------------------

Programmer: Sireesha Thati / Sanee Maphungphong
Date      : 03/27/2006
Note      : To check financials and units here, we have created temporary variables to
            change them to numeric (in general the conversion from character to numeric
            is done which is done in pxxx_enc1.sas), to apply our formats.
--------------------------------------------------------------------------------
Programmer: Sanee Maphungphong
Date      : 08/09/2007
Note      : Make sure that in addition to updating format used in ENC1.SAS, we need to
            update "format_icd9dx.sas" and "usmdzip.sas".
            Also added "31 - Skilled Nursing Facility" to "$svcplac" and update
"$unitind"
            according to changes submission manual.
*================================================================================*;

options center obs = max pageno = 1 pagesize = 79 linesize = 132 error = 2;
options nomprint;

%let payernum = P020;
%let rptYr    = 2006;
%let drive    = I;

libname in    "J:\mcdb_raw";
libname out   "&drive:\mcdb\datacol\&payernum\sasdata";

options nosource2;
%include "I:\MCdb\datacol\Formats\format_icd9dx.sas";
%include "I:\MCdb\datacol\Formats\usmdzip.sas";
options source2;

data check;
  set in.&payernum._Enc_raw;
  title1 " &payernum.";
     ********************************
     Billed Amount
     BILL_RAW ==> BILL_ORIG
     ********************************;
     length bill_orig 8;
     bill_orig = bill_raw;

     ********************************
     Allowed Amount
     ALLOW_RAW ==> ALLOW_ORIG
     ********************************;
     length allow_orig 8;
     allow_orig = allow_raw;

     ********************************
     Reimbursed Amount
     REIMB_RAW ==> REIMB_ORIG
     ********************************;

     length reimb_orig 8;
     reimb_orig = reimb_raw;

     ********************************
     Patient Liability
```

```
     PATLIAB_RAW ==> PATLIAB_ORIG
     *********************************;
     length patliab_orig 8;
     patliab_orig = patliab_raw;

     ****************************
     Units of Service
     UNITS_RAW ==> UNITS
     ****************************;
     length units 3;
     units = units_raw;

run;

proc format;
     value agecat
        .M     = 'missing'
        low-<0 = '<0'
        0-20   = '0-20'
        21-30  = '21-30'
        31-40  = '31-40'
        41-50  = '41-50'
        51-64  = '51-64'
        65-110 = '65-110';
     value miss
        .      = 'missing'
        other  = 'not missing';
         value $missyr
            ' ' = 'Year Missing';
         value $monfmt
            '01' - '12' = '01-12'
            ' '         = 'Missing';
         value $dayfmt
            '01' - '31' = '01-31'
            ' '         = 'Missing';
         value $by
        1900-2007 = '1900-2007'
        ' '       = 'Missing';
         value $cdhpind
            '0' = '0: No'
            '1' = '1: Yes';
         value $miss
            ' '   = 'Value Missing'
            other = 'Value Reported';
         value $clmrelc
            '0'   = '0: Non-Accident'
            '1'   = '1: Work Accident'
            '2'   = '2: Auto Accident'
            '3'   = '3: Other Accident'
            '9'   = '9: Payer Code=9 (Not Coded)'
            ' '   = 'Code Missing';
     value $covothr
            '0'   = '0: No-No Other Insurance Coverage'
            '1'   = '1: Yes-Other Coverage is Primary'
            '2'   = '2: Yes-Other Coverage is Secondary'
            '9'   = '9: Payer Code=9 (Not Coded)'
            ' '   = 'Code Missing';
         value $covtype
            '1'   = '1: Medicare Supplemental(Indvl,Grp,WRAP)'
            '2'   = '2: Individual Plan'
            '3'   = '3: Priv Employr Sponsord Fully Self-Insu'
            '4'   = '4: Private Employer Sponsored Insured'
            '5'   = '5: Public Employee'
            '6'   = '6: Comprehensive Std Health Benefit Plan'
            '7'   = '7: Mcare srv provided by a Mcare HMO/HCFA'
            '8'   = '8: Taft Hartley Jointly Managed Trust Fund'

            '9'   = '9: Payer Code=9 (Unknown)'
            ' '   = 'Code Missing';
          value $delivty
            '1'   = '1: HMO (Non-Medicaid,Includes Medicare)'
```

```
   '2'   = '2: PPO-POS'
   '3'   = '3: PPO or Other Managed Care'
   '4'   = '4: Indemnity Care'
   '5'   = '5: HMO-POS Rider'
   '9'   = '9: Payer Code=9 (Not Coded)'
   ' '   = 'Code Missing';
value $amamod
  'TC'   = 'TC'
  '26'   = '26'
  ' '    = 'missing'
  other  = 'other';
value $provflg
   '1'   = '1: Yes-Participating Provider'
   '2'   = '2: No-Not a Participating Provider'
   '3'   = '3: Payer Code=3 (Not Coded)'
   ' '   = 'Missing'
  other  = 'Invalid Code';
value $sex
   '1'   = '1: Male'
   '2'   = '2: Female'
   '3'   = '3: Payer Code=3 (Not Coded)'
   ' '   = 'Missing';
value $svcplac
   '11'  = '11: Provider''s Office'
   '12'  = '12: Patient''s Home'
   '13'  = '13: Assisted Living Facility'
   '20'  = '20: Urgent Care Facility'
   '21'  = '21: Inpatient Hospital'
   '22'  = '22: Outpatient Hospital'
   '23'  = '23: Emergency Room (Hospital portion)'
   '24'  = '24: Ambulatory Surgical Center'
   '25'  = '25: Birthing Center'
   '26'  = '26: Military Treatment Facility'
   '31'  = '31: Skilled Nursing Facility'
   '32'  = '32: Nursing Facility'
   '33'  = '33: Custodial Care Facility'
   '34'  = '34: Hospice'
   '41'  = '41: Ambulance-Land'
   '42'  = '42: Ambulance-Air or Water'
   '51'  = '51: Inpatient Psychiatric Facility'
   '52'  = '52: Psychiatric Facility Partial Hospitaln'
   '53'  = '53: Community Mental Health Center'
   '54'  = '54: Intermedi Care FCLTY/Mentally Retarded'
   '55'  = '55: Residential Subst Abuse Treatment Fcty'
   '56'  = '56: Psychiatric Residential Treatment Ctr'
   '57'  = '57: Non-Resident Subst Abuse Treatmnt Fcty'
   '60'  = '60: Mass Immunization Center'
   '61'  = '61: Comprehensive Inpat Rehabilitatn Fclty'
   '62'  = '62: Comprehensive Outpat Rehabilitatn Fcty'
   '65'  = '65: End Stage Renal Disease Treatment Fcty'
   '71'  = '71: State or Local Public Health Clinic'
   '81'  = '81: Independent Laboratory & Imaging'
   '99'  = '99: Other Unlisted Facility'
   '  '  = 'Missing';
 value $unitind
        '0'   = '0: Reported as zero, no allowed activities'
   '1'   = '1: Transportation Miles'
   '2'   = '2: Anesthesia Time Units'
   '3'   = '3: Services'
   '4'   = '4: Oxygen Units'
   '5'   = '5: Blood Units'
   '6'   = '6: Allergy Tests'
   '8'   = '8: Minutes of Anesthesia'
   '9'   = '9: Not Coded'
   ' '   = 'Missing';
 value finan
  .             = 'missing'
  -999          = '-999'
  low - < -999,
  -998 - -1     = '<0'
  0 = '0'
```

```
          1-high        = '>0';
         value units
           .      = 'missing'
           low-<0 = '<0'
           0      = '0'
           1      = '1'
           2-high = '>1';

run;

proc freq data = check;
 tables begyr_raw begmo_raw begday_raw / list missing;
 format begyr_raw $missyr. begmo_raw $monfmt. begday_raw $dayfmt.;
 title2 "Frequencies of Service From Date";
run;

proc freq data = check;
 tables birthyr_raw birthmo_raw birthdd_raw / list missing;
 format birthyr_raw $by. birthmo_raw $monfmt. birthdd_raw $dayfmt.;
 title2 "Frequencies of Patient Date of Birth";
run;

proc freq data = check;
 tables cdhpind_raw / list missing;
 format cdhpind_raw $cdhpind.;
 title2 "Frequencies of Consumer Directed Health Plan";
run;

proc freq data = check;
 tables claimcn_raw / list missing;
 format claimcn_raw $miss.;
 title2 "Frequencies of Claim Control Number";
run;

proc freq data = check;
 tables claimpyr_raw claimpmo_raw claimpdd_raw / list missing;
 format claimpyr_raw $missyr. claimpmo_raw $monfmt. claimpdd_raw $dayfmt.;
 title2 "Frequencies of Claim Paid Date";
run;

proc freq data = check;
 tables clmrelco_raw / list missing;
 format clmrelco_raw $clmrelc.;
 title2 "Frequencies of Claim Related Condition";
run;

proc freq data = check;
 tables covothr_raw / list missing;
 format covothr_raw $covothr.;
 title2 "Frequencies of Patient Covered by Other Insurance";
run;

proc freq data = check;
 tables covtype_raw / list missing;
 format  covtype_raw $covtype.;
 title2 "Frequencies of Coverage Type";
run;

proc freq data = check;
 tables cpt_raw / list missing;
 format cpt_raw $miss.;
 title2 "Frequencies of Procedure Code";
run;

proc freq data = check;
 tables delivtyp_raw / list missing;
 format delivtyp_raw $delivty.;
 title2 "Frequencies of Delivery System Type";
run;

proc freq data = check;
```

```
 tables dx1_raw dx2_raw dx3_raw dx4_raw dx5_raw dx6_raw dx7_raw dx8_raw dx9_raw dx10_raw/
list missing;
 format dx1_raw dx2_raw dx3_raw dx4_raw dx5_raw dx6_raw dx7_raw dx8_raw dx9_raw dx10_raw
$icd9dx.;
 title2 "Frequencies of Diagnosis Codes";
run;

proc freq data = check;
 tables endyr_raw endmo_raw endday_raw / list missing;
 format endyr_raw $missyr. endmo_raw $monfmt. endday_raw $dayfmt.;
 title2 "Frequencies of Service Thru Date";
run;

proc freq data = check;
 tables fedtaxid_raw / list missing;
 format fedtaxid_raw $miss.;
 title2 "Frequencies of Practitioner Federal Tax Id";
run;

proc freq data = check;
 tables mod1_raw / list missing;
 format mod1_raw $amamod.;
 title2 "Frequencies of Modifier - I";
run;

proc freq data = check;
 tables mod2_raw / list missing;
 format mod2_raw $amamod.;
 title2 "Frequencies of Modifier - II";
run;

proc freq data = check;
 tables numdx_raw / list missing;
 title2 "Frequencies of Number of Diagnosis Codes";
run;

proc freq data = check;
 tables numline_raw / list missing;
 title2 "Frequencies of Number of Line Items";
run;

proc freq data = check;
 tables partprov_raw / list missing;
 format partprov_raw $provflg.;
 title2 "Frequencies of Participating Provider Flag";
run;

proc freq data = check;
 tables patid_raw / list missing;
 format patid_raw $miss.;
 title2 "Frequencies of Patient ID";
run;

proc freq data = check;
 tables provid_raw / list missing;
 format provid_raw $miss.;
 title2 "Frequencies of Servicing Practitioner ID";
run;

proc freq data = check;
 tables sex_raw / list missing;
 format sex_raw $sex.;
 title2 "Frequencies of Patient Sex";
run;

proc freq data = check;
 tables svcplace_raw / list missing;
 format svcplace_raw $svcplac.;
 title2 "Frequencies of Place of Service";
run;
```

```
proc freq data = check;
 tables unitind_raw / list missing;
 format unitind_raw $unitind.;
 title2 "Frequencies of Service Unit Indicator";
run;

proc freq data = check;
 tables units / list missing;
 format units units.;
 title2 "Frequencies of Service Unit Indicator";
run;

proc freq data = check;
 tables zip_p_raw / list missing;
 format zip_p_raw $usmdzip.;
 title2 "Frequencies of Patient Zip Code";
run;

proc freq data = check;
 tables zip_s_raw  / list missing;
 format zip_s_raw $usmdzip.;
 title2 "Frequencies of Service Location Zip Code";
run;

proc freq data = check;
 tables enroll_yr_raw enroll_mm_raw enroll_dd_raw
        disenroll_yr_raw disenroll_mm_raw disenroll_dd_raw / list missing;
 title2 "Frequencies of Enrollment and Disenrollment Dates";
run;

proc freq data = check;
 tables billtype_raw * bill_orig * allow_orig * reimb_orig * patliab_orig /list missing;
 format bill_orig allow_orig reimb_orig patliab_orig finan.;
 title2 "Frequencies of Financials";
run;

proc freq data = check;
 tables encnt_id file_id pnum  / list missing;
 format encnt_id $miss.;
 title2 "Frequencies of the variables created in &payernum._enc_read program";
run;

%jobend;

dm "log;    print file = I:\Mcdb\datacol\&payernum.\programs\&payernum._enc_check.log
replace";
dm "output; print file = I:\Mcdb\datacol\&payernum.\programs\&payernum._enc_check.lst
replace";




proc datasets kill;
run;

%jobstart;

*==============================================================================*
Program :  I:\Mcdb\datacol\P020\programs\P020_prv_read.sas
Purpose :  1) Read in the provider data
           2) To Confirm Practitioner Federal Tax ID # matches Encounter file
              Practitioner Federal Tax ID #.
Input   :  P020prva.txt
Output  :  P020_prv_raw.sas7bdat
By      :  Sireesha Thati / Sanee Maphungphong
Date    :  03/28/2006
-------------------------------------------------------------------------------
Programmer: Sanee Maphungphong
Date        : 08/09/2007
Purpose : Change the report year from 2005 to 2006 and then read in the Provider
```

```
          raw data.
Note    : 1. Added a new variable 'npi_raw' for 'National Provider Indentifier'.
                                      2. Added 'npi_raw' to frequency list.
                                      3. This payer did not provide the new added variable
National Provider
                                              Identifier this year.
                                      4. Added one more checking up for fedtaxid_raw both
of 'Pxxx_enc_raw.sas7bdat'
                                         'Pxxx_prv_raw.sas7bdat' due to the low matching
rate on 'provid_raw'.

--------------------------------------------------------------------------------
Modified : Kasey chen
Date       : 08/30/2007
Note     : Run 2nd submission.
*================================================================================*;
options center obs = max pageno = 1 errors = 2 pagesize = 86 linesize = 123;

%let payernum    = P020;
%let sbmssn    = 002;
%let drive     = I;
%let file      = a;
%let rptYr     = 2006;

libname out "J:\Mcdb_raw";

filename filein "&drive:\Mcdb\datacol\&payernum.\rawdata\&payernum.PrvA.txt";

title1 "Maryland Medical Care Database (MCDB)";
title2 "Year &rptYr Provider Data";
title3 "&payernum (&Sbmssn) - Prv_Read";
title4 " ";

footnote2 "program: &drive:\Mcdb\datacol\&payernum.\programs\&payernum._Prv_read.sas";

****************
Read Raw Data
****************;
data out.&payernum._prv_raw;
     infile filein lrecl = 200 truncover;
     length provid_raw $13;
     input @  1  provid_raw      $char11.
           @ 12  fedtaxid_raw    $char9.
           @ 21  lname_raw       $char31.
           @ 52  fname_raw       $char19.
           @ 71  midinit_raw     $char1.
           @ 72  suffix_raw      $char4.
           @ 76  credent_raw     $char5.
           @ 81  spec1_raw       $char3.
           @ 84  spec2_raw       $char3.
           @ 87  spec3_raw       $char3.
           @ 90  deanum_raw      $char11.
           @101  business_raw    $char1.
                  @102  npi_raw               $char10.;

     recnum = _n_;

     length pnum $ 4;
     pnum = upcase("&payernum");

     length file_id $ 1;
     file_id = upcase("&file");
run;

proc contents data = out.&payernum._prv_raw;
run;

proc print data = out.&payernum._prv_raw (obs = 20);
  title "Sample print of the raw provider file";
run;
```

```
****************************************
Check Link to Encounter by provider id
****************************************;

*** unique list from provider file, by provid_raw ***;
proc summary data = out.&payernum._prv_raw nway;
     class provid_raw;
     output out = prov_f (keep = provid_raw);
run;

*** unique list from encounter file, by provid_raw ***;
proc summary data = out.&payernum._enc_raw(keep = pnum file_id encnt_id provid_raw) nway
missing;
     class provid_raw;
     output out = enc_f (drop = _type_);
run;

*** check by provid_raw ***;
data enc_prov_f;
     merge prov_f (in = inprov)
           enc_f (in = inenc);
     by provid_raw;

     length match $ 9;
     if inenc and inprov then match = 'Both';
     else if inenc and not inprov then match = 'Only_Enc';
     else if inprov and not inenc then match = 'Only_Prv';
run;

proc freq data = enc_prov_f;
  tables match / list missing;
  title "Check Link to Encounter -- By Provider ID";
run;

/****** 07/26/2007 Kasey Chen ******/
*******************************************************
Check Link to Encounter by practitioner federal Tax id

*******************************************************;

*** unique list from provider file, by fedtaxid_raw ***;
proc summary data = out.&payernum._prv_raw nway;
     class fedtaxid_raw;
     output out = prov_f1 (keep = fedtaxid_raw);
run;

*** unique list from encounter file, by fedtaxid_raw ***;
proc summary data = out.&payernum._enc_raw(keep = pnum file_id encnt_id fedtaxid_raw)
nway missing;
     class fedtaxid_raw;
     output out = enc_f1 (drop = _type_);
run;

*** check by fedtaxid_raw ***;
data enc_prov_f1;
     merge prov_f1 (in = inprov)
           enc_f1 (in = inenc);
     by fedtaxid_raw;

     length match $ 9;
     if inenc and inprov then match = 'Both';
     else if inenc and not inprov then match = 'Only_Enc';

     else if inprov and not inenc then match = 'Only_Prv';
run;

proc freq data = enc_prov_f1;
  tables match / list missing;
  title "Check Link to Encounter -- By Federal Tax id";
run;
```

```
*************************
Diagnostics on Variables
*************************;

proc format;
 value $miss
            ' '                    = 'Value Missing'
    '000000000' - '00000000000'    = 'Zero'
        other                      = 'Value Reported';
 value $spectyp
        '001',
        '002',
        '003',
        '004',
        '005',
        '006',
        '007',
        '008',
        '009',
        '010',
        '011',
        '012',
        '013',
        '014',
        '015',
        '016',
        '017',
        '018',
        '019',
        '020',
        '021',
        '022',
        '023',
        '024',
        '025',
        '026',
        '027',
        '028',
        '029',
        '030',
        '031',
        '032',
        '033',
        '035',
        '036',
        '039',
        '100',
        '040',
        '041',
        '042',
        '043',
        '044',
        '045',
        '046',
        '047',
        '048',
        '049',
        '050',
        '051',
        '052',
        '053',
        '054',
        '055',
        '056',
        '057',
        '058',
        '059',
        '060',
        '061',
        '063',
        '070',
```

```
        '071',
        '072',
        '073',
        '074',
        '075',
        '076',
        '080',
        '081',
        '082',
        '083',
        '084',
        '090',
        '091',
        '092',
        '102',
        '101',
        '103' = '1: Valid, not miscellaneous'
        '034',
        '037',
        '038',
        '062',
        '085',
        '093' = '2: Valid, Miscellaneous'
        ' '   = '4: Missing'
        other = '3: Invalid';
run;

*********************
Initial Checks
*********************;

data prv_raw;
   set out.&payernum._prv_raw;
   len_fedtax = length(fedtaxid_raw);
run;

title 'Frequencies';
proc freq data = prv_raw compress;
     tables provid_raw * fedtaxid_raw/list missing;
     tables spec1_raw * spec2_raw * spec3_raw/list missing;
     tables len_fedtax/list missing;
     tables deanum_raw/list missing;
     tables midinit_raw/list missing;
     tables suffix_raw/list missing;
     tables credent_raw/list missing;
     tables business_raw/list missing;
     tables file_id/list missing;
     tables pnum/list missing;
     tables npi_raw/list missing;
     format provid_raw fedtaxid_raw deanum_raw $miss. spec1_raw spec2_raw spec3_raw
$spectyp.;
run;


%jobend;

dm "log;    print file = I:\Mcdb\datacol\&payernum.\programs\&payernum._prv_read.log
replace";
dm "output; print file = I:\Mcdb\datacol\&payernum.\programs\&payernum._prv_read.lst
replace";

proc datasets kill;
run;
quit;


%jobstart;

*==============================================================================*
Program:    I:\MCdb\datacol\P020\programs\P020_enc1.sas
Project:    Maryland, Encounter Preprocessing
```

```
Purpose:
-------------------------------------------------------------------------------
input:      P020_Enc_Raw.sas7bdat
output:     P020_Enc1.sas7bdat
            P020_Enc1_Deleted.sas7bdat
            P020_Enc1_D_Flags.sas7bdat
-------------------------------------------------------------------------------
global macros to adjust:
A. Once for the base program
   1.
sampint
B. For each payer
   1. paynum
   2. subnum
   3. file
-------------------------------------------------------------------------------
Programmer:  Laurie Hamliton / Po-Lun Chou
date:        04/07/2005
modified:    Sireesha Thati
date:        05/22/2006
purpose:     Per the september 29, 2005 email from Sophie to Ben, we shall flag the
             medical records for 'NON-MD' residents but will not delete them.
             Hence we have d_zip_f='1' also included in enc1
Note:        From the output of the check program check all the years, if there is
             any invalid year the mdy function would give an error
-------------------------------------------------------------------------------
Modified By: Po-Lun Chou
Date:        August 6, 2007
Purpose:     1. Modified the lists, hmolst & mhmolst,
             2. Generated variable AGE12**_NR,
             3. Kept the records with deletion flag = 1 (D_***_F = 1) in Enc1 data
                file
             4. Added variable F_FAC
*===============================================================================*;

options center obs = max pageno = 1 pagesize = 90 linesize = 120 error = 2 nodate ;
options nomprint;

%let payernum = P020;
%let subnum   = 001;
%let file     = a;
%let yrenddt  = '31DEC2006'd;
%let rptYr    = 2006;
%let prevYr   = 05;
%let curYr    = 06;
%let nextYr   = 07;
%let sbmssn   = Submission &subnum;
%let sampint  = 8;
%let drive    = I;

* libname library "&drive:\mcdb\datacol\formats\permanent";
libname library ("&drive:\MCdb\datacol\Formats\PERMANENT\BETOS_formats",
                 "&drive:\MCdb\datacol\Formats\PERMANENT\RVU_formats");
libname in      "J:\mcdb_raw";
libname out     "&drive:\Mcdb\datacol\&payernum.\sasdata";
libname outj    "J:\mcdb_raw";

%let hmolst = 'P030','P130','P160','P480','P520','P620','P680','P870';
%let mhmolst = 'P480';

title1 "Maryland Medical Care Database (MCDB)";
title2 "Year &rptYr Encounter Data";
title3 "&Payernum. (&Sbmssn) - Enc1";
title4 " ";

footnote2 "program: &drive:\mcdb\datacol\&payernum.\programs\&payernum._Enc1.sas";

options nosource2;

*==============================*
```

```
            FORMATS

*==============================*;
*** cpt procedure codes ***;
%include "&drive:\MCdb\DataCol\formats\CPTFMT.SAS";

*** hcpcs procedure codes ***;
%include "&drive:\MCdb\DataCol\formats\HCPCSFMT.SAS";

*** home grown procedure codes ***;
%include "&drive:\MCdb\DataCol\formats\HMGFMT.SAS";

*** format to determine whether a zipcode is valid MD ***;
%include "&drive:\MCdb\DataCol\formats\MDSTZIPFMT.SAS";

*** MCDB Global Screen Flag: Routine dental procedure codes (cpt) ***;
%include "&drive:\MCdb\DataCol\formats\DENTAL.SAS";

*** MCDB Global Screen Flag: High cost cpt procedure codes to identify services with cost
> 7000 ***;
%include "&drive:\MCdb\DataCol\formats\HICOST.SAS";

*** MCDB Global Screen Flag: Revenue codes to identify hospital inpatient services ***;
%include "&drive:\MCdb\DataCol\formats\REVCODE.SAS";

*** MCDB Global Screen Flag:: ICD9 procedure codes to identify hospital inpatient
services ***;
%include "&drive:\MCdb\DataCol\formats\icd9pr_&curYr..sas";

*** icd9 diagnosis codes codes to validate diagnosis codes ***;
%include "&drive:\MCdb\DataCol\formats\icd9dx_&curYr..sas";

*** format to determine whether zipcode is valid US zipcode ***;
%include "&drive:\MCdb\DataCol\formats\USZIPFMT.SAS";

*** format to determine county code for each MD zipcode ***;
%include "&drive:\MCdb\DataCol\formats\MDCYZIPFMT.SAS";

*** format for payer name ***;
%include "&drive:\MCdb\DataCol\formats\PAYNAME.SAS";

*** format for scrambled patient id ***;
%include "&drive:\MCdb\DataCol\formats\ENCRYPTION_FORMAT.SAS";

*** format for BETOS ***;
* %include "&drive:\MCdb\DataCol\formats\FORMAT_BETOS_2005.SAS";

*** format for NEWMOD ***;
* %include "&drive:\MCdb\DataCol\formats\FORMAT_RVU_2005.SAS";

*** format for this program (Enc1) ***;
%include "&drive:\MCdb\DataCol\formats\formats_&curYr..SAS";


*==============================================*

              INCLUDE PROGRAMS

*==============================================*;
%include "&drive:\MCdb\DataCol\Include_Program\MZDCONV.SAS";

options source2;

*====================================================*

                 READ RAW DATA

*====================================================*;

data temp1
     (keep = age
```

```
age12&curyr
allow_orig
begday
begindt
begmo
begyr
betosagg
betosdet
bill_orig
billtype
birthdate
birthmo
birthyr
cdhpind
claimcn
claimpdd
claimpmo
claimpyr
clmpdate
clmrelco
county_p
covothr
covtype
cpt
delivtyp
dx1
dx2
dx3
dx4
dx5
dx6
dx7
dx8
dx9
dx10

dxdot1
dxdot2
dxdot3
dxdot4
dxdot5
dxdot6
dxdot7
dxdot8
dxdot9
dxdot10
dxval1
dxval2
dxval3
dxval4
dxval5
dxval6
dxval7
dxval8
dxval9
dxval10
encnt_id
endday
enddt
endmo
endyr
f_baddt
f_cap_orig
f_cpt
f_cpt_orig
f_fac
f_insamp
f_yr&nextyr
fedtaxid
file_id
instate
```

```
        los
        mod1
        mod2
        newmod
        numdxc
        partprov
        patid
        patid_raw
        patliab_orig
        payertyp
        payment_orig
        pidbdg
        plan
        plan2
        pnum
        provid
        reimb_orig
        sex
        svcplace
        svcplace_raw
        unitind
        units
        zip_p
        zip_s
        partprov_raw
        sex_raw
        newbetag
        newbetdt
        enrolldt
        disenrolldt
        age12&curyr._nr)

outj.&payernum._Enc1_Deleted
(keep = age
        age12&curyr
        allow_orig
        begday
        begindt
        begmo
        begyr
        betosagg
        betosdet
        bill_orig
        billtype
        birthdate
        birthmo
        birthyr
        cdhpind_raw
        claimcn
        claimpdd
        claimpmo
        claimpyr
        clmpdate
        clmrelco
        county_p
        covothr
        covtype
        cpt
        delivtyp
        dx1
        dx2
        dx3
        dx4
        dx5
        dx6
        dx7
        dx8
        dx9
        dx10
        encnt_id
        endday
```

```
            enddt
            endmo
            endyr
            f_baddt
            f_cap_orig
            f_cpt
            f_cpt_orig
            f_fac
            f_insamp
            f_yr&nextyr
            fedtaxid
            file_id
            instate
            los
            mod1
            mod2
            newmod
            numdxc
            partprov
            patid
            patliab_orig
            payertyp
            payment_orig
            pidbdg
            plan
            plan2
            pnum
            provid
            reimb_orig
            sex
            svcplace
            unitind
            units
            zip_p
            zip_s
            age12&curyr._nr)

     outj.&payernum._Enc1_D_Flags
     (keep = pnum
            file_id
            encnt_id
            d_dnt_f
            d_inp_f
            d_i7k_f
            d_zip_f
            d_dat_f
            f_yr&nextYr)

     outj.&payernum._EncProv
     (keep = pnum
            file_id
            encnt_id
            fedtaxid
            provid
            cpt);

     set in.&payernum._enc_raw;


*=========================================================================*

                      PATIENT IDENTIFICATION

*=========================================================================*;
     ********************************************
     Encrypt Patient ID
     PATID_RAW ==> PATID
     ********************************************;
     length holdpid $14;
     holdpid = left(patid_raw);
```

```
*** encrypt ***;
length old1-old14 $1;
length new1-new14 $1;
length patid $14;

array aold old1-old14;
array anew new1-new14;

old1  = substr(holdpid,1,1);
new1  = put(old1,$ecode.);
patid = new1;

do i = 2 to 14;
   aold[i] = substr(holdpid,i,1);
   anew[i] = put(aold[i],$ecode.);
   patid   = trim(patid) || anew[i];
end;

**********************************************
Patient Birth Date Information
BIRTHYR_RAW ==> BIRTHYR
BIRTHMO_RAW ==> BIRTHMO
Note: Birth Day (birthdd_raw) is supposed to
be coded 00 and is not moved into the MCDB
**********************************************;
*** Convert if ZD data ***;
%zdconv2(invar = birthyr_raw);
%zdconv2(invar = birthmo_raw);

length birthyr $4;
birthyr = birthyr_raw;

length birthmo $2;
birthmo = birthmo_raw;

**********************************************
Patient Gender
SEX_RAW ==> SEX
Automatic recodes from M and F to 1 and 2
**********************************************;
*** convert zoned decimal if necessary ***;
if sex_raw in ('A','B') then do;
   %zdconv2(invar = sex_raw);
end;

*** convert M/F to 1/2 ***;
length sex $1;
sex = sex_raw;
if upcase(sex) = 'M' then sex = '1';
else
if upcase(sex) = 'F' then sex = '2';

**********************************************
Assign Unique Patient Identifier
PIDBDG
This is a concatination of:
   1. Payer number (PNUM)
   2. Scrambled patient id (PATID)
   2. Patient Birth Year (BIRTHYR)
   4. Patient Birth Month (BIRTHMO)
   5. Patient Gender (SEX)
**********************************************;
length pidbdg $25;

if patid ne ' ' then do;
   pidbdg = pnum    ||
            patid   ||
            birthyr ||
            birthmo ||
            sex;
end;
```

```
     drop old1-old14 new1-new14;

     ********************************************
     Patient Zip Code
     ZIP_P_RAW ==> ZIP_P
     and
     Assign Patient County of Residence
     COUNTY_P
     ********************************************;
     *** convert if zoned decimal ***;
     %zdconv2(invar=zip_p_raw);

     length zip_p $5;
     zip_p = zip_p_raw;

     *** assign code for patient county of residence ***;
     length county_p $3;

     county_p = put(zip_p,$mdcyzip.);


*=====================================================================*

                       COVERAGE ATTRIBUTES

*=====================================================================*;
     ********************************************
     Coverage Type
     COVTYPE_RAW ==> COVTYPE
     ********************************************;
     if covtype_raw in ('A','B','C','D','E','F','I') then do;
        %zdconv2(invar=covtype_raw);
     end;

     length covtype $1;
     covtype = covtype_raw;

     ********************************************
     CDHPIND_RAW ==> CDHPIND
     ********************************************;
     if cdhpind_raw in ('A','{') then do;
     %zdconv2(invar=cdhpind_raw);
     end;

     length cdhpind $1;
     cdhpind = cdhpind_raw;

     ********************************************
     Delivery System
     DELIVTYP_RAW ==> DELIVTYP
     ********************************************;
     if delivtyp_raw in ('A','B','C','D','E','I') then do;
        %zdconv2(invar=delivtyp_raw);
     end;

     length delivtyp $1;
     delivtyp = delivtyp_raw;

/** Note from Po-Lun on 07-25-2007:                                    **/
/** I commented the codes below out because it's not suitable for this year **/

/**
     if pnum = 'P131' then do;
        if delivtyp = '5' then delivtyp = '4';
     end;
**/

     ********************************************
     Patient Other Insurance Coverage
     COVOTHR_RAW ==> COVOTHR
```

```
     ***********************************************;
     *** convert zoned decimal ***;
     if covothr_raw in ('A','B','I','{') then do;
        %zdconv2(invar=covothr_raw);
     end;

     length covothr $1;
     covothr = covothr_raw;


*====================================================================*

                        CLAIM ATTRIBUTES

*====================================================================*;
     **********************************************
     Claim Control Number
     CLAIMCN_RAW ==> CLAIMCN
     **********************************************;
     length claimcn $23;
     claimcn = claimcn_raw;

     **********************************************
     Claim Paid Day
     CLAIMPDD_RAW ==> CLAIMPDD
     **********************************************;
     *** convert zoned decimal ***;
     %zdconv2(invar=claimpdd_raw);

     length claimpdd $2;
     claimpdd = claimpdd_raw;

     **********************************************
     Claim Paid Month
     CLAIMPMO_RAW ==> CLAIMPMO
     **********************************************;
     *** convert zoned decimal ***;
     %zdconv2(invar=claimpmo_raw);

     length claimpmo $2;
     claimpmo = claimpmo_raw;

     **********************************************
     Claim Paid Year
     CLAIMPYR_RAW ==> CLAIMPYR
     **********************************************;
     *** convert zoned decimal ***;
     %zdconv2(invar=claimpyr_raw);

     length claimpyr $4;
     claimpyr = claimpyr_raw;

     **********************************************
     Assign Claim Paid Data
     CLMPDATE
     **********************************************;
     clmpdate = mdy(claimpmo, claimpdd, claimpyr);
     format clmpdate mmddyy8.;

     **********************************************
     Claim Related Condition
     CLMRELCO_RAW ==> CLMRELCO
     **********************************************;
     if clmrelco_raw in ('A','B','C','I','{') then do;
        %zdconv2(invar=clmrelco_raw);
     end;

     length clmrelco $1;
     clmrelco = clmrelco_raw;
```

```
*============================================================*

                      SERVICE DATES

*============================================================*;
     ***********************************************
     Service Begin Month
     Enroll Begin Month
     disenroll Begin Month
     BEGMO_RAW ==> BEGMO
     ENROLL_MM_RAW ==> ENROLLMM
     DISENROLL_MM_RAW ==> DISENROLLMM
     ***********************************************;
     *** convert zoned decimal ***;
     %zdconv2(invar=begmo_raw);
     %zdconv2(invar=enroll_mm_raw);
     %zdconv2(invar=disenroll_mm_raw);

     length begmo enrollmm disenrollmm $2;
     begmo = begmo_raw;
     enrollmm = enroll_mm_raw;
     disenrollmm = disenroll_mm_raw;

     ***********************************************
     Service Begin Day
     Enroll Begin Month
     disenroll Begin Month
     BEGDAY_RAW ==> BEGDAY
     ENROLL_DD_RAW ==> ENROLLDD
     DISENROLL_DD_RAW ==> DISENROLLDD
     ***********************************************;
     *** convert zoned decimal ***;
     %zdconv2(invar=begday_raw);
     %zdconv2(invar=enroll_dd_raw);
     %zdconv2(invar=disenroll_dd_raw);

     length begday enrolldd disenrolldd $2;
     begday = begday_raw;
     enrolldd = enroll_dd_raw;
     disenrolldd = disenroll_dd_raw;

     ***********************************************
     Service Begin Year
     Enroll Begin Year
     disenroll Begin Year
     BEGYR_RAW ==> BEGYR
     ENROLL_YR_RAW ==> ENROLLYR
     DISENROLL_YR_RAW ==> DISENROLLYR
     ***********************************************;
     *** convert zoned decimal ***;
     %zdconv2(invar=begyr_raw);
     %zdconv2(invar=enroll_yr_raw);
     %zdconv2(invar=disenroll_yr_raw);

     length begyr enrollyr disenrollyr $4;
     begyr = begyr_raw;
     enrollyr = enroll_yr_raw;
     disenrollyr = disenroll_yr_raw;

     ***********************************************
     Service Begin Date
     BEGINDT
     ***********************************************;
     *** the mdy function will only convert if year, month and day are valid ***;
     if begyr not in ('9999','0000') then do;
        begindt = mdy(begmo,begday,begyr);
     end;
     format begindt mmddyy8.;

     ***********************************************
     Service End Month
```

```
ENDMO_RAW ==> ENDMO
*********************************************;
*** convert zoned decimal ***;
%zdconv2(invar=endmo_raw);

length endmo $2;
endmo = endmo_raw;

*********************************************
Service End Day
ENDDAY_RAW ==> ENDDAY
*********************************************;
*** convert zoned decimal ***;
%zdconv2(invar=endday_raw);

length endday $2;
endday = endday_raw;

*********************************************
Service End Year
ENDYR_RAW ==> ENDYR
*********************************************;
*** convert zoned decimal ***;
%zdconv2(invar=endyr_raw);

length endyr $4;
endyr = endyr_raw;

*********************************************
Service End Date
ENDDT
*********************************************;
***  the mdy function will only convert if year, month and day are valid ***;
if endyr not in ('9999','0000') then do;
   enddt = mdy(endmo,endday,endyr);
end;
format enddt mmddyy8.;

****************************************************
Adjust begdt and enddt per Bens email of 10/31/01
Add flag to indicate adjustment status
    F_BADDT

1 = Bad beginning and ending dates
2 = Bad ending date, beginning date assigned
3 = Bad beginning date, ending date assigned
4 = Both dates good
****************************************************;
length f_baddt $1;

*** bad beginning and ending dates ***;
if begindt = . and
   enddt = . then f_baddt = '1';

*** bad end date, good beginning date, assign begdt to enddt ***;
else if enddt = . then do;
   enddt = begindt;
   f_baddt = '2';
end;

*** bad beginning date, good end date, assign enddt to begdt ***;
else if begindt = .  then do;
   begindt = enddt;
   f_baddt = '3';
end;

*** both dates good ***;
else f_baddt = '4';

****************************************************
Flag Services Occurring Later than Reporting Year
```

```
F_YRXX
****************************************************;
length f_yr&nextYr $1;

f_yr&nextYr = '0';
if begindt ne . then do;
   if year(begindt) > 20&curYr then f_yr&nextYr = '1';
end;
else do;
   if year(enddt) > 20&curYr then f_yr&nextYr = '1';
end;

**********************************************
Calculate Age at Service
AGE
**********************************************;
*****************************************************************************
For 2004 MCDB, we had a problem with the calculation of the variable Age
(it was set to missing for all obs) for the payers with variable format.
This problem is corrected for 2005 MCDB by creating the variable
begindate first and then the variable age.
*****************************************************************************;
length age 3;

*** Have birth year ***;
if '1700' < birthyr < '9999' then do;

*** Have month, use the 15th to calculate date ***;
   if ('01' <= birthmo< = '12') then birthdate=mdy(birthmo,15,birthyr);

*** don't have month, use July 15 to calculate date ***;
   else birthdate=mdy(7,15,birthyr);

end;

*** No birth year information, birthdate set to missing ***;
else birthdate = .;

if birthdate = . then age = .;
else age = round((begindt-birthdate)/365.25,1);

if age < 0 or age > 110 then age= .M;
format birthdate mmddyy8.;

**********************************************
Calculate Age as of December 31 of the
Reporting Year
AGE12xx
**********************************************;
length age12&curyr 3;

*** Have birth year ***;
if '1700' < birthyr < '9999' then do;

*** Have month, use the 15th to calculate date ***;
   if ('01' <= birthmo <= '12') then birthdate = mdy(birthmo,15,birthyr);

*** Don't have month, use July 15 to calculate date ***;
   else birthdate = mdy(7,15,birthyr);

end;

*** No birth year information, birthdate set to missing ***;
else birthdate = .;

if birthdate = . then age12&curyr = .;
else age12&curyr = round((&yrenddt-birthdate)/365.25,1);

if birthyr = . then age12&curyr = .;
else if age12&curyr < 0 or age12&curyr > 110 then age12&curyr= .M;
```

```
     *===> Note: In few cases we get age as of December 31 of the reporting year -
AGE12xx;
     *===>       less than the age at service - AGE. This happens when service dates
fall;
     *===>       in between January to April of the next year.;

     ***********************************************
     Calculate Non Rounded Age as of December 31
     of the Reporting Year
     AGE12xx_NR
     ***********************************************;

     *===> Note from Po-Lun on 07-25-2007;
     *===>      Per Sanee's request, I moved the section below regarding generating
variable;
     *===>      AGE12xx_NR from P***_Enc3.sas to here;

     length age12&curyr._nr 3;

     dobyr=input(birthyr,5.);
     if birthdate = . then age12&curyr._nr = .;
     else age12&curyr._nr = (&yrenddt-birthdate)/365.25;

     *===> Note: Patients born after December 31 of the reporting year will be set to .M;
     *===>       However rounded age12&curyr (Age as of December,31 of Current Year);
     *===>       will be set to zero if patients were born in the four months after;
     *===>       December, 31 of Current Year.;

     if dobyr= . then age12&curyr._nr = .;
     else if age12&curyr._nr < 0 or age12&curyr._nr > 110 then age12&curyr._nr = .M;

     *************************************
     Enroll and Disenroll Dates
     ENROLLDT and DISENROLLDT
     *************************************;

     format enrolldt disenrolldt mmddyy8.;

     if enrollyr not in ('9999','0000') then do;
        enrolldt = mdy(enrollmm, enrolldd, enrollyr);
     end;

     if disenrollyr not in ('9999','0000','0023','0024') then do;
        disenrolldt = mdy(disenrollmm, disenrolldd, disenrollyr);
     end;

     ***********************************************
     Length of Service, Days
     LOS
     ***********************************************;
     length los 3;

     *** calculate los ***;
     if begindt ne . and enddt ne . then do;

     *** same day ***;
        if begindt = enddt then los = 1;

     *** more than one day with ending date greatest ***;
        else if enddt gt begindt then los = enddt - begindt;

     *** all other cases missing ***;
        else los = .;

     end;


*=====================================================================*

                          SERVICE LOCATION
```

```
*=====================================================================*;
     ****************************************
     Place of Service
     SVCPLACE_RAW ==> SVCPLACE

     Left justify and zero-fill.

     Convert MHCC to CMS coding
     ****************************************;
     *** convert zoned decimal ***;
     %zdconv2(invar=svcplace_raw);

     length svcplace $2;
     svcplace = svcplace_raw;
     svcplace = translate(right(svcplace),'0',' ');

     *** convert MHCC to  CMS codes ***;
     if svcplace = "01" then svcplace = "21";
     else
     if svcplace = "02" then svcplace = "22";
     else
     if svcplace = "03" then svcplace = "81";
     else
     if svcplace = "04" then svcplace = "11";
     else
     if svcplace = "05" then svcplace = "12";
     else
     if svcplace = "06" then svcplace = "71";
     else
     if svcplace = "07" then svcplace = "34";
     else
     if svcplace = "08" then svcplace = "54";
     else
     if svcplace = "09" then svcplace = "61";
     else
     if svcplace = "10" then svcplace = "99";
     else
     if svcplace = "31" then svcplace = "32";
     else
     if svcplace = "72" then svcplace = "71";

     ******************************************
     Identify Service Performed in a Facility
     F_FAC
     ******************************************;
     ** Identify facility-based claims (IP,OP,ASC,SNF) **;

     length f_fac $1;
     if svcplace in ('21', '22', '23', '24',
                     '31', '34',
                     '51', '52')
     then f_fac = 1;
     else f_fac = 0;

     ****************************************
     Service Location Zip Code
     ZIP_S_RAW ==> ZIP_S
     ****************************************;
     *** convert zoned decimal ***;
     %zdconv2(invar=zip_s_raw);

     length zip_s $5;
     zip_s = zip_s_raw;

     ***************************************************************************
     Flag service location MD/other
     INSTATE

     1 = practitioner site of service is a valid MD zipcode
     0 = practitioner site of service is a valid non-MD zipcode
     Missing  = practitioner site of service is a invalid zipcode
```

```
     ************************************************************************;
     length instate $1;
     if put(zip_s,$mdstzip.) = '1' then instate = '1';
     else
     if put(zip_s,$uszip.) = '1'  then instate = '0';
     else
     instate = ' ';


*=====================================================================*

                        SERVICE UNITS

*=====================================================================*;
     ****************************************************************
     Units of Service
     UNITS_RAW ==> UNITS

     For year 2006, data reported include 1 implied decimal point.
     ****************************************************************;
     length units 3;
     %zdconv1(invar=units_raw, outvar=units_raw);
     units = units_raw;

     ****************************
     Service Unit Indicator
     UNITIND_RAW ==> UNITIND
     ****************************;
     length unitind $1;
     unitind = unitind_raw;


*=====================================================================*

                       DIAGNOSIS CODES

*=====================================================================*;
     ****************************************
     Diagnosis Codes
     DX1_RAW-DX10_RAW ==> DX1-DX10

     Remove '.'s

     Diagnosis Code Validity Flags - Dots in
     Codes
     DXDOT1-DXDOT10
     0 = No dots
     1 = Dots

     Diagnosis Code Validity Flags - Invalid
     Codes
     DXVAL1-DXVAL10
     0 = Invalid
     1 = Valid
     ' ' = Missing
     ****************************************;
     length dx1 - dx10 $5;
     length dxdot1 - dxdot10 $1;
     length dxval1 - dxval10 $1;

     array adxraw dx1_raw dx2_raw dx3_raw dx4_raw dx5_raw dx6_raw
                  dx7_raw dx8_raw dx9_raw dx10_raw;
     array adx    dx1 - dx10;
     array adxdot dxdot1 - dxdot10;
     array adxval dxval1 - dxval10;

     do i = 1 to 10;
        *** assign ***;
        adx[i] = left(adxraw[i]);

        *** special case for State Farm (P760) ***;
```

```
      if pnum = 'P760' then do;
         length pos1 - pos5 $1.;
         array apos pos1 - pos5;
         do j = 1 to 5;
            apos[j] = substr(adx[i],j,1);
         end;
         if apos[5]='0' then adx[i] = substr(adx[i], 1, 4);
      end;

      *** validity check for dots, remove if present ***;
      adxdot[i] = '0';
      if index(adx[i],'.') > 0 then do;
         adxdot[i] = '1';
         adx[i] = compress(adx[i],'.');
      end;

      *** validity check code ***;
      if adx[i] = ' ' then adxval[i] = ' ';
      else
      if put(adx[i],icd9dx.) = '1' then adxval[i] = '1';
      else
      adxval[i] = '0';
   end;

   *********************************************
   Count Number of Diagnosis Codes
   NUMDXC
   Note: The payer's count, NUMDX_RAW, is not
   carried into the MCDB.
   *********************************************;

   length numdxc 3;

   *** initialize count to 0 ***;
   numdxc=0;

   array diag dx1 - dx10;

   *** loop through the 10 diagnosis code variables and count all nonmissingn ***;
   do i = 1 to 10;
      if diag(i) notin ('00'x,'','00000','0','00','000','0000')
         and compress(diag(i),'00'x) ne ''  then numdxc+1;
   end;

   drop i;


*==============================================================================*

                          PROCEDURE ATTRIBUTES
                                  and
                             BETOS CODING

*==============================================================================*;
   ************************************************
   Procedure Code
   CPT_RAW == > CPT

   Assign flag for procedure code type
   F_CPT
   1 = HomeGrown
   2 = HCPCS
   3 = CPT
   4 = ICD-9-CM
   5 = Revenue Code
   6 = missing/blank
   7 = Invalid Procedure Code

   Assign another flag for procedure code length
   F_CPT_ORIG
   ************************************************;
```

```
length cpt $6;
cpt = left(cpt_raw);

    length f_cpt_orig $1;
    if length(trim(left(cpt))) = 5 then f_cpt_orig = '1';
    else if length(trim(left(cpt))) = 6 then f_cpt_orig = '0';

*** left-adjust procedure code if necessary ***;
if length(trim(left(cpt))) = 5 then cpt = trim(left(cpt));

length f_cpt $1;

if cpt=' '                                then f_cpt='6';
else
if put(cpt,$&payernum.hmg.) = 'HomeGrown'  then f_cpt='1';
else
if put(cpt,$hcpcs.) = '1'                 then f_cpt='2';
else
if put(cpt,$cptfmt.) = '1'                then f_cpt='3';
else
if put(cpt,$icd9pr.) = '1'                then f_cpt='4';
else
if put(cpt,$revcode.)= '1'                then f_cpt='5';
else                                           f_cpt='7';

************************************************
Procedure Code Modifiers
MOD1_RAW ==> MOD1
MOD2_RAW ==> MOD2
************************************************;
length mod1 $2;
mod1 = mod1_raw;

length mod2 $2;
mod2 = mod2_raw;

*******************************************************
Imputed Modifier
Work On Modifiers For 26 And Blank For Radiology
Impute Them By Site Of Service
*******************************************************;
length newmod $2;
length id_26_&prevyr id_tc_&prevyr id_26_&curyr id_tc_&curyr $2;

id_26_&prevyr = put(cpt,$y&prevyr._26m.);
id_tc_&prevyr = put(cpt,$y&prevyr._tcm.);
id_26_&curyr = put(cpt,$y&curyr._26m.);
id_tc_&curyr = put(cpt,$y&curyr._tcm.);

newmod = 'XX';

*** ONLY PROCESS THE CLAIMS WHERE 26 OR TC MODIFIER MIGHT APPLY ***;
*** special exceptions noted by inspection of the data 93555,   ***;
*** 93556 are out of line -- what is the solution to that       ***;
*** ASSIGN BASED ON EXISTING MODIFIER OR ON PLACE OF SERVICE    ***;
*** MENTION OF INDIVIDUAL CPTS BASED ON INSPECTION OF 1999 DATA ***;

if id_26_&prevyr = '26' or id_26_&curyr = '26' or
   id_tc_&prevyr = 'TC' or id_tc_&curyr = 'TC' then do;
   if cpt in ('93555', '93556') then newmod = '26';
   else
   if mod1 = '26' or mod2 = '26' then newmod = '26';
   else
   if mod1 = 'TC' or mod2 = 'TC' then newmod = 'TC';
   else
   if f_fac = 1 then newmod = '26';
   else
   newmod = '  ';
end;

if newmod not in ('26','TC') then newmod = '  ';
```

```
     ****************************************************************
     Assign BETOS codes
     BETOSDET
     BETOSAGG

     Note: Aggregated category 8, Childhood Immunizations, is not
     used after 2002
     ****************************************************************;
     *** Staring with 2003 MCDB, O1H - Childhood immunization is not computed. ***;
     *** Per Ben's e-mail on 9/2/2004. No category 8 for 2003.                 ***;

     length betosdet $3 betosagg $1;

     *** assign detailed betos category ***;
     if f_cpt in ("2", "3") then do;
          betosdet = put(cpt,PRC2BET.);
      end;

     *** assign aggregated betos category ***;
     if      substr(betosdet,1,1) = "M"         then betosagg = "1";
     else if substr(betosdet,1,1) = "P"         then betosagg = "2";
     else if substr(betosdet,1,1) = "I"         then betosagg = "3";
     else if substr(betosdet,1,1) = "T"         then betosagg = "4";
     else if substr(betosdet,1,1) = "D"         then betosagg = "5";
     else if substr(betosdet,1,1) = "O"         then betosagg = "6";
     else if substr(betosdet,1,1) in ("Y", "Z") then betosagg = "7";
     else if betosdet = ' ' then betosagg = " ";

     newbetag=betosagg;
     newbetdt=betosdet;


*==========================================================================*

                          FINANCIALS

*==========================================================================*;
     ********************************
     Billed Amount
     BILL_RAW ==> BILL_ORIG
     *********************************;
     length bill_orig 8;
     %zdconv1(invar=bill_raw, outvar=bill_raw);
     bill_orig = bill_raw;

     ********************************
     Allowed Amount
     ALLOW_RAW ==> ALLOW_ORIG
     *********************************;
     length allow_orig 8;
     %zdconv1(invar=allow_raw, outvar=allow_raw);
     allow_orig = allow_raw;

     ********************************
     Reimbursed Amount
     REIMB_RAW ==> REIMB_ORIG
     *********************************;
     length reimb_orig 8;
     %zdconv1(invar=reimb_raw, outvar=reimb_raw);
     reimb_orig = reimb_raw;

     ********************************
     Patient Liability
     PATLIAB_RAW ==> PATLIAB_ORIG
     *********************************;
     length patliab_orig 8;
     %zdconv1(invar=patliab_raw, outvar=patliab_raw);
     patliab_orig = patliab_raw;

     ********************************
```

```
     Bill Type
     BILLTYPE_RAW ==> BILLTYPE_ORIG
     and
     Capitation Flag
     F_CAP_ORIG
     0 = Not Capitated
     1 = Capitated
     *********************************;
     length billtype $1;
     billtype = billtype_raw;

     length f_cap_orig $1;
     f_cap_orig = '0';
     if billtype = '8' then f_cap_orig = 1;

     **********************************************
     Calculate FFS Payment
     PAYMENT_ORIG
     **********************************************;
     length payment_orig 8;
     if f_cap_orig = '0' then payment_orig = sum(patliab_orig, reimb_orig);


*==========================================================================*

                        PROVIDER ATTRIBUTES

*==========================================================================*;
     **********************************************
     Provider Federal Tax ID
     FEDTAXID_RAW ==> FEDTAXID
     **********************************************;
     length fedtaxid $9;
     fedtaxid = left(fedtaxid_raw);

     **********************************************
     Provider Identification Number
     PROVID_RAW ==> PROVID
     **********************************************;
     length provid $13;
     provid = left(provid_raw);

     **********************************************
     Participating Provider Flag
     PARTPROV_RAW ==> PARTPROV
     **********************************************;
     length partprov $1;
     partprov = partprov_raw;

     if partprov in ('0' '9') then partprov = '3';


*==============================================================*

                  MCDB GLOBAL SCREEN FLAG

*==============================================================*;
     **************************************************
     Delete Routine Dental Services
     D_DNT_F

     Compare to format containing list of routine
     dental codes.
     **************************************************;
     length d_dnt_f $1;
     d_dnt_f='9';

     *** exclude homegrown procedure codes ***;
     if put(cpt,$&payernum.hmg.) ne 'HomeGrown' then do;
        if put(cpt,$dentexc.) = '1' then d_dnt_f='1';
     end;
```

```
     ************************************************
     Delete Inpatient Services
     D_INP_F

     Based on place of service and type of procedure
     code.
     ************************************************;
     length d_inp_f $1;
     d_inp_f='9';

     if svcplace = '21' then do;
        if put(cpt,$&payernum.hmg.) ne 'HomeGrown' then do;
           if put(cpt,$icd9pr.) = '1' or
              put(cpt,$revcode.) = '1' then d_inp_f='1';
        end;
     end;

     ******************************************************
     Delete Financial Outliers
     D_I7K_F

     Outliers are services with if allowed amount >$7,000
     or <-$7,000 which are not on the list of high cost
     procedures
     ******************************************************;
     length d_i7k_f $1;
     d_i7k_f = '9';

     if ((allow_orig<-7000 and allow_orig>.Z) or allow_orig>7000) and
        put(cpt,$hicost.) = '0'                          and
        put(cpt,$&payernum.hmg.) ne 'HomeGrown'          then d_i7k_f='1';

     ************************************************
     Delete Non-Maryland Residents Services
     D_ZIP_F
     ************************************************;
     length d_zip_f $1;
     d_zip_f = '9';

     if put(zip_p,$mdstzip.) ne '1' then d_zip_f='1';

     ****************************************************
     Delete Services with Date Out of Range
     D_DAT_F

     Case 1. Service Date before 1/1/Service Year
     Case 2. both Begin and End Date are missing

     Note: use service from date if available otherwise
     use service thru date.
     ****************************************************;
     length d_dat_f $1;
     d_dat_f='9';

     if begindt = . and
        enddt   = . then d_dat_f = '1';
     else
     if begindt ne . then do;
        if year(begindt) < &curYr then d_dat_f = '1';
     end;
     else
     if year(enddt) < &curYr then d_dat_f = '1';


*==============================================================================*

                          PLAN and PAYER TYPE

*==============================================================================*;
     ******************************************************************
```

```
Create variable to flag a service provided by Medicare HMO
MCRHMO
*****************************************************************;
length mcrhmo $1;
mcrhmo = '0';

if pnum in (&mhmolst)    and
   delivtyp in ('1','5') and
   age12&curyr >= 65     and
   age12&curyr <= 110    then mcrhmo='1';

*****************************************************************
Create variable to flag Medicare service other than Medicare HMO
MCROTH
*****************************************************************;
length mcroth $1;
mcroth = '0';

if mcrhmo ne '1' then do;
   if covtype in ('1','7') and
      age12&curyr >= 65    and
      age12&curyr <= 110   then mcroth='1';
end;

*****************************************************************
Create variable to flag private HMO, non-Medicare Service
PRVHMO
*****************************************************************;
length prvhmo $1;
prvhmo = '0';

***
1. payer must be a known hmo
2. patient age must be between 0 and 64, or unknown
3. coverage type must be known and non-medicare
4. delivery system must be hmo
***;

if pnum in (&hmolst)                           and
           (0<=age12&curyr<65 or age12&curyr=.) and
           delivtyp in ('1','5')               then do;

   *** special case for Kaiser ***;
       if pnum = 'P480' and covtype = '7' then prvhmo = '1';
   *** everybody else ***;
       else
       if covtype in ('2','3','4','5','6') then prvhmo = '1';

end;

*************************************************************************
Create variable to flag private non-HMO service
PRVNHMO
*************************************************************************;
length prvnhmo $1;
prvnhmo = '0';

***
1. payer is not an hmo provider and
   age is between 0 and 64 or missing and
   coverage type is known, non-medicare
2. payer is a private medicare provider and
   age is between 0 and 64 or missing
3. Special cases for p850 and p480
***;

*** not an hmo provider ***;
if pnum notin (&hmolst)                 and
   (0<=age12&curyr<65 or age12&curyr=.) then do;
   *** special case for payor P850 which is all covtype=8(Taft-Hartley) ***;
       if pnum = 'P850' and covtype = '8' then prvnhmo = '1';
```

```
   *** everybody else ***;
       else
       if covtype in ('2','3','4','5','6') then prvnhmo = '1';
end;

*** hmo providers ***;
else do;
     if pnum in (&hmolst)                 and
     (0<=age12&curyr<65 or age12&curyr=.) and
     delivtyp in ('2','3','4')            then do;
     *** special case for 480 ***;
         if pnum = 'P480' and covtype = '7' then prvnhmo = '1';
     *** everybody else ***;
         else
         if covtype in ('2','3','4','5','6') then prvnhmo = '1';
     end;
end;

*****************************************************************
Plan Type (HMOFFS/NONHMO/HMOCAP)
PLAN
and
Plan Type 2 (internal use only)
PLAN2
and
Payer Type (1=PRVHMO 2=PRVNHMO 3=MCRALL 9=HMOCAP 5=UNKNOWN FFS)
PAYERTYP
*****************************************************************;
length plan $6 plan2 $6 payertyp $1;

*** all capitated ***;
if f_cap_orig = '1' then do;
   plan2 = 'HMOCAP';
   payertyp = '9';
end;

*** private hmo ffs ***;
else
if prvhmo = '1' then do;
   plan2 = 'HMOFFS';
   payertyp = '1';
end;

*** private non-hmo ffs ***;
else
if prvnhmo = '1' then do;
   plan2 = 'NONHMO';
   payertyp = '2';
end;

*** medicare hmo ffs ***;
else
if mcrhmo = '1' then do;
   plan2 = 'MCRALL';
   payertyp = '3';
end;

*** other medicare ***;
else
if mcroth = '1' then do;
   plan2 = 'MCRALL';
   payertyp = '5';
end;

*** unknown ffs ***;
else do;
     plan2 = ' ';
     payertyp = '5';
end;

************************
```

```
     ASSIGN PLAN
     ************************;
     if 0 <= age12&curyr < 65 or age12&curyr = . then plan = plan2;
     else plan = ' ';


*=====================================================================*

                       CREATE OUTPUT DATASETS

*=====================================================================*;
     ******************************************************
     Identify Records for 10% Sample
     F_INSAMP
     ******************************************************;
     length dig01 - dig14 $1;
     length hex01 - hex14 $2;
     length int01 - int14 8;
     length f_insamp $1;

     array digits (*) dig01 - dig14 ;
     array hexes  (*) hex01 - hex14 ;
     array ints   (*) int01 - int14 ;

     do j = 1 to 14 ;
        digits(j) = substr(patid,j,1) ;
        hexes(j)  = put(digits(j),$hex2.) ;
        ints(j)   = input(hexes(j),hex2.) ;
     end;

     label sampler = 'Sampling integer 0 to 9' ;
     sampler = sum(of int01 - int14) ;
     sampler = mod(sampler,10) ;

     if sampler = &sampint then f_insamp = '1';
     else
     f_insamp = '0';

     drop dig01 - dig14
          hex01 - hex14
          int01 - int14
          i
          j
          sampler
          holdpid
          id_26_&prevyr id_tc_&prevyr
          id_26_&curyr id_tc_&curyr
          ;

     ************************************
     Labels
     ************************************;
     options nosource2;
     %include "&drive:\MCdb\DataCol\Include_Program\labels_&curYr..sas";
     options source2;

     *********************************
     Create output datasets
     *********************************;

     *==> Note from Po-Lun on 07-25-2007:
     *==> In prior years, records with MCDB Global Flags set to one 1 are excluded from
the ENC1.;
     *==> For 2005 submission, MHCC asked that we keep "NON_MD residence" records (d-zip-
f = '1');
     *==> in the MCDB but deleting routine dental, inpatient, highcost and invalid date
of services;
     *==> records.
     *==> For 2006 submission, MHCC asked that we continue to apply MCDB Global Screen
flags but;
     *==> not deleting them from the data base.;
```

```
     *==> See e-mail regarding 2006 issues from MHCC, dated 07-05-2007;

     *==> Note: We do not delete from ENC1 records with d_zip_f = '1' or d_dnt_f = '1'
or;
     *==>       d_inp_f = '1' or d_i7k_f = '1' or d_dat_f = '1'.;
     *==>       However, we continue the output these records to delete file
(pxxx_Enc1_D_Flags.sas7bdat);
     *==>       This is for report and data chek purposes;

     output outj.&payernum._Enc1_D_Flags;
     output temp1;
     output outj.&payernum._EncProv;

     if d_zip_f = '1' or d_dnt_f = '1' or d_inp_f = '1' or d_i7k_f = '1' or
        d_dat_f = '1' then output outj.&payernum._Enc1_Deleted;

run;


*=========================================================================*

          CREATE MAIN DATASETS WITH NEW ADDED VARIABLES

*=========================================================================*;

proc sort data = temp1 out = tempsort tagsort;
     by pidbdg begindt;
run;

data out.&payernum._Enc1;
     set tempsort;
     by pidbdg;
     length holdcnty $3;
     retain holdcnty;

     *****************************************
     Identify demographic donor record
     F_DEMOG

     For each unique id, this is the record
     for the first date of service, either
     capitated or not
     *****************************************;
     *** create variable indicating record used to spread demographics -
         it will be the first analysis record if one is available,
         otherwise just the first record ***;
     length f_demog  $1;
     f_demog = '0';

     *** first record gets the demog donor flag, put demog vars in hold ***;
     if first.pidbdg then do;
        f_demog = '1';
        holdcnty = county_p;
     end;
     else if holdcnty='' then do;
          f_demog='1';
          holdcnty=county_p;
     end;


     ************************************************
     County to Use for Practitioner Report Analysis
     CNTY_P2
     ************************************************;
     *** all records get the hold values ***;
     length cnty_p2 $3;
     cnty_p2 = holdcnty;

     *** initialize the hold vars on the last record for a pidbdg ***;
     if last.pidbdg then do;
        holdcnty = ' ';
```

```
     end;

     label CNTY_P2 = 'County to Use for Pract. Report Analysis'
           F_DEMOG = 'Demographic Donor for Analysis File?';
     drop holdcnty;
run;


*=====================================================================*

                PRINT CONTENTS AND SAMPLE RECORDS

*=====================================================================*;

proc contents data = out.&payernum._Enc1;
     title5 " ";
run;

proc print data = out.&payernum._Enc1 (obs = 20);
     format begindt enddt birthdate clmpdate enrolldt disenrolldt mmddyy10.;
     title5 "20 Records of &payernum._Enc1";
run;


proc contents data = outj.&payernum._Enc1_Deleted;
     title5 " ";
run;

proc print data = outj.&payernum._Enc1_Deleted (obs = 20);
     format begindt enddt birthdate clmpdate mmddyy10.;
     title5 "20 Records of &payernum._Enc1_Deleted";
run;


proc contents data = outj.&payernum._Enc1_D_Flags;
     title5 " ";
run;

proc print data = outj.&payernum._Enc1_D_Flags (obs = 20);
     title5 "20 Records of &payernum._Enc1_D_Flags";
run;

proc freq data = outj.&payernum._Enc1_D_Flags;
     tables pnum file_id d_dnt_f d_inp_f d_i7k_f d_zip_f d_dat_f / list missing;
     format d_dnt_f $d_dnt_f. d_inp_f $d_inp_f. d_i7k_f $d_i7k_f.
            d_zip_f $d_zip_f. d_dat_f $d_dat_f.;
     title5 "Frequencies of &payernum._Enc1_D_Flags";
run;

proc freq data = outj.&payernum._Enc1_D_Flags;
     tables d_dnt_f * d_inp_f * d_i7k_f * d_zip_f * d_dat_f / list missing;
     title5 "Frequencies of &payernum._Enc1_D_Flags";
run;

proc contents data = outj.&payernum._EncProv;
     title5 " ";
run;

proc print data = outj.&payernum._EncProv (obs = 20);
     title5 "20 Records of &payernum._EncProv";
run;


%jobend;


dm "log;    print file = I:\Mcdb\datacol\&payernum.\programs\&payernum._enc1.log
replace";
dm "output; print file = I:\Mcdb\datacol\&payernum.\programs\&payernum._enc1.lst
replace";
```

```
*==============================================================================*
*
* Program     : I:\MCdb\datacol\Formats\formats_06.sas
* Project     : Maryland
* Programmer  : Laurie Hamilton / Po-Lun Chou
* Date Created: May 20, 2005
* Purpose     : Create formats in permanent library for using in the new
*               version of 2004 programs
* Modified    : June 12, 2006
* Purpose     : Create formats in permanent library for using in the new
*               version of 2005 programs
*
* Modified    : Sanee Maphungphong / Po-Lun chou
* Date        : July 18, 2007
* Purpose     : Modified formats in permanent library using in the new
*               version of 2006 'Data Dictionary - Encounter'
* Note        : 1. $SVCPLAC - Added value '31'.
*               2. $UNITIND - Modified value '3' - from 'Visit' to 'Service'
*                             Modified value '8' - 'Minutes of Anesthesia Time'
*                             Added value '0' - 'Values reported as zero
*                                               (no allowed activities)'
*                             Added value '7' - 'Lab Services'
*                   (Based on the waiver list, it's only for P130 & P131)
*               3. $BIRTHYR - Add 2007
*
* Modified    : Kasey Chen
* Date        : 09/05/2007
* Note        : Modified format $UNITIND:
*               1. Added '7' (P130/P131 only)
*               2. Added '9' /DME for P510
*
* Modified    : Kasey Chen
* Date        : 11/07/2007
* Note        : Modified format $UNITIND according to the 2006 Submission Manual:
*               1. Revised '7' and '9' for all payer.
*               2. Removed '0'.
*==============================================================================*;

proc format;

***************************
Payer, File and Record
Identifiers
***************************
;
     value $FILE_ID;

     value $MISS
           ' '   = 'Value Missing'
           other = 'Value Reported';


************************

Raw Input Variables
************************
;
     value $CDHPIND
           '0' = '0: No'
           '1' = '1: Yes';


**************************
Intermediate Variables
**************************
;
     value $D_DAT_F
           '1' = '1:Service Before 2006/Unknown,Exclude from MCDB'
           '9' = '9:Service in 2006+';

     value $D_DNT_F
```

```
            '1' = '1:Routine Dental Service,Exclude from MCDB'
            '9' = '9:Other';

      value $D_I7K_F
            '1' = '1:ALLOAMT>7000 or <-7000 and not Hicost,Exclude from MCDB'
            '9' = '9:ALLOAMT Between -$7000 and +$7000 or Hicost';

      value $D_INP_F
            '1' = '1:Inpatient Service,Exclude from MCDB'
            '9' = '9:Outpatient Service';

      value $D_ZIP_F
            '1' = '1:Non-MD Patient Zip,Include in MCDB'
            '9' = '9:MD Patient Zip';

      value $F_CAP_O
            '0' = '0: No-Not a Capitated Service'
            '1' = '1: Yes-A Capitated Service';

      value $PLAN_b
            'HMOCAP' = 'HMOCAP: HMO Capitated Plan'
            'HMOFFS' = 'HMOFFS: HMO Fee-For-Services Plan'
            'NONHMO' = 'NONHMO: Non-HMO Plan'
            'MCRHMO' = 'MCRHMO: Medicare HMO Plan'
            ' '      = 'Not Assigned';

*******************
MCDB Variables
*******************
;
      value $YEAR
            ' ' = 'Year = Missing';

      value $MONTH
            '00'        = 'Month = 00'
            '01' - '12' = 'Month = 01-12'
            ' '         = 'Month = Missing';

      value $DAY
            '00'        = 'Day = 00'
            '01' - '31' = 'Day = 01-31'
            ' '         = 'Day = Missing';

      value $BETOSAG
            '1'   = '1: Evaluation and Management'
            '2'   = '2: Procedures'
            '3'   = '3: Imaging'
            '4'   = '4: Tests'
            '5'   = '5: Durable Medical Equipment'
            '6'   = '6: Other'
            '7'   = '7: Exceptions/Unclassified'
            ' '   = 'Code Not Assigned';

      value $BETOSDE
            'M1A' = 'M1A: Office Visits - New'
            'M1B' = 'M1B: Office Visits - Established'
            'M2A' = 'M2A: Hospital Visit - Initial'
            'M2B' = 'M2B: Hospital Visit - Subsequent'
            'M2C' = 'M2C: Hospital Visit - Critical Care'
            'M3'  = 'M3: Emergency Room Visit'
            'M4A' = 'M4A: Home Visit'
            'M4B' = 'M4B: Nursing Home Visit'
            'M5A' = 'M5A: Specialist - Pathology'
            'M5B' = 'M5B: Specialist - Psychiatry'
            'M5C' = 'M5C: Specialist - Ophthalmology'
            'M5D' = 'M5D: Specialist - Other'
            'M6'  = 'M6: Consultations'
            'P0'  = 'P0: Anesthesia'
            'P1A' = 'P1A: Major Procedure - Breast'
            'P1B' = 'P1B: Major Procedure - Colectomy'
            'P1C' = 'P1C: Major Procedure - Cholecystectomy'
```

```
'P1D' = 'P1D: Major Procedure - TURP'
'P1E' = 'P1E: Major Procedure - Hysterectomy'
'P1F' = 'P1F: Major Procedure - Explor/Decompr/ExcisDisc'
'P1G' = 'P1G: Major Procedure - Other and Major Maternal Procedures'
'P2A' = 'P2A: Major Procedure, Cardiovascular - CABG'
'P2B' = 'P2B: Major Procedure, Cardiovascular - Aneurysm Repair'
'P2C' = 'P2C: Major Procedure, Cardiovascular - Thromboendarterectomy'
'P2D' = 'P2D: Major Procedure, Cardiovascular - Coronary Angioplasty (PTCA)'
'P2E' = 'P2E: Major Procedure, Cardiovascular - Pacemaker Insertion'
'P2F' = 'P2F: Major Procedure, Cardiovascular - Other'
'P3A' = 'P3A: Major Procedure, Orthopedic - Hip Fracture Repair'
'P3B' = 'P3B: Major Procedure, Orthopedic - Hip Replacement'
'P3C' = 'P3C: Major Procedure, Orthopedic - Knee Replacement'
'P3D' = 'P3D: Major Procedure, Orthopedic - Other'
'P4A' = 'P4A: Eye Procedures - Corneal Transplant'
'P4B' = 'P4B: Eye Procedures - Cataract Removal/Lens Insertion'
'P4C' = 'P4C: Eye Procedures - Retinal Detachment'
'P4D' = 'P4D: Eye Procedures - Treatment of Retinal Lesions'
'P4E' = 'P4E: Eye - Other'
'P5A' = 'P5A: Ambulatory Procedures - Skin'
'P5B' = 'P5B: Ambulatory Procedures - Musculoskeletal'
'P5C' = 'P5C: Ambulatory Procedures - Inguinal Hernia Repair'
'P5D' = 'P5D: Ambulatory Procedures - Lithotripsy'
'P5E' = 'P5E: Ambulatory Procedures - Other'
'P6A' = 'P6A: Minor Procedures - Skin'
'P6B' = 'P6B: Minor Procedures - Musculoskeletal'
'P6C' = 'P6C: Minor Procedures - Other (Medicare Fee Schedule)'
'P6D' = 'P6D: Minor Procedures - Other (Non-Medicare Fee Schedule)'
'P7A' = 'P7A: Oncology - Radiation Therapy'
'P7B' = 'P7B: Oncology - Other'
'P8A' = 'P8A: Endoscopy - Arthroscopy'
'P8B' = 'P8B: Endoscopy - Upper Gastrointestinal'
'P8C' = 'P8C: Endoscopy - Sigmoidoscopy'
'P8D' = 'P8D: Endoscopy - Colonoscopy'
'P8E' = 'P8E: Endoscopy - Cystoscopy'
'P8F' = 'P8F: Endoscopy - Bronchoscopy'
'P8G' = 'P8G: Endoscopy - Lararoscopic Cholecystectomy'
'P8H' = 'P8H: Endoscopy - Laryngoscopy'
'P8I' = 'P8I: Endoscopy - Other'
'P9A' = 'P9A: Dialysis Services (Medicare Fee Schedule)'
'P9B' = 'P9B: Dialysis Services (Non-Medicare Fee Schedule)'
'I1A' = 'I1A: Standard Imaging - Chest'
'I1B' = 'I1B: Standard Imaging - Musculoskeletal'
'I1C' = 'I1C: Standard Imaging - Breast'
'I1D' = 'I1D: Standard Imaging - Contrast Gastrointestinal'
'I1E' = 'I1E: Standard Imaging - Nuclear Medicine'
'I1F' = 'I1F: Standard Imaging - Other'
'I2A' = 'I2A: Advanced Imaging - CAT: Head'
'I2B' = 'I2B: Advanced Imaging - CAT: Other'
'I2C' = 'I2C: Advanced Imaging - MRI: Brain'
'I2D' = 'I2D: Advanced Imaging - MRI: Other'
'I3A' = 'I3A: Echography - Eye'
'I3B' = 'I3B: Echography - Abdomen/Pelvis'
'I3C' = 'I3C: Echography - Heart'
'I3D' = 'I3D: Echography - Carotid Arteries'
'I3E' = 'I3E: Echography - Prostate, Transrectal'
'I3F' = 'I3F: Echography - Other'
'I4A' = 'I4A: Imaging/Procedure - Heart, including Cardiac Catheterization'
'I4B' = 'I4B: Imaging/Procedure - Other'
'T1A' = 'T1A: Lab Tests - Routine Venipuncture (Non-Medicare Fee Schedule)'
'T1B' = 'T1B: Lab Tests - Automated General Profiles'
'T1C' = 'T1C: Lab Tests - Urinalysis'
'T1D' = 'T1D: Lab Tests - Blood Counts'
'T1E' = 'T1E: Lab Tests - Glucose'
'T1F' = 'T1F: Lab Tests - Bacterial Cultures'
'T1G' = 'T1G: Lab Tests - Other (Medicare Fee Schedule)'
'T1H' = 'T1H: Lab Test, Other (Non-Medicare Fee Schedule)'
'T2A' = 'T2A: Other Tests - Electrocardiograms'
'T2B' = 'T2B: Other Tests - Cardiovascular Stress Tests'
'T2C' = 'T2C: Other Tests - EKG Monitoring'
'T2D' = 'T2D: Other Tests - Other'
```

```
      'D1A' = 'D1A: Medical/Surgical Supplies'
      'D1B' = 'D1B: Hospital Beds'
      'D1C' = 'D1C: Oxygen and Supplies'
      'D1D' = 'D1D: Wheelchairs'
      'D1E' = 'D1E: Other DME'
      'D1F' = 'D1F: Orthotic Devices'
      'O1A' = 'O1A: Ambulance'
      'O1B' = 'O1B: Chiropractic'
      'O1C' = 'O1C: Enteral and Parenteral'
      'O1D' = 'O1D: Chemotherapy'
      'O1E' = 'O1E: Other Drugs'
      'O1F' = 'O1F: Vision, Hearing and Speech Services'
      'Y1'  = 'Y1: Medicare Fee Schedule'
      'Y2'  = 'Y2: Other - Non-Medicare Fee Schedule'
      'Z1'  = 'Z1: Local Codes'
      'Z2'  = 'Z2: Undefined Codes'
      '  '  = 'Code Not Assigned';

value $BILLTYP
      '1'   = '1: Final Bill'
      '8'   = '8: Capitated'
      ' '   = 'Code Missing';

value $BIRTHMO
      '00'        = 'Month = 00'
      '01' - '12' = 'Month = 01-12'
      ' '         = 'Month = Missing';

value $BIRTHYR
      '1880' - '1899' = '1880-1899'
      '1900' - '1919' = '1900-1919'
      '1920' - '1939' = '1920-1939'
      '1940' - '1959' = '1940-1959'
      '1960' - '1979' = '1960-1979'
      '1980' - '1999' = '1980-1999'
      '2000' - '2005' = '2000-2005'
      '2006'          = '2006'
      '2007'          = '2007'
      ' '             = 'Missing';

value $CLMRELC
      '0'   = '0: Non-Accident'
      '1'   = '1: Work Accident'
      '2'   = '2: Auto Accident'
      '3'   = '3: Other Accident'
      '9'   = '9: Payer Code=9 (Not Coded)'
      ' '   = 'Code Missing';

value $COUNTY
      '001' = '001: Allegany'
      '003' = '003: Anne Arundel'
      '005' = '005: Baltimore'
      '009' = '009: Calvert'
      '011' = '011: Caroline'
      '013' = '013: Carroll'
      '015' = '015: Cecil'
      '017' = '017: Charles'
      '019' = '019: Dorchester'
      '021' = '021: Frederick'
      '023' = '023: Garrett'
      '025' = '025: Harford'
      '027' = '027: Howard'
      '029' = '029: Kent'
      '031' = '031: Montgomery'
      '033' = '033: Prince Georges'
      '035' = '035: Queen Annes'
      '037' = '037: Saint Marys'
      '039' = '039: Somerset'
      '041' = '041: Talbot'
      '043' = '043: Washington'
      '045' = '045: Wicomico'
```

```
      '047' = '047: Worcester'
      '510' = '510: Baltimore City'
      ' '   = 'Patient Residence County Unknown';

value $COVOTHR
      '0'   = '0: No-No Other Insurance Coverage'
      '1'   = '1: Yes-Other Coverage is Primary'
      '2'   = '2: Yes-Other Coverage is Secondary'
      '9'   = '9: Payer Code=9 (Not Coded)'
      ' '   = 'Code Missing';

value $COVTYPE
      '1'   = '1: Medicare Supplemental(Indvl,Grp,WRAP)'
      '2'   = '2: Individual Plan'
      '3'   = '3: Priv Employr Sponsord Fully Self-Insu'
      '4'   = '4: Private Employer Sponsored Insured'
      '5'   = '5: Public Employee'
      '6'   = '6: Comprehensive Std Health Benefit Plan'
      '7'   = '7: Mcare srv provided by a Mcare HMO/HCFA'
      '8'   = '8: Taft Hartley Jointly Managed Trust Fund'
      '9'   = '9: Payer Code=9 (Unknown)'
      ' '   = 'Code Missing';

value $DELIVTY
      '1'   = '1: HMO (Non-Medicaid,Includes Medicare)'
      '2'   = '2: PPO-POS'
      '3'   = '3: PPO or Other Managed Care'
      '4'   = '4: Indemnity Care'
      '5'   = '5: HMO-POS Rider'
      '9'   = '9: Payer Code=9 (Not Coded)'
      ' '   = 'Code Missing';

value $F_BADDT
      '1' = '1: Bad Beginning and Ending Dates'
      '2' = '2: Bad Ending Date, Beginning Date Assigned'
      '3' = '3: Bad Beginning Date, Ending Date Assigned'
      '4' = '4: Both Dates Good';

value $F_CPT
      '1' = '1: Homegrown Code'
      '2' = '2: HCPCS Code'
      '3' = '3: CPT-4 Code'
      '4' = '4: ICD-9-CM Code'
      '5' = '5: Revenue Code'
      '6' = '6: Procedure Code Missing'
      '7' = '7: Procedure Code Invalid';

value $F_CPT_O
      '0' = '0: No-Raw Procedure Code Length > 5'
      '1' = '1: Yes-Raw Procedure Code Length = 5';

value $F_DEMOG
      '0' = '0: No-Demographics Not Used in Analysis'
      '1' = '1: Yes-Demographics Used in Analysis';

value $F_FAC
      '0' = '0: Service Not Provided in a Facility'
      '1' = '1: Service Provided in a Facility'
      ' ' = 'Not Coded (Procedure Code Not CPT/CMS)';

value $F_INSAM
      '1' = '1: In 10% Sample'
      '0' = '0: Not in 10% Sample';

value $F_YR
      '0' = '0: No - Service Provided in 2006'
      '1' = '1: Yes - Service Provided in 2007+';

value $INSTATE
      '0' = '0: Service Location Valid, Not MD (Based on Zipcode)'
```

```
      '1' = '1: Service Location in MD (Based on Zipcode)'

      ' ' = 'Service Location Unknown (Zipcode Missing)';

value $MOD
      '20'  = '20: Microsurgery'
      '21'  = '21: Prolonged Evaluatn/Mgmt(E/M) Serv'
      '22'  = '22: Unusual Procedural Services'
      '23'  = '23: Unusual Anesthesia'
      '24'  = '24: Unrel E/M Serv by Same Phys Postop'
      '25'  = '25: Sep Ident. E/M by Same Phys/day'
      '26'  = '26: Professional Component'
      '32'  = '32: Mandated Services'
      '47'  = '47: Anesthesia by Surgeon'
      '50'  = '50: Bilateral Procedure'
      '51'  = '51: Multiple Procedures'
      '52'  = '52: Reduced Services'
      '54'  = '54: Surgical Care Only'
      '55'  = '55: Postoperative Management Only'
      '56'  = '56: Preoperative Management Only'
      '57'  = '57: Decision for Surgery'
      '58'  = '58: Staged/Rel Proc by Same Phys Postop'
      '62'  = '62: Two Surgeons'
      '65'  = '65: Surgical Team'
      '76'  = '76: Repeat Procedure by Same Physician'
      '77'  = '77: Repeat Procedure by Another Physician'
      '78'  = '78: Return to Op Rm for Rel Proc Postop'
      '79'  = '79: Unrelated Proc by Same Phy Postop'
      '80'  = '80: Assistant Surgeon'
      '81'  = '81: Minimum Assisting Surgeon'
      '82'  = '82: Assist Surgeon(qual res phy unav)'
      '90'  = '90: Reference (Outside) Laboratory'
      '99'  = '99: Multiple Modifiers'
      'AN'  = 'AN: Anesthesia(surg. codes: 10000 thru 69999)'
      'TC'  = 'TC: Radiology: Technical Component'
      'SG'  = 'SG: HCFA Ambulatory Surgery'
      'QZ'  = 'QZ: Cert Reg Nurse Anesth (CRNA)'
      'QX'  = 'QX: Cert Reg Nurse Anesth. w/o doc dir'
      '  '  = 'Missing Code';

value $NEWMOD
      '26'  = '26: Professional Component'
      'TC'  = 'TC: Radiology: Technical Component'
      '  '  = 'Missing Code';

value $PARTPRO
      '1'   = '1: Yes-Participating Provider'
      '2'   = '2: No-Not a Participating Provider'
      '3'   = '3: Payer Code=3 (Not Coded)'
      ' '   = 'Code Missing';

value $PAYERTY
      '1' = '1: Private HMO FFS'
      '2' = '2: Private Non-HMO'
      '3' = '3: Medicare HMO FFS'
      '4' = '4: Medicare Non-HMO'
      '5' = '5: Not Classified'
      '9' = '9: Capitated Services'
      ' ' = 'Unknown - FFS Services';

value $PLAN
      'HMOCAP' = 'HMOCAP: HMO Capitated Plan'
      'HMOFFS' = 'HMOFFS: HMO Fee-For-Services Plan'
      'NONHMO' = 'NONHMO: Non-HMO Plan'
      ' '      = 'Not Assigned';

value $SEX
      '1' = '1: Male'
      '2' = '2: Female'
      '3' = '3: Payer Code=3 (Not Coded)'
      ' ' = 'Missing';
```

```
     value $SVCPLAC
           '11'  = '11: Provider''s Office'
           '12'  = '12: Patient''s Home'
           '13'  = '13: Assisted Living Facility'
           '20'  = '20: Urgent Care Facility'
           '21'  = '21: Inpatient Hospital'
           '22'  = '22: Outpatient Hospital'
           '23'  = '23: Emergency Room (Hospital portion)'
           '24'  = '24: Ambulatory Surgical Center'
           '25'  = '25: Birthing Center'
           '26'  = '26: Military Treatment Facility'
           '31'  = '31: Skilled Nursing Facility'
           '32'  = '32: Nursing Facility'
           '33'  = '33: Custodial Care Facility'
           '34'  = '34: Hospice'
           '41'  = '41: Ambulance-Land'
           '42'  = '42: Ambulance-Air or Water'
           '51'  = '51: Inpatient Psychiatric Facility'
           '52'  = '52: Psychiatric Facility Partial Hospitaln'
           '53'  = '53: Community Mental Health Center'
           '54'  = '54: Intermedi Care FCLTY/Mentally Retarded'
           '55'  = '55: Residential Subst Abuse Treatment Fcty'
           '56'  = '56: Psychiatric Residential Treatment Ctr'
           '57'  = '57: Non-Resident Subst Abuse Treatmnt Fcty'
           '60'  = '60: Mass Immunization Center'
           '61'  = '61: Comprehensive Inpat Rehabilitatn Fclty'
           '62'  = '62: Comprehensive Outpat Rehabilitatn Fcty'
           '65'  = '65: End Stage Renal Disease Treatment Fcty'
           '71'  = '71: State or Local Public Health Clinic'
           '81'  = '81: Independent Laboratory & Imaging'
           '99'  = '99: Other Unlisted Facility'
           '  '  = 'Missing';

     value $UNITIND
           '1'   = '1: Transportation (Ambulance Air or Ground)Miles'
           '2'   = '2: Anesthesia Time Units'
           '3'   = '3: Services'
           '4'   = '4: Oxygen Units'
           '5'   = '5: Units of Blood'
           '6'   = '6: Allergy Tests'
           '7'   = '7: Lab Services'
           '8'   = '8: Minutes of Anesthesia'
           '9'   = '9: Durable and Non-Durable Medical Equipment and Supplies Including
Drugs'
           ' '   = 'No Information';

run;

*==============================================================================
*PROGRAM:       mzdconv.sas
*PROJECT:       Maryland
*PURPOSE:       Macros to perform conversions from zoned decimal format
*               Include in step4-7 programs as needed
*PROGRAMMER:    Raymond Hu
*DATE MODIFIED: 7/25/01 lsh
*               code moved from 1999 zdconv.sas to this separate include program
*==================================================================================
;
*******************************************************************************
                        MACROS FOR ZONED DECIMAL CONVERSIONS
*******************************************************************************
;

%let char_ls1 = ABCDEFGHIJKLMNOPQR{};
%let char_ls2 = 0 -+.123456789;
%let char_ls3 = 0123456789;

************************************
MACRO ZDCONV1
Convert a character var with ASCII:
```

```
   a) 00000-123,
   b) +00000123 or 00000+123,
      but not 00123+123
   c) 000000123 or '123   ' or '    123'
   d) zoned decimal -> 000000012C
to a numeric var
************************************
;
%macro zdconv1(invar=,outvar=);

     if compress(&invar,"&CHAR_LS2")  = '' then do;

       %*** a) negative numeric ascii data ***;
       if index(&invar,"-")>0 then do;
         &outvar=input( substr(&invar,index(&invar,"-")) ,9.);
       end;

       %*** b) positive numeric ascii data ***;

       %**else if 0<index(&invar,"+")<length(&invar) then do;
       %**  &outvar=input(substr(&invar,index(&invar,"+")+1),9.);
       %**end;
       else if 0<index(&invar,"+")<length(&invar) then do;
              if index(&invar,"+")=1 then
&outvar=input(substr(&invar,index(&invar,"+")+1),9.);
              else if compress(substr(&invar,1,index(&invar,"+")-1),'0')='' then
                  &outvar=input(substr(&invar,index(&invar,"+")+1),9.);
       end;

       %*** c) positive numeric ascii data ***;
       else if 0=index(&invar,"+") then &outvar  = input(&invar,9.);
     end;

     %*** d) numeric zoned decimal data ***;
     else if length(trim(left(&invar)))=length(&invar) AND
compress(substr(&invar,1,length(&invar)-1),"&CHAR_LS3")  = '' AND
             index(substr(&invar,1,length(&invar)),"")  = 0 AND


length(compress("&CHAR_LS1",substr(&invar,length(&invar),1)))+1=length("&CHAR_LS1") then
             &outvar  = input(repeat('0',20-length(&invar)-1)||&invar,zd20.);

     %*** e) missing data ***;
     else &outvar=.;

%mend zdconv1;

************************************
MACRO ZDCONV2
Convert a zoned decimal character var
to a character var
Convert only if it is zd format:
   ('A' --> '1' or '012C' --> '0123')
************************************
;

%macro zdconv2(invar=);

     %*** a) one character var with zd format;
     if length(&invar)=1 then do;
       if compress(&invar,"&CHAR_LS1")  = '' and &invar ne '' then do;
          &invar  = put(input(&invar,zd1.),z1.);
       end;
     end;
     %*** b) more character zoned decimal data ***;
     else if index(substr(&invar,1,length(&invar)),"")  = 0 AND
             compress(substr(&invar,1,length(&invar)-1),"&CHAR_LS2")  = '' AND

length(compress("&CHAR_LS1",substr(&invar,length(&invar),1)))+1=length("&CHAR_LS1") then
do;
             ** put back as right adjustified, zero filled and the same length;
```

```
             &invar  = substr(put(input(&invar,zd9.),z20.),20-length(&invar)+1);
     end;
%mend zdconv2;

%jobstart;
*==================================================================================
Program   : i:\mcdb\datacol\P020\programs\P020_enc1_freqs.sas
Purpose   : To calculate frequency counts
----------------------------------------------------------------------------------
Input     : P020_enc1.sas7bdat
----------------------------------------------------------------------------------
By        : sireesha thati
Date      : June 13, 2005

Modified  : June 22, 2006 to add xtab of ENROOLDT and DISENROLLDT.
*==================================================================================;
options linesize = 132 pagesize = 72 errors = 2 nodate pageno=1 nolabel;

%let payer = P020;
%let drive = I;

title1 "MCDB Encounter Preprocessing";
title2 "January 2005 - April 2006 Data";
title3 "Frequency Counts (Source: Enc1 Dataset-D_Flag Services Revised)";
title4 "&payer";

footnote1 "program: &drive:\mcdb\datacol\&payer.\programs\&payer._enc1_freqs.sas";
libname in "&drive:\mcdb\datacol\&payer.\sasdata";
libname out "&drive:\mcdb\datacol\&payer.\sasdata";

options nosource2;
%include "I:\mcdb\datacol\formats\formats_05.sas";
%include "I:\MCdb\datacol\Formats\usmdzip.sas";
options source2;

proc format;
     value agecat
        .M     = 'missing'
        low-<0 = '<0'
        0-20   = '0-20'
        21-30  = '21-30'
        31-40  = '31-40'
        41-50  = '41-50'
        51-64  = '51-64'
        65-110 = '65-110';
     value miss
        .      = 'missing'
        other  = 'not missing';
     value $missa
       ' '                                          = 'Missing'
       '000000000' - '00000000000'   = 'Zero'
        other                        = 'Present';
     value $amamod
        'TC'   = 'TC'
        '26'   = '26'
        ' '     = 'missing'
        other  = 'other';
     value ageplan
        .M     = 'missing'
        0-64   = '0-64'
        65-110 = '65-110'
        low-<0 = '<0';
     value mydate
        .      = 'missing/invalid'
        other  = 'valid';
     value units
        .      = 'missing'
        low-<0 = '<0'
        0      = '0'
        1      = '1'
        2-high = '>1';
```

```
     value $by
        1900-2006 = '1900-2006';
     value $bm
        0-12 = '0-12';

proc freq data=in.p020_enc1;
     tables pnum
            file_id
            encnt_id
            birthyr
            birthmo
                            billtype
            patid
            sex_raw*sex
            cdhpind
            cnty_p2*county_p
            covothr
            covtype
            delivtyp
            clmrelco
            provid*fedtaxid
            partprov*partprov_raw
            claimcn
            claimpyr
            clmpdate
            begyr*endyr
            f_baddt
            dxval1*dxval2*dxval3*dxval4*dxval5*dxval6*dxval7*dxval8*dxval9*dxval10
            dxdot1*dxdot2*dxdot3*dxdot4*dxdot5*dxdot6*dxdot7*dxdot8*dxdot9*dxdot10
            svcplace_raw*svcplace
            instate
            unitind
            units
            f_cpt
            newmod*mod1*mod2
            age*age1205
            betosagg*newbetag
            f_demog
            f_insamp
            f_yr06*begyr*endyr
            los
            pidbdg
            plan*payertyp*plan2*age1205
                            zip_p
                            zip_s / list missing;
    format  encnt_id   $miss.
            birthyr    $by.
            birthmo    $bm.
            billtype   $billtyp.
            patid      $miss.
            cdhpind    $cdhpind.
            covothr    $covothr.
            covtype    $covtype.
            delivtyp   $delivty.
            clmrelco   $clmrelc.
            provid     $missa.
            fedtaxid   $missa.
            claimcn    $miss.
            clmpdate   mydate.
            svcplace   $svcplac.
            unitind    $unitind.
            units      units.
            f_cpt      $f_cpt.
            newmod     $amamod.
            mod1       $amamod.
            mod2       $amamod.
            age        agecat.
            age1205    agecat.
            f_baddt    $f_baddt.
            los        units.
            pidbdg     $miss.
```

```
            payertyp   $plan.
            instate    $instate.
            zip_p        $usmdzip.
            zip_s        $usmdzip.;
run;

proc freq data=in.p020_enc4;
   where (enrolldt GE disenrolldt);
   tables enrolldt*disenrolldt / missing list;
title3 "Check where Enroll is greater than or equal to Disenroll Dates";
run;

proc freq data=in.p020_enc1;
   where unitind = '8';
   tables units / missing list;
title3 " freq for Units where UNITIND = 8";
run;

%jobend;

dm "log;    print file = I:\Mcdb\datacol\&payer.\programs\&payer._enc4_freqs.log
replace";
dm "output; print file = I:\Mcdb\datacol\&payer.\programs\&payer._enc5_freqs.lst
replace";

proc datasets kill;
run;

%jobstart;

*==================================================================================*
Program:    I:\MCdb\datacol\P020\programs\P020_enc1_financials.sas
Project:    Maryland, Encounter Preprocessing
Purpose:    Run diagnostics on
Date:       June 21, 2006.
----------------------------------------------------------------------------------
Date:       June 21, 2006.
Programmer: Sanee Maphungphong
Purpose:    Check financials in ENC1
*==================================================================================*;
options linesize = 132 pagesize = 72 errors = 2 nodate pageno=1 nolabel;

%let pnum = P020;
%let drive = I;

libname dsin  "i:\MCDB\DATACOL\&pnum\sasDATA\";

*** include format for payer name ***;
%include 'i:\mcdb\DataCol\formats\PayName.sas';

title1 "MCDB Encounter Preprocessing";
title2 "January 2006 - April 2007 Data";
title3 "Frequency Counts (Source: Enc1 Dataset-D_Flag Services Revised)";
title4 "&pnum";

footnote1 "program: &drive:\mcdb\datacol\&pnum\programs\&pnum._enc1_financials.sas";
libname in "&drive:\mcdb\datacol\&pnum\sasdata";
libname out "&drive:\mcdb\datacol\&pnum\sasdata";

proc format;
  value fmtNine
                       .C = '.C'
                        . = '.'
                       .M = '.M'
                     -999 = '-999'
                    -9999 = '-9999'
                   -99999 = '-99999'
                  -999999 = '-999999'
                 -9999999 = '-9999999'
                -99999999 = '-99999999'
                   -99900 = '-99900'
```

```
                 -999900 = '-999900'
                -9999900 = '-9999900'
               -99999900 = '-99999900'
                     999 = '999'
                    9999 = '9999'
                   99999 = '99999'
                  999999 = '999999'
                 9999999 = '9999999'
                99999999 = '99999999'
               999999999 = '999999999'
                   99900 = '99900'
                  999900 = '999900'
                 9999900 = '9999900'
                99999900 = '99999900'
               999999900 = '999999900'
                   other = 'others';
  value chargex
                      .C = '.C'
                       . = '.'
      LOW-<-99999900,
      -99999900<-<-20000 = '<-20000, not -99999900'
         -20000 - <-7000 = '-20000 - <-7000'
  -7000 - <-999,-999<-<0 = '-7000 - <0,not -999'
                    -999 = '-999'
                       0 = '0'
                  1-7000 = '1-7000'
             7000<-20000 = '7000-20000'
      20000<-<999999900,
         999999900<-HIGH = '20000+, not 999999900';
  value charge
                      .C = '.C'
                       . = '.'
      LOW-<-99999900,
      -99999900<-<-20000 = '< -20000'
         -20000 - <-7000 = '-20000 - <-7000'
   -7000 - <-999,-999<-<0 = '-7000 - -1'
                    -999 = '-999'
                       0 = '0'
                   0<-<1 = '0<-<1'
                  1-7000 = '1 - 7000'
             7000<-20000 = '7000 - 20000'
      20000<-<999999900,
         999999900<-HIGH = '20000+'  ;
  value $delvsys
                     '1' = '1: HMO (Non-Medicaid)'
                     '2' = '2: PPO-POS'
                     '3' = '3: PPO/Other Managed Care'
                     '4' = '4: Indemnity Care'
                     '5' = '5: HMO-POS Rider'
                     '9' = '9: Not Coded'
                     ' ' = 'Blank';

  value $covtyp
                     '1' = '1: Medicare Supp. (Indvl,Grp,WRAP)'
                     '2' = '2: Individual Plan'
                     '3' = '3: Private Employer, Fully Self-Ins.'
                     '4' = '4: Private Employer'
                     '5' = '5: Public Employee'
                     '6' = '6: Comp Std Health Benefit Plan'
                     '7' = '7: Med. Serv provided by Med. HMO/HCFA'
                     '8' = '8: Taft Hartley Trust'
                     '9' = '9: Not Coded'
                     'Y' = 'Y: Blank/Missing'
                   other = 'Code Invalid';
  value finan
                       .  = 'missing'
                    -999  = '-999'
 low - < -999, -998 - -1  = '<0'
                       0 = '0'
                  1-high = '>0';
run;
```

```
data enc1_ck;
  set dsin.&pnum._enc1 (keep=encnt_id claimcn begindt enddt delivtyp covtype cpt
                            dx1 billtype patid pidbdg f_cap_orig plan payertyp
                            los f_cpt svcplace_raw Bill_orig Allow_orig Reimb_orig
Patliab_orig);

  pnum="&pnum";
run;
title1 "MCDB Encounter Preprocessing";
title2 "January 2006 - April 2007 Data";
title2 "&pnum -- Check Bill Type, Delivery Systems and Financials";
title4 "Dataset is &pnum._ENC1";
proc freq data=enc1_ck;
  tables billtype*delivtyp*covtype / list missing;
  tables billtype*(bill_orig allow_orig reimb_orig patliab_orig) / list missing;
  format bill_orig allow_orig reimb_orig patliab_orig charge.
         delivtyp $delvsys.
         covtype $covtyp.;
run;

title3 "&pnum -- Financial Variables";
proc freq data = enc1_ck;
     tables billtype * f_cap_orig * bill_orig * allow_orig * reimb_orig * patliab_orig /
list missing;
     format bill_orig allow_orig reimb_orig patliab_orig finan.;
run;

title3 "&pnum -- Financial Variables, Billtype <> 8";
proc means data=enc1_ck n nmiss min max mean median sum;
  where billtype ne '8';
  class pnum covtype;
  var bill_orig allow_orig reimb_orig patliab_orig;
run;

*========================*;
* Print outliers if any  *;
*========================*;

proc sort data=enc1_ck;
  by pnum patid begindt enddt;
run;

data chkit;
  set enc1_ck (where=(billtype <> "8"));
  by pnum patid begindt enddt;

if (bill_orig >20000 or .Z<bill_orig <-20000) or
   (allow_orig >20000 or .Z<allow_orig <-20000) or
   (reimb_orig >20000 or .Z<reimb_orig <-20000) or
   (patliab_orig >20000 or .Z<patliab_orig <-20000) then output;
run;

title3 "&pnum -- Print of Outlier Financials, Excluding Records with BILLTYPE <> 8";
proc print data=chkit;
  by pnum;
  id pidbdg;
  var claimcn begindt enddt cpt bill_orig allow_orig reimb_orig patliab_orig;
run;

%jobend;

dm "log;    print file = I:\Mcdb\datacol\&pnum.\programs\&pnum._enc1_financials.log
replace";
dm "output; print file = I:\Mcdb\datacol\&pnum.\programs\&pnum._enc1_financials.lst
replace";

proc datasets kill;
run;

%jobstart;
```

```
*==================================================================================*
Program :  I:\MCdb\datacol\P020\programs\P020_prv1.sas
Project :  Maryland, Provider Preprocessing
Purpose :  1. Create 1st working file after deleting exact duplicates
           2. Variable processing:
              a. pad specialty codes with leading 0's if needed
              b. left justify all other character variables
              c. flag specialty codes as valid-nonmisc, valid-misc, invalid
                 missing
           3. Initial QC checks on name fields, specialty codes and business
              flagging
           4. 2nd try to link to Encounter
----------------------------------------------------------------------------------
input   : P020_prv_raw.sas7bdat
output  : P020_prv1_deletes.sas7bdat
          p020_prv1.sas7bdat
----------------------------------------------------------------------------------
by      : Laurie Hamilton / Sireesha Thati
Date    : 06/23/2005
modified: Sireesha Thati
Date    : 04/21/2006
----------------------------------------------------------------------------------
modified: Sanee Maphungphong
Date    : 08/09/2007
Note           : 1. Changed report year from 2005 to 2006.
                                    2. Added npi_raw as a char variable.
                                    3. Addes deanum and npi in freq list.
----------------------------------------------------------------------------------
modified: Sanee Maphungphong
Date    : 08/21/2007
Note           : WE RUN THIS PROGRAM TO EXECUTE THE ONLY THE LAST PART OF OF THE CODE
          WHICH CHECK THE LINK BETWEEN ENCOUNTER AND PROVIDER FILES (Quick 2nd Check
          on Link to Encounter)

          WE DO NOT WANT TO RECREATE "P020_PRV1.SAS7BDAT" AS THE ERROR IN THE CODE
          DOES NOT AFFECT THE DATASET CREATION PART OF THE PROGRAM. IF WE RECREATE
          "P020_PRV1.SAS7BDAT" AT THIS POINT, WE WILL HAVE TO RERUN THE ALL PROGRAMS
          IN STEP1 AND STEP2 OF THE "SPECIALTY ASSIGNMENT" PROCESS.

          WE ADDED A LINE TO SORT THE TEMPORARY DATASET "PROV_PF" TO KEEP ONE
          RECORD PER "PROVID".

          NEST YEAR, WE NEED TO IMPLEMENT THIS FIX TO PXXX_PRV1.SAS FOR ALL PAYERS.
----------------------------------------------------------------------------------
modified: Kasey Chen
Date    : 08/30/2007
Note    : Run 2nd PrvA submission.
*==================================================================================*;
options center obs = max pageno = 1 errors = 2 pagesize = 86 linesize = 123;

%let payernum = P020;
%let sbmssn   = 002;
%let drive    = I;
%let file     = a;
%let rptYr    = 2006;

libname out "&drive:\mcdb\datacol\&payernum.\sasdata";
libname in "J:\mcdb_raw";

title1 "Maryland Medical Care Database (MCDB)";
title2 "Year &rptYr Provider Data";
title3 "&payernum (&Sbmssn) - Prv1";
title4 " ";

footnote2 "program: &drive:\mcdb\datacol\&payernum\programs\&payernum._Prv1.sas";

proc format;
     value $smisc
        '034',
```

```
   '037',
   '038',
   '062',
   '085',
   '093' = '1'
   other = '0';

value $spectyp
   '001',
   '002',
   '003',
   '004',
   '005',
   '006',
   '007',

   '008',
   '009',
   '010',
   '011',
   '012',
   '013',
   '014',
   '015',
   '016',
   '017',
   '018',
   '019',
   '020',
   '021',
   '022',
   '023',

   '024',
   '025',
   '026',
   '027',
   '028',
   '029',
   '030',
   '031',
   '032',
   '033',
   '035',
   '036',
   '039',
   '100',
   '040',
   '041',

   '042',
   '043',
   '044',
   '045',
   '046',
   '047',
   '048',
   '049',
   '050',
   '051',
   '052',
   '053',
   '054',
   '055',
   '056',
   '057',

   '058',
   '059',
   '060',
   '061',
```

```
        '063',
        '070',
        '071',
        '072',
        '073',
        '074',
        '075',
        '076',
        '080',
        '081',
        '082',
        '083',

        '084',
        '090',
        '091',
        '092',
        '102',
        '101',
        '103' = 1 /* valid, not miscellaneous */
        '034',
        '037',
        '038',
        '062',
        '085',
        '093' = 2 /* valid, miscellaneous */
        ' '   = 4 /* missing */
        other = 3 /* invalid*/;

     value $ sflag
            1 = 'Valid, Not Misc'
            2 = 'Valid, Misc'
            3 = 'Invalid'
            4 = 'Missing';

     value $ miss
          ' ' = 'Missing'
        other = 'Present';
run;


/***********************

****************
Read Raw Data
****************;
data temp (keep = provid
             fedtaxid
             lname
             fname
             midinit

             suffix
             credent
             spec1
             spec2
             spec3
             sflag1
             sflag2
             sflag3
             spec1_raw
             spec2_raw
             spec3_raw
             deanum
             business
             npi
             pnum
             recnum
             file_id);
     set in.&payernum._prv_raw;
```

```
     length provid_raw
            provid      $13;

     length provid $ 13;
     provid = upcase(left(provid_raw));

     length fedtaxid $ 9;
     fedtaxid = upcase(left(fedtaxid_raw));

     length lname $ 31;
     lname = upcase(left(lname_raw));

     length fname $ 19;
     fname = upcase(left(fname_raw));

     length midinit $ 1;
     midinit = upcase(midinit_raw);

     length suffix $ 4;
     suffix = upcase(left(suffix_raw));

     length credent $ 5;
     credent = upcase(left(credent_raw));

     length spec1 $ 3;
     spec1 = upcase(right(spec1_raw));
     if spec1 ne ' ' then spec1 = translate(spec1,'0',' ');

     length spec2 $ 3;
     spec2 = upcase(right(spec2_raw));
     if spec2 ne ' ' then spec2 = translate(spec2,'0',' ');

     length spec3 $ 3;
     spec3 = upcase(right(spec3_raw));
     if spec3 ne ' ' then spec3 = translate(spec3,'0',' ');

     length deanum $ 11;
     deanum = upcase(left(deanum_raw));

     length business $ 1;
     business = upcase(business_raw);

     length npi $ 10;
     npi = upcase(npi_raw);

     *** validate specialty codes ***;
     array aspec(3) spec1-spec3;

     length sflag1-sflag3 $ 1;
     array aflag(3) sflag1-sflag3;

     do i = 1 to 3;
        aflag(i) = put(aspec(i),$spectyp.);
     end;

     label
      PROVID   = '1. Practitioner ID'
      FEDTAXID = '2. Practitioner Federal Tax ID'
      LNAME    = '3. Practitioner Last Name'
      FNAME    = '4. Practitioner First Name'
      SPEC1    = '5. Practitioner Specialty-1'
      SPEC2    = '6. Practitioner Specialty-2'
      SPEC3    = '7. Practitioner Specialty-3'
      DEANUM   = '8. Practitioner DEA Number'
      NPI      = '9. National Provider Identifier';

run;

***********************
Remove Exact Duplicates
```

```
***********************;
*** remove exact dups ***;
proc sort data = temp out = tempsort;
     by provid fedtaxid
        lname fname midinit suffix credent
        spec1 spec2 spec3
        deanum
        business;
run;

data in.&payernum._prv1_deletes
     out.&payernum._prv1;
     set tempsort;
     by provid fedtaxid
        lname fname midinit suffix credent
        spec1 spec2 spec3
        deanum
        business;
     if first.business then output out.&payernum._prv1;
     else
     output in.&payernum._prv1_deletes;
run;

proc contents data = out.&payernum._prv1; run;

*********************
Initial Checks
*********************;

title4 'Frequencies';
proc freq data = out.&payernum._prv1 compress;
     tables provid * fedtaxid * deanum/list missing;
     tables midinit/list missing;
     tables suffix/list missing;
     tables credent/list missing;
     tables spec1_raw/list missing;
     tables spec2_raw/list missing;
     tables spec3_raw/list missing;
     tables spec1/list missing;
     tables spec2/list missing;

     tables spec3/list missing;
     tables deanum/list missing;
     tables sflag1 * sflag2 * sflag3/list missing;
     tables business/list missing;
     tables npi/list missing;

     format provid
            fedtaxid
            deanum        $miss.
            sflag1-sflag3 $sflag.;
run;

proc summary data = out.&payernum._prv1 nway missing;
     where business = '0';
     class fname lname;
     output out = indivnames;
run;

title4 'Individual Names -- BUSINESS FLAG = 0';
proc print data = indivnames (obs = 60);
run;

proc summary data = out.&payernum._prv1 nway missing;
     where business = '1';
     class fname lname;
     output out = busnames;
run;

title4 'Business Name -- BUSINESS FLAG = 1';
proc print data = busnames (obs = 60);
```

```
run;

proc summary data = out.&payernum._prv1 nway missing;
     class fname lname;
     output out = anynames;
run;

title4 'Sample records -- 60 obs.';
proc print data = anynames (obs = 60);
run;

data dups_pf dups_pf2;
     set out.&payernum._prv1;
     by provid fedtaxid;
     if not (first.fedtaxid and last.fedtaxid) then do;
        if provid='00000000000' and fedtaxid='000000000' then output dups_pf;
        else output dups_pf2;
     end;
run;

title4 'Duplicates by Provid and Fedtaxid - 1st 60';
title5 'Both PROVID and FEDTAXID are coded as zeros';
proc print data = dups_pf (obs = 60);
     by provid fedtaxid;
     id provid fedtaxid;
     var fname lname midinit suffix credent spec1-spec3;
run;


title4 'Duplicates by Provid and Fedtaxid - 1st 60';
title5 'PROVID and/or FEDTAXID are not coded as zeros';
proc print data = dups_pf2 (obs = 60);
     by provid fedtaxid;
     id provid fedtaxid;
     var fname lname midinit suffix credent spec1-spec3;
run;


data dups_p;
     set out.&payernum._prv1;
     by provid;
     if not (first.provid and last.provid) then output;
run;

title4 'Duplicates by Provid - 1st 60';
title5 'PROVID are not coded as zeros';
proc print data = dups_pf (obs = 60);
  where provid ne '00000000000';
     by provid;
     id provid;
     var fname lname midinit suffix credent fedtaxid spec1-spec3;
run;

**************************/



***************************
Quick 2nd Check on Link
to Encounter
***************************;

*** unique list from provider file, by provid fedtaxid ***;
proc summary data = out.&payernum._prv1 nway;
     class provid fedtaxid;
     output out = prov_pf (keep = provid fedtaxid);
run;

*** unique list from encounter file, by provid fedtaxid ***;
proc summary data = in.&payernum._encprov nway missing;
     class provid fedtaxid;
```

```
     output out = enc_all (drop = _type_
                           rename = (_freq_ = numserv)
                           );
run;


*** check by provid fedtaxid ***;

data enc_ok_pf
     enc_check_p;
     merge prov_pf (in = inprov)
           enc_all (in = inenc);
     by provid fedtaxid;
     length prov_match $ 2;
     if inenc then do;
        if inprov then do;
           prov_match = 'pf';
           output enc_ok_pf;
        end;
        else output enc_check_p;
     end;
run;


*** check remainder - link by provid only ***;

proc sort data = prov_pf (keep=provid) nodupkey;
  by provid;
run;

data enc_ok_p
     enc_ok_not;
     merge prov_pf     (in = inprov)
           enc_check_p (in = inenc);
     by provid;

     length prov_match $ 2;

     if inenc then do;
        if inprov then do;
           prov_match = 'p';
           output enc_ok_p;
        end;
        else do;
             prov_match = ' ';
             output enc_ok_not;
        end;
     end;
run;

*** combine ***;

data encmatch;
     set enc_ok_pf
         enc_ok_p
         enc_ok_not;
run;

title4 '2nd Attempt to Link to Encounter';
proc freq data = encmatch;
     tables prov_match /list missing;
run;

proc sort data = encmatch;
     by prov_match;
run;

proc means data = encmatch
           noprint
           sum;
     by prov_match;
```

```
     var numserv;
     output out = match_stat
            sum = numserv;
run;

proc print data = match_stat;
run;

%jobend;

dm "log;    print file = I:\mcdb\datacol\&payernum.\programs\&payernum._prv1.log
replace";
dm "output; print file = I:\mcdb\datacol\&payernum.\programs\&payernum._prv1.lst
replace";


proc datasets kill;
run;

%jobstart;


*=================================================================*
Program  : I:\mcdb\datacol\pxxx\reports\pxxx_enc1_codebook_crunch.sas
Purpose  : To calculate frequency counts for codebook
-------------------------------------------------------------------
Input    : pxxx_enc1.sas7bdat
Output   : pxxx_enc1_codebook_crunch.sas7bdat
-------------------------------------------------------------------
By       : Sireesha Thati
Date     : June 14, 2005

Modified : Po-Lun Chou
Date     : July 11, 2005

Modified : Sireesha Thati
Date     : June 21, 2006 for 2005 MCDB
-------------------------------------------------------------------
Modified : Kasey chen
Date     : 09/05/2007 for 2006 MCDB
*=================================================================*
;

options linesize = 165 pagesize = 68 errors = 2 nomprint;

%let payer = P070;
%let drive = I;
%let curryr= 06;

title1 "MCDB Encounter Preprocessing";
title2 "January 2006 - April 2007 Data";
title3 "Code Book(Source: Enc1 Dataset)";
title4 "&payer";

footnote1 "program:
&drive:\mcdb\datacol\&payer.\reports\&payer._enc1_codebook_crunch.sas";

libname in "&drive:\mcdb\datacol\&payer.\sasdata";
libname out "&drive:\mcdb\datacol\&payer.\sasdata\codebook";
*************************************************************************
Note:        icd9dxfmt.sas is the format that is used for the Codebook Processing
                                            This program is created by copying
icd9dx_06 and changing
                                            1 to valid, other to invalid and missing to
blank
*************************************************************************
Note : Changed 'codebookfmt.sas'
               1. $srvunit '3' from 'visits' to 'services'
               2. Added '8' in this format list.
*************************************************************************;
```

```
********************************
Re-set Blank or Missing Value
********************************
;

/** force blank or missing value to the bottom of the frequency tables **/

proc format;
     value blank_n
           . = 9998;
     value blank_nm
           .m = 9998;
     value blank_na
           . = 0.5;
     value $blank_a
           ' ' = 'Y';
     value $blank_b
           ' ' = 'ZY';
     value $blank_c
           ' ' = 'ZZY';
     value $blank_d
           ' ' = 'ZZZY';
     value $blank_e
           ' ' = 'ZZZZY';
     value $blank_f
           ' ' = 'ZZZZZY';
     value $blank_g
           ' ' = 'ZZZZZZZZY';
     value $blank_h
           ' ' = 'ZZZZZZZZZZZZY';
     value $blank_i

           ' ' = 'ZZZZZZZZZZZZZZZZZZZZZZZY';
     value $blank_j
           ' ' = 'ZZZZZZZZZZZZZZZZZZZZZZZZZY';
run;

data enc1;
     set in.&payer_enc1;
     age          = put(age, blank_nm.);
     age1205      = put(age1205, blank_nm.);
     begday       = put(begday, $blank_b.);
     begindt      = put(begindt, blank_n.);
     begmo        = put(begmo, $blank_b.);
     begyr        = put(begyr, $blank_d.);
     betosagg     = put(betosagg, $blank_a.);
     betosdet     = put(betosdet, $blank_c.);
     billtype     = put(billtype, $blank_a.);
     birthmo      = put(birthmo, $blank_b.);
     birthyr      = put(birthyr, $blank_d.);
     cdhpind      = put(cdhpind, $blank_a.);
     claimcn      = put(claimcn, $blank_i.);
     claimpdd     = put(claimpdd, $blank_b.);
     claimpmo     = put(claimpmo, $blank_b.);
     claimpyr     = put(claimpyr, $blank_d.);
     clmpdate     = put(clmpdate, blank_n.);
     clmrelco     = put(clmrelco, $blank_a.);
     cnty_p2      = put(cnty_p2, $blank_c.);
     county_p     = put(county_p, $blank_c.);
     covothr      = put(covothr, $blank_a.);
     covtype      = put(covtype, $blank_a.);
     cpt          = put(cpt, $blank_f.);
     delivtyp     = put(delivtyp, $blank_a.);
     dx1          = put(dx1, $blank_e.);
     dx2          = put(dx2, $blank_e.);
     dx3          = put(dx3, $blank_e.);
     dx4          = put(dx4, $blank_e.);
     dx5          = put(dx5, $blank_e.);
     dx6          = put(dx6, $blank_e.);
     dx7          = put(dx7, $blank_e.);
```

```
     dx8          = put(dx8, $blank_e.);
     dx9          = put(dx9, $blank_e.);
     dx10         = put(dx10, $blank_e.);
     endday       = put(endday, $blank_b.);
     enddt        = put(enddt, blank_n.);
     endmo        = put(endmo, $blank_b.);
     endyr        = put(endyr, $blank_d.);
     fedtaxid     = put(fedtaxid, $blank_g.);
     instate      = put(instate, $blank_a.);
     los          = put(los, blank_na.);
     mod1         = put(mod1, $blank_b.);
     mod2         = put(mod2, $blank_b.);
     newbetag     = put(newbetag, $blank_a.);
     newbetdt     = put(newbetdt, $blank_c.);
     newmod       = put(newmod, $blank_b.);
     numdxc       = put(numdxc, blank_n.);
     partprov     = put(partprov, $blank_a.);
     payertyp     = put(payertyp, $blank_a.);
     pidbdg       = put(pidbdg, $blank_j.);
     plan         = put(plan, $blank_f.);
     plan2        = put(plan2, $blank_f.);
     provid       = put(provid, $blank_h.);
     sex          = put(sex, $blank_a.);
     svcplace     = put(svcplace, $blank_b.);
     unitind      = put(unitind, $blank_a.);
     units        = put(units, blank_na.);
     zip_p        = put(zip_p, $blank_e.);
     zip_s        = put(zip_s, $blank_e.);
     enrolldt     = put(enrolldt, blank_n.);
     disenrolldt  = put(disenrolldt, blank_n.);
run;


**********************
Get Frequency Table
**********************
;

proc freq data = enc1 order = formatted noprint;
     tables age           / list missing outcum out = age;
     tables age1206       / list missing outcum out = age1206;
     tables begday        / list missing outcum out = begday;
     tables begindt       / list missing outcum out = begindt;
     tables begmo         / list missing outcum out = begmo;
     tables begyr         / list missing outcum out = begyr;
     tables betosagg      / list missing outcum out = betosagg;
     tables betosdet      / list missing outcum out = betosdet;
     tables billtype      / list missing outcum out = billtype;
     tables birthmo       / list missing outcum out = birthmo;
     tables birthyr       / list missing outcum out = birthyr;
     tables cdhpind       / list missing outcum out = cdhpind;
     tables claimcn       / list missing outcum out = claimcn;
     tables claimpdd      / list missing outcum out = claimpdd;
     tables claimpmo      / list missing outcum out = claimpmo;
     tables claimpyr      / list missing outcum out = claimpyr;
     tables clmpdate      / list missing outcum out = clmpdate;
     tables clmrelco      / list missing outcum out = clmrelco;
     tables cnty_p2       / list missing outcum out = cnty_p2;
     tables county_p      / list missing outcum out = county_p;
     tables covothr       / list missing outcum out = covothr;
     tables covtype       / list missing outcum out = covtype;
     tables cpt           / list missing outcum out = cpt;
     tables delivtyp      / list missing outcum out = delivtyp;
     tables dx1           / list missing outcum out = dx1;
     tables dx2           / list missing outcum out = dx2;
     tables dx3           / list missing outcum out = dx3;
     tables dx4           / list missing outcum out = dx4;
     tables dx5           / list missing outcum out = dx5;
     tables dx6           / list missing outcum out = dx6;
     tables dx7           / list missing outcum out = dx7;
     tables dx8           / list missing outcum out = dx8;
```

```
tables dx9           / list missing outcum out = dx9;
tables dx10          / list missing outcum out = dx10;
tables endday        / list missing outcum out = endday;
tables enddt         / list missing outcum out = enddt;
tables endmo         / list missing outcum out = endmo;
tables endyr         / list missing outcum out = endyr;
tables fedtaxid      / list missing outcum out = fedtaxid;
tables file_id       / list missing outcum out = file_id;
tables f_baddt       / list missing outcum out = f_baddt;
tables f_cap_orig    / list missing outcum out = f_cap_orig;
tables f_cpt         / list missing outcum out = f_cpt;
tables f_demog       / list missing outcum out = f_demog;
tables f_insamp      / list missing outcum out = f_insamp;
tables f_yr07        / list missing outcum out = f_yr07;
tables instate       / list missing outcum out = instate;
tables los           / list missing outcum out = los;
tables mod1          / list missing outcum out = mod1;
tables mod2          / list missing outcum out = mod2;
tables newbetag      / list missing outcum out = newbetag;
tables newbetdt      / list missing outcum out = newbetdt;
tables newmod        / list missing outcum out = newmod;
tables numdxc        / list missing outcum out = numdxc;
tables partprov      / list missing outcum out = partprov;
tables payertyp      / list missing outcum out = payertyp;
tables pidbdg        / list missing outcum out = pidbdg;
tables plan          / list missing outcum out = plan;
tables plan2         / list missing outcum out = plan2;
tables pnum          / list missing outcum out = pnum;
tables provid        / list missing outcum out = provid;
tables sex           / list missing outcum out = sex;
tables svcplace      / list missing outcum out = svcplace;
tables unitind       / list missing outcum out = unitind;
tables units         / list missing outcum out = units;
tables zip_p         / list missing outcum out = zip_p;
tables zip_s         / list missing outcum out = zip_s;
tables enrolldt      / list missing outcum out = enrolldt;
tables disenrolldt   / list missing outcum out = disenrolldt;


format age           agecat.
       age1206       agecat.
       begday        $dayfmt.
       begindt       dtfmt.
       begmo         $monfmt.
       begyr         $missyr.
       betosagg      $betosa.
       betosdet      $betosd.
       billtype      $btype.
       birthmo       $monfmt.
       birthyr       $byr.
       cdhpind       $cdhpind.
       claimcn       $unk_c.
       claimpdd      $dayfmt.
       claimpmo      $monfmt.
       claimpyr      $missyr.
       clmpdate      dtfmt.
       clmrelco      $claimrc.
       cnty_p2       $cyfmt.
       county_p      $cyfmt.
       covothr       $othinsu.
       covtype       $covrtyp.
       cpt           $rmiss_b.
       delivtyp      $delvsys.
       dx1           $icd9dx.
       dx2           $icd9dx.
       dx3           $icd9dx.
       dx4           $icd9dx.
       dx5           $icd9dx.
       dx6           $icd9dx.
       dx7           $icd9dx.
       dx8           $icd9dx.
```

```
            dx9           $icd9dx.
            dx10          $icd9dx.
            endday        $dayfmt.
            enddt         dtfmt.
            endmo         $monfmt.
            endyr         $missyr.
            fedtaxid      $unk_a.
            file_id       $file_id.
            f_baddt       $bddt.
            f_cap_orig    $fcap.
            f_cpt         $f_proc.
            f_demog       $demog.
            f_insamp      $insamp.
            f_yr07        $fyr_b.
            instate       $instate.
            los           los.
            mod1          $modf.
            mod2          $modf.
            newbetag      $betosa.
            newbetdt      $betosd.
            newmod        $modf.
            numdxc        miss.
            partprov      $provflg.
            payertyp      $paygp.
            pidbdg        $unk_d.
            plan          $plan.
            plan2         $plan_b.
            pnum          $pnum.
            provid        $unk_b.
            sex           $sex.
            svcplace      $srvplac.
            unitind       $srvunit.
            units         units.
            zip_p         $rmiss_a.
            zip_s         $rmiss_a.
            enrolldt      dtfmt.
            disenrolldt   dtfmt.;
run;

************************************
Create Standard Dataset Structure
************************************
;

%macro freq (var =, fmt =);

data use_&var (keep = varname code explcode count percent cum_freq cum_pct order);
     set &var;
     length VARNAME $10 VALUE $100 CODE $30 EXPLCODE $60 ORDER 3;
     varname = upcase("&var");
     value = (put(&var,&fmt.));
     code = scan(value, 1, "?");
     explcode = scan(value, 2, "?");
     order = scan(value, 3, "?");
run;

proc append base = freq_1 data = use_&var;
run;

%mend;

%freq (var = age,           fmt = agecat.);
%freq (var = age1206,       fmt = agecat.);
%freq (var = begday,        fmt = $dayfmt.);
%freq (var = begindt,       fmt = dtfmt.);
%freq (var = begmo,         fmt = $monfmt.);
%freq (var = begyr,         fmt = $missyr.);
%freq (var = betosagg,      fmt = $betosa.);
%freq (var = betosdet,      fmt = $betosd.);
%freq (var = billtype,      fmt = $btype.);
%freq (var = birthmo,       fmt = $monfmt.);
```

```
%freq (var = birthyr,       fmt = $byr.);
%freq (var = cdhpind ,      fmt = $cdhpind.);
%freq (var = claimcn,       fmt = $unk_c.);
%freq (var = claimpdd,      fmt = $dayfmt.);
%freq (var = claimpmo,      fmt = $monfmt.);
%freq (var = claimpyr,      fmt = $missyr.);
%freq (var = clmpdate,      fmt = dtfmt.);
%freq (var = clmrelco,      fmt = $claimrc.);
%freq (var = cnty_p2,       fmt = $cyfmt.);
%freq (var = county_p,      fmt = $cyfmt.);
%freq (var = covothr,       fmt = $othinsu.);
%freq (var = covtype,       fmt = $covrtyp.);
%freq (var = cpt,           fmt = $rmiss_b.);
%freq (var = delivtyp,      fmt = $delvsys.);
%freq (var = dx1,           fmt = $icd9dx.);
%freq (var = dx2,           fmt = $icd9dx.);
%freq (var = dx3,           fmt = $icd9dx.);
%freq (var = dx4,           fmt = $icd9dx.);
%freq (var = dx5,           fmt = $icd9dx.);
%freq (var = dx6,           fmt = $icd9dx.);
%freq (var = dx7,           fmt = $icd9dx.);
%freq (var = dx8,           fmt = $icd9dx.);
%freq (var = dx9,           fmt = $icd9dx.);
%freq (var = dx10,          fmt = $icd9dx.);
%freq (var = endday,        fmt = $dayfmt.);
%freq (var = enddt,         fmt = dtfmt.);
%freq (var = endmo,         fmt = $monfmt.);
%freq (var = endyr,         fmt = $missyr.);
%freq (var = fedtaxid,      fmt = $unk_a.);
%freq (var = file_id,       fmt = $file_id.);
%freq (var = f_baddt,       fmt = $bddt.);
%freq (var = f_cap_orig,    fmt = $fcap.);
%freq (var = f_cpt,         fmt = $f_proc.);
%freq (var = f_demog,       fmt = $demog.);
%freq (var = f_insamp,      fmt = $insamp.);
%freq (var = f_yr07,        fmt = $fyr_b.);
%freq (var = instate,       fmt = $instate.);
%freq (var = los,           fmt = los.);
%freq (var = mod1,          fmt = $modf.);
%freq (var = mod2,          fmt = $modf.);
%freq (var = newbetag,      fmt = $betosa.);
%freq (var = newbetdt,      fmt = $betosd.);
%freq (var = newmod,        fmt = $modf.);
%freq (var = numdxc,        fmt = miss.);
%freq (var = partprov,      fmt = $provflg.);
%freq (var = payertyp,      fmt = $paygp.);
%freq (var = pidbdg,        fmt = $unk_d.);
%freq (var = plan,          fmt = $plan.);
%freq (var = plan2,         fmt = $plan_b.);
%freq (var = pnum,          fmt = $pnum.);
%freq (var = provid,        fmt = $unk_b.);
%freq (var = sex,           fmt = $sex.);
%freq (var = svcplace,      fmt = $srvplac.);
%freq (var = unitind,       fmt = $srvunit.);
%freq (var = units,         fmt = units.);
%freq (var = zip_p,         fmt = $rmiss_a.);
%freq (var = zip_s,         fmt = $rmiss_a.);
%freq (var = enrolldt,      fmt = dtfmt.);
%freq (var = disenrolldt,   fmt = dtfmt.);

******************************
Create Final Codebook Table
******************************
;

proc contents data = in.&payer_enc1 out = enc1 noprint;
run;

data cont_1 (rename = (NAME = VARNAME));
     length NAME $10 LABEL $60;
     set enc1 (keep = name label);
```

```
     name = upcase(name);
run;

proc sort data = cont_1;
     by varname ;
run;

proc sort data = freq_1;
     by varname;
run;

data &payer._enc1_codebook_crunch;
     merge freq_1 (in = infreq)
           cont_1 (in = incont);
     by varname;
     code = left(code);
     explcode = left(explcode);
     if varname in ("BEGYR", "BIRTHYR", "CLAIMPYR", "ENDYR") then do;
        explcode = "Year = " || code;
     end;
     if varname in ("BEGMO", "BIRTHMO", "CLAIMPMO", "ENDMO") then do;
        if code not in ("01 thru 12", " ") then explcode = "Month = " || code;
     end;
     if varname in ("BEGDAY", "BIRTHDD", "CLAIMPDD", "ENDDAY") then do;
        if code not in ("01 thru 31", " ") then explcode = "Day = " || code;
     end;
     if code = "ZZZY(Blank)" then code = "Blank";
     else if code = "ZZZY(Missing)" then code = "Missing";
     else if code = "ZZZZ(Invalid)" then code = "Invalid";
     else if code = "ZZZZ(Other)" then code = "Other";
     if infreq then output;
     label code     = "Variable Value"
           explcode = "Value Description";
run;

proc sort data = &payer._enc1_codebook_crunch
          out = tempsort (drop = order)
;

     by varname label order code;
run;

data out.&payer._enc1_codebook_crunch;
     set tempsort;
     by varname label;

     if not (first.varname) then varname = ' ';
     if not (first.label) then label = ' ';
run;

proc print data = out.&payer._enc1_codebook_crunch;
     var varname label code explcode count percent cum_freq cum_pct;
run;

proc contents data = out.&payer._enc1_codebook_crunch;
run;


%jobend;

dm "log;    print file = I:\mcdb\datacol\&payer.\reports\&payer._enc1_codebook_crunch.log
replace";
dm "output; print file = I:\mcdb\datacol\&payer.\reports\&payer._enc1_codebook_crunch.lst
replace";


*****************************************************************************
Program  : I:\mcdb\datacol\pxxx\reports\pxxx_enc1_codebook_0506.sas
Purpose  : To compare the services of 2006 MCDB with 2005 MCDB
-----------------------------------------------------------------------------
Input    : pxxx_enc1_codebook_crunch.sas7bdat
```

```
                      pxxx_enc1_codebook_crunch_05.sas7bdat
Output   : pxxx_enc1_codebook_0506.sas7bdat
---------------------------------------------------------------------------
By       : Sireesha Thati
Date     : June 22, 2006
---------------------------------------------------------------------------
By       : Kasey Chen
Date     : 09/05/2007
Note     : 1. Changed service year from 0405 to 0506;
           2. Modified 'payer' to a macro variable.
***************************************************************************;

options linesize = 165 pagesize = 68 errors = 2 nomprint;

%let payer  = P070;
%let drive  = I;
%let curryr = 06;
%let lastyr = 05;

title1 "MCDB Encounter Preprocessing";
title2 "January 2005 - April 2006 Data";
title3 "Code Book(Source: Enc1 Dataset)";
title4 "&payer";

footnote1 "program:
&drive:\mcdb\datacol\&payer.\reports\&payer._enc1_codebook_&lastyr.&curryr..sas";

libname in "I:\MCdb\datacol\&payer\sasdata\codebook";
libname out "I:\MCdb\datacol\&payer\sasdata\codebook";

data cb_&curryr.(drop = varname label rename = (count = count&curryr. percent =
percent&curryr.));
  set in.&payer._enc1_codebook_crunch (keep = code count explcode varname percent label);
  retain dvarname dlabel;
  if varname ne ' ' then dvarname = varname;
  if label ne ' ' then dlabel = label;
run;

data cb_&lastyr.(drop = varname label rename = (count = count&lastyr. percent =
percent&lastyr.));
  set in.&payer._enc1_codebook_crunch_05 (keep = code count explcode percent varname
label);
  retain dvarname dlabel;
  if varname ne ' ' then dvarname = varname;
  if label ne ' ' then dlabel = label;
run;

proc sort data = cb_&curryr.;
  by dvarname code;
run;
proc sort data = cb_&lastyr.;
  by dvarname code;
run;

data cb_&lastyr.&curryr.;
  merge cb_&curryr. cb_&lastyr.;
  by dvarname code;
  per_chg = (count&curryr. - count&lastyr.)/count&lastyr.*100;
run;

proc sort data = cb_&lastyr.&curryr.;
     by dvarname dlabel code;
run;

data out.&payer._enc1_codebook_&lastyr.&curryr.;
     set cb_&lastyr.&curryr.;
     by dvarname dlabel;
     if not (first.dvarname) then dvarname = ' ';
     if not (first.dlabel) then dlabel = ' ';
run;
```

```
proc print data = out.&payer._enc1_codebook_&lastyr.&curryr.;
  var dvarname dlabel code explcode count&curryr. percent&curryr. count&lastyr.
percent&lastyr. per_chg;
run;

proc contents data = out.&payer._enc1_codebook_&lastyr.&curryr.;
run;


dm "log;    print file = I:\mcdb\datacol\&payer.\reports\&payer._enc1_codebook_0506.log
replace";
dm "output; print file = I:\mcdb\datacol\&payer.\reports\&payer._enc1_codebook_0506.lst
replace";

*%jobstart;

*******************************************************************
Program  : pxxx_enc1_codebook_xls.sas
Project  : Maryland
Purpose  : For the codebook from the Enc1 Data, create an Excel
           spreadsheet from the crunched sas dataset

Notes    : 1. This program includes both the macro to create the spreadsheet
               and the macro call
           2. Programmer change payer, titles, filenames, and literals for
               titles in macro call at the end of the program.
-------------------------------------------------------------------

Input    : pxxx_enc1_codebook_0506.sas7bdat
Output   : pxxx_enc1_codebook.xls

--------------------------------------------------------------------
Date     : 07/06/2005
By       : Raymond Hu
Modified : 07/22/2005
           Laurie Hamilton
           1. Delete global macros and move to macro call at end
               of program
           2. Send complete titles to spreadsheet

Modified : 06/21/2006
                 Sireesha Thati

Modified : 09/05/2007
                 Kasey Chen
Note     : 1. Changed years from '0405' to '0506'
           2. Changed codebook creating date (fmonday and fyear).
*******************************************************************
;
*options mprint
        mlogic
        symbolgen
;

%let payer   = P070;
%let pname   = American Republic Insurance Co;
%let elserv  = 29969;
%let subserv = 30124
;
%let fmonday = september 05
;
%let fyear   = 2007
;

libname in "i:\mcdb\datacol\&payer.\sasdata\codebook";

********************************
Format to handle fix
for contents of variable
value column
```

```
1 = Use contents of EXPLODE
2 = Concatenate code and EXPLCODE
    with exception of Missings
3 = Use contents of CODE
********************************;
;
proc format;
     value $exptype
     'AGE     '  = '1'
        'AGE1204 '  = '1'
     'AGE1205 '  = '1'
     'BEGDAY  '  = '1'
     'BEGINDT '  = '1'
     'BEGMO   '  = '1'
     'BEGYR   '  = '1'
     'BETOSAGG'  = '2'
     'BETOSDET'  = '2'
     'BILLTYPE'  = '2'
     'BIRTHMO '  = '1'
     'BIRTHYR '  = '1'
     'CDHPIND '  = '2'
     'CLAIMCN '  = '1'
     'CLAIMPDD'  = '1'
     'CLAIMPMO'  = '1'
     'CLAIMPYR'  = '1'
     'CLMPDATE'  = '1'
     'CLMRELCO'  = '2'
     'CNTY_P2 '  = '2'
     'COUNTY_P'  = '2'
     'COVOTHR '  = '2'
     'COVTYPE '  = '2'
     'CPT     '  = '1'
     'DELIVTYP'  = '2'
     'DISENROLLD'= '1'
     'DX1     '  = '1'
     'DX10    '  = '1'
     'DX2     '  = '1'
     'DX3     '  = '1'
     'DX4     '  = '1'
     'DX5     '  = '1'
     'DX6     '  = '1'
     'DX7     '  = '1'
     'DX8     '  = '1'
     'DX9     '  = '1'
     'ENDDAY  '  = '1'
     'ENDDT   '  = '1'
     'ENDMO   '  = '1'
     'ENDYR   '  = '1'
     'ENROLLDT'  = '1'
     'FEDTAXID'  = '1'
     'FILE_ID '  = '2'
     'F_BADDT '  = '2'
     'F_CAP_ORIG'= '2'
     'F_CPT   '  = '2'
     'F_DEMOG '  = '2'
     'F_INSAMP'  = '2'
        'F_YR05  '  = '2'
        'F_YR06  '  = '2'
     'INSTATE '  = '2'
     'LOS     '  = '3'
     'MOD1    '  = '2'
     'MOD2    '  = '2'
     'NEWBETAG'  = '2'
     'NEWBETDT'  = '2'
     'NEWMOD  '  = '2'
     'NUMDXC  '  = '1'
     'PARTPROV'  = '2'
     'PAYERTYP'  = '2'
     'PIDBDG  '  = '1'
     'PLAN    '  = '2'
     'PLAN2   '  = '2'
```

```
     'PNUM    '  = '2'
     'PROVID  '  = '1'
     'SEX     '  = '2'
     'SVCPLACE'  = '2'
     'UNITIND '  = '2'
     'UNITS   '  = '3'
     'ZIP_P   '  = '1'
     'ZIP_S   '  = '1'
;
run;

*************************
Adjust Code Descriptions

Note: Next year, do
this using different
formats in the crunch
program
************************
;
data temp;
     set in.&payer._enc1_codebook_0405;

     length holdname $ 12;
     retain holdname;

     if dvarname ne ' ' then holdname = dvarname;

     length value $ 70;

     valtype = put(holdname,$exptype.);

     if valtype = '1' then value = explcode;
     else
     if valtype = '2' then do;
        if code in ('Missing', 'Blank', 'Invalid') then value = explcode;
        else value = trim(code) || ':' || explcode;
     end;
     else
     if valtype = '3' then value = code;

     *** fixes to betos descriptions, will be done through crunch
         format next year **;
     if holdname in ('BETOSDET', 'NEWBETDT') then do;
        value = tranwrd(value," - ","-");
        value = tranwrd(value,"Gastrointestinal","GI");
        value = tranwrd(value,"Procedure, ","Proc,");
        value = tranwrd(value,"Catheterization","Cath");
        value = tranwrd(value,"Procedures","Procs");
        value = tranwrd(value,", ",",");
        value = tranwrd(value,"Imaging/Procedure","Imaging/Proc");
        value = tranwrd(value,"Cardiovascular","Cardiovasc");
        value = tranwrd(value,"Proc, ","Proc,");
        value = tranwrd(value,"Medicare Fee Schedule","MCR Fee Sched");
        value = tranwrd(value,"Venipuncture","VP");
     end;

     *if dvarname in ('F_YR06', 'F_YR05') then delete;
     *else output;
run;

*proc print data = temp;
*     var varname code explcode value;
*run;

*proc print data = temp;
*     where holdname in ("BETOSDET", "NEWBETDT");
*     var holdname value;
*run;

********************
```

```
Spreadsheet Macro
*********************
;
%macro Create_Xls_Table(ThisYear              =,
                        SASDataLib            =,
                        SASDataName           =,
                        pnum                  =,
                        pname                 =,
                        elserv                =,
                        subserv               =,
                        text8b                =,
                        fmonday               =,
                        fyear                 =,
                        ExcelOutPathName      =
                        )
;
   %***put  &SASDataLib &SASDataName;

   options noxwait noxsync;

   ******************************************************************
      1.       Open Microsoft Excel
   ******************************************************************
;
   *** excel location on maryland room computer ***;
   x "C:\Progra~1\Micros~2\Office10\EXCEL.EXE ";


   ** Sleep to allow time for Excel to come up **;
   data _null_;
     x=sleep(3);
   run;

   ******************************************************************
     2.        Open another Excel worksheet
   ******************************************************************
;
   %let cmopen2='[OPEN("i:\mcdb\datacol\include_program\enc1_codebook_shell.xls")]';

   %put &cmopen2;

   filename cmds dde 'excel|system';

   data _null_;
     file cmds;
     put &cmopen2;
   run;

   data _null_;
     x=sleep(3);
   run;

   ******************************************************************
     3.        Write data to Excel by DDE
   ******************************************************************
;

   ******************************************************************
     3.1       Write titles to Excel by DDE
   ******************************************************************
;
   %let
cmxcpt='Excel|i:\mcdb\datacol\include_program\[enc1_codebook_shell.xls]sheet1!R4C1:R9C18'
;

   ** Establish the DDE link **;
   filename caseman dde &cmxcpt notab;

   data _null_;
      file caseman ;
```

```
      *** concatenate payer number and payer name ***;
      pnumname = "&pnum" || ": " || "&pname";

      *** create text for Eligible Services line ***;
      text8a = "Eligible Services    :" || put(&elserv,comma12.);

      *** create text for Submitted Services line ***;
      text9a = "Submitted Services:" || put(&subserv,comma12.);

      *** create text for File Date line ***;
      text9b = "File Date    : " || "&fmonday" || ', ' || "&fyear";

      put pnumname;
      put "Codebook - Data After Initial Processing Step(Enc1)";
      put "Non-MD Patients are not included in the 2004 Service Counts";
      put;
      put text8a '09'x '09'x '09'x "&text8b." ;
      put text9a '09'x '09'x '09'x text9b ;
   run;

   ******************************************************************
     3.2       Write data to Excel by DDE
   ******************************************************************
;
   %let
cmxcpt='Excel|i:\mcdb\datacol\include_program\[enc1_codebook_shell.xls]sheet1!R12C1:R2000
C18';

   ** Establish the DDE link **;
   filename caseman dde &cmxcpt notab;


   data _null_;
    file caseman ;
    set temp end=eof;

       put dvarname    '09'x
           dlabel      '09'x
           value       '09'x
           count04      comma12.0 '09'x
           percent04    comma8.1  '09'x
           count05      comma12.0 '09'x
           percent05    comma8.1  '09'x
           per_chg      comma8.1  '09'x;
   run;

   ******************************************************************
     5.        Write footnote to Excel by DDE
   ******************************************************************
;
   %let
cmxcpt='Excel|i:\mcdb\datacol\include_program\[enc1_codebook_shell.xls]sheet1!R10C1:R2000
C18';

   ** Establish the DDE link **;
   filename caseman dde &cmxcpt notab;

   data _null_;
    file caseman ;
    **put "&foot1";
   run;

   *********************************************************
     6.  Save as and close Xls
   *********************************************************
;
   %let rc3 = %sysfunc(fileexist("&exceloutPathName"));
   data _null_;
     if &rc3 = 1 then do; x " del &exceloutPathName"; end;
   run;
```

```
   data _null_;
     x=sleep(3);
   run;

   data _null_;
     file cmds;

     put "[SAVE.AS(%bquote("&exceloutPathName"))]";
     put '[FILE.CLOSE()]';
     put '[QUIT()]';
   run;

%mend;

*********************
Create Spreadsheet
*********************
;

%Create_Xls_Table(ThisYear         = 2005,
                  pnum             = %str(&payer),
                  pname            = %str(American Republic Insurance Co),
                  elserv           = %str(&elserv),
                  subserv          = %str(&subserv),
                  text8b           = %str(Source File: p070_enc1_codebook_0405.sas7bdat),
                  fmonday          = %str(&fmonday),
                  fyear            = %str(&fyear),
                  ExceloutPathName =
%str(i:\mcdb\datacol\&payer.\reports\&payer._enc1_codebook.xls)
                  );

*%jobend;

dm "log;    print file = I:\mcdb\datacol\&payer.\reports\&payer._enc1_codebook_xls.log
replace";
dm "output; print file = I:\mcdb\datacol\&payer.\reports\&payer._enc1_codebook_xls.lst
replace";


%jobstart;
proc datasets kill;
run;


*=============================================================================================
===
Program   :I:\mcdb\datacol\pxxx\reports\pxxx_enc1_data_completeness.sas.
Notes     :For recipient counts using Proc Summary, do NOT include the Missing option.
           We do not want to consider records with a missing patient ID as a known
recipient.
---------------------------------------------------------------------------------------------
---
input     :pxxx_enc1.sas7bdat
output    :delivtp_pxxx_20xx.sas7bdat
           covtype_pxxx_20xx.sas7bdat
           pxxx_enc1_dc_crunch.sas7bdat (for 2005 version: final_pxxx_20xx.sas7bdat).
           plan_pxxx_20xx.sas7bdat
           plan2_pxxx_20xx.sas7bdat
           totrecip_20xx.sas7bdat
---------------------------------------------------------------------------------------------
--
Programmer:Laurie Hamilton
Date      :June 22, 2005
Revised   :Sireesha Thati
Date      :June 20, 2006
purpose   :For 2005 MCDB

---------------------------------------------------------------------------------------------
---
Programmer: Kasey Chen
Date      : September 10, 2007
```

```
Purpose   : Generate data Completeness Report on 2007 unedited data.
Note      : 1. Changed last year from 04 to 05 and this year from 05 to 06.
            2. Changed 'libname stats&lastyear' from C:\ to I:\.
            3. Changed Payer number in libname to a marco variable '&payer'.
**=================================================================================
===
;

options obs = max        errors = 2;

%let thisyear = 06;
%let lastyear = 05;
%let payer = P020;
%let drive = I;

libname in "&drive:\mcdb\datacol\&payer.\sasdata";
libname stats&thisyear "&drive:\mcdb\datacol\&payer.\sasdata\dc_enc1";
**libname stats&lastyear "C:\2004_MD\MCDB_2004\datacol\&payer.\sasdata\dc_enc1";
libname stats&lastyear "I:\MCdb\datacol\&payer\sasdata\2005_data\dc_enc1";
libname inj "J:\mcdb_raw";

title1 "Data Completeness Summary Report Statistics - 20&thisyear";
title2 "Source - &payer._enc1.sas7bdat";

*********************************************************************************

                                    Statistics

*********************************************************************************
;
**********************
Adjust missing values

For this version, there
are no RVUs in the database
yet. Create a variable equal
to 0 in all cases.
**********************
;
data temp;
     set in.&payer._enc1 (keep = covtype delivtyp plan2 pidbdg payment_orig);

     *** adjust missing coverage type ***;
     if covtype not in ('1', '2', '3', '4', '5', '6', '7','8','9') then covtype = 'X';

     *** adjust missing delivery system ***;
     if delivtyp not in ('1', '2', '3', '4', '5') then delivtyp = 'X';

     *** adjust missing plan ***;
     if plan = ' ' then  plan = 'MISS';

     *** adjust missing plan 2 ***;
     if plan2 = ' ' then  plan2 = 'MISS';

     rvu_i = 0;
run;

**********************
Delivery System
**********************
;
** number of recipients **;
proc summary data = temp nway;
     class delivtyp pidbdg;
     output out = tempdr (drop = _type_);
run;

proc summary data = tempdr nway missing;
     class delivtyp;
     output out = drecip (drop = _type_ rename = (_freq_ = nrecip_&thisyear));
run;
```

```
*** number of services ***;
proc summary data = temp nway missing;
     class delivtyp;
     output out = dserv (drop = _type_ rename = (_freq_ = nserv&thisyear));
run;

*** total payments and total rvus ***;
proc summary data = temp nway missing;
     class delivtyp;
     var payment_orig rvu_i;
     output out = dpayrvu (drop = _type_ _freq_)
                           sum   = totpay&thisyear
                                   totrvu&thisyear
;
run;

*** combine stats and output permanent dataset ***;
data stats&thisyear..delivtyp_&payer._20&thisyear;
     merge drecip
           dserv
           dpayrvu
;
     by delivtyp;
run;

title3 "Delivery System - &thisyear";
proc print data = stats&thisyear..delivtyp_&payer._20&thisyear;
run;

********************
Coverage Type
********************
;
** number of recipients **;
proc summary data = temp nway;
     class covtype pidbdg;
     output out = tempcr (drop = _type_);
run;

proc summary data = tempcr nway missing;
     class covtype;
     output out = crecip (drop = _type_ rename = (_freq_ = nrecip_&thisyear));
run;

*** number of services ***;
proc summary data = temp nway missing;
     class covtype;
     output out = cserv (drop = _type_ rename = (_freq_ = nserv&thisyear));
run;

*** total payments and total rvus ***;
proc summary data = temp nway missing;
     class covtype;
     var payment_orig rvu_i;
     output out = cpayrvu (drop = _type_ _freq_)
                           sum   = totpay&thisyear
                                   totrvu&thisyear
;
run;

*** combine stats and output permanent dataset ***;
data stats&thisyear..covtype_&payer._20&thisyear;
     merge crecip
           cserv
           cpayrvu
;
     by covtype;
run;

title3 "Coverage Type - &thisyear";
```

```
proc print data = stats&thisyear..covtype_&payer._20&thisyear;
run;

*********************************************************
plan2

plan2 is created for all services, and includes Medicare
coverage as a separate category
*********************************************************
;
** number of recipients **;
proc summary data = temp nway;
     class plan2 pidbdg;
     output out = temppr (drop = _type_);
run;

proc summary data = temppr nway missing;
     class plan2;
     output out = precip (drop = _type_ rename = (_freq_ = nrecip_&thisyear));
run;

*** number of services ***;
proc summary data = temp nway missing;
     class plan2;
     output out = pserv (drop = _type_ rename = (_freq_ = nserv&thisyear));
run;

*** total payments and total rvus ***;
proc summary data = temp nway missing;
     class plan2;
     var payment_orig rvu_i;
     output out = ppayrvu (drop = _type_ _freq_)
            sum  = totpay&thisyear totrvu&thisyear;
run;

*** combine stats and output permanent dataset ***;
data stats&thisyear..plan2_&payer._20&thisyear;
     merge precip
           pserv
           ppayrvu
;
     by plan2;
run;

title3 "Plan2 - &thisyear";
proc print data = stats&thisyear..plan2_&payer._20&thisyear;
run;

*****************
Total Recipients
Overall
*****************
;
proc summary data = temp nway;
     class pidbdg;

     output out = temprecip (drop = _type_);
run;

proc summary data = temprecip nway missing;
     output out = stats&thisyear..totrecip_&payer._20&thisyear (drop = _type_ rename =
(_freq_ = tnrecip&thisyear));
run;

title3 "Total Recipients - &thisyear";
proc print data = stats&thisyear..totrecip_&payer._20&thisyear;
run;

*******************************************************************************************
*********
```

```
*****************************************************************************************
*********
;
*****************
Delivery System
*****************
;
data devs_temp1;
     merge stats&lastyear..delivtyp_&payer._20&lastyear
                                (rename = (nrecip_&lastyear = nr_lastyr
                                           nserv&lastyear   = ns_lastyr
                                           totpay&lastyear  = tp_lastyr
                                           totrvu&lastyear  = tr_lastyr
                                          )
                                )

           stats&thisyear..delivtyp_&payer._20&thisyear
                                (rename = (nrecip_&thisyear  = nr_thisyr
                                           nserv&thisyear    = ns_thisyr
                                           totpay&thisyear   = tp_thisyr
                                           totrvu&thisyear   = tr_thisyr
                                          )
                                )
;
     by delivtyp;

     nr_chg = (nr_thisyr - nr_lastyr)/nr_lastyr*100;
     ns_chg = (ns_thisyr - ns_lastyr)/ns_lastyr*100;
     tp_chg = (tp_thisyr - tp_lastyr)/tp_lastyr*100;
     *tr_chg = (tr_thisyr - tr_lastyr)/tr_lastyr*100;

     length rownum 8;
     if delivtyp = "X" then rownum = 6;
     else rownum = delivtyp;

     tablenum = 1;
run;

*proc print data=devs_temp1;
*title3 "Both Years - Delivery System";
*run;

*****calculate total for row 8***;
proc means data = devs_temp1 noprint;
     var ns_lastyr
         ns_thisyr
         tp_lastyr
         tp_thisyr
         tr_lastyr
         tr_thisyr
;
     output out = devs_tot
            sum = ns_lastyr
                  ns_thisyr
                  tp_lastyr
                  tp_thisyr
                  tr_lastyr
                  tr_thisyr
;
run;

data devs_temp2;
     merge devs_tot
           stats&lastyear..totrecip_&payer._20&lastyear
           stats&thisyear..totrecip_&payer._20&thisyear
;

     nr_lastyr = tnrecip&lastyear;
     nr_thisyr = tnrecip&thisyear;
```

```
     nr_chg = (nr_thisyr - nr_lastyr)/nr_lastyr*100;
     ns_chg = (ns_thisyr - ns_lastyr)/ns_lastyr*100;
     tp_chg = (tp_thisyr - tp_lastyr)/tp_lastyr*100;
    * tr_chg = (tr_thisyr - tr_lastyr)/tr_lastyr*100;


     tablenum = 1;
     rownum = 8;
     drop _type_ _freq_ tnrecip&lastyear tnrecip&thisyear;
run;

*proc print data=devs_temp2;
*run;

data devs_final;
     set devs_temp1
         devs_temp2
;
run;

title3 "Delivery System - Final Stats";
proc print data=devs_final;
     var tablenum rownum
         nr_lastyr nr_thisyr nr_chg
         ns_lastyr ns_thisyr ns_chg
         tp_lastyr tp_thisyr tp_chg
         tr_lastyr tr_thisyr /*tr_chg*/
;
run;

************************
Coverage Type
************************
;
data covtype_temp1;
     merge stats&lastyear..covtype_&payer._20&lastyear
                                 (rename = (nrecip_&lastyear = nr_lastyr
                                            nserv&lastyear   = ns_lastyr
                                            totpay&lastyear  = tp_lastyr
                                            totrvu&lastyear  = tr_lastyr
                                           )
                                 )

           stats&thisyear..covtype_&payer._20&thisyear
                                 (rename = (nrecip_&thisyear = nr_thisyr
                                            nserv&thisyear   = ns_thisyr
                                            totpay&thisyear  = tp_thisyr
                                            totrvu&thisyear  = tr_thisyr
                                           )
                                 )
;
     by covtype;

     nr_chg = (nr_thisyr - nr_lastyr)/nr_lastyr*100;
     ns_chg = (ns_thisyr - ns_lastyr)/ns_lastyr*100;
     tp_chg = (tp_thisyr - tp_lastyr)/tp_lastyr*100;
     *tr_chg = (tr_thisyr - tr_lastyr)/tr_lastyr*100;

     length rownum 8;
     if covtype = "X" then rownum = 9;
     else rownum = covtype;


     tablenum = 3;
run;

*****calculate total for row 11***;
proc means data = covtype_temp1 noprint;
     var ns_lastyr
```

```
         ns_thisyr
         tp_lastyr
         tp_thisyr
         tr_lastyr
         tr_thisyr
     ;
     output out = covtype_tot
            sum = ns_lastyr
                  ns_thisyr
                  tp_lastyr
                  tp_thisyr
                  tr_lastyr
                  tr_thisyr
     ;
run;

data covtype_temp2;
     merge covtype_tot
           stats&lastyear..totrecip_&payer._20&lastyear
           stats&thisyear..totrecip_&payer._20&thisyear;

     nr_lastyr = tnrecip&lastyear;
     nr_thisyr = tnrecip&thisyear;


     nr_chg = (nr_thisyr - nr_lastyr)/nr_lastyr*100;
     ns_chg = (ns_thisyr - ns_lastyr)/ns_lastyr*100;
     tp_chg = (tp_thisyr - tp_lastyr)/tp_lastyr*100;
     *tr_chg = (tr_thisyr - tr_lastyr)/tr_lastyr*100;


     tablenum = 3;
     rownum = 11;
     drop _type_ _freq_ tnrecip&lastyear tnrecip&thisyear;
     run;

data covtype_final;
     set covtype_temp1
         covtype_temp2
;
run;

title3 "Coverage Type - Final Stats";
proc print data=covtype_final;
     var tablenum rownum
         nr_lastyr nr_thisyr nr_chg
         ns_lastyr ns_thisyr ns_chg
         tp_lastyr tp_thisyr tp_chg
         tr_lastyr tr_thisyr /*tr_chg*/
;
run;

************************
PLAN2
************************
;
data plan2_temp1;
     merge stats&lastyear..plan2_&payer._20&lastyear
           (rename = (nrecip_&lastyear = nr_lastyr
                      nserv&lastyear   = ns_lastyr
                      totpay&lastyear  = tp_lastyr
                      totrvu&lastyear  = tr_lastyr))

           stats&thisyear..plan2_&payer._20&thisyear
           (rename = (nrecip_&thisyear = nr_thisyr
                      nserv&thisyear   = ns_thisyr
                      totpay&thisyear  = tp_thisyr
                      totrvu&thisyear  = tr_thisyr))
;
     by plan2;
```

```
     nr_chg = (nr_thisyr - nr_lastyr)/nr_lastyr*100;
     ns_chg = (ns_thisyr - ns_lastyr)/ns_lastyr*100;
     tp_chg = (tp_thisyr - tp_lastyr)/tp_lastyr*100;
     *tr_chg = (tr_thisyr - tr_lastyr)/tr_lastyr*100;

     length rownum 8;
     if plan2 = "NONHMO" then rownum = 1;
     else
     if plan2 = "HMOFFS" then rownum = 2;
     else
     if plan2 = "HMOCAP" then rownum = 3;
     else
     if plan2 in ("MCRALL", "MCRHMO") then rownum = 4;

     if plan2 = "MISS" then rownum = 5;

     tablenum = 2;
run;

*****calculate total for row 6***;
proc means data = plan2_temp1 noprint;
     var ns_lastyr
         ns_thisyr
         tp_lastyr
         tp_thisyr
         tr_lastyr
         tr_thisyr
;
     output out = plan2_tot
            sum = ns_lastyr
                  ns_thisyr

                  tp_lastyr
                  tp_thisyr
                  tr_lastyr
                  tr_thisyr
;
run;

data plan2_temp2;
     merge plan2_tot
           stats&lastyear..totrecip_&payer._20&lastyear
           stats&thisyear..totrecip_&payer._20&thisyear;

     nr_lastyr = tnrecip&lastyear;
     nr_thisyr = tnrecip&thisyear;


     nr_chg = (nr_thisyr - nr_lastyr)/nr_lastyr*100;
     ns_chg = (ns_thisyr - ns_lastyr)/ns_lastyr*100;
     tp_chg = (tp_thisyr - tp_lastyr)/tp_lastyr*100;
     *tr_chg = (tr_thisyr - tr_lastyr)/tr_lastyr*100;


     tablenum = 2;
     rownum = 6;
     drop _type_ _freq_ tnrecip&lastyear tnrecip&thisyear;
run;

data plan2_final;
     set plan2_temp1
         plan2_temp2
;
run;

title "Plan2 - Final Stats";
proc print data=plan2_final;
     var tablenum rownum
         nr_lastyr nr_thisyr nr_chg
         ns_lastyr ns_thisyr ns_chg
         tp_lastyr tp_thisyr tp_chg
```

```
         tr_lastyr tr_thisyr /*tr_chg*/
;

*****************************************************************************************
*********************

                                          Create the Table

*****************************************************************************************
**********************
;
******************************
Create Dummy dataset to ensure
that the table data contains all
possible rows, regardless of the
particular payer
*********************************
;
data dummy;

     tablenum=1;
     do rownum = 1 to 8;
        output;
     end;

     tablenum=2;
     do rownum = 1 to 6;
        output;
     end;

     tablenum=3;
     do rownum = 1 to 11;
        output;
     end;
run;

data all_in_all;
     set devs_final
         covtype_final
         plan2_final
;
run;

proc sort data=all_in_all;
     by tablenum rownum;
run;

data stats&thisyear..&payer._enc1_dc_crunch;
     merge dummy (in=indummy)
           all_in_all
;
     by tablenum rownum;

     if indummy then output;
run;

title3 "Data Ready for Spreadsheet";
proc print data=stats&thisyear..&payer._enc1_dc_crunch;
var delivtyp covtype plan2
    tablenum rownum
    nr_lastyr nr_thisyr nr_chg
    ns_lastyr ns_thisyr ns_chg
    tp_lastyr tp_thisyr tp_chg
    tr_lastyr tr_thisyr /*tr_chg*/
    ;

run;

%jobend;
```

```
dm "log;    print file =
I:\mcdb\datacol\&payer.\reports\&payer._enc1_data_completeness.log replace";
dm "output; print file =
I:\mcdb\datacol\&payer.\reports\&payer._enc1_data_completeness.lst replace";


*****************************************************************
Program  : pxxx_enc1_DC_xls.sas
Project  : Maryland
Purpose  : For the report on the Enc1 Data, create an Excel
           spreadsheet from the crunched sas dataset

Notes    : 1. This program includes both the macro to create the spreadsheet
              and the macro call
           2. Programmer change payer, titles, filenames, and literals for
              titles in macro call at the end of the program.
-------------------------------------------------------------------

Input    : i:\mcdb\datacol\pxxx\sasdata\dc_enc1\pxxx_enc1_crunch.sas7bdat
Output   : i:\mcdb\datacol\pxxx\reports\pxxx_enc1_dc_crunch.xls

-------------------------------------------------------------------
Date     : 07/06/2005
By       : Raymond Hu
Modified : 07/27/2005
           Sireesha Thati
Modified : Sireesha Thati
               06/20/2006

---------------------------------------------------------------------------
Date     : 09/10/2007
By       : Kasey Chen
Note     : 1. Changed program name as 'pxxx_enc1_DC_xls.sas'.
           2. Changed last year as 2005 and this year as 2006.
           3. Inserted current report date and servise number
              from Pxxx_Enc1.sas7bdat.
           4. Reduce the number of tabs ('09'x) between "text8a" and "test8b".
           5. Reduce the number of tabs ('09'x) between "text9a" and "test9b".
****************************************************************************
;
options mprint nomlogic nosymbolgen;


****************************
Spreadsheet Macro
****************************
;
%macro Create_Xls_Table(This_year        =,
                        SASDataLib       =,
                        SASDataName      =,
                        Pnum             =,
                        Pname            =,
                        elserv           =,
                        subserv          =,
                        fmonday          =,
                        fyear            =,
                        ExcelOutPathName =
                        )
;

   options noxwait noxsync;

   ****************************************************************
      1.       Open Microsoft Excel
         (There are 2 sets of statements to open the Excel file
          because different machine may have different setup.
          The codes have to be modified appropriately.)
   ****************************************************************
;
```

```
   *** excel location on maryland room computer (#11076) ***;
*   x "C:\Progra~1\Micros~2\Office10\EXCEL.EXE ";

   *** excel location on maryland room computer (#11077) ***;
   x "C:\Progra~1\Micros~2\Office10\EXCEL.EXE ";


   ** Sleep to allow time for Excel to come up **;
   data _null_;
     x=sleep(3);
   run;

   ******************************************************************
     2.        Open another Excel worksheet
   ******************************************************************
;
   %let cmopen2='[OPEN("i:\mcdb\datacol\include_program\enc1_dc_shell_2006.xls")]';

   %put &cmopen2;

   filename cmds dde 'excel|system';

   data _null_;
     file cmds;
     put &cmopen2;
   run;

   data _null_;
     x=sleep(3);
   run;

   ******************************************************************
    3.        Write data to Excel by DDE
   ******************************************************************
;
   ******************************************************************
    3.1       Write header to Excel by DDE
   ******************************************************************
;

   %let
cmxcpt='Excel|i:\mcdb\datacol\include_program\[enc1_dc_shell_2006.xls]Data_Completeness_R
eport!R4C1:R9C18';
   ** Establish the DDE link **;
   filename caseman dde &cmxcpt notab;
   %let elserv = %sysfunc(putn(&elserv,comma12.));
   %let subserv = %sysfunc(putn(&subserv,comma12.));

   data _null_;
      file caseman ;
          *** concatenate payer number and payer name ***;
      pnumname = "&pnum" || ": " || "&pname";

      *** Create text for source file ***;
      text8a = "Eligible Services: " || "&elserv";
      text8b = "Source File: " || "&payer._enc1_dc_crunch.sas7bdat";
      text9a = "Services Submitted: " || "&subserv";
      text9b = "File Date: "|| "&fmonday" || ', '|| "&fyear";

      put pnumname;
      put "Data After Initial Processing Step (Enc1)";
      put "Data Completeness Summary Report";
      put ;
      put text8a '09'x '09'x '09'x '09'x '09'x  text8b  ;
      put text9a '09'x '09'x '09'x '09'x '09'x  text9b  ;
      run;


   ******************************************************************
    3.2       Write data to Excel-table1 by DDE
   ******************************************************************
```

```
;

  %let
cmxcpt='Excel|i:\mcdb\datacol\include_program\[enc1_dc_shell_2006.xls]Data_Completeness_R
eport!R14C2:R21C18';

   ** Establish the DDE link **;
   filename caseman dde &cmxcpt notab;


   data _null_;
    file caseman ;
    set &SASDataLib..&SASDataName end=eof;

    if tablenum=1 then do;
       if nr_LastYr ne . or nr_ThisYr ne . or nr_chg ne . or
          ns_LastYr ne . or ns_ThisYr ne . or ns_chg ne . or
          tp_LastYr ne . or tp_ThisYr ne . or tp_chg ne .
          then do;

          put nr_LastYr    '09'x
              nr_ThisYr    '09'x
              nr_chg       '09'x
              ns_LastYr    '09'x
              ns_ThisYr    '09'x
              ns_chg       '09'x
              tp_LastYr    '09'x
              tp_ThisYr    '09'x
              tp_chg       '09'x
              ;
          end;

       else put;

      end;

   run;


   ******************************************************************
    3.3       Write data to Excel-table2 by DDE
   ******************************************************************
;

  %let
cmxcpt='Excel|i:\mcdb\datacol\include_program\[enc1_dc_shell_2006.xls]Data_Completeness_R
eport!R26C2:R31C18';

   ** Establish the DDE link **;
   filename caseman dde &cmxcpt notab;


   data _null_;
    file caseman ;
    set &SASDataLib..&SASDataName end=eof;

    if tablenum=2 then do;
       if nr_LastYr ne . or nr_ThisYr ne . or nr_chg ne . or
          ns_LastYr ne . or ns_ThisYr ne . or ns_chg ne . or
          tp_LastYr ne . or tp_ThisYr ne . or tp_chg ne .
          then do;

          put nr_LastYr    '09'x
              nr_ThisYr    '09'x
              nr_chg       '09'x
              ns_LastYr    '09'x
              ns_ThisYr    '09'x
              ns_chg       '09'x
              tp_LastYr    '09'x
              tp_ThisYr    '09'x
              tp_chg       '09'x
```

```
            ;
         end;

       else put;

      end;

   run;

   ******************************************************************
    3.4       Write data to Excel-table3 by DDE
   ******************************************************************
;

  %let
cmxcpt='Excel|i:\mcdb\datacol\include_program\[enc1_dc_shell_2006.xls]Data_Completeness_R
eport!R36C2:R48C18';

   ** Establish the DDE link **;
   filename caseman dde &cmxcpt notab;


   data _null_;
    file caseman ;
    set &SASDataLib..&SASDataName end=eof;

    if tablenum=3 then do;
       if nr_LastYr ne . or nr_ThisYr ne . or nr_chg ne . or
          ns_LastYr ne . or ns_ThisYr ne . or ns_chg ne . or
          tp_LastYr ne . or tp_ThisYr ne . or tp_chg ne .
          then do;

          put nr_LastYr    '09'x
              nr_ThisYr    '09'x
              nr_chg       '09'x
              ns_LastYr    '09'x
              ns_ThisYr    '09'x
              ns_chg       '09'x
              tp_LastYr    '09'x
              tp_ThisYr    '09'x
              tp_chg       '09'x

              ;
          end;

       else put;

      end;

   run;


   *********************************************************
     7.  Save as and close Xls
   *********************************************************
;
   %let rc3 = %sysfunc(fileexist("&exceloutPathName"));
   data _null_;
     if &rc3 = 1 then do; x " del &exceloutPathName"; end;
   run;

   data _null_;
     x=sleep(3);
   run;

   data _null_;
     file cmds;

     put "[SAVE.AS(%bquote("&exceloutPathName"))]";
     put '[FILE.CLOSE()]';
     put '[QUIT()]';
```

```
   run;

%mend;


   ******************************************************************
    Macro Call
   ******************************************************************
;

%let payer = P020;
%let pname = Aetna Life and Health Insurance Co;
%let fmonday = September 11;
%let fyear   = 2007;
libname in2 "i:\mcdb\datacol\&payer.\sasdata\dc_enc1\";

%Create_Xls_Table(This_year = 2006,
                  SASDataLib = in2,
                  SASDataName = %str(&payer._enc1_dc_crunch),
                  Pnum = %str(&payer),
                  Pname = %str(&pname),
                  elserv  = 6735681,
                  subserv = 6735681,
                  fmonday = %str(&fmonday),
                  fyear = %str(&fyear),
                  ExceloutPathName =
%str(i:\mcdb\datacol\&payer.\reports\&payer._enc1_dc_crunch.xls)
                  );

dm "log;    print file = I:\mcdb\datacol\&payer.\reports\&payer._enc1_DC_xls.log
replace";
dm "output; print file = I:\mcdb\datacol\&payer.\reports\&payer._enc1_DC_xls.lst
replace";

proc datasets kill;
run;
quit;


%jobstart;

*=========================================================================================
======
Program   : I:\mcdb\datacol\p020\reports\p020_enc1_dflags.sas.
Notes     : To get the summary of Deletion Flags in the form of dataset.
            The final dataset created in this program is exported into excel by
            p020_enc1_dflags_xls.sas.
-----------------------------------------------------------------------------------------
------
input     : p020_enc5.sas7bdat
output    : p020_enc5_dflags.sas7bdat
-----------------------------------------------------------------------------------------
------
Programmer: Sanee Maphungphong / Sireesha Thati
Date      : June 9, 2006
-----------------------------------------------------------------------------------------
------
Programmer: Po-Lun Chou
Date:       August 29, 2007
Note:       Per the decision from MHCC, all of the records should be kept in the
            MCDB database this year (2006 data), including the records which have
            deletion flags (D_***_F) = 1, so I modified this program based on this
            decision to reflect this change.
*=========================================================================================
======;

options obs = max errors = 2 missing = ' ';

%let payernum = P020;

libname inj "J:\mcdb_raw";
```

```
libname out "I:\MCdb\datacol\P020\sasdata\dc_enc1";

*******************************************************************************************
**************
Use ODS to capture output stats

"onewayfreqs" is SAS internal name for dataset for one-way freq

"list" is SAS internal name for dataset containing 2-way and multiple freqs when /list
option is used

"list" is also used for proc summary output
*******************************************************************************************
**************;

***** for all deletion flags *****;

ods output list = temp2;

proc freq data = inj.&payernum._enc1_d_flags;
     tables d_dnt_f * d_inp_f * d_i7k_f * d_zip_f * d_dat_f / list missing;
     title "Crossed Frequencies of Deletion Flags - &payernum";
run;

ods output close;

data all_dflags (keep = d_zip_f d_dat_f d_dnt_f d_inp_f d_i7k_f test frequency percent
                        cumfrequency cumpercent);
     set temp2 ;
     length test $1;
     if d_zip_f = '9' and d_dat_f = '9' and d_dnt_f = '9' and d_inp_f = '9' and d_i7k_f =
'9'
     then test = 'x';
run;


%MACRO NOTUSE;

***** for deletion flag d_zip_f only *****;

ods output onewayfreqs = temp3;

proc freq data = inj.&payernum._enc1_d_flags;
     where d_zip_f = '1' and d_dat_f = '9' and d_dnt_f = '9' and d_inp_f = '9' and
d_i7k_f = '9';
     tables  d_zip_f / list missing;
run;

ods output close;

** We want the number of records that are flagged as d_zip_f and are included in the MCDB
   eligible service count **;

data zip_flag ( drop = d_zip_f rename = (frequency = d_zip_f) );
     set temp3 ( keep =  d_zip_f  frequency);
     length test $1;
     test = 'x';
run;

%MEND NOTUSE;


data out.&payernum._enc1_dflags (drop = d_zip_f d_dat_f d_dnt_f d_inp_f d_i7k_f test
frequency
                                       percent CumFrequency cumpercent);
     set all_dflags;
     length zip_f dnt_f inp_f i7k_f dat_f $1 freq pct cfreq cpct 8;
     zip_f = d_zip_f;
     dat_f = d_dat_f;
     dnt_f = d_dnt_f;
     inp_f = d_inp_f;
```

```
     i7k_f = d_i7k_f;
     freq  = frequency;
     pct   = percent;
     cfreq = cumfrequency;
     cpct  = cumpercent;
     format pct cpct 8.2;
run;

proc print data = out.&payernum._enc1_dflags noobs;
     var zip_f dnt_f inp_f i7k_f dat_f freq pct cfreq cpct;
     title "Summary of Deletion Flags - &payernum";
run;

%jobend;

dm "log;    print file = I:\MCdb\datacol\P020\reports\&payernum._enc1_dflags.log
replace";
dm "output; print file = I:\MCdb\datacol\P020\reports\&payernum._enc1_dflags.lst
replace";

proc datasets kill;
run;
quit;


%jobstart;

************************************************************************
Program : p020_enc1_dflags_xls.sas
Project : Maryland
Purpose : For the report on the deletion flags from the Enc1 Data,
          create an Excel spreadsheet from the crunched sas dataset

Notes   : 1. This program includes both the macro to create the
             spreadsheet and the macro call
          2. Programmer change payer, titles, filenames, and
             literals for titles in macro call at the end of the
             program.
------------------------------------------------------------------------
Input   : i:\mcdb\datacol\p020\sasdata\dc_enc1\p020_enc1_dflags.sas7bdat
Output  : i:\mcdb\datacol\p020\reports\p020_enc1_dflags.xls
------------------------------------------------------------------------
Date    : 07/06/2005
By      : Raymond Hu

Modified: 07/27/2005
          Sireesha Thati
------------------------------------------------------------------------
Modified By: Po-Lun Chou
Date:        August 29, 2007
Note:        1. Per MHCC's request, we apply MCDB Global Screen flags
                (variables D_***_F) as usual for year 2006 data, but
                not deleting them from the data base.  So this program
                has to be modified to include all flags information in
                the Excel spreadsheet.
             2. Excel shell file, enc1_dflags_shell_2006.xls, has to be
                modified before running this program, and the related
                Macro, DeleteRows, has to be generated with it.  This
                Macro can be generated by following the steps below in
                this program, and the contents of it can be got from
                previous year's shell file, but have to be modified.
************************************************************************;

options obs = max nomprint nomlogic nosymbolgen;

%let payer = P020;
%let pname = Aetna Life & Health Insurance Co.;


libname in "I:\MCdb\datacol\&payer.\sasdata\dc_enc1";
```

```
data _null_;
  current = today();
  call symput('curyr', trim(left(year(current))));
  call symput('curmon', trim(left(put(current, monname9.)))));
  call symput('curday', trim(left(day(current))));
run;


**************************************
Any needed fixes to input dataset
**************************************;

data temp;
  set in.&payer._enc1_dflags;
run;


****************************
Spreadsheet Macro
****************************;

%macro Create_Xls_Table(This_Year = ,
                        SASDataLib = ,
                        SASDataName = ,
                        ExcelOutPathName = );

   options noxwait noxsync;

   ******************************************************************
     1.  Open Microsoft Excel
   ******************************************************************;

   *** excel location on maryland room computer ***;
   x "C:\Progra~1\Micros~2\Office10\EXCEL.EXE ";

   *** excel location on maryland room computer (#11077) ***;
/* x "C:\Progra~1\Micros~2\Office10\EXCEL.EXE ";  */

   ** Sleep to allow time for Excel to come up **;
   data _null_;
     x=sleep(3);
   run;

   ******************************************************************
     2.  Open another Excel worksheet
   ******************************************************************;

   %let cmopen2='[OPEN("I:\MCdb\datacol\Include_Program\enc1_dflags_shell_2006.xls")]';

   %put &cmopen2;

   filename cmds dde 'excel|system';

   data _null_;
     file cmds;
     put &cmopen2;
   run;

   data _null_;
     x=sleep(3);
   run;

   ******************************************************************
     3.  Write data to Excel by DDE
   ******************************************************************;

   ******************************************************************
     3.1  Write header to Excel by DDE
   ******************************************************************;
```

```
   %let
cmxcpt='Excel|I:\MCdb\datacol\Include_Program\[enc1_dflags_shell_2006.xls]MCDB_Screening_
Report!R4C1:R4C10';
/*  %let
cmxcpt='Excel|I:\MCdb\datacol\Include_Program\[enc1_dflags_shell_2006.xls]MCDB_Screening_
Report!R4C1:R9C18';  */

   ** Establish the DDE link **;
   filename caseman dde &cmxcpt notab;

   data _null_;
     file caseman ;

     *** concatenate payer number and payer name ***;
     pnumname = "&payer" || ": " || "&pname";
     put pnumname;

  run;

   %let
cmxcpt='Excel|I:\MCdb\datacol\Include_Program\[enc1_dflags_shell_2006.xls]MCDB_Screening_
Report!R8C1:R8C10';
/*  %let
cmxcpt='Excel|I:\MCdb\datacol\Include_Program\[enc1_dflags_shell_2006.xls]MCDB_Screening_
Report!R4C1:R9C18';  */

   ** Establish the DDE link **;

   filename caseman dde &cmxcpt notab;

   data _null_;
     file caseman ;

     *** Create text for source file ***;
     text8a = "Source File: " || "&payer._enc1_dflags.sas7bdat";
     text8b = "File Date: " || "&curmon" || ' ' || "&curday" || ', ' || "&curyr";
     put text8a '09'x '09'x '09'x '09'x '09'x '09'x '09'x '09'x text8b;

   run;


   ******************************************************************
     3.2  Write data to Excel by DDE
   ******************************************************************;

/*  %let
cmxcpt='Excel|i:\mcdb\datacol\data_completeness_summaries\[shell_3.xls]Data_Completeness_
Report!R25C2:R30C18';  */

   %let
cmxcpt='Excel|I:\MCdb\datacol\Include_Program\[enc1_dflags_shell_2006.xls]MCDB_Screening_
Report!R12C1:R28C10';

/*  %let
cmxcpt='Excel|I:\MCdb\datacol\Include_Program\[enc1_dflags_shell_2006.xls]MCDB_Screening_
Report!R12C1:R28C18';  */

   ** Establish the DDE link **;
   filename caseman dde &cmxcpt notab;

   data _null_;
     file caseman ;
     set &SASDataLib..&SASDataName end=eof;
     length xAction $6;

     xd_zip_f = '   ';
     xd_dnt_f = '   ';
     xd_inp_f = '   ';
     xd_i7k_f = '   ';
     xd_dat_f = '   ';
```

```
     if zip_f=1 then xd_zip_f='Yes';
     if dnt_f=1 then xd_dnt_f='Yes';
     if inp_f=1 then xd_inp_f='Yes';
     if i7k_f=1 then xd_i7k_f='Yes';
     if dat_f=1 then xd_dat_f='Yes';
     xAction = ' ';
     if zip_f=9 and dat_f=9 and dnt_f=9 and inp_f=9 and i7k_f=9
     then xAction = 'Yes';

     put xd_zip_f         '09'x
         xd_dnt_f         '09'x
         xd_inp_f         '09'x
         xd_i7k_f         '09'x
         xd_dat_f         '09'x
         xAction          '09'x
         freq   comma12.0 '09'x
         pct    comma8.2  '09'x
         cfreq  comma12.0 '09'x
         cpct   comma8.2  '09'x;

     if eof then do;
         put ;
         put ;
         put ;
         put ;

         *** pass the number of obs to a macro var to decide how many blank row needs to
be left***;
         call symput('NumRow',_N_);
     end;

   run;


   ******************************************************************
     4.  Pass the number obs of data into Excel
   ******************************************************************;

/*  %let
cmxcpt='Excel|I:\mcdb\datacol\data_completeness_summaries\[shell_3.xls]Data_Completeness_
Report!R50C1:R50C1';  */
/*  %let
cmxcpt='Excel|C:\work_HCACC\xls\[shell_3.xls]Data_Completeness_Report!R50C1:R50C1';  */

   %let
cmxcpt='Excel|I:\MCdb\datacol\Include_Program\[enc1_dflags_shell_2006.xls]MCDB_Screening_
Report!R24C1:R24C10';
/*  %let
cmxcpt='Excel|I:\MCdb\datacol\Include_Program\[enc1_dflags_shell_2006.xls]MCDB_Screening_
Report!R39C1:R39C18';  */

   ** Establish the DDE link **;
   filename caseman dde &cmxcpt notab;

   data _null_;
     file caseman ;
     put "&NumRow";
   run;


   *******************************************************
     5.  Run Excel Macro
   *******************************************************;

   filename cmds dde 'excel|system';

*******************************************************************************
Error: DDE session not ready
FATAL: Unrecoverable I/O error detected in the execution of the data step program.
       Aborted during execution phase
For the kind of error above---- check the excel shell which we are calling
```

```
  Go to tools -> Macro -> Security and set the security to Low
  And also check the macro name in Tools -> Macro -> macros
  (For this program the macro name should be DeleteRows)
*******************************************************************************;

   data _null_;
     file cmds;
     put '[RUN("DeleteRows")]';
   run;


   *******************************************************
     6.  Save as and close Xls
   *******************************************************;

   %let rc3 = %sysfunc(fileexist("&exceloutPathName"));

   data _null_;
     if &rc3 = 1 then do; x " del &exceloutPathName"; end;
   run;

   data _null_;
     x=sleep(3);
   run;

   data _null_;
     file cmds;

     put "[SAVE.AS(%bquote("&exceloutPathName"))]";
     put '[FILE.CLOSE()]';
     put '[QUIT()]';
   run;

%mend;


*****************************************************************
Macro Call
*****************************************************************;

libname in2 "I:\MCdb\datacol\&payer.\sasdata\dc_enc1\";

%Create_Xls_Table(This_Year = 2006,
                  SASDataLib = in2,
                  SASDataName = %str(&payer._enc1_dflags),
                  ExceloutPathName =
%str(i:\mcdb\datacol\&payer.\reports\&payer._enc1_dflags.xls));

%jobend;


dm "log;    print file = I:\mcdb\datacol\&payer.\reports\&payer._enc1_dflags_xls.log
replace";
dm "output; print file = I:\mcdb\datacol\&payer.\reports\&payer._enc1_dflags_xls.lst
replace";

proc datasets kill;
run;
quit;

%jobstart;


**==========================================================================**;
** Program:     Specialty_Check_2006_All_Payers.sas (SPECIALTY PROGRAM #1)   **;
** Date:        June 7, 2006                                                 **;
** Purpose:     Match specialty records in provider roster file with         **;
**              encounter reocrds by                                         **;
**              - PROVID and FEDTAXID                                        **;
**              - PROVID                                                     **;
** Input Ds:    pxxx_enc1.sas7bdat                                           **;
```

```
**              pxxx_prv1.sas7bdat                                          **;
** Output Ds:   pxxx_pf_ok_all (single spec. matched by PROVID/FEDTAXID)     **;
**              pxxx_pf_chk_all (multiple spec. matched by PROVID/FEDTAXID)  **;
**              pxxx_p_ok_all (single spec. matched by PROVID)               **;
**              pxxx_p_chk_all (multiple spec. matched by PROVID)            **;
** Note:        Due to our decision to change processinf sequence, the input **;
**              file from the encoounter side is ENC1 rather than ENC2.      **;
**==========================================================================**;
**                                                                          **;
** Program:     Specialty_Check_2004_All_Payers.sas                          **;
** Programmer:  Po-Lun Chou                                                 **;
** Date:        September 1, 2005                                           **;
**                                                                          **;
** Modified by: Po-Lun Chou                                                 **;
** Date:        September 30, 2005                                          **;
** Note:        Modified this program to include the original               **;
**              SPEC1 - SPEC3 in Provider data file.                        **;
**==========================================================================**;
** Date:        August 9, 2006                                              **;
** Note:        Run on 2006 Data Submission. No modification made to code.    **;
**==========================================================================**;
options obs = max errors = 2 ls = 123 ps = 96;

***** format *****;

proc format;
     value $smisc
           '034',
           '037',
           '038',
           '062',
           '085',
           '093' = '1'
           other = '0';
     value $spectyp
           '001','002','003','004','005','006','007','008','009','010',
           '011','012','013','014','015','016','017','018','019','020',
           '021','022','023','024','025','026','027','028','029','030',
           '031','032','033','035','036','039','100','040',
           '041','042','043','044','045','046','047','048','049','050',
           '051','052','053','054','055','056','057','058','059','060',
           '061','063','070',
           '071','072','073','074','075','076','080',
           '081','082','083','084','090',
           '091','092','102','101',
           '103' = '1' /* valid, not miscellaneous */
           '034','037','038','062','085',
           '093' = '2' /* valid, miscellaneous */
           ' '   = '4' /* missing */
           other = '3' /* invalid */;
     value $sflag
           '1' = '1: Valid, Not Misc'
           '2' = '2: Valid, Misc'
           '3' = '3: Invalid'
           '4' = '4: Missing';
     value $miss
           ' '   = 'Missing'
           other = 'Present';
     value $f_id
           ' '              = 'Blank'
           '0000000000000',

           '000000000000',
           '00000000000',
           '0000000000',
           '000000000',
           '00000000',
           '0000000',
           '000000',
           '00000',
           '0000',
```

```
           '000',
           '00',
           '0'              = 'All 0s'
           '9999999999999',
           '999999999999',
           '99999999999',
           '9999999999',
           '999999999',
           '99999999',
           '9999999',
           '999999',
           '99999',
           '9999',
           '999',
           '99',
           '9'              = 'All 9s'
           other            = 'Valid';
run;


%macro A1(payer);

libname out "I:\MCdb\datacol\&payer\sasdata";

*=================================================================*;
* Create a subset of encounter data and sort by PROVID & FEDTAXID *;
*=================================================================*;

proc sort data = out.&payer._enc1  (keep = pnum provid fedtaxid encnt_id pidbdg) nodupkey
          out = &payer._enc_pf;
  by provid fedtaxid;
run;

*=====================================================*;
* Create a subset of encounter data and sort by PROVID *;
*=====================================================*;


proc sort data = out.&payer._enc1 (keep = pnum provid fedtaxid encnt_id pidbdg) nodupkey
          out = &payer._enc_p;
  by provid;
run;

*=================================================================*;
* Create a subset of provider data and sort by PROVID & FEDTAXID *;
*=================================================================*;

data &payer._prv_pf_k &payer._prv_pf_d;
  set out.&payer._prv1 (drop = spec1_raw spec2_raw spec3_raw);

if provid = " " or
   compress(provid, "0") = " " or
   compress(provid, "9") = " " or
   fedtaxid = " " or
   compress(fedtaxid, "0") = " " or
   compress(fedtaxid, "9") = " " then output &payer._prv_pf_d;
else output &payer._prv_pf_k;
run;

title "Frequencies of Invalid PROVID / FEDTAXID in %upcase(&payer) Provider File";
proc freq data = &payer._prv_pf_d;
  tables provid * fedtaxid / list missing;
  format provid fedtaxid $f_id.;
run;

data &payer._prv_pf_k1 (drop = sflag1 - sflag3);
  length spec_n $3;
  set &payer._prv_pf_k;

if spec1 ne " " then do;
   spec_n = spec1;
```

```
   output;
end;
if spec2 ne " " then do;
   spec_n = spec2;
   output;
end;
if spec3 ne " " then do;
   spec_n = spec3;
   output;
end;
run;

proc sort data = &payer._prv_pf_k1 nodupkey;
  by provid fedtaxid spec_n;
run;

*=======================================================*;
* Create a subset of provider data and sort by PROVID. *;
*=======================================================*;

data &payer._prv_p_k &payer._prv_p_d;
  set out.&payer._prv1 (drop = spec1_raw spec2_raw spec3_raw);
  if provid = " " or compress(provid, "0") = " " or compress(provid, "9") = " "
     then output &payer._prv_p_d;
  else output &payer._prv_p_k;
run;

proc freq data = &payer._prv_p_d;
  tables provid / list missing;
  format provid $f_id.;
  title "Frequencies of Invalid PROVID in %upcase(&payer) Provider File";
run;

data &payer._prv_p_k1 (drop = sflag1 - sflag3);
  length spec_n $3;
  set &payer._prv_p_k;
  if spec1 ne " " then do;
    spec_n = spec1;
    output;
  end;
  if spec2 ne " " then do;
    spec_n = spec2;
    output;
  end;
  if spec3 ne " " then do;
    spec_n = spec3;
    output;
  end;
run;

proc sort data = &payer._prv_p_k1 nodupkey;
  by provid spec_n;
run;



*================================================================*;
* Check for PROVID & FEDTAXID with more than one SPECIALTY code *;
* Output two files:                                              *;
* - Specialty in Provider roster and in Encounter.               *;
* - Specialty in Provider roster and not in Encounter.           *;
*================================================================*;

data &payer._pf_ok &payer._pf_chk;
  set &payer._prv_pf_k1;
  by provid fedtaxid;

if first.fedtaxid & last.fedtaxid then output &payer._pf_ok;
else output &payer._pf_chk;
run;
```

```
title "Samples of Problem Cases in %upcase(&payer) - more than 1 SPECIALTY based on
PROVID & FEDTAXID";
proc print data = &payer._pf_chk (obs = 30);
  id pnum provid fedtaxid;
  by pnum provid fedtaxid;
run;

data &payer._pf_chk1  /* in Encounter file */
     &payer._pf_chk2  /* not in Encounter file */;
  length f_specn $1;
  merge &payer._enc_pf (in = inenc)
        &payer._pf_chk (in = inchk);
  by provid fedtaxid;

f_specn = put(spec_n, $spectyp.);
if inchk then do;
   if inenc then output &payer._pf_chk1;
   else if not inenc then output &payer._pf_chk2;
end;
run;

*=========================================================================-====*;
* Freq and Print Specialty (SPEC1) in Provider roster and in Encounter file *;
*=========================================================================*;

title "Frequencies of SPEC1 in %upcase(&payer) - PROVID & FEDTAXID are in Encounter
File";
proc freq data = &payer._pf_chk1;
  tables spec_n f_specn / list missing;
  tables f_specn * spec_n / list missing;
  format f_specn $sflag.;
run;

data &payer._pf_chk1_k &payer._pf_chk1_d;
  set &payer._pf_chk1;

if f_specn = "1" then output &payer._pf_chk1_k;
else output &payer._pf_chk1_d;
run;

data &payer._pf_chk1_k_ok &payer._pf_chk1_k_chk;
  set &payer._pf_chk1_k;
  by provid fedtaxid;

if first.fedtaxid & last.fedtaxid then output &payer._pf_chk1_k_ok;
else output &payer._pf_chk1_k_chk;
run;

title "Samples of SPEC1 is MISC/Invalid/Blank in %upcase(&payer) - PROVID & FEDTAXID are
in Encounter File";
proc print data = &payer._pf_chk1_d (obs = 40);
  id pnum provid fedtaxid;
  by pnum provid fedtaxid;
  var spec_n recnum lname fname midinit suffix credent recnum deanum business f_specn;
run;

*=========================================================================-====*;
* Freq and Print Specialty (SPEC1) in Provider roster not in Encounter file *;
*=========================================================================*;

title "Frequencies of SPEC1 in %upcase(&payer) - PROVID & FEDTAXID are not in Encounter
File";
proc freq data = &payer._pf_chk2;
  tables spec_n f_specn / list missing;
  tables f_specn * spec_n / list missing;
  format f_specn $sflag.;
run;

proc print data = &payer._pf_chk2 (obs = 40);
  id pnum provid fedtaxid;
  by pnum provid fedtaxid;
```

```
  var spec_n recnum lname fname midinit suffix credent recnum deanum business f_specn;
  title "SPEC1 in %upcase(&payer) - PROVID & FEDTAXID are not in Encounter File";
run;


*============================*;
* create permanent data file *;
*============================*;

data &payer._pf_ok_all;
  set &payer._pf_ok &payer._pf_chk1_k_ok;
run;

proc sort data = &payer._pf_ok_all out = out.&payer._pf_ok_all;
  by pnum provid fedtaxid;
run;

title "Good SPEC1 in %upcase(&payer) - based on PROVID & FEDTAXID";
proc print data = out.&payer._pf_ok_all (obs = 40);
  id pnum provid fedtaxid;
  by pnum provid fedtaxid;
  var spec_n recnum lname fname midinit suffix credent recnum deanum business f_specn;
run;

data out.&payer._pf_chk_all;
  set &payer._pf_chk1_k_chk;
run;

proc print data = out.&payer._pf_chk_all (obs = 40);
  id pnum provid fedtaxid;
  by pnum provid fedtaxid;
  var spec_n recnum lname fname midinit suffix credent recnum deanum business f_specn;
  title "Multiple SPEC1 in %upcase(&payer) - based on PROVID & FEDTAXID";
run;


*=====================================================*;
* Check for PROVID with more than one SPECIALTY code *;
* Output two files:                                  *;
* - Specialty in Provider roster and in Encounter    *;
* - Specialty in Provider roster and not in Encounter*;
*=====================================================*;

data &payer._p_ok &payer._p_chk;
  set &payer._prv_p_k1;
  by provid;

if first.provid & last.provid then output &payer._p_ok;
else output &payer._p_chk;
run;

title "Samples of Problem Cases in %upcase(&payer) - more than 1 SPECIALTY based on
PROVID";
proc print data = &payer._p_chk (obs = 30);
  id pnum provid;
  by pnum provid;
run;

data &payer._p_chk1  /* in Encounter file */
     &payer._p_chk2  /* not in Encounter file */;
  length f_specn $1;
  merge &payer._enc_p (in = inenc)
        &payer._p_chk (in = inchk);
  by provid;

f_specn = put(spec_n, $spectyp.);
if inchk then do;
   if inenc then output &payer._p_chk1;
   else if not inenc then output &payer._p_chk2;
end;
run;
```

```
*=========================================================================-====*;
* Freq and Print Specialty (SPEC1) in Provider roster and in Encounter file *;
*=========================================================================*;

title "Frequencies of SPEC1 in %upcase(&payer) - PROVID is in Encounter File";
proc freq data = &payer._p_chk1;
  tables spec_n f_specn / list missing;
  tables f_specn * spec_n / list missing;
  format f_specn $sflag.;
run;

data &payer._p_chk1_k &payer._p_chk1_d;
  set &payer._p_chk1;

if f_specn = "1" then output &payer._p_chk1_k;
else output &payer._p_chk1_d;
run;

data &payer._p_chk1_k_ok &payer._p_chk1_k_chk;
  set &payer._p_chk1_k;
  by provid;

if first.provid & last.provid then output &payer._p_chk1_k_ok;
else output &payer._p_chk1_k_chk;
run;

title "Samples of SPEC1 is MISC/Invalid/Blank in %upcase(&payer) - PROVID is in Encounter
File";
proc print data = &payer._p_chk1_d (obs = 40);
  id pnum provid;
  by pnum provid;
  var spec_n fedtaxid recnum lname fname midinit suffix credent recnum deanum business
f_specn;
run;

*=========================================================================-====*;
* Freq and Print Specialty (SPEC1) in Provider roster not in Encounter file *;
*=========================================================================*;

title "Frequencies of SPEC1 in %upcase(&payer) - PROVID is not in Encounter File";
proc freq data = &payer._p_chk2;
  tables spec_n f_specn / list missing;
  tables f_specn * spec_n / list missing;
  format f_specn $sflag.;
run;

title "SPEC1 in %upcase(&payer) - PROVID is not in Encounter File";
proc print data = &payer._p_chk2 (obs = 40);
  id pnum provid;
  by pnum provid;
  var spec_n fedtaxid recnum lname fname midinit suffix credent recnum deanum business
f_specn;
run;

*============================*;
* create permanent data file *;
*============================*;

data &payer._p_ok_all;
  set &payer._p_ok &payer._p_chk1_k_ok;
run;

proc sort data = &payer._p_ok_all out = out.&payer._p_ok_all;
  by pnum provid;
run;

title "Good SPEC1 in %upcase(&payer) - based on PROVID";
proc print data = out.&payer._p_ok_all (obs = 40);
  id pnum provid;
  by pnum provid;
```

```
  var spec_n fedtaxid recnum lname fname midinit suffix credent recnum deanum business
f_specn;
run;

data out.&payer._p_chk_all;
  set &payer._p_chk1_k_chk;
run;

title "Multiple SPEC1 in %upcase(&payer) - based on PROVID";
proc print data = out.&payer._p_chk_all (obs = 40);
  id pnum provid;
  by pnum provid;
  var spec_n fedtaxid recnum lname fname midinit suffix credent recnum deanum business
f_specn;
run;


%mend A1;

%A1(payer = P020);


%jobend;


***** save log and output files *****;

dm "log;    print file =
I:\MCdb\datacol\P020\programs\Specialty_Check_2006_All_Payers.log replace";
dm "output; print file =
I:\MCdb\datacol\P020\programs\Specialty_Check_2006_All_Payers.lst replace";

proc datasets kill;
run;
quit;

%jobstart;


**=================================================================**;
**                                                                 **;
** Program:     pass1.sas (SPECIALTY PROGRAM #2)                   **;
** Date:        June 6, 2006                                       **;
** Purpose:     Match provider records with "single" specialty code **;
**              with records in ENC1, first by PROVID/FEDTACID and  **;
**              then by PROVID.                                    **;
** Input Ds     pxxx_enc1.sas7bdat                                 **;
**              pxxx_pf_ok_all.sas7bdat                            **;
**              pxxx_p_ok_all.sas7bdat                             **;
** Output Ds:   pxxx_roster_pass1.sas7bdat                         **;
** Note:        Update for 2005 data processing.                   **;
**=================================================================**;
**                                                                 **;
** Program:     pass1.sas                                          **;
** Programmer:  Po-Lun Chou                                        **;
** Date:        September 13, 2005                                 **;
**                                                                 **;
** Input Data:  pxxx_enc2.sas7bdat                                 **;
** Output Data: pxxx_roster_pass1.sas7bdat                         **;
**                                                                 **;
** Purpose:     Got SPECIALTY from separate data files.            **;
** Note:        The basic source of SPECIALTY is the Provider data  **;
**              file, and we used 2 ways to try to get the correct  **;
**              SPECIALTY: by both PROVID & FEDTAXID or by only     **;
**              PROVID.                                            **;
**                                                                 **;
** Modified by: Po-Lun Chou                                        **;
** Date:        September 30, 2005                                 **;
** Note:        Modified this program to include the original SPEC1 **;
**              - SPEC3.                                           **;
**                                                                 **;
```

```
** Modified by: Kasey Chen                                          **;
** Date:        August 13, 2006                                     **;
** Note:        Modified service year from 2005 to 2006             **;
**=================================================================**;

options obs = max center errors = 2 linesize = 123 pagesize = 96;

libname library "I:\mcdb\datacol\formats\permanent";

title1 "MARYLAND HEALTH CARE COMMISSION";

footnote1 "%sysfunc(putn(%sysfunc(time()), hhmm.)) %sysfunc(putn(%sysfunc(date()),
weekdate.))";
footnote2 "program: I:\MCdb\datacol\2006_Specialty\pass1.sas";


***** format *****;

proc format;
  value $spectyp

        '001','002','003','004','005','006','007','008','009','010',
        '011','012','013','014','015','016','017','018','019','020',
        '021','022','023','024','025','026','027','028','029','030',
        '031','032','033','035','036','039','100','040',
        '041','042','043','044','045','046','047','048','049','050',
        '051','052','053','054','055','056','057','058','059','060',
        '061','063','070',
        '071','072','073','074','075','076','080',
        '081','082','083','084','090',
        '091','092','102','101',
        '103' = '1' /* valid, not miscellaneous */
        '034','037','038','062','085',
        '093' = '2' /* valid, miscellaneous */
        ' '   = '4' /* missing */
        other = '3' /* invalid */;
   value $f_spec
        "1" = "1: Valid, Not Misc"
        "2" = "2: Valid, Misc"
        "3" = "3: Invalid"
        "4" = "4: Missing";
  value $f_linka
        "1" = "1: match with PROVID & FEDTAXID"
        "2" = "2: not match with PROVID & FEDTAXID";
  value $f_linkb
        "1" = "1: match with only PROVID"
        "2" = "2: not match with only PROVID";
  value $f_sour
        "PF" = "PF: SPEC based on PROVID & FEDTAXID"
        "P"  = "P: SPEC based on only PROVID";
  value $f_same
        "Y" = "Y: SPEC_PF matches SPEC_P"
        "N" = "N: SPEC_PF not Matches SPEC_P";
run;


%macro pass1(payer);

libname in  "I:\mcdb\datacol\&payer\sasdata";
libname out "I:\mcdb\datacol\&payer\sasdata";

title2 "Year 2006 Encounter File - &payer._Roster_Pass1";


***** bring in the Encounter data *****;

proc sort data = in.&payer._enc1 (keep = provid fedtaxid cpt pnum encnt_id)
          out = &payer._enc;
  by provid fedtaxid;
run;
```

```
***** bring in SPECIALTY *****;

/*** based on PROVID & FEDTAXID ***/

data &payer._spec_pf;
  set in.&payer._pf_ok_all (keep = pnum provid fedtaxid spec_n spec1 - spec3);

f_specn = put(spec_n, $spectyp.);
if f_specn in ("1", "2") then output;

rename spec_n  = spec_n_pf
       f_specn = f_specn_pf;
run;


/*** based on only PROVID ***/

data &payer._spec_p;
  set in.&payer._p_ok_all (keep = pnum provid fedtaxid spec_n spec1 - spec3);

f_specn = put(spec_n, $spectyp.);
if f_specn in ("1", "2") then output;

rename spec_n  = spec_n_p
       f_specn = f_specn_p;
run;


***** merge to get SPECIALTY *****;

*** first, merge files based on PROVID & FEDTAXID ***;

data &payer._enc_pf;
  length f_link1 $1;
  merge &payer._spec_pf (in = inspec)
        &payer._enc (in = inenc);
  by provid fedtaxid;

if inenc then do;
   if inspec then do;
      f_link1 = "1";
      output;
   end;
   else do;
      f_link1 = "2";
      output;
   end;
end;
run;

title4 "Frequencies in %upcase(&payer._enc_pf)";
proc freq data = &payer._enc_pf;
  tables spec_n_pf f_specn_pf f_link1 / list missing;
  format f_specn_pf $f_spec. f_link1 $f_linka.;
run;

proc freq data = &payer._enc_pf order = freq noprint;
  tables provid / list missing out = &payer._enc_pf_x;
  where spec_n_pf = " ";
run;

title4 "Records count of PROVID GE 100 and SPEC_N_PF = Blank in %upcase(&payer._enc_pf)";
proc print data = &payer._enc_pf_x;
  where count ge 100;
run;


*** second, merge files based on only PROVID ***;

data &payer._enc_pf_p;
```

```
  length f_link2 $1;
  merge &payer._spec_p (in = inspec)
        &payer._enc_pf (in = inenc);
  by provid;

if inenc then do;
   if inspec then do;
      f_link2 = "1";
      output;
   end;
   else do;
      f_link2 = "2";
      output;
   end;
end;
run;

title4 "Frequencies in %upcase(&payer._enc_pf_p)";
proc freq data = &payer._enc_pf_p;
  tables spec_n_p f_specn_p f_link2 / list missing;
  format f_specn_p $f_spec. f_link2 $f_linkb.;
run;

proc freq data = &payer._enc_pf_p order = freq noprint;
  tables provid / list missing out = &payer._enc_pf_p_x;
  where spec_n_p = " ";
run;

title4 "Records with count of PROVID GE 100 and SPEC_N_P = Blank in
%upcase(&payer._enc_pf_p)";
proc print data = &payer._enc_pf_p_x;
  where count ge 100;
run;


***** create final data file *****;

data &payer._enc_final;
  length newspec $3 f_newspec $1 f_source $2 f_same $1;
  set &payer._enc_pf_p;

/** if SPEC_N_PF = SPEC_N_P **/

  if spec_n_pf = spec_n_p then do;
    if spec_n_pf ne " " then do;
       newspec = spec_n_pf;
       f_source = "PF";
       f_same = "Y";
    end;
    else if spec_n_pf = " " then do;
       newspec = " ";
       f_source = " ";
       f_same = " ";
    end;
  end;

/** if SPEC_N_PF ~= SPEC_N_P **/

  else if spec_n_pf ne spec_n_p then do;
    if spec_n_pf ne " " & spec_n_p = " " then do;
       newspec = spec_n_pf;
       f_source = "PF";
       f_same = "N";
    end;
    else if spec_n_pf = " " & spec_n_p ne " " then do;
       newspec = spec_n_p;
       f_source = "P";
       f_same = "N";
    end;
    else if spec_n_pf ne " " & spec_n_p ne " " then do;
       if put(spec_n_pf, $spectyp.) = "1" and  put(spec_n_p, $spectyp.) = "2" then do;
```

```
          newspec = spec_n_pf;
          f_source = "PF";
          f_same = "N";
       end;
       else if put(spec_n_pf, $spectyp.) = "2" and  put(spec_n_p, $spectyp.) = "1" then
do;
          newspec = spec_n_p;
          f_source = "P";
          f_same = "N";
       end;
       else if put(spec_n_pf, $spectyp.) = "1" and  put(spec_n_p, $spectyp.) = "1" then
do;
          newspec = spec_n_pf;
          f_source = "PF";
          f_same = "N";
       end;
       else if put(spec_n_pf, $spectyp.) = "2" and  put(spec_n_p, $spectyp.) = "2" then
do;
          newspec = spec_n_pf;
          f_source = "PF";
          f_same = "N";
      end;
    end;
  end;

  f_newspec = put(newspec, $spectyp.);
run;


*** check ***;

title4 "Frequencies of SPEC Variables in %upcase(&payer._enc_final)";
proc freq data = &payer._enc_final;
  tables newspec * spec_n_pf * spec_n_p / list missing;
run;


*** final permanent data file ***;

data out.&payer._roster_pass1 (keep = encnt_id newspec provid fedtaxid pnum cpt
                                      f_source f_same f_newspec spec1 - spec3);
  set &payer._enc_final;
run;

title4 "Contents of %upcase(&payer._roster_pass1)";
proc contents data = out.&payer._roster_pass1;
run;

title4 "Frequencie in %upcase(&payer._roster_pass1)";
proc freq data = out.&payer._roster_pass1;
  tables newspec f_newspec f_source f_same / list missing;
  tables f_newspec * f_source / list missing;
  tables f_source * f_same / list missing;
  tables f_newspec * f_source * f_same / list missing nocum;
  format f_newspec $f_spec. f_source $f_sour. f_same $f_same.;
run;

title4 "Samples of Records in %upcase(&payer._roster_pass1)";
proc print data = out.&payer._roster_pass1 (obs = 20);
  var pnum provid fedtaxid encnt_id newspec f_newspec f_source f_same;
run;

%mend pass1;

%pass1(P020);


%jobend;


***** save log and output files *****;
```

```
dm "log;    print file = I:\MCdb\datacol\P020\programs\pass1.log replace";
dm "output; print file = I:\MCdb\datacol\P020\programs\pass1.lst replace";

proc datasets kill;
run;
quit;


%jobstart;

**=================================================================**;
** Program:     cpt_spec_check_2006.sas (SPECIALTY PROGRAM #3)      **;
** Date:        August 14, 2007                                     **;
** Input Data:  pxxx_roster_pass1.sas7bdat                          **;
** Output Data: all_cpt_spec1.sas7bdat                              **;
** Purpose:     Modify program for processing 2005 MCDB data.       **;
**=================================================================**;
**                                                                 **;
** Program:     CPT_Spec_Check_2004.sas                             **;
** Programmer:  Po-Lun Chou                                         **;
** Date:        September 14, 2005                                  **;
**                                                                 **;
** Input Data:  pxxx_roster_pass1.sas7bdat                          **;
** Output Data: all_cpt_spec1.sas7bdat                              **;
**                                                                 **;
** Purpose:     Got SPECIALTY from separate data files and added it **;
**              into the Encounter data file.                       **;
** Note:        The basic source of SPECIALTY is the Provider data  **;
**              file, and we used 2 ways to try to get the correct  **;
**              SPECIALTY: by both PROVID & FEDTAXID or by only     **;
**              PROVID.                                             **;
**                                                                 **;
** Modified by: Po-Lun Chou                                         **;
** Date:        September 26, 2005                                  **;
**                                                                 **;
** Modified by: Po-Lun Chou                                         **;
** Date:        September 30, 2005                                  **;
** Note:        Modified this program to include the original SPEC1 **;
**              - SPEC3.                                            **;
**                                                                 **;
** Modified by: Kasey Chen                                          **;
** Date:        August 14, 2007                                     **;
** Note:        Modified Service year from 2005 to 2006             **;
**                                                                 **;
** Modified by: Kasey Chen                                          **;
** Date:        September 6, 2007                                   **;
** Note:        P330 is excluded from this time check as Sophie     **;
**              notified.                                           **;
**                                                                 **;
** Modified by: Kasey Chen                                          **;
** Date:        September 12, 2007                                  **;
** Note:        P330 is excluded from this time check as Sophie     **;
**              notified.                                           **
**              Included P471 2ndsubmission                         **;
**=================================================================**;

options obs = max errors = 2 ls = 123 ps = 96;

%let yr = 2006;

libname out "I:\mcdb\datacol\&yr._specialty\data";

title1 "MARYLAND HEALTH CARE COMMISSION";
title2 "Year &yr Encounter File - All_CPT_SPEC1";

footnote1 "%sysfunc(putn(%sysfunc(time()), hhmm.)) %sysfunc(putn(%sysfunc(date()),
weekdate.))";
footnote2 "program: I:\MCdb\datacol\&yr._Specialty\programs\cpt_spec_check_&yr.sas";
```

```
***** get SPEC based on CPT *****;

%macro A1 (payer);

libname in "I:\mcdb\datacol\&payer\sasdata";

proc freq data = in.&payer._roster_pass1 noprint;
  tables cpt * newspec / out = &payer._cpt1;
  where newspec ne " ";
run;

proc append base = all_cpt1 data = &payer._cpt1;
run;

%mend A1;

%A1(P020);
%A1(P030);
%A1(P070);
%A1(P130);
%A1(P131);
%A1(P160);
%A1(P280);
%A1(P320);
%A1(P325);
*%A1(P330);
%A1(P350);
%A1(P471);
%A1(P480);
%A1(P500);
%A1(P510);
%A1(P520);
%A1(P530);
%A1(P620);
%A1(P680);
%A1(P760);
%A1(P820);
%A1(P830);
%A1(P850);
%A1(P870);



proc summary data = all_cpt1 (drop = count percent) nway missing;
  class cpt newspec;
  output out = out.all_cpt_spec1 (drop = _type_ rename = (_freq_ = freq));
run;

proc sort data = out.all_cpt_spec1 out = all_cpt2;
  by cpt descending freq;
run;


proc print data = all_cpt2 (obs = 500);
  by cpt;
  id cpt;
  var newspec freq;
  title1 "Crossed Frequencies of CPT & NEWSPEC for All Payers from
%upcase(all_cpt_spec1)";
run;


%jobend;


***** save log and output files *****;

dm "log;    print file = I:\MCdb\datacol\2006_Specialty\programs\cpt_spec_check_2006.log
replace";
dm "log;    print file = I:\MCdb\datacol\2006_Specialty\programs\cpt_spec_check_2006.log
replace";
```

```
dm "output; print file = I:\MCdb\datacol\2006_Specialty\programs\cpt_spec_check_2006.lst
replace";
dm "output; print file = I:\MCdb\datacol\2006_Specialty\programs\cpt_spec_check_2006.lst
replace";

proc datasets kill;
run;
quit;


%jobstart;

**=====================================================================**;
** Program:     pass2.sas (SPECIALTY PROGRAM #4)                       **;
** Date:        June 6, 2006                                           **;
** Input Data:  pxxx_roster_pass1.sas7bdat                             **;
**              all_cpt_spec1.sas7bdat                                 **;
** Output Data: pxxx_roster_pass2.sas7bdat                             **;
** Note:        Update for 2005 data processing.                       **;
**                                                                     **;
** Modified by: Po-Lun Chou                                            **;
** Date:        August 24, 2006                                        **;
** Purpose:     Assigned Specialty for the records which have multiple **;
**              SPECs for 2005 data.                                   **;
** Note:        1. MACRO A1 - For the payers which have multiple SPECs **;
**                            by using both PROVID & FEDTAXID and only **;
**                            PROVID to match Encounter data and       **;
**                            Provider data.  P070, P130, P131, P160,  **;
**                            P280, P330, P350, P471, P530, P680, P760 **;
**                            & P830 should use macro A1.              **;
**              2. MACRO A2 - For the payers which have multiple SPECs **;
**                            by using only PROVID to match Encounter  **;
**                            data and Provider data.  P320, P510,     **;
**                            P820, P850, & P870 should use macro A2.  **;
**              3. Following payers won't have Roster_Pass2 data file  **;
**                 because they don't have multiple SPECs:             **;
**                 P020, P030, P325, P480.
        **;
**              4. Following payers won't have Roster_Pass2 data file  **;
**                 because they don't have any records with NEWSPEC =  **;
**                 " " in Roster_Pass1 data file:                      **;
**                 P500, P520, & P620.                                 **;
**=====================================================================**;
**                                                                     **;
** Program:     pass2.sas                                              **;
** Programmer:  Tom Bell / Po-Lun Chou                                 **;
** Date:        September 19, 2005                                     **;
**                                                                     **;
** Input Data:  pxxx_roster_pass1.sas7bdat                             **;
** Output Data: pxxx_roster_pass2.sas7bdat                             **;
**                                                                     **;
** Purpose:     Assigned Specialty for the records which have multiple **;
**              SPECs.                                                 **;
** Note:        1. MACRO A1 - for the payers which have multiple SPECs **;
**                            using both PROVID & FEDTAXID and only    **;
**                            by PROVID to match Encounter data and    **;
**                            Provider data.                           **;
**              2. MACRO A2 - for the payers which have multiple SPECs **;
**                            by using only PROVID to match Encounter  **;
**                            data and Provider data.                  **;
**              3. Following payers won't have Roster_Pass2 data file  **;
**                 because they don't have multiple SPECs:             **;
**                 P020, P030, P130, P325, P480, P500, P760, p820, &   **;
**                 P870.                                               **;
**                                                                     **;
** Modified by: Po-Lun Chou                                            **;
** Date:        September 30, 2005                                     **;
** Note:        Modified this program to include the original SPEC1 -  **;
**              SPEC3.                                                 **;
**                                                                     **;
** Modified by: Kasey Chen                                             **;
```

```
** Date:        September 6, 2007                                    **;
** Note:        1. Changed service year from 2005 to 2006.           **;
**              2. P330 is excluded from this time check as          **;
**                 Sophe notified.                                   **;
**                                                                   **;
** Modified by: Kasey Chen                                           **;
** Date:        September 12, 2007                                   **;
** Note:        Included P471 2nd submission.                        **;
**===================================================================**;

options obs = max center errors = 2 linesize = 123 pagesize = 96;

libname in "I:\mcdb\datacol\2006_specialty\data";

title1 "MARYLAND HEALTH CARE COMMISSION";

footnote1 "%sysfunc(putn(%sysfunc(time()), hhmm.)) %sysfunc(putn(%sysfunc(date()),
weekdate.))";
footnote2 "program: I:\mcdb\datacol\2006_specialty\programs\pass2.sas";


***** format *****;

proc format;
  value $spectyp
        '001','002','003','004','005','006','007','008','009','010',
        '011','012','013','014','015','016','017','018','019','020',
        '021','022','023','024','025','026','027','028','029','030',
        '031','032','033','035','036','039','100','040',
        '041','042','043','044','045','046','047','048','049','050',
        '051','052','053','054','055','056','057','058','059','060',
        '061','063','070',
        '071','072','073','074','075','076','080',
        '081','082','083','084','090',
        '091','092','102','101',
        '103' = '1' /* valid, not miscellaneous */
        '034','037','038','062','085',
        '093' = '2' /* valid, miscellaneous */
        ' '   = '4' /* missing */
        other = '3' /* invalid */;
     value $f_spec
           "1" = "1: Valid, Not Misc"
           "2" = "2: Valid, Misc"
           "3" = "3: Invalid"
           "4" = "4: Missing";
     value $f_linka
           "1" = "1: match with PROVID & FEDTAXID"
           "2" = "2: not match with PROVID & FEDTAXID";
     value $f_linkb
           "1" = "1: match with only PROVID"
           "2" = "2: not match with only PROVID";
     value $f_sour
           "PF" = "PF: SPEC based on PROVID & FEDTAXID"
           "P"  = "P: SPEC based on only PROVID";
     value $f_same
           "Y" = "Y: SPEC_PF matches SPEC_P"
           "N" = "N: SPEC_PF not Matches SPEC_P";
run;


proc sort data = in.all_cpt_spec1 out = all_cpt;
     by cpt newspec;
run;


%macro A1(payer);

libname out "I:\MCdb\datacol\&payer\sasdata";

title2 "Year 2006 Encounter File - &payer._Roster_Pass2";
```

```
/** based on PROVID & FEDTAXID **/

proc sort data = out.&payer._roster_pass1 (keep = provid fedtaxid newspec cpt pnum
encnt_id)
          out = payer_roster_pf (drop = newspec) nodupkey;
  by provid fedtaxid;
  where newspec = " ";
run;

proc sort data = out.&payer._pf_chk_all (keep = provid fedtaxid spec_n)
          out = pf_duplicate_newspec;
  by provid fedtaxid;
run;

data pf_withspec;
  merge payer_roster_pf (in = a)
        pf_duplicate_newspec (in = b);
  by provid fedtaxid;

if a and b then output;
run;

proc sort data = pf_withspec;
  by cpt spec_n;
run;

data pf_withcptfreq (rename = (newspec = spec_n_pf));
  merge pf_withspec (in = a rename = (spec_n = newspec))
        all_cpt (in = b);
  by cpt newspec;

if a then output;
run;

proc sort data = pf_withcptfreq;
  by provid fedtaxid descending freq;
run;

data payer_roster_2;
  set pf_withcptfreq;
  by provid fedtaxid;

if first.fedtaxid then do;
   f_specn_pf = put(spec_n_pf, $spectyp.);
   if spec_n_pf ne " " then f_link1 = "1";
   else f_link1 = "2";
   output;
end;
run;

title4 "Frequencies after Provider and Taxid Match - &payer";
proc freq data = payer_roster_2;
  tables spec_n_pf f_specn_pf f_link1 / list missing;
  format f_specn_pf $f_spec. f_link1 $f_linka.;
run;


/** based on PROVID **/

proc sort data = out.&payer._roster_pass1 (keep = provid fedtaxid newspec cpt pnum
encnt_id)
          out = payer_roster_p (drop = newspec) nodupkey;
  by provid;
  where newspec = " ";
run;

proc sort data = out.&payer._p_chk_all (keep = provid spec_n)
          out = p_duplicate_newspec;
  by provid;
run;
```

```
data p_withspec;
  merge payer_roster_p (in = a)
        p_duplicate_newspec (in = b);
  by provid;


if a and b then output;
run;

proc sort data = p_withspec;
  by cpt spec_n;
run;

data p_withcptfreq (rename = (newspec = spec_n_p));
  merge p_withspec (in = a rename = (spec_n = newspec))
           all_cpt (in = b);
  by cpt newspec;

if a then output;
run;

proc sort data = p_withcptfreq;
  by provid descending freq;
run;

data payer_roster_3;
  set p_withcptfreq;
  by provid;

if first.provid then do;
   f_specn_p = put(spec_n_p, $spectyp.);
   if spec_n_p ne " " then f_link2 = "1";
   else f_link2 = "2";
   output;
end;
run;

title4 "Frequencies after Provider Match - &payer";
proc freq data = payer_roster_3;
  tables spec_n_p f_specn_p f_link2 / list missing;
  format f_specn_p $f_spec. f_link2 $f_linkb.;
run;



***** create final data file *****;

proc sort data = out.&payer._roster_pass1 (keep = provid fedtaxid newspec cpt pnum
encnt_id
                                                   spec1 - spec3)
  out = payer_roster (drop = newspec);
  by provid fedtaxid;
  where newspec = " ";
run;

data &payer._enc_f1;
  merge payer_roster_2 (in = a drop = pnum encnt_id cpt)
        payer_roster (in = b);
  by provid fedtaxid;

if b then output;
run;

data &payer._enc_f2;
  merge payer_roster_3 (in = a drop = pnum encnt_id cpt)
        &payer._enc_f1 (in = b);
  by provid;

if b then output;
run;
```

```
data &payer._enc_final;
  length newspec $3 f_nspec $1 f_source $2 f_same $1;
  set &payer._enc_f2;

/** if SPEC_N_PF = SPEC_N_P **/

if spec_n_pf = spec_n_p then do;
   if spec_n_pf ne " " then do;
      newspec = spec_n_pf;
      f_source = "PF";
      f_same = "Y";
   end;
   else if spec_n_pf = " " then do;
      newspec = " ";
      f_source = " ";
      f_same = " ";
   end;
end;

/** if SPEC_N_PF ~= SPEC_N_P **/

else if spec_n_pf ne spec_n_p then do;
   if spec_n_pf ne " " & spec_n_p = " " then do;
      newspec = spec_n_pf;
      f_source = "PF";
      f_same = "N";
   end;
   else if spec_n_pf = " " & spec_n_p ne " " then do;
      newspec = spec_n_p;
      f_source = "P";
      f_same = "N";
   end;
   else if spec_n_pf ne " " & spec_n_p ne " " then do;
      if put(spec_n_pf, $spectyp.) = "1" and  put(spec_n_p, $spectyp.) = "2" then do;
         newspec = spec_n_pf;
         f_source = "PF";
         f_same = "N";
      end;
      else if put(spec_n_pf, $spectyp.) = "2" and  put(spec_n_p, $spectyp.) = "1" then
do;
         newspec = spec_n_p;
         f_source = "P";
         f_same = "N";
      end;
      else if put(spec_n_pf, $spectyp.) = "1" and  put(spec_n_p, $spectyp.) = "1" then
do;
         newspec = spec_n_pf;
         f_source = "PF";
         f_same = "N";
      end;
      else if put(spec_n_pf, $spectyp.) = "2" and  put(spec_n_p, $spectyp.) = "2" then
do;
         newspec = spec_n_pf;
         f_source = "PF";
         f_same = "N";
      end;
   end;
end;

f_nspec = put(newspec, $spectyp.);
run;


*** check ***;

title4 "Frequencies of SPEC Variables in %upcase(&payer._enc_final)";
proc freq data = &payer._enc_final;
  tables newspec * spec_n_pf * spec_n_p / list missing;
run;
```

```
***** final permanent data file *****;

data out.&payer._roster_pass2 (keep = encnt_id newspec provid fedtaxid pnum cpt
                                      f_source f_same f_nspec spec1 - spec3);
  set &payer._enc_final;
run;

title4 "Contents of %upcase(&payer._roster_pass2)";
proc contents data = out.&payer._roster_pass2;
run;

title4 "Frequencie in %upcase(&payer._roster_pass2)";
proc freq data = out.&payer._roster_pass2;
  tables newspec f_nspec f_source f_same / list missing;
  tables f_nspec * f_source / list missing;
  tables f_source * f_same / list missing;
  tables f_nspec * f_source * f_same / list missing nocum;
  format f_nspec $f_spec. f_source $f_sour. f_same $f_same.;
run;

title4 "Samples of Records in %upcase(&payer._roster_pass2)";
proc print data = out.&payer._roster_pass2 (obs = 20);
  var pnum provid fedtaxid encnt_id newspec f_nspec f_source f_same;
run;

%mend A1;

%A1(P070);
%A1(P130);
%A1(P131);
%A1(P160);
%A1(P280);
*%A1(P330);
%A1(P350);
%A1(P471);
%A1(P510);
%A1(P530);
%A1(P620);
%A1(P680);
%A1(P760);
%A1(P830);


%macro A2(payer);

libname out "I:\MCdb\datacol\&payer\sasdata";

title2 "Year 2006 Encounter File - &payer._Roster_Pass2";

/** based on PROVID **/

proc sort data = out.&payer._roster_pass1 (keep = provid fedtaxid newspec cpt pnum
encnt_id)
          out = payer_roster_p (drop = newspec) nodupkey;
     by provid;
     where newspec = " ";
run;

proc sort data = out.&payer._p_chk_all (keep = provid spec_n)
          out = p_duplicate_newspec;
     by provid;
run;

data p_withspec;
     merge payer_roster_p (in = a)
           p_duplicate_newspec (in = b);
     by provid;
     if a and b then output;
run;
```

```
proc sort data = p_withspec;
     by cpt spec_n;
run;

data p_withcptfreq (rename = (newspec = spec_n_p));
     merge p_withspec (in = a rename = (spec_n = newspec))
           all_cpt (in = b);
     by cpt newspec;
     if a then output;
run;

proc sort data = p_withcptfreq;
     by provid descending freq;
run;

data payer_roster_3;
     set p_withcptfreq;
     by provid;
     if first.provid then do;
        f_specn_p = put(spec_n_p, $spectyp.);
        if spec_n_p ne " " then f_link2 = "1";
        else f_link2 = "2";
        output;
     end;
run;

proc freq data = payer_roster_3;
     tables spec_n_p f_specn_p f_link2 / list missing;

     format f_specn_p $f_spec. f_link2 $f_linkb.;
     title4 "Frequencies after Provider Match - &payer";
run;


***** create final data file *****;

proc sort data = out.&payer._roster_pass1 (keep = provid fedtaxid newspec cpt pnum
encnt_id
                                                    spec1 - spec3)
          out = payer_roster (drop = newspec);
     by provid fedtaxid;
     where newspec = " ";
run;

data &payer._enc_f1;
     merge payer_roster_3 (in = a drop = pnum encnt_id cpt)
           payer_roster (in = b);
     by provid;
     if b then output;
run;

data &payer._enc_final;
     length newspec $3 f_nspec $1 f_source $2 f_same $1;
     set &payer._enc_f1;

     if spec_n_p ne " " then do;
        newspec = spec_n_p;

        f_source = "P";
        f_same = " ";
     end;
     else if spec_n_p = " " then do;
             newspec = " ";
             f_source = " ";
             f_same = " ";
     end;

     f_nspec = put(newspec, $spectyp.);
run;
```

```
*** check ***;

proc freq data = &payer._enc_final;
     tables newspec * spec_n_p / list missing;
     title4 "Frequencies of SPEC Variables in %upcase(&payer._enc_final)";
run;


***** final permanent data file *****;

data out.&payer._roster_pass2 (keep = encnt_id newspec provid fedtaxid pnum cpt
                                      f_source f_same f_nspec spec1 - spec3);
     set &payer._enc_final;
run;

title4 "Contents of %upcase(&payer._roster_pass2)";
proc contents data = out.&payer._roster_pass2;
run;

title4 "Frequencie in %upcase(&payer._roster_pass2)";
proc freq data = out.&payer._roster_pass2;
     tables newspec f_nspec f_source f_same / list missing;
     tables f_nspec * f_source / list missing;
     tables f_source * f_same / list missing;
     tables f_nspec * f_source * f_same / list missing nocum;
     format f_nspec $f_spec. f_source $f_sour. f_same $f_same.;
run;

title4 "Samples of Records in %upcase(&payer._roster_pass2)";
proc print data = out.&payer._roster_pass2 (obs = 20);
     var pnum provid fedtaxid encnt_id newspec f_nspec f_source f_same;
run;

%mend A2;

%A2(P320);
%A2(P500);
%A2(P520);
%A2(P820);
%A2(P870);


%jobend;


***** save log and output files *****;

dm "log;    print file = I:\MCdb\datacol\2006_Specialty\programs\pass2.log replace";
dm "log;    print file = I:\MCdb\datacol\2006_Specialty\programs\pass2.log replace";
dm "output; print file = I:\MCdb\datacol\2006_Specialty\programs\pass2.lst replace";
dm "output; print file = I:\MCdb\datacol\2006_Specialty\programs\pass2.lst replace";

proc datasets kill;
run;
quit;


%jobstart;

**================================================================**;
** Program:     pass3.sas (SPECIALTY PROGRAM #5)                  **;
** Date:        June 7, 2006                                      **;
** Input Ds:    all_cpt_spec1.sas7bdat                            **;
**              pxxx_roster_pass2.sas7bdat                        **;
** Output Ds:   pxxx_roster_pass3.sas7bdat                        **;
**                                                                **;
** Modified by: Po-Lun Chou                                       **;
** Date:        August 25, 2006                                   **;
** Purpose:     Assigned specialties for the records which don't **;
**              have specialties after using PROVID/FEDTAXID to  **;
**              detect them by using 2 ways, matched on CPT or   **;
```

```
**              randomly assigned, for 2005 data.                **;
** Note:        Following payers don't have to ne ran here since **;
**              they don't have any records with NEWSPEC = " "   **;
**              in either Roster_Pass1 or Roster_Pass2:          **;
**              1. none in Roster_Pass1:                         **;
**                 P030, P500, P520, & P620.                     **;
**              2. none in Roster_Pass2:                         **;
**                 P680.                                         **;
**================================================================**;
**                                                                **;
** Program:     pass3.sas                                        **;
** Programmer:  Tom Bell                                         **;
** Date:        September 23, 2005                               **;
**                                                                **;
** Purpose:     Assigned specialties for the records which don't **;
**              have specialties after using PROVID/FEDTAXID to  **;
**              detect them by using 2 ways: matched on CPT or   **;
**              randomly assigned.                               **;
**                                                                **;
** Modified by: Po-Lun Chou                                      **;
** Date:        September 30, 2005                               **;
** Note:        Modified this program to include the original    **;
**              SPEC1 - SPEC3.                                   **;
**                                                                **;
**                                                                **;
** Modified by: Kasey Chen                                       **;
** Date:        September 7, 2007                                **;
** Note:        Changed service year for 2006                    **;
**              1. Added Payers P620 and P680 as pass2.          **;
**              2. Added Payers P850 as pass1.                   **;
**              3. None in Roster_Pass1:                         **;
**                 P030, P500, and P520.                         **;
**              4. none in Roster_Pass2:                         **;
**                 P500, P520, and P850.                         **;
**              5. P330 was excluded for cpt_spec_check          **;
**                                                                **;
** Modified by: Kasey Chen                                       **;
** Date:        September 12, 2007                               **;
** Note:        Run P471 2nd submission                          **;
**================================================================**;

options obs = max errors = 2;

libname in "I:\MCdb\datacol\2006_Specialty\data";

title1 "MARYLAND HEALTH CARE COMMISSION";

footnote1 "%sysfunc(putn(%sysfunc(time()), hhmm.)) %sysfunc(putn(%sysfunc(date()),
weekdate.))";
footnote2 "program: I:\mcdb\datacol\2006_Specialty\programs\pass3.sas";


***** format *****;

proc format;
     value fnewspec
           1 = "1: Matched on CPT"
           2 = "2: Randomly Assigned";
run;


***** create working file based on CPT Spec file *****;

proc summary data = in.all_cpt_spec1 ;
  var freq;
  class cpt;
  output out = sum_by_cpt (keep = cpt cptsum) sum = cptsum;
run;

data cpt_spec (keep = cpt newspec random);
  merge in.all_cpt_spec1 (in = a)
```

```
        sum_by_cpt (in = b);
  by cpt;

if cptsum < 100 then multiplier = ceil(100/cptsum);
else multiplier = 1;
  loop_end = freq * multiplier;
   do i = 1 to loop_end;
      random = ranuni(2006);
   output;
end;
run;

proc sort data = cpt_spec;
   by cpt random;
run;

data in.cpt_spec_pass3 (keep = cpt newspec rnumber);
   set cpt_spec;
   by cpt;

retain counter;
if first.cpt = 1 then counter = 1;
else counter = counter + 1;
if counter le 100 then do;
   rnumber = counter;
   output;
end;
run;

proc sort data = cpt_spec;
  by random;
run;

data in.cpt_random_pass3 (keep = rnumber newspec);
   set cpt_spec (obs = 100);

rnumber = _n_;
run;



***** assign new SPEC *****;

%macro pass3(payer, file);

libname out "I:\MCdb\datacol\p&payer\sasdata";

title2 "Year 2006 Encounter File - p&payer._Roster_Pass3";

data payer_p&payer (keep = encnt_id cpt rnumber spec1 - spec3);
  set out.p&payer._roster_&file;

if newspec = '' then do;
   rnumber = int((ranuni(&payer) * 100));
   if rnumber = 0 then rnumber = 1;
   output;
end;
run;

proc sort data = payer_p&payer;
  by cpt rnumber;
run;

data p&payer._roster_pass3a p&payer._cpt_not_found (drop = newspec);
  merge payer_p&payer (in = a)
        in.cpt_spec_pass3 (in = b);
  by cpt rnumber;

if a and b then output p&payer._roster_pass3a ;
else if a then output p&payer._cpt_not_found;
run;
```

```
title4 "CPT Codes not in Look-up Table for Payer p&payer";
proc freq data = p&payer._cpt_not_found;
  tables cpt / list missing;
run;

proc sort data = p&payer._cpt_not_found;
  by rnumber;
run;

data p&payer._roster_pass3b;
  merge p&payer._cpt_not_found (in = a)
        in.cpt_random_pass3 (in = b);
  by rnumber;

if a then output;
run;

data out.p&payer._roster_pass3;
  set p&payer._roster_pass3a (in = a)
      p&payer._roster_pass3b (in = b);

if a then f_newspec = 1;
else f_newspec = 2;
run;

title4 "PASS 3 Results for p&payer";
proc freq data = out.p&payer._roster_pass3;
  tables newspec f_newspec / list missing;
  format f_newspec fnewspec.;
run;

%mend pass3;


*** payers have PASS2 file ***;

%pass3(070, pass2);
%pass3(130, pass2);
%pass3(131, pass2);
%pass3(160, pass2);
%pass3(280, pass2);
%pass3(320, pass2);
*%pass3(330, pass2);
%pass3(350, pass2);
%pass3(471, pass2);
%pass3(510, pass2);
%pass3(530, pass2);
%pass3(620, pass2);
%pass3(680, pass2);
%pass3(820, pass2);
%pass3(830, pass2);
%pass3(870, pass2);
%pass3(760, pass2);

*** payers do not have PASS2 file ***;

%pass3(020, pass1);
%pass3(325, pass1);
%pass3(480, pass1);
%pass3(850, pass1);


%jobend;


***** save log and output files *****;

dm " ";
dm "log;    print file = I:\MCdb\datacol\2006_Specialty\programs\pass3.log replace";
dm " ";
```

```
dm "output; print file = I:\MCdb\datacol\2006_Specialty\programs\pass3.lst replace";

proc datasets kill;
run;
quit;


%jobstart;

**============================================================**;
**                                                            **;
** Program:     final_specailty.sas                           **;
** Date:        June 6, 2006                                  **;
** Input Ds:    pxxx_roster_pass1.sas7bdat                    **;
**              pxxx_roster_pass2.sas7bdat                    **;
**              pxxx_roster_pass3.sas7bdat                    **;
** Output Ds:   pxxx_final_specialty.sas7bdat                 **;
**============================================================**;
**                                                            **;
** Program:     final_specialty.sas                           **;
** Programmer:  Tom Bell                                      **;
** Date:        September 23, 2005                            **;
**                                                            **;
** Purpose:     Put all specialty together for each payer.    **;
**                                                            **;
** Modified by: Po-Lun Chou                                   **;
** Date:        September 30, 2005                            **;
** Note:        Modified this program to include the original **;
**              SPEC1 - SPEC3.                                **;
**                                                            **;
** Modified by: Kasey Chen                                    **;
** Date:        September 7 , 2007                            **;
** Note:        1. Changed service year from 2005 to 2006     **;
**              2. Excluded P330.                             **;
**              3. Added P620 and P680 in category 'a'        **;
**              4. Removed P850 from category 'a' to 'b'      **;
**              5. Added P500 and P520 in category 'c'        **;
**              6. Left only P030 in category 'd'             **;
**                                                            **;
** Modified by: Kasey Chen                                    **;
** Date:        September 12 , 2007                           **;
** Note:        Run P471 2nd submission                       **;
**============================================================**;

options obs = max errors = 2;

libname in "I:\MCdb\datacol\2006_Specialty\data";

title1 "MARYLAND HEALTH CARE COMMISSION";

footnote1 "%sysfunc(putn(%sysfunc(time()), hhmm.)) %sysfunc(putn(%sysfunc(date()),
weekdate.))";
footnote2 "program: I:\mcdb\datacol\2006_Specialty\programs\final_specialty.sas";


***** formats *****;

proc format;
     value source
           1 = "1: Provider with Single Specialty Code"
           2 = "2: Provider with Multiple Specialty Codes"
           3 = "3: No Provider - Random";
run;

**=========================================================**;
** create data file with final SPEC                        **;
** for the payers have data files PASS1, PASS2, and PASS3 **;

**=========================================================**;

%macro final_a(payer);
```

```
libname out "I:\MCdb\datacol\&payer\sasdata";

title2 "Year 2006 Encounter File - &payer._final_specialty";

data out.&payer._final_specialty (keep = encnt_id newspec spec_source spec1 - spec3);
     set out.&payer._roster_pass1 (in = a keep = encnt_id newspec spec1 - spec3
                                   where = (newspec ne " "))
         out.&payer._roster_pass2 (in = b keep = encnt_id newspec spec1 - spec3
                                   where = (newspec ne " "))
         out.&payer._roster_pass3 (in = c keep = encnt_id newspec spec1 - spec3);
     if a then spec_source = 1;
     else if b then spec_source = 2;
     else if c then spec_source = 3;
run;

proc contents data = out.&payer._final_specialty;
     title4 "Contents of %upcase(&payer._final_specialty)";
run;

proc freq data = out.&payer._final_specialty;
     tables newspec spec_source / list missing;
     format spec_source source.;
     title4 "Final Specialties and Sources for &payer";
run;

proc print data = out.&payer._final_specialty (obs = 25);
     var encnt_id newspec spec_source spec1;
     title4 "Samples of Records from %upcase(&payer._final_specialty)";
run;

%mend final_a;

%final_a(P070);
%final_a(P130);
%final_a(P131);
%final_a(P160);
%final_a(P280);
%final_a(P320);
**%final_a(P330);
%final_a(P350);
%final_a(P471);
%final_a(P510);
%final_a(P530);
%final_a(P620);
%final_a(P680);
%final_a(P760);
%final_a(P820);
%final_a(P830);
**%final_a(P850);
%final_a(P870);



**==================================================**;
** for the payers have data files PASS1 and PASS3 **;
**==================================================**;

%macro final_b(payer);

libname out "I:\MCdb\datacol\&payer\sasdata";

title2 "Year 2006 Encounter File - &payer._final_specialty";

data out.&payer._final_specialty (keep = encnt_id newspec spec_source spec1 - spec3);
     set out.&payer._roster_pass1 (in = a keep = encnt_id newspec spec1 - spec3
                                   where = (newspec ne " "))
         out.&payer._roster_pass3 (in = c keep = encnt_id newspec spec1 - spec3);
     if a then spec_source = 1;
     else if c then spec_source = 3;
run;
```

```
proc contents data = out.&payer._final_specialty;
     title4 "Contents of %upcase(&payer._final_specialty)";
run;

proc freq data = out.&payer._final_specialty;
     tables newspec spec_source / list missing;
     format spec_source source.;
     title4 "Final Specialties and Sources for &payer";
run;

proc print data = out.&payer._final_specialty (obs = 25);
     var encnt_id newspec spec_source spec1;
     title4 "Samples of Records from %upcase(&payer._final_specialty)";
run;

%mend final_b;

%final_b(P020);
%final_b(P325);
%final_b(P480);
%final_b(P850);

**==================================================**;
** for the payers have data files PASS1 and PASS2 **;
**==================================================**;

%macro final_c(payer);

libname out "I:\MCdb\datacol\&payer\sasdata";

title2 "Year 2006 Encounter File - &payer._final_specialty";

data out.&payer._final_specialty (keep = encnt_id newspec spec_source spec1 - spec3);
     set out.&payer._roster_pass1 (in = a keep = encnt_id newspec spec1 - spec3
                                   where = (newspec ne " "))
         out.&payer._roster_pass2 (in = b keep = encnt_id newspec spec1 - spec3
                                   where = (newspec ne " "));
     if a then spec_source = 1;
     else if b then spec_source = 2;
run;

proc contents data = out.&payer._final_specialty;
     title4 "Contents of %upcase(&payer._final_specialty)";
run;

proc freq data = out.&payer._final_specialty;
     tables newspec spec_source / list missing;
     format spec_source source.;
     title4 "Final Specialties and Sources for &payer";
run;

proc print data = out.&payer._final_specialty (obs = 25);
     var encnt_id newspec spec_source spec1;
     title4 "Samples of Records from %upcase(&payer._final_specialty)";
run;

%mend final_c;

%final_c(P500);
%final_c(P520);

**=============================================**;
** for the payers have data file PASS1 only **;
**=============================================**;

%macro final_d(payer);

libname out "I:\MCdb\datacol\&payer\sasdata";

title2 "Year 2006 Encounter File - &payer._final_specialty";
```

```
data out.&payer._final_specialty (keep = encnt_id newspec spec_source spec1 - spec3);
     set out.&payer._roster_pass1 (in = a keep = encnt_id newspec spec1 - spec3
                                   where = (newspec ne " "));
     if a then spec_source = 1;
run;

proc contents data = out.&payer._final_specialty;
     title4 "Contents of %upcase(&payer._final_specialty)";
run;

proc freq data = out.&payer._final_specialty;
     tables newspec spec_source / list missing;
     format spec_source source.;
     title4 "Final Specialties and Sources for &payer";
run;

proc print data = out.&payer._final_specialty (obs = 25);
     var encnt_id newspec spec_source spec1;
     title4 "Samples of Records from %upcase(&payer._final_specialty)";
run;

%mend final_d;

%final_d(P030);
*%final_d(P500);
*%final_d(P520);
*%final_d(P620);


%jobend;


***** save log and output files *****;

dm "log;    print file = I:\MCdb\datacol\2006_Specialty\programs\final_specialty.log
replace";
dm "log;    print file = I:\MCdb\datacol\2006_Specialty\programs\final_specialty.log
replace";
dm "output; print file = I:\MCdb\datacol\2006_Specialty\programs\final_specialty.lst
replace";
dm "output; print file = I:\MCdb\datacol\2006_Specialty\programs\final_specialty.lst
replace";

proc datasets kill;
run;
quit;


%jobstart;

**=================================================================**;
**                                                                 **;
** Program:    P020_enc2.sas (was P020_enc4.sas)                   **;
** Programmer: Adrien Ndikumwami/Chris Hogan                       **;
** Date:       September 25, 2001                                  **;
**                                                                 **;
** This program assign RVUs (Work,Practice, and Malpractice) to    **;
** valid procedures. Medicare Payment Rates are also computed.     **;
** The components are estimated for procedures with imputed RVUs   **;
** The following format is needed to run the program:              **;
** Format_RVU_2006.sas                                             **;
**                                                                 **;
** Adapted to 2004 MCDB on June 9, 2005-Adrien                     **;
** Adapted to 2005 MCDB on July 10, 2006 - Sireesha                **;
**                                                                 **;
** Modified by: Po-Lun Chou                                        **;
** Date:        October 4, 2006                                    **;
** Purpose:     1. Assigned RVU                                    **;
**              2. Added odd service flag                          **;
** Note:        We used to assign RVU in Pxxx_enc2.sas, but since we **;
```

```
**              changed processing order this year for 2005 data and **;
**              decided to assign RVU later, we changed this program **;
**              from Pxxx_enc2.sas to Pxxx_enc4.sas.                 **;
**                                                                   **;
** Modified by: Po-Lun Chou                                          **;
** Date:        August 16, 2007                                      **;
** Purpose:     Assign RVU and odd service flag, and bring back      **;
**              deletion flags.                                      **;
** Note:        1)  Per Ben's request, we have to have RVU included  **;
**                  in the Enc1 changing report. For having RVU, we  **;
**                  modified the original Enc4.sas (in 2005) and     **;
**                  saved it as a new program, Enc2.sas.             **;
**              2)  The programs we use to create new RVUs have to   **;
**                  be completed before running this program. Those  **;
**                  programs are as below:                           **;
**                    1. rvu_imputation_subset.sas,                  **;
**                    2. Compute_Imputed_RVU_2006.sas, &             **;
**                    3. Compute_Impute_RVU_Components_2006.sas.     **;
**                  (Please refer to Rvu_Process.xls for details.)   **;
**              3)  The deletion flag variables (d_***_f) should be  **;
**                  included in the Enc1 data file, but we forgot to **;
**                  modified the codes to include them in program    **;
**                  P***_enc1.sas this time, so we have to bring     **;
**                  these variables back from P***_Enc1_D_Flags data **;
**                  file. This issue has to be modified in           **;
**                  P***_enc1.sas next year.                         **;
**              4)  For comparing this year's RVU values with        **;
**                  previous year's values, Adrien and I discussed   **;
**                  it and decided to add the whole section of       **;
**                  assigning odd services flags here.  However,     **;
**                  because NEWSPEC has not been created yet (it     **;
**                  will be created in Enc3), I commented the        **;
**                  screens regarding to NEWSPEC out below, and for  **;
**                  this reason, variable F_ODDSVC which will be     **;
**                  created later in this progeam is just a          **;
**                  temporary variable, it has to be re-created as a **;
**                  permanent variable in Enc4.                      **;
**                                                                   **;
**===================================================================**;

options obs = max nomprint nosource2 errors = 2 linesize = 120 pagesize = 92;

%let payernum = P020;
%let yr = 2006;
%let year = 06;
%let nextyr = 07;

libname library "I:\MCdb\datacol\Formats\PERMANENT\RVU_formats";
libname inout   "I:\MCdb\datacol\&payernum\sasdata";
libname outj    "J:\mcdb_raw";

%let sasDTin  = &payernum._enc1;
%let sasDTout = &payernum._enc2;


**========================================**

               Assign RVU

**========================================**;

/** Note from Po-Lun on 08-06-2007:                                    **/
/** format program Format_RVU_2006.sas should be ran before running this program **/

* rvu and payment formats ;

*************************************************************************;
* NOTES FOR UPDATING THE CODE TO PROCESS 2006 CLAIMS DATA               *;
* Since 2002, MHCC wants not only the total RVU, but also the individual    *;
* components of the RVU (work, malpractice, practice expense) and,          *;
* what is far more difficult, the actual Medicare price for the service.    *;
```

```
* That requires:                                                        *;
*  1) Adding the GPCI (geographic pricing) data by service              *;
*     location (ZIP code mapped to Part B carrier locality),             *;
*  2) Gap filling that as required,                                      *;
*  3) Adding the actual lab fee schedule amounts, which are separate     *;
*     for each state                                                     *;
*  4) Gap filling lab fees.                                              *;
*  5) Assigning the 2006 RVU components.                                 *;
*  6) Identifying modifiers that would change the Medicare price, and    *;
*     either changing the Medicare price there or flagging it as not     *;
*     the correct price                                                  *;
*  7) Updating all the service-specific things like the price for        *;
*     mammograms                                                         *;
*     If we go back to redo older files, we have to re-create            *;
*     all the gpci and lab fee schedule information for the earlier years *;
*     The code is not written to price out all of the pricing-relevant   *;
*     modifiers under Medicare.These will typically be marked with F_ODDSVC *;
*     values.  The user must know enough to throw out the records with   *;
*     F_ODDSVC,and not to use the RVU or Medicare price data marked on   *;
*     those records.                                                     *;
*                                                                        *;
*  Other aspects of Medicare payment are also not modeled.  For example, *;
*  the status T codes are only paid if they are the only thing the doc does *;
*  for the patient on that day.  I have not done any such checking of    *;
*  relationships among all procedures billed by a doc on a date for a    *;
*  patient.                                                              *;
*************************************************************************;

*********************************************************************;
*1. Crosswalk provider zip to carrier locality                      *;
*   Use zip of location of service (service location zip)           *;
*   If that does not work, use beneficiary's zip as gap fill        *;
*   If still no valid carrier locality, wait and put in imputed values *;
*   for the numerical values that you need, using the all-MD averages *;
*   rather than pick one carrier-locality and use that .            *;
*********************************************************************;

data &sasDTout._1 (compress = binary
                   drop = ratio_mp ratio_pe ratio_w
                          ascgrp carrier carrloc cpt_4_rvu
                          facbet
                          gpci_mal gpci_pe gpci_work
                          labcode
                          rvu_fac rvu_imp rvu_lab rvu_nfac
                          rvu_pf rvu_pn tempval
                          mcr_stat);
  set inout.&sasDTin;

     *** medicare carrier and locality, concatenated ***;
     length carrloc $ 7 ;
     carrloc = put(zip_s,$zipcarr.) ;
     if carrloc = '' then carrloc = put(zip_p,$zipcarr.) ;

     *** if still missing, do not impute yet ***;
     *** add GPCI components based on that   ***;
     gpci_work = input(put(carrloc,$g06work.),6.2); *medicare work gpci, 2006;
     gpci_pe   = input(put(carrloc,$g06pe.  ),6.2); *medicare practice gpci, 2006;
     gpci_mal  = input(put(carrloc,$g06mal. ),6.2); *medicare malpractice gpci, 2006;

     ******************************************************************;
     * Note:                                                          *;
     * There are only a small fraction of records for which location info *;
     * is missing for 2006, once the missing provider locations have been *;
     * gap-filled with the beneficiary locations                      *;
     * We can gap-fill with the average GPCI values for Maryland, after *;
     * running through this once                                      *;
     * After running the code once, the dollar-weighted means were    *;
     * computed  and plug in as the gap-fill values                   *;
     ******************************************************************;

     /* Year 2005 --> */ *if gpci_work = . then gpci_work = 1.0261;
```

```
     /* Year 2005 --> */ *if gpci_pe   = . then gpci_pe   = 1.0929;
     /* Year 2005 --> */ *if gpci_mal  = . then gpci_mal  = 0.9213;

/***** THE GPCI COMPONENTS BELOW HAVE TO BE CHANGED FOR 2007  *****/

     if gpci_work = . then gpci_work = 1.0000;     /* <-- Year 2006 */
     if gpci_pe   = . then gpci_pe   = 1.1082;     /* <-- Year 2006 */
     if gpci_mal  = . then gpci_mal  = 0.9217;     /* <-- Year 2006 */

     ********************************************************************;
     * Calculate Lab Fee Schedule values here                            *;
     * The program identifies which procedure codes are for lab/radiology *;
     * and assigns  payment rate for each lab codes based on the carrier. *;
     ********************************************************************;

     length labcode 3 ;
     labcode = input(put(cpt,$L&year.CODE.),4.);  ***temp flag to identify lab/rad
codes*;

     if labcode = 1 then do ;

        ********************************************************************;
        * This block is ordered by state to speed execution, with MD,DC,VA   *;
        * being the first three states on the list. Note that ND and SD have *;
        * the same carrier but different payment rate. The assignment uses   *;
        * carrier location instead of carrier. $L00821B is the format for ND *;
        * and $L00822B is the format for SD (see Fomat_RVU_2005.sas)         *;
        * TEMPVAL is the temporary variable to hold pmt value for lab code.  *;
        ********************************************************************;

        length carrier $ 5 ;
        carrier = substr(carrloc,1,5) ;     ***medicare carrier number;

        if      carrier = '00901' then tempval = input(put(cpt,$l00901b.),6.2) ;
        else if carrier = '00903' then tempval = input(put(cpt,$l00903b.),6.2) ;
        else if carrier = '00904' then tempval = input(put(cpt,$l00904b.),6.2) ;
        else if carrier = '00902' then tempval = input(put(cpt,$l00902b.),6.2) ;
        else if carrier = '00865' then tempval = input(put(cpt,$l00865b.),6.2) ;
        ELSE IF CARRIER = '00883' THEN TEMPVAL = INPUT(PUT(CPT,$L00883B.),6.2) ;
        ELSE IF CARRIER = '00884' THEN TEMPVAL = INPUT(PUT(CPT,$L00884B.),6.2) ;
        else if carrier = '00831' then tempval = input(put(cpt,$l00831b.),6.2) ;
        else if carrier = '00510' then tempval = input(put(cpt,$l00510b.),6.2) ;
        else if carrier = '00520' then tempval = input(put(cpt,$l00520b.),6.2) ;
        else if carrier = '00832' then tempval = input(put(cpt,$l00832b.),6.2) ;
        else if carrier = '31140' then tempval = input(put(cpt,$l31140b.),6.2) ;
        else if carrier = '31146' then tempval = input(put(cpt,$l31146b.),6.2) ;
        else if carrier = '00824' then tempval = input(put(cpt,$l00824b.),6.2) ;
        else if carrier = '00591' then tempval = input(put(cpt,$l00591b.),6.2) ;
        else if carrier = '00590' then tempval = input(put(cpt,$l00590b.),6.2) ;
        else if carrier = '00511' then tempval = input(put(cpt,$l00511b.),6.2) ;
        else if carrier = '00833' then tempval = input(put(cpt,$l00833b.),6.2) ;
        else if carrier = '00826' then tempval = input(put(cpt,$l00826b.),6.2) ;
        else if carrier = '05130' then tempval = input(put(cpt,$l05130b.),6.2) ;
        else if carrier = '00952' then tempval = input(put(cpt,$l00952b.),6.2) ;
        else if carrier = '00630' then tempval = input(put(cpt,$l00630b.),6.2) ;
        else if carrier = '00650' then tempval = input(put(cpt,$l00650b.),6.2) ;
        else if carrier = '00660' then tempval = input(put(cpt,$l00660b.),6.2) ;
        else if carrier = '00528' then tempval = input(put(cpt,$l00528b.),6.2) ;
        else if carrier = '31143' then tempval = input(put(cpt,$l31143b.),6.2) ;
        else if carrier = '31142' then tempval = input(put(cpt,$l31142b.),6.2) ;
        else if carrier = '00953' then tempval = input(put(cpt,$l00953b.),6.2) ;
        else if carrier = '00954' then tempval = input(put(cpt,$l00954b.),6.2) ;
        else if carrier = '00740' then tempval = input(put(cpt,$l00740b.),6.2) ;
        else if carrier = '00523' then tempval = input(put(cpt,$l00523b.),6.2) ;
        else if carrier = '00512' then tempval = input(put(cpt,$l00512b.),6.2) ;
        else if carrier = '00751' then tempval = input(put(cpt,$l00751b.),6.2) ;
        else if carrier = '05535' then tempval = input(put(cpt,$l05535b.),6.2) ;
        else if carrloc='0082001' then tempval = input(put(cpt,$l00821b.),6.2) ; *ND;
        else if carrloc='0082002' then tempval = input(put(cpt,$l00822b.),6.2) ; *SD;
        else if carrier = '00655' then tempval = input(put(cpt,$l00655b.),6.2) ;
        else if carrier = '31144' then tempval = input(put(cpt,$l31144b.),6.2) ;
```

```
        else if carrier = '00805' then tempval = input(put(cpt,$l00805b.),6.2) ;
        else if carrier = '00521' then tempval = input(put(cpt,$l00521b.),6.2) ;
        else if carrier = '00834' then tempval = input(put(cpt,$l00834b.),6.2) ;
        else if carrier = '00801' then tempval = input(put(cpt,$l00801b.),6.2) ;
        else if carrier = '00803' then tempval = input(put(cpt,$l00803b.),6.2) ;
        else if carrier = '14330' then tempval = input(put(cpt,$l14330b.),6.2) ;
        else if carrier = '00522' then tempval = input(put(cpt,$l00522b.),6.2) ;
        else if carrier = '00835' then tempval = input(put(cpt,$l00835b.),6.2) ;
        else if carrier = '00973' then tempval = input(put(cpt,$l00973b.),6.2) ;
*       else if carrier = '00870' then tempval = input(put(cpt,$l00870b.),6.2) ;
        else if carrier = '00880' then tempval = input(put(cpt,$l00880b.),6.2) ;
        else if carrier = '05440' then tempval = input(put(cpt,$l05440b.),6.2) ;
        else if carrier = '00900' then tempval = input(put(cpt,$l00900b.),6.2) ;
*       else if carrier = '00910' then tempval = input(put(cpt,$l00910b.),6.2) ;
        else if carrier = '31145' then tempval = input(put(cpt,$l31145b.),6.2) ;
        else if carrier = '00836' then tempval = input(put(cpt,$l00836b.),6.2) ;
        else if carrier = '00951' then tempval = input(put(cpt,$l00951b.),6.2) ;
        else if carrier = '00825' then tempval = input(put(cpt,$l00825b.),6.2) ;
     end ;

     ************************************************************************;
     * Compute the 2006 Medicare rates using the 2006 conversion factor.      *;
     * Keep track of what type of code this is for payment calculation        *;
     * purposes.  Rather than sprinkle the payment calculation through the    *;
     * program, just flag what needs to be done and calculate payments at end *;
     * For example, GPCI does not apply to lab values, etc.                   *;
     ************************************************************************;

*    cf2005=37.8975;     /* 2005 medicare fee schedule conversion factor - was 37.3374 in
2004, a 1.5% increase */

     cf2006=37.8975;     /* 2006 medicare fee schedule conversion factor - same as 2005
conversion factor */

     length pay_calc $1;   /* flag for what payment calculation to use */
     pay_calc = '0';       /* value of zero means do not calculate payment */

     **Remember that the end of the loop is long way down there*;
     if f_cpt in ('2','3') then do;

     *******************************************************;
     * Work On Modifiers For 26 And Blank For Radiology    *;
     * Impute Them By Site Of Service                      *;
     *******************************************************;

     length id_26_05 id_tc_05 id_26_06 id_tc_06 $2;
     label id_26_05 = 'CPT for which mod of 26 appears MFS 2005'
           id_tc_05 = 'CPT for which mod of TC appears MFS 2005'
           id_26_06 = 'CPT for which mod of 26 appears MFS 2006'
           id_tc_06 = 'CPT for which mod of TC appears MFS 2006';

     id_26_05 = put(cpt,$y05_26m.);
     id_tc_05 = put(cpt,$y05_tcm.);
     id_26_06 = put(cpt,$y06_26m.);
     id_tc_06 = put(cpt,$y06_tcm.);

     *** DETERMINE Y2006 MEDICARE ASC RATE based on the file ASCPUF05 from CMS(HCFA)*;

     length ascgrp $ 1;
     if svcplace = '24' then  ascgrp = put(cpt,$y06_asc.);

     ***************************************************************************;
     * ADD 2006 TRANSITIONED RVUS, AND PUT ON LAB FEES AS WELL:                 *;
     * FOR 2006, THIS ARE MORE COMPLEX.  MUST KEEP THE COMPONENTS OF THE RVU    *;
     * FOR AN EVENTUAL ANALYSIS OF SECTION 805, AND FOR CALCULATION OF PAYMENTS *;
     * ONLY USE THE MEDICARE RVUS WHEN MEDICARE USES THEM FOR PAYMENT,          *;
     * INCLUDING THE OBSOLETE CODES (STATUS D) THAT ARE HELD OVER INTO          *;
     * THE CURRENT YEAR.                                                        *;
     ***************************************************************************;

     *cpt || pricing mod for rvu merge**;
```

```
     length cpt_4_rvu $ 7 ;
     cpt_4_rvu = left(trim(cpt))|| left(trim(newmod));

     length rvu_w rvu_pf rvu_pn rvu_mp 4 ;
     label rvu_w  = 'Medicare Phys Fee Schedule RVU, Work component'
           rvu_pf = 'Medicare Phys Fee Schedule RVU, Pract exp facility'
           rvu_pn = 'Medicare Phys Fee Schedule RVU, Pract exp non-facility'
           rvu_mp = 'Medicare Phys Fee Schedule RVU, Malpractice'
           rvu_pe = 'Medicare Phys Fee Schedule RVU, Pract Expense'
           rvu_i  = 'Assigned/Imputed RVU';

     length mcr_stat $ 1 ;
     mcr_stat = put(cpt_4_rvu,$rvu06st.) ;
     If mcr_stat in ('A','D','H','R','T','N') then do ;
        rvu_w  = INPUT(PUT(CPT_4_rvu,$rvu06w.),6.2);
        rvu_pf = INPUT(PUT(CPT_4_rvu,$rvu06pf.),6.2);
        rvu_pn = INPUT(PUT(CPT_4_rvu,$rvu06pn.),6.2);
        rvu_mp = INPUT(PUT(CPT_4_rvu,$rvu06mp.),6.2);
     end ;

     ****************************************************************;
     * NOTE THAT THE COMPONENTS WILL ALL BE MISSING FOR CODES OF THE *;
     * WRONG STATUS. DO NOT SET INDIVIDUAL COMPONENTS TO MISSING.    *;
     * THERE ARE NUMEROUS TRUE ZEROS AMONG THESE VALUES              *;
     * BUT IT IS OK TO SET THE TOTALS (RVU_FAC, RVU_NFAC) TO MISSING *;
     ****************************************************************;

     rvu_fac  = sum(rvu_w,rvu_pf,rvu_mp) ;
     rvu_nfac = sum(rvu_w,rvu_pn,rvu_mp) ;

     if rvu_fac le 0 then rvu_fac = . ;
     if rvu_nfac le 0 then rvu_nfac = . ;

     **** For lab rvu, use the Maryland lab fee schedule ;
     * approx rvu for lab based on medicare lfs *;

     rvu_lab = input(put(cpt,$100901b.),6.2) ;
     if rvu_lab ne . then rvu_lab = rvu_lab/cf2006;
     if rvu_lab le 0 then rvu_lab = . ;

     rvu_imp = input(put(cpt_4_rvu,$rvuimp.),6.2);    *y2006 imputed rvu***;

     *** for hogan test run ;
*    rvu_imp = input(put(cpt_4_rvu,gaprvu.),6.2);

     if rvu_imp le 0 then rvu_imp = . ;

     * Starting from 2002, mammography procedures are not edited ***;
     * Code for the edits was deleted in 2002 **********************;

     * IMPLEMENT RULE FOR MEDICARE RVUS BY SITE ;

     if f_fac = '1' then do ;

       if svcplace = '24' then do ;
           if ascgrp ne '' then do ;
              f_fac='1';
              rvu = rvu_fac ;
              rvu_pe=rvu_pf;
              pay_calc = '2';  * value for pmt based on facility rvu ;
           end;
           else do ;
               f_fac='0';
               rvu = rvu_nfac ;
               rvu_pe=rvu_pn;
               pay_calc = '3';  * value for non-facility rvu ;
           end ;
       end; *end of svcplace='24' do loop;
       else do ;
           pay_calc = '2' ;
```

```
      f_fac='1';
      rvu = rvu_fac ;
      rvu_pe=rvu_pf;
  end ;

end ;  *end of f_fac='1' do loop;
else do ;

  f_fac='0';
  pay_calc = '3';
  rvu = rvu_nfac;
  rvu_pe=rvu_pn;

end ;

* SPECIAL MEDICARE RULE FOR THERAPY CODES ;
* THIS IS MY BEST GUESS AT THERAPY BILLED UNDER MEDICARE FEE SCHEDULE;

if cpt gt '97001' and cpt le '97456' then do ;
   f_fac='0';
   pay_calc = '3' ;
   rvu = rvu_nfac ;
   rvu_pe=rvu_pn;
end;

if rvu ne . then f_imprvu='0';        ****no-- rvu not imputed****;
rvu_i=rvu;

**impute lab rvus using medicare lab fee data**;
if rvu_i = . and rvu_lab ne . then do ;
   rvu_i = rvu_lab ;
   pay_calc = '4' ; * calc medicare rate based no lab fee schedule ;
   rvu_pe=rvu_lab;  *All lab rvu is set to pract.expense***;
   rvu_w=0;
   rvu_mp=0;
end ;
else if rvu_i = . and rvu_imp ne . then do ;
     rvu_i = rvu_imp ;
     pay_calc = '5' ; * calc based on imputed total rvu ;
end ;

if rvu_i ne . and rvu=. then f_imprvu='1';  *yes--rvu imputed;
if rvu_i=.    and rvu=. then f_imprvu='2';  *no--rvu not imputed -- no donor;

****************** Impute component RVUs ***************;

length facbet $5;

facbet=Put(f_fac,1.)||left(trim(betosdet));

Ratio_pe=input(put(facbet,$rat_pe.),6.4);
Ratio_mp=input(put(facbet,$rat_mp.),6.4);
Ratio_w =input(put(facbet,$rat_w.),6.4);

***New code starts here****;
*************************************************************;
* Set components to missing for procs with wrong status code *;
* (D=Deleted,N=Non covered services,R=Restricted services    *;
* Note: Some have good values, do this only when RVU=.       *;
* See above the statement [if rvu_fac le 0 then rvu_fac = .] *;
*************************************************************;
if rvu=. and mcr_stat in ('D','N','R')then do;
   rvu_mp=.;
   rvu_pe=.;
   rvu_w=.;
end;


***New code ends here****;
```

```
     If rvu_i ne . and (rvu_w in (.,0) and rvu_mp in (.,0) and rvu_pe in (.,0)) then do;
*open do loop 1;
        rvu_pe=Ratio_pe*RVU_I;
        rvu_mp=Ratio_mp*RVU_I;
        rvu_w=Ratio_w*RVU_I;

        *******if components still not imputed***********;
        if (rvu_w=. and rvu_mp=. and rvu_pe=.) then do;      *open do loop 2;
           if betosdet in ("T1A","T1B") then do;             *open do loop 3;
              rvu_pe=rvu_i;                       *All lab rvu_i set to pract.
expense***;
              rvu_w=0;
              rvu_mp=0;
           end;                                              *close do loop 3;
           else do;                                          *open do loop 4;

                 *****use facility total ratios from
Compute_Impute_RVU_Components_2006.lst;***;
               if f_fac='1' then do;
                  RVU_pe= 0.3073*RVU_I;
                  RVU_mp= 0.0606*RVU_I;
                  RVU_w = 0.6321*RVU_I;
               end;
               else do;
                    RVU_pe = 0.5658*RVU_I;
                    RVU_mp = 0.0314*RVU_I;
                    RVU_w  = 0.4028*RVU_I;
               end;
           end;                                              *close do loop 4;
        end;                                                 *close do loop 2;
     end;                                                    *close do loop 1;


     ************************************************************ ;
     * NOW CALCULATE MEDICARE PAYMENTS BASED ON TYPE OF CODE AND ;
     * PAYMENT CALCULATION FLAG ;
     * VALUES FOR PAY_CALC ARE:
     * 0 = DO NOT CALCULATE PAYMENT ;
     * 1 = THIS IS SCREENING MAMMOGRAPHY (PRE-2002) ;
     * 2 = THIS IS TO BE CALCULATED USING FACILTIY RVUS ;
     * 3 = THIS IS TO BE CALCUALTED USING NONFACILTIY RVUS ;
     * 4 = THIS IS A LAB FEE SCHEDULE CODE  ;
     * 5 = THIS CODE HAS AN IMPUTED TOTAL RVU BUT NO RVU COMPONENTS ;
     * 6 = (SEE BELOW) - THIS WAS FLAGGED AS AN ODD SERVICE -- DO NOT USE MEDICARE PMT
RATE ;
     ************************************************************** ;

     label pay_mcr = 'Prelim Calc MCR Rate-DO NOT USE AT THIS POINT' ;

     if pay_calc = '1' then do;  * screening mammography, no gpci-obsolete starting 2002;
         pay_mcr = rvu_fac*cf2006;
     end ;
     else if pay_calc = '2' then do;  * facilty rvus to be used ;
         pay_mcr = sum(gpci_work*rvu_w,gpci_pe*rvu_pf,gpci_mal*rvu_mp)*cf2006;
     end ;
     else if pay_calc = '3' then do;  * nonfacility rvus to be used ;
        pay_mcr = sum(gpci_work*rvu_w,gpci_pe*rvu_pn,gpci_mal*rvu_mp)*cf2006;
     end ;
     else if pay_calc = '4' then do;  * lab fee schedule -- no gpci adjustment ;
        pay_mcr = rvu_lab*cf2006;
     end ;
     else if pay_calc = '5' then do;  * imputed rvu, gap fill gpci using average weights
;
       pay_mcr = rvu_i*sum(gpci_work*.4724,gpci_pe*.4874,gpci_mal*.0403)*cf2006;
     end ;

     if pay_mcr le 0 then pay_mcr = . ;
     drop id_26_05 id_tc_05;

  end;         ****end of the big loop[if f_cpt in ('2','3')]*****;
```

```
  else f_imprvu='0';  **non cpt/hcpcs codes are classified as 0-no-- rvu not imputed**;

run;


**=====================================**

       Assign Odd Services Flags

**=====================================**;

/* Note from Po-Lun on 08-16-2007:
/* For comparing this year's RVU values with previous year's values, Adrien and I */
/* discussed it and decided to add the whole section of assigning odd services    */
/* flags here.  However, because NEWSPEC has not been created yet (it will be     */
/* created in Enc3), I commented the screens regarding to NEWSPEC out below, and  */
/* for this reason, variable F_ODDSVC which will be created later in this progeam */
/* is just a temporary variable; it has to be re-created as a permanent variable  */
/* in Enc4.                                                                        */

*** create condition code flags ***;

data &sasDTout._2;
  set &sasDTout._1;
  length cond01 - cond15 3;
  label cond01  = 'Member of Debit-Credit Pair'
        cond02  = 'Dental Orthodontic'
        cond03  = 'FFS Payment <= 1, Allowed Amount <= 0'
        cond04  = 'Identified Anesthesia Outside Of Range'
        cond05  = 'Identified Facility Bill'
        cond06  = 'Minor Service Assoc With Major Service'
        cond07  = 'Adjustment Record'
        cond08  = 'Payer not NONHMO,HMOFFS,HMOCAP'
        cond09  = 'Not First Dupe by Key, Descending Payment'
        cond10  = 'Actuaries Trim, Payment, Low'
        cond11  = 'Actuaries Trim, Payment, High'
        cond12  = 'Actuaries Trim, RVU_I, Low'
        cond13  = 'Actuaries Trim, RVU_I, High'
        cond14  = 'Covoth_I = 2, Possible Secondary Payer'
        cond15  = 'RVU_I <= 0';

  array acond cond01-cond15;
  do i = 1 to 15;
     acond[i] = 0;
  end;

*********************************************************
COND01
Member of Debit/Credit Pair

PARTIAL BILLS ARE NO LONGER ALLOWED IN THE DATABASE.
THIS FLAG IS NO LONGER ASSIGNED.
*********************************************************;


*********************************************************
COND02
Dental Orothodontic CPT Codes
*********************************************************;
  if (cpt ge 'D0001' and cpt le 'D7420') or
     (cpt ge 'D8001' and cpt le 'D8999') then cond02 = 1;


*********************************************************
COND03
For Fee-For-Service (FFS) records only
Allowed Amount <= 0
or
Payment <= 1
or
Billed Amount is the only financial information provided
```

```
*********************************************************;
  if plan in ('HMOFFS','NONHMO') then do;
     if allow_orig le 0   or
        payment_orig le 1 then cond03 = 1;

     else
     if bill_orig ne 0 and
        (allow_orig in (.,0) and reimb_orig in (.,0) and patliab_orig in (.,0)) then
cond03 = 1;
  end;


*********************************************************
COND04
Anesthesia Services Not Identified
*********************************************************;
  %let aneslist = 'AN', '47', 'P1', 'P2', 'P3', 'P4', 'P5', '30', 'QX', 'QZ',
                  '23', 'QK', 'QS', 'QY', 'G8', 'G9', 'AA', 'AB', 'AC', 'AD',
                  'AE', 'AF', 'AG';

  *** first find all anesthesia procedures and flag ***;
  length anesth $ 1;
  label anesth = 'Anesthesia Procedure';
  anesth = '0';

  if (cpt ge '00100' and cpt le '01999')         or
      cpt in ('99100','99116','99135','99140')   or
      unitind = '2'                              or
/*    newspec in ('004','049')                   or     */   /** comment NEWSPEC out in
Enc2 temporary  **/
      mod1 in (&aneslist)                        or
      mod2 in (&aneslist)                        then anesth = '1';

  *** services that are legitimate anesthesia codes are ok, ***;
  *** others have to be marked as odd services              ***;
  if anesth   = '1' then do;
     if newbetdt ne 'P0 ' then do;
        cond04   = 1;
        newbetdt = 'P0 ';
        newbetag = '2';
     end;
  end ;
  drop anesth;


*********************************************************
COND06
Minor Services Associated With Major Services
*********************************************************;
  length split assist bilat $1;

  *** minor/split surgery ***;
  if  mod1 in ('52','53','54','55','56','62','57','62','65') or
      mod2 in ('52','53','54','55','56','62','57','62','65') then split = '1';

  *** assistance at surgery ***;
  if mod1 in ('80','81','82') or
     mod2 in ('80','81','82') then assist = '1';

  *** bilateral/multiple procedures ***;
  if mod1 = '51' or
     mod2 = '51' or
     mod1 = '50' or
     mod2 = '50' then bilat = '1';

  if split  = '1'  or
     assist = '1'  or
     bilat  = '1'  then cond06 = 1;

  drop split assist bilat;
```

```
*******************************************************
COND07
Adjustment Record
*******************************************************;
  if billtype not in ('1', '8') then cond07 = 1;


*******************************************************
Create MCR_STAT
Add a flag for Medicare Payment Status Code, we are
specifically interested in identifying services with
Status Code C
(miscellaneous coded priced by carrier report)
*******************************************************;
  length mcr_stat $ 1 ;
  label mcr_stat = 'Medicare Payment Status Code';

  mcr_stat = put(cpt,$RVU05ST.) ;


*******************************************************
COND05
Facility Fees
*******************************************************;
  if (/* newspec in ('090','091','092','093') or */   /** comment NEWSPEC out in Enc2
temporary  **/
      mod1 = 'SG'                          or
      mod2 = 'SG')
  or (newmod = 'TC')                       then cond05 = 1;


*******************************************************
COND08
Ambiguous Payer Type
*******************************************************;
  if plan eq '' then cond08 = 1;


*******************************************************
COND14
Possible Secondary Payer Involved
*******************************************************;

/* Note from Po-Lun on 08-16-2007:                              */
/* for generating COND14, I have to copy the section below from */
/* other program to set variables COVOTH_I & F_IMPOTH first     */

  ***************************************************************************
  Created imputed variable for other insurance coverage, and associated flag
  covoth_i -- 0 = not imputed 2 = imputed, other insurance secondary
  f_impoth
  ***************************************************************************;

  length covoth_i $ 1;
  length f_impoth $ 1;

  *** if other insurance coverage is already known, keep the value and ***;
  *** set the flag for no imputation                                   ***;
  if covothr in ('0','1','2') then do;
     covoth_i = covothr;
     f_impoth = '0';
  end;

  *** otherwise, check to see if we can impute other insurance coverage ***;
  else do;

     *** impute if record has positive amounts for allowable amount, ***;
     *** reimbursement amount, and patient liability                  ***;
     if allow_orig > 0 and sum(reimb_orig, patliab_orig) >= 0 then do;
```

```
        *** impute no other insurance coverage if the sum of the reimbursemant ***;
        *** amount and patient liability is >+ 35% of the allowable amount     ***;
        if sum(reimb_orig, patliab_orig) * 100 / allow_orig >= 35 then do;
           covoth_i = '0';
           f_impoth = '1';
        end;

        *** otherwise, impute other insurance coverage is present and is secondary ***;
        else do;
           covoth_i = '2';
           f_impoth = '1';
        end;
     end;

     *** if the record doesnt have positive amounts for the financial information, ***;
     *** do not impute, and use value of original variable                         ***;
     else do;
        covoth_i = covothr;
        f_impoth = '0';
     end;
  end;

/* assign COND14 now */

  if covoth_i = '2' then cond14 = 1;


*********************************************************
COND15
RVU <= 0
*********************************************************;
  if rvu_i le 0 then cond15 = 1;


*********************************************************
Add a Variable to Speed Sorting in the Next Step
*********************************************************;
  length sortvar $ 100;
  length chardate $ 7;
  length odd $ 2;

  *** this (prematurely figures) out the equivalent of f_oddsvc so it can be part of the
sort ***;

  array conds(15) cond01-cond15;
  condcnt= sum(of cond01-cond15);

  if condcnt=0 then odd=' ';
  else
  if condcnt gt 1 then odd = '99';
  else do j = 1 to 15;
       if conds{j}=1 then do;
          odd=put(j,z2.);
       end;
  end;

  *** make date character ***;

  chardate = put(enddt,julian7.);
  sortvar = trim(pidbdg) || trim(chardate) || trim(cpt) || odd || trim(newmod);
run;


****************************************
COND09
Flag Duplicates By Service Date/CPT
Keep Only Highest Payment Record
****************************************;
proc sort data = &sasDTout._2 out = cond_a tagsort;
  by sortvar descending payment_orig;
run;
```

```
data &sasDTout._3 (drop = sortvar);
  set cond_a;
  by sortvar descending payment_orig;
  if first.sortvar ne 1 then cond09 = 1;
run;


**********************************************
COND12
COND13
Impose The HCFA Actuaries' Trim
-- Screen out bills with allowed less
   than one third or more than three
   times the mean allowed charge for the
   entire database
-- restrict to only good records (those
   which will be used in the pricing
   analysis
-- Note: This will not catch major outliers
   in the miscellaneous codes with
   Medicare status = 'C'
**********************************************;

*** cond_b is good records only, used for the means ***;

data cond_b;
  set &sasDTout._3;
  if plan in ('HMOFFS', 'NONHMO') and
     sum (of cond01-cond15) = 0 then output;
run;

proc sort data = cond_b;
  by cpt newmod;
run;

*** use only valid records to calculate means *** ;

proc means data = cond_b mean noprint;
  var payment_orig;
  output out = cptsum mean = payment_;
  by cpt newmod;
run;

*** note asymmetric actuaries trim ***;
*** note this step also assigns condition flags cond10-cond13 ***;

proc sort data = &sasDTout._3;
  by cpt newmod;
run;

data &sasDTout._4 (drop = _type_ _freq_);
  merge &sasDTout._3
        cptsum;
  by cpt newmod ;

  cf&yr = 37.8975;       /* 2006 medicare fee schedule conversion factor */

  *** do not use odd services ***;
  if sum (of cond01-cond15) = 0 then ok = '1';

  if mcr_stat ne 'C'             and
     ok = '1'                    and
     plan in ('HMOFFS','NONHMO') then do ;
       if payment_orig < .33*payment_ then cond10 = 1;
       else if payment_orig >   6*payment_ then cond11 = 1;
       if rvu_i gt 0 and payment_orig < .25*cf&yr*rvu_i then cond12 = 1;
       if rvu_i gt 0 and payment_orig >   8*cf&yr*rvu_i then cond13 = 1;
  end;
run;
```

```
****************************
Assign Odd Services Flag
****************************;

data &sasDTout._5 (compress = binary);
     set &sasDTout._4;

     length f_oddsvc $ 2;
     label f_oddsvc = 'Odd Service Flag';

     array conds(15) cond01-cond15;
     condcnt = sum(of cond01-cond15);

     if condcnt = 0 then f_oddsvc = ' ';
     else if condcnt gt 1 then f_oddsvc = '99';
     else do j = 1 to 15;
          if conds{j} = 1 then do;
             f_oddsvc = put(j,z2.);
          end;
     end;

     ************************************************************
     Final Adjustments Based on Odd Services Flagging
     ************************************************************;

     if pay_calc > 0 and f_oddsvc ne '' then do ;
        pay_mcr = . ;
        pay_calc = '6' ; * odd service -- do not use medicare payment rate;
     end;

run;


***** check the data *****;

proc means data = &sasDTout._5 n nmiss sum mean min max;
  var rvu_pe rvu_w rvu_mp rvu_i;
  where f_cpt in ('2','3');
  title4 "The sum of components must be equal to RVU_I";
run ;

proc summary data = &sasDTout._5;
  class betosagg;
  var pay_mcr payment_orig bill_orig;
  where f_cap_orig = '0' and payment_orig > 0 and pay_mcr > 0 and f_cpt in ('2','3');
  output out = paysum3
         n = mnobs pnobs bnobs
         nmiss = mmiss pmiss bmiss
         mean = mmcr mpvt mbill
         sum = tmcr tpvt tbill;
run;

proc print data = paysum3;
  title4 "Sum of PAY_MCR, PAYMENT, and BILL by BETOSAGG";
run;

proc freq data = &sasDTout._5;
  tables cpt;
  where f_cpt in ('2','3') and rvu_i = .;
  title4 'Remember that many procs with certain or missing status code are not assigned
rvus';
run;


**=======================================**

       Create Final Permanent Data

**=======================================**;
```

```
proc sort data = &sasDTout._5;
     by pnum encnt_id;
run;

*** bring the deletion flag variables in ***;

proc sort data = outj.&payernum._Enc1_D_Flags (drop = file_id f_yr&nextYr)
          out = &payernum._DFlag;
     by pnum encnt_id;
run;

data inout.&sasDTout (compress = binary
                      keep = age
                             age12&year
                             age12&year._nr
                             allow_orig
                         /* allow */
                             begday
                             begindt
                             begmo
                             begyr
                             betosagg
                             betosdet
                             bill_orig
                         /* bill */
                             billtype
                             birthdate
                             birthmo
                             birthyr
                             cdhpind
                             cf&yr
                             claimcn
                             claimpdd
                             claimpmo
                             claimpyr
                             clmpdate
                             clmrelco
                             cnty_p2
                             county_p
                             covoth_i
                             covothr
                             covtype
                             cpt
                             d_dat_f
                             d_dnt_f
                             d_i7k_f
                             d_inp_f
                             d_zip_f
                             delivtyp
                             disenrolldt
                             dx1 - dx10
                             dxdot1 - dxdot10
                             dxval1 - dxval10
                             encnt_id
                             endday
                             enddt
                             endmo
                             endyr
                             enrolldt
                             f_baddt
                         /* f_benrule */
                         /* f_cap */
                         /* f_cap_a */
                             f_cap_orig
                             f_cpt
                             f_cpt_orig
                             f_demog
                             f_fac
                             f_impoth
                             f_imprvu
                             f_insamp
```

```
                          /* f_ncap_a */
                          /* f_nspec */
                             f_oddsvc
                             f_yr&nextyr
                             fedtaxid
                             file_id
                             id_26_&year
                             id_tc_&year
                             instate
                             los
                             mod1
                             mod2
                             newbetag
                             newbetdt
                             newmod
                          /* newspec */
                             numdxc
                             partprov
                             partprov_raw
                             patid
                             patid_raw
                             patliab_orig
                          /* patliab */
                             pay_calc
                             pay_mcr
                             payertyp
                             payment_orig
                          /* payment */
                             pidbdg
                             plan
                             plan2
                             pnum
                             provid
                             reimb_orig
                          /* reimb */
                             rvu
                             rvu_i
                             rvu_mp
                             rvu_pe
                             rvu_w
                             sex
                             sex_raw
                          /* spec1 */
                          /* spec2 */
                          /* spec3 */
                          /* spec_source */
                             svcplace
                             svcplace_raw
                             unitind
                             units
                             zip_p
                             zip_s);
     merge &sasDTout._5 (in = in1)
           &payernum._DFlag (in = in2);
     by pnum encnt_id;
     if in1 and in2 then output;
run;

proc contents data = inout.&sasDTout;
  title4 "Contents of Final Data Set &sasDTout";
run;

proc print data = inout.&sasDTout (obs = 20);
  title4 "Sample of Records of Final Data Set &sasDTout";
run;


%jobend;


dm "log;    print file = I:\MCdb\datacol\&payernum\programs\&payernum._enc2.log replace";
```

```
dm "output; print file = I:\MCdb\datacol\&payernum\programs\&payernum._enc2.lst replace";

proc datasets kill;
run;
quit;


%jobstart;

**=================================================================**;
**                                                                 **;
** Program:     P020_enc3.sas (was P020_enc2.sas)                  **;
** Programmer:  Po-Lun Chou                                        **;
** Date:        September 26, 2005                                 **;
**                                                                 **;
** Input Data:  p020_enc2.sas7bdat                                 **;
**              p020_final_specialty.sas7bdat                      **;
** Output Data: p020_enc3.sas7bdat                                 **;
**                                                                 **;
** Purpose:     Got SPECIALTY from separate data files and added it **;
**              into the Encounter data file.                      **;
** Note:        The basic source of SPEC is the Provider data file, **;
**              we also used CPT to decide the final SPEC for some  **;
**              records.                                           **;
**                                                                 **;
** Modified:    Sireesha Thati -- 09/11/2006 --- For 2005 MCDB     **;
**                                                                 **;
** Modified by: Po-Lun Chou                                        **;
** Date:        August 8, 2006                                     **;
** Purpose:     Get new specialties for 2006 from data file,       **;
**              p***_final_specialty.sas7bdat.                     **;
** Note:        1)  Per Ben's request, we have to have RVU included **;
**                  in the Enc1 changing report. For having RVU, we **;
**                  modified the original Enc4.sas (in 2005) and    **;
**                  saved it as a new program, Enc2.sas, and then   **;
**                  modified the original Enc2.sas (in 2005) and    **;
**                  saved it as Enc3.sas.                           **;
**              2)  The programs we use to create new specialties   **;
**                  have to be completed before running this        **;
**                  program. Those programs are as below:           **;
**                    1. specialty_check_2006_all_payers.sas,       **;
**                    2. pass1.sas,                                 **;
**                    3. cpt_spec_check_2006.sas,                   **;
**                    4. pass2.sas,                                 **;
**                    5. pass3.sas, &                               **;
**                    6. final_specialty.sas.                       **;
**                                                                 **;
** Modified by: Kasey Chen                                         **;
** Date:        September 13, 2007                                 **;
** Purpose:     Get new specialties for 2007 from data file,       **;
**              p***_final_specialty.sas7bdat.                     **;
** Note:        1)  This program was run as Enc2 in 2005.  This    **;
**                  year, we switch it back to step 3 of the ENC    **;
**                  processing.                                     **;
**              2)  The programs we use to create new specialties   **;
**                  have to be completed before running this        **;
**                  program. Those programs are as below:           **;
**                    1. specialty_check_2006_all_payers.sas,       **;
**                    2. pass1.sas,                                 **;
**                    3. cpt_spec_check_2006.sas,                   **;
**                    4. pass2.sas,                                 **;
**                    5. pass3.sas, &                               **;
**                    6. final_specialty.sas.                       **;
**                                                                 **;
**=================================================================**;

options obs = max center errors = 2 linesize = 120 pagesize = 92;

%let payer = P020;
%let drive = I;
%let rptYr = 2006;
```

```
* libname library "&drive:\mcdb\datacol\formats\permanent";
libname library ("&drive:\MCdb\datacol\Formats\PERMANENT\BETOS_formats",
                 "&drive:\MCdb\datacol\Formats\PERMANENT\RVU_formats");
libname in      "&drive:\mcdb\datacol\&payer\sasdata";
libname out     "&drive:\mcdb\datacol\&payer\sasdata";

title1 "Maryland Medical Care Database (MCDB)";
title2 "Year &rptYr Encounter Data";
title3 "&payer. - Enc3";
title4 " ";

footnote1 "%sysfunc(putn(%sysfunc(time()), hhmm.)) %sysfunc(putn(%sysfunc(date()),
weekdate.))";
footnote2 "program: &drive:\mcdb\datacol\&payer\programs\&payer._Enc3.sas";


***** format *****;

proc format;
     value $spectyp
           "001",
           "002",
           "003",
           "004",
           "005",
           "006",
           "007",
           "008",
           "009",
           "010",
           "011",
           "012",
           "013",
           "014",
           "015",
           "016",
           "017",
           "018",
           "019",
           "020",
           "021",
           "022",
           "023",
           "024",
           "025",
           "026",
           "027",
           "028",
           "029",
           "030",
           "031",
           "032",
           "033",
           "035",
           "036",
           "039",
           "100",
           "040",
           "041",
           "042",
           "043",
           "044",
           "045",
           "046",
           "047",
           "048",
           "049",
           "050",
           "051",
           "052",
```

```
            "053",
            "054",
            "055",
            "056",
            "057",
            "058",
            "059",
            "060",
            "061",
            "063",
            "070",
            "071",
            "072",
            "073",
            "074",
            "075",
            "076",
            "080",
            "081",
            "082",
            "083",
            "084",
            "090",
            "091",
            "092",
            "102",

            "101",
            "103" = '1' /* valid, not miscellaneous */
            "034",
            "037",
            "038",
            "062",
            "085",
            "093" = '2' /* valid, miscellaneous */
            " "   = '4' /* missing */
            other = '3' /* invalid */;

      value $f_spec
            "1" = "1: Valid, Not Misc"
            "2" = "2: Valid, Misc"
            "3" = "3: Invalid"
            "4" = "4: Missing";

      value source
            1 = "1: Provider with Single Specialty Code"
            2 = "2: Provider with Multiple Specialty Codes"
            3 = "3: No Provider - Random";
run;


***** bring in the Encounter data *****;

proc sort data = in.&payer._enc2 out = &payer._enc2;
      by encnt_id;
run;


***** bring in SPECIALTY *****;

proc sort data = in.&payer._final_specialty out = &payer._spec;
      by encnt_id;
run;


***** create permanent data file with final SPEC *****;

data out.&payer._enc3 (compress = binary);
      length f_nspec $1;
      merge &payer._enc2 (in = ina)
            &payer._spec (in = inb);
```

```
     by encnt_id;
     f_nspec = put(newspec, $spectyp.);
     if ina and inb then output;
run;

proc contents data = out.&payer._enc3;
     title4 "Contents of %upcase(&payer._enc3)";
run;

proc freq data = out.&payer._enc3;
     tables newspec f_nspec spec_source / list missing;
     format f_nspec $f_spec. spec_source source.;
     title4 "Frequencies in %upcase(&payer._enc3)";
run;

proc summary data = out.&payer._enc3 nway missing;
     class newspec spec_source;
     output out = final_spec (drop = _type_ rename = (_freq_ = freq));
run;

proc print data = final_spec;
     id newspec;
     by newspec;
     var spec_source freq;
     format spec_source source.;
     title4 "Print of Final Specialties for &payer";
run;

proc print data = out.&payer._enc3 (obs = 20);
     var pnum provid fedtaxid encnt_id newspec f_nspec spec_source;
     title4 "Samples of Records in %upcase(&payer._enc3)";
run;


%jobend;


dm "log;    print file = I:\MCdb\DataCol\&payer\programs\&payer._enc3.log replace";
dm "output; print file = I:\MCdb\DataCol\&payer\programs\&payer._enc3.lst replace";

proc datasets kill;
run;
quit;


%jobstart;

**=================================================================**;
**                                                                 **;
** Program:    P020_enc4.sas (was P020_enc3.sas & P020_enc4.sas)   **;
** Programmer: Adrien Ndikumwami/Chris Hogan                       **;
** Date:       September 25, 2001                                  **;
**                                                                 **;
** This program assign RVUs (Work,Practice, and Malpractice) to    **;
** valid procedures. Medicare Payment Rates are also computed.     **;
** The components are estimated for procedures with imputed RVUs   **;
** The following format is needed to run the program:              **;
** Format_RVU_2005.sas                                             **;
**                                                                 **;
** Adapted to 2004 MCDB on June 9, 2005-Adrien                     **;
** Adapted to 2005 MCDB on July 10, 2006 - Sireesha                **;
**                                                                 **;
** Modified by: Po-Lun Chou                                        **;
** Date:        October 4, 2006                                    **;
** Purpose:     1. Assigned RVU                                    **;
**              2. Added odd service flag                          **;
** Note:        We used to assign RVU in Pxxx_enc2.sas, but since we **;
**              changed processing order this year for 2005 data and **;
**              decided to assign RVU later, we changed this program **;
**              from Pxxx_enc2.sas to Pxxx_enc4.sas.               **;
**                                                                 **;
```

```
** Modified by: Po-Lun Chou                                          **;
** Date:        October 1, 2007                                      **;
** Purpose:  1. Modify PLAN, PLAN2, & PAYERTYP based on AGE1206_NR.  **;
**           2. Modify F_CPT based modified format.                  **;
**           3. Modify BETOSDET, BETOSAGG, NEWBETDT, & NEWBETAG      **;
**              based on F_CPT.                                      **;
**           4. Create imputed variable for other insurance cov.,    **;
**              COVOTH_I, and associated flag, F_IMPOTH.             **;
**           5. Create non capitated flag, F_NCAP_A.                 **;
**           6. Create capitated flag, F_CAP_A.                      **;
**                                                                   **;
**===================================================================**;

options obs = max errors = 2 linesize = 120 pagesize = 92;

%let drive = I;
%let payernum = P020;
%let yr = 2006;
%let year = 06;
%let nextyr = 07;

%let sasDTin = &payernum._enc3;
%let sasDTout = &payernum._enc4;

%let hmolst = 'P030','P130','P160','P480','P520','P620','P680','P870';
%let mhmolst = 'P480';

libname inout "I:\MCdb\datacol\&payernum\sasdata";
libname library ("I:\MCdb\datacol\Formats\PERMANENT\BETOS_formats",
                 "I:\MCdb\datacol\Formats\PERMANENT\RVU_formats");

title1 "MARYLAND HEALTH CARE COMMISSION";
title2 "Year 2006 &payernum Encounter File - &payernum._enc4";

footnote1 "%sysfunc(putn(%sysfunc(time()), hhmm.)) %sysfunc(putn(%sysfunc(date()),
weekdate.))";
footnote2 "program: I:\mcdb\datacol\&payernum\programs\&payernum._enc4.sas";

options nosource2;

*** cpt procedure codes ***;
%include "&drive:\MCdb\DataCol\formats\CPTFMT.SAS";

*** hcpcs procedure codes ***;
%include "&drive:\MCdb\DataCol\formats\HCPCSFMT.SAS";*

*** home grown procedure codes ***;
%include "&drive:\MCdb\DataCol\formats\HMGFMT.SAS";*

*** revenue codes to identify hospital inpatient services for exclusion from the mcdb
***;
%include "&drive:\MCdb\DataCol\formats\REVCODE.SAS";*

*** icd9 procedure codes to identify hospital inpatient services for exclusion from the
mcdb ***;
%include "&drive:\MCdb\DataCol\formats\icd9pr_&year..sas";

*** icd9 diagnosis codes codes to validate diagnosis codes ***;
%include "&drive:\MCdb\DataCol\formats\icd9dx_&year..sas";

options source2;


*****************************************;
**  Re-Assign PLAN, PLAN2, and PAYERTYP  **;
*****************************************;

/* Note from Po-Lun on 09-28-2007:                                          */
/* According to last year's note, we should assign PLAN, PLAN2, & PAYERTYP based on */
/* AGE12**_NR in program Enc1 this year; however, we still used AGE12** to assign   */
/* these variables in Enc1.  After talked to Sanee on 09-21-2007, we decided to fix */
```

```
/* these here in program Enc4, and we have to remember to assign these variables    */
/* right next year in Enc1.                                                          */

data &sasDTout._1;
  set inout.&sasDTin (rename = (f_oddsvc = f_oddsvc_o f_cpt = f_cpt_o plan = plan_o
                                plan2 = plan2_o payertyp = payertyp_o
                                betosagg = betosagg_o newbetag = newbetag_o
                                betosdet = betosdet_o newbetdt = newbetdt_o));

** create flag variables **;

/* for Medicare HMO (MCRHMO) */

  length mcrhmo $1;
  mcrhmo = '0';
  if pnum in (&mhmolst)      and
     delivtyp in ('1', '5')  and
     age12&year._nr >= 65    and
     age12&year._nr <= 110   then mcrhmo = '1';

/* for Medicare service other than Medicare HMO (MCROTH) */

  length mcroth $1;
  mcroth = '0';
  if mcrhmo ne '1' then do;
    if covtype in ('1', '7')  and
       age12&year._nr >= 65   and
       age12&year._nr <= 110  then mcroth = '1';
  end;

/* for private HMO, non-Medicare Service (PRVHMO) */

  length prvhmo $1;
  prvhmo = '0';

  if pnum in (&hmolst)                                and
     (0 <= age12&year._nr < 65 or age12&year._nr = .)  and
     delivtyp in ('1','5')                            then do;
       if pnum = 'P480' and covtype = '7' then prvhmo = '1';  /* special case for Kaiser
*/
       else if covtype in ('2','3','4','5','6') then prvhmo = '1';  /* everybody else */
  end;

/* for private non-HMO service (PRVNHMO) */

  length prvnhmo $1;
  prvnhmo = '0';

/* not an hmo provider */
  if pnum notin (&hmolst)                        and
     (0<=age12&year._nr<65 or age12&year._nr=.)  then do;
       if pnum = 'P850' and covtype = '8' then prvnhmo = '1';  /* special case for P850
which is all covtype=8 (Taft-Hartley) */
       else if covtype in ('2','3','4','5','6') then prvnhmo = '1';  /* everybody else */
  end;

/* hmo providers */
  else do;
     if pnum in (&hmolst)                            and
        (0<=age12&year._nr<65 or age12&year._nr=.)  and
        delivtyp in ('2','3','4')                    then do;
          if pnum = 'P480' and covtype = '7' then prvnhmo = '1';  /* special case for 480
*/
          else if covtype in ('2','3','4','5','6') then prvnhmo = '1';  /* everybody else
*/
     end;
  end;

** re-assignPLAN, PLAN2, and PAYERTYP **;

  length plan $6 plan2 $6 payertyp $1;
```

```
/* all capitated */
  if f_cap_orig = '1' then do;
     plan2 = 'HMOCAP';
     payertyp = '9';
  end;

/* private hmo ffs */
  else if prvhmo = '1' then do;
     plan2 = 'HMOFFS';
     payertyp = '1';
  end;

/* private non-hmo ffs */
  else if prvnhmo = '1' then do;
     plan2 = 'NONHMO';
     payertyp = '2';
  end;

/* medicare hmo ffs */
  else if mcrhmo = '1' then do;
     plan2 = 'MCRALL';
     payertyp = '3';
  end;

/* other medicare */
  else if mcroth = '1' then do;
     plan2 = 'MCRALL';
     payertyp = '5';
  end;

/* unknown ffs */
  else do;
     plan2 = ' ';
     payertyp = '5';
  end;

  if 0 <= age12&year._nr < 65 or age12&year._nr = . then plan = plan2;
  else plan = ' ';

run;


*************************************************************;
**  check F_CPT, BETOSDET, BETOSAGG, NEWBETDT, & NEWBETAG  **;
*************************************************************;

data &sasDTout._2;
  set &sasDTout._1;

** re-assign F_CPT - based on modified format **;

***********************************************
F_CPT ---
1 = HomeGrown
2 = HCPCS
3 = CPT
4 = ICD-9-CM
5 = Revenue Code
6 = missing/blank
7 = Invalid Procedure Code
***********************************************;
  length f_cpt $1;

  if cpt=' '                                then f_cpt = '6';
  else
  if put(cpt,$&payernum.hmg.) = 'HomeGrown'  then f_cpt = '1';
  else
  if put(cpt,$hcpcs.) = '1'                 then f_cpt = '2';
  else
  if put(cpt,$cptfmt.) = '1'                then f_cpt = '3';
```

```
  else
  if put(cpt,$icd9pr.) = '1'                 then f_cpt = '4';
  else
  if put(cpt,$revcode.)= '1'                 then f_cpt = '5';
  else                                            f_cpt = '7';

** re-assign BETOS codes, BETOSDET, BETOSAGG, NEWBETDT, & NEWBETAG **;
  length betosdet newbetdt $3 betosagg newbetag $1;

  if f_cpt in ("2", "3") then do;
    betosdet = put(cpt,PRC2BET.);
  end;

  if      substr(betosdet,1,1) = "M"         then betosagg = "1";
  else if substr(betosdet,1,1) = "P"         then betosagg = "2";
  else if substr(betosdet,1,1) = "I"         then betosagg = "3";
  else if substr(betosdet,1,1) = "T"         then betosagg = "4";
  else if substr(betosdet,1,1) = "D"         then betosagg = "5";
  else if substr(betosdet,1,1) = "O"         then betosagg = "6";
  else if substr(betosdet,1,1) in ("Y", "Z") then betosagg = "7";
  else if betosdet = " "                     then betosagg = " ";

  newbetag = betosagg;
  newbetdt = betosdet;

run;


****************************;
**  Create New Variables  **;
****************************;

data &sasDTout._3;
  set &sasDTout._2;

*************************************************************************
Created imputed variable for other insurance coverage, and associated flag
COVOTH_I -- 0 = not imputed 2 = imputed, other insurance secondary
F_IMPOTH
*************************************************************************;
  length covoth_i f_impoth $1;

/* if other insurance coverage is already known, keep the value and set the flag for no
imputation */
   if covothr in ('0','1','2') then do;
      covoth_i = covothr;
      f_impoth = '0';
   end;

/* otherwise, check to see if we can impute other insurance coverage */
   else do;

  /* impute if record has positive amounts for allowable amount, reimbursement amount, */
  /* and patient liability                                                              */
     if allow_orig > 0 and sum(reimb_orig, patliab_orig)>=0 then do;

    /* impute no other insurance coverage if the sum of the reimbursemant amount */
    /* and patient liability is >+ 35% of the allowable amount                   */
       if sum(reimb_orig, patliab_orig) * 100 / allow_orig >= 35 then do;
          covoth_i = '0';
          f_impoth = '1';
       end;

    /* otherwise, impute other insurance coverage is present and is secondary */
       else do;
          covoth_i = '2';
          f_impoth = '1';
       end;

     end;
```

```
  /* if the record doesnt have positive amounts for the financial information, */
  /* do not impute, and use value of original variable                          */
     else do;
        covoth_i = covothr;
        f_impoth = '0';
     end;

   end;

*************************************************************
Flag services eligible for Practitioner Report/SHEA-MCDB
analysis file
F_NCAP_A
F_CAP_A
*************************************************************;
  length f_ncap_a f_cap_a $1;

  f_ncap_a = '0';
  f_cap_a = '0';

/* flag uncovered services */
   length f_uncv $ 1;
   f_uncv = '0';

   if allow_orig in (0,.,.M)   and
      reimb_orig in (0,.,.M)   then f_uncv = '1';

/* flag services with TC or SG modifier */
   length f_modf $ 1;
   f_modf = '0';

   if mod1 in ('TC','SG')   or
      mod2 in ('TC','SG')   then f_modf = '1';

/* non capitated and used in analysis */
   if f_cap_orig = '0' then do;
       if (f_yr&nextyr eq '0'       and
           f_uncv = '0'             and
           covoth_i in ('0','2')    and
           f_cpt in ('1','2','3')   and
           f_modf eq '0'            and
           newbetag ne '5')         then f_ncap_a = '1';
   end;

/* capitated and used in analysis, do not apply f_uncv or other insurance info */
   else if f_cap_orig = '1' then do;
      if (f_yr&nextyr eq '0'        and
          f_cpt in ('1','2','3')    and
          f_modf eq '0'             and
          newbetag ne '5')          then f_cap_a = '1';
   end;

run;


***********************************************;
**  Re-Assign F_ODDSVC (Odd Services Flags)  **;
***********************************************;

/* Note from Po-Lun on 09-28-2007:                                                  */
/* We assigned variable F_ODDSVC (odd services flag) in program Enc2 (PXXX_enc2.sas) */
/* for 2006 data this year already because we wanted to check those assigned RVU     */
/* variables (RVU_I, RVU_MP, RVU_PE, & RVU_W) at that time; however, 2 variables     */
/* (COND04 & COND05) for odd services flag were not assigned properly since NEWSPEC  */
/* was not generated then.  For this reason, Sanee and I decided to drop the current */
/* F_ODDSVC and then re-generated it by following the original and complete way we   */
/* used last yesr.                                                                  */

** create condition code flags **;

data &sasDTout._4;
```

```
  set &sasDTout._3;

  length cond01 - cond15 3;
  label cond01  = 'Member of Debit-Credit Pair'
        cond02  = 'Dental Orthodontic'
        cond03  = 'FFS Payment <= 1, Allowed Amount <= 0'
        cond04  = 'Identified Anesthesia Outside Of Range'
        cond05  = 'Identified Facility Bill'
        cond06  = 'Minor Service Assoc With Major Service'
        cond07  = 'Adjustment Record'
        cond08  = 'Payer not NONHMO,HMOFFS,HMOCAP'
        cond09  = 'Not First Dupe by Key, Descending Payment'
        cond10  = 'Actuaries Trim, Payment, Low'
        cond11  = 'Actuaries Trim, Payment, High'
        cond12  = 'Actuaries Trim, RVU_I, Low'
        cond13  = 'Actuaries Trim, RVU_I, High'
        cond14  = 'Covoth_I = 2, Possible Secondary Payer'
        cond15  = 'RVU_I <= 0';

  array acond cond01-cond15;
  do i = 1 to 15;
     acond[i] = 0;
  end;


*********************************************************
COND01
Member of Debit/Credit Pair

PARTIAL BILLS ARE NO LONGER ALLOWED IN THE DATABASE.
THIS FLAG IS NO LONGER ASSIGNED.
*********************************************************;


*********************************************************
COND02
Dental Orothodontic CPT Codes
*********************************************************;
  if (cpt ge 'D0001' and cpt le 'D7420') or
     (cpt ge 'D8001' and cpt le 'D8999') then cond02 = 1;


*********************************************************
COND03
For Fee-For-Service (FFS) records only
Allowed Amount <= 0
or
Payment <= 1
or
Billed Amount is the only financial information provided
*********************************************************;
  if plan in ('HMOFFS','NONHMO') then do;
     if allow_orig le 0   or
        payment_orig le 1 then cond03 = 1;
     else
     if bill_orig ne 0 and
        (allow_orig in (.,0) and reimb_orig in (.,0) and patliab_orig in (.,0)) then
cond03 = 1;
  end;


*********************************************************
COND04
Anesthesia Services Not Identified
*********************************************************;
  %let aneslist = 'AN', '47', 'P1', 'P2', 'P3', 'P4', 'P5', '30', 'QX', 'QZ',
                  '23', 'QK', 'QS', 'QY', 'G8', 'G9', 'AA', 'AB', 'AC', 'AD',
                  'AE', 'AF', 'AG';

  *** first find all anesthesia procedures and flag ***;
  length anesth $ 1;
```

```
  label anesth = 'Anesthesia Procedure';
  anesth = '0';

  if (cpt ge '00100' and cpt le '01999')         or
      cpt in ('99100','99116','99135','99140')   or
      unitind = '2'                              or
      newspec in ('004','049')                   or
      mod1 in (&aneslist)                        or
      mod2 in (&aneslist)                        then anesth = '1';

  *** services that are legitimate anesthesia codes are ok, ***;
  *** others have to be marked as odd services              ***;
  if anesth   = '1' then do;
     if newbetdt ne 'P0 ' then do;
        cond04   = 1;
        newbetdt = 'P0 ';
        newbetag = '2';
     end;
  end ;
  drop anesth;


*********************************************************
COND06
Minor Services Associated With Major Services
*********************************************************;
  length split assist bilat $1;

  *** minor/split surgery ***;
  if  mod1 in ('52','53','54','55','56','62','57','62','65') or
      mod2 in ('52','53','54','55','56','62','57','62','65') then split = '1';

  *** assistance at surgery ***;
  if mod1 in ('80','81','82') or
     mod2 in ('80','81','82') then assist = '1';

  *** bilateral/multiple procedures ***;
  if mod1 = '51' or
     mod2 = '51' or
     mod1 = '50' or
     mod2 = '50' then bilat = '1';

  if split  = '1'  or
     assist = '1'  or
     bilat  = '1'  then cond06 = 1;

  drop split assist bilat;


*********************************************************
COND07
Adjustment Record
*********************************************************;
  if billtype not in ('1', '8') then cond07 = 1;


*********************************************************
Create MCR_STAT
Add a flag for Medicare Payment Status Code, we are
specifically interested in identifying services with
Status Code C
(miscellaneous coded priced by carrier report)
*********************************************************;
  length mcr_stat $ 1 ;
  label mcr_stat = 'Medicare Payment Status Code';

  mcr_stat = put(cpt,$RVU06ST.) ;


*********************************************************
COND05
```

```
Facility Fees
*********************************************************;
  if (newspec in ('090','091','092','093') or
      mod1 = 'SG'                          or
      mod2 = 'SG')
  or (newmod = 'TC')                       then cond05 = 1;


*********************************************************
COND08
Ambiguous Payer Type
*********************************************************;
  if plan eq '' then cond08 = 1;


*********************************************************
COND14
Possible Secondary Payer Involved
*********************************************************;
  if covoth_i = '2' then cond14 = 1;


*********************************************************
COND15
RVU <= 0
*********************************************************;
  if rvu_i le 0 then cond15 = 1;


*********************************************************
Add a Variable to Speed Sorting in the Next Step
*********************************************************;
  length sortvar $ 100;
  length chardate $ 7;
  length odd $ 2;

/* this (prematurely figures) out the equivalent of f_oddsvc so it can be part of the
sort */

  array conds(15) cond01-cond15;
  condcnt= sum(of cond01-cond15);

  if condcnt=0 then odd=' ';
  else
  if condcnt gt 1 then odd = '99';
  else do j = 1 to 15;
       if conds{j}=1 then do;
          odd=put(j,z2.);
       end;
  end;

/* make date character */

  chardate = put(enddt,julian7.);
  sortvar = trim(pidbdg) || trim(chardate) || trim(cpt) || odd || trim(newmod);
run;


****************************************
COND09
Flag Duplicates By Service Date/CPT
Keep Only Highest Payment Record
****************************************;

proc sort data = &sasDTout._4 out = cond_a /* tagsort */;
  by sortvar descending payment_orig;
run;

data cond_b (drop = sortvar);
  set cond_a;
  by sortvar descending payment_orig;
```

```
  if first.sortvar ne 1 then cond09 = 1;
run;


*********************************************
COND10
COND11
COND12
COND13
Impose The HCFA Actuaries' Trim
-- Screen out bills with allowed less
   than one third or more than three
   times the mean allowed charge for the
   entire database
-- restrict to only good records (those
   which will be used in the pricing
   analysis
-- Note: This will not catch major outliers
   in the miscellaneous codes with
   Medicare status = 'C'
*********************************************;

/* cond_c is good records only, used for the means */

data cond_c;
  set cond_b;
  if plan in ('HMOFFS', 'NONHMO')  and
     sum (of cond01-cond15) = 0    then output;
run;

proc sort data = cond_c;
  by cpt newmod;
run;

/* use only valid records to calculate means */

proc means data = cond_c mean noprint;
  var payment_orig;
  output out = cptsum mean = payment_;
  by cpt newmod;
run;

/* note asymmetric actuaries trim */
/* note this step also assigns condition flags cond10-cond13 */

proc sort data = cond_b;
  by cpt newmod;
run;

data &sasDTout._5 (drop = _type_ _freq_);
  merge cond_b
        cptsum;
  by cpt newmod ;

  cf&yr = 37.8975;    /* 2006 medicare fee schedule conversion factor - same as 2005
conversion factor */

  *** do not use odd services ***;
  if sum (of cond01-cond15) = 0 then ok = '1';

  if mcr_stat ne 'C'             and
     ok = '1'                    and
     plan in ('HMOFFS','NONHMO') then do ;
       if payment_orig < .33*payment_ then cond10 = 1;
       else if payment_orig >   6*payment_ then cond11 = 1;
       if rvu_i gt 0 and payment_orig < .25*cf&yr*rvu_i then cond12 = 1;
       if rvu_i gt 0 and payment_orig >   8*cf&yr*rvu_i then cond13 = 1;
  end;
run;
```

```
** assign Odd Services Flag **;

data &sasDTout._6;
  set &sasDTout._5;
  length f_oddsvc $ 2;
  label f_oddsvc = 'Odd Service Flag';

  array conds(15) cond01-cond15;
  condcnt = sum(of cond01-cond15);

  if condcnt = 0 then f_oddsvc = ' ';
  else if condcnt gt 1 then f_oddsvc = '99';
  else do j = 1 to 15;
       if conds{j} = 1 then do;
          f_oddsvc = put(j,z2.);
       end;
  end;
run;


*************************;
**  Final Adjustments  **;
*************************;

data inout.&sasDTout (compress = binary
                      keep = age
                             age12&year._nr
                             allow_orig
                             begday
                             begindt
                             begmo
                             begyr
                             betosagg
                             betosdet
                             bill_orig
                             billtype
                             birthdate
                             birthmo
                             birthyr
                             cdhpind
                             cf2006
                             claimcn
                             claimpdd
                             claimpmo
                             claimpyr
                             clmpdate
                             clmrelco
                             cnty_p2
                             county_p
                             covoth_i
                             covothr
                             covtype
                             cpt
                             d_dat_f
                             d_dnt_f
                             d_i7k_f
                             d_inp_f
                             d_zip_f
                             delivtyp
                             disenrolldt
                             dx1 - dx10
                             dxdot1 - dxdot10
                             dxval1 - dxval10
                             encnt_id
                             endday
                             enddt
                             endmo
                             endyr
                             enrolldt
                             f_baddt
                             f_cap_a
```

```
                              f_cap_orig
                              f_cpt
                              f_cpt_orig
                              f_demog
                              f_fac
                              f_impoth
                              f_imprvu
                              f_insamp
                              f_ncap_a
                              f_nspec
                              f_oddsvc
                              f_yr&nextyr
                              fedtaxid
                              file_id
                              id_26_06
                              id_tc_06
                              instate
                              los
                              mod1
                              mod2
                              newbetag
                              newbetdt
                              newmod
                              newspec
                              numdxc
                              partprov
                              patid
                              patliab_orig
                              pay_calc
                              pay_mcr
                              payertyp
                              payment_orig
                              pidbdg
                              plan
                              plan2
                              pnum
                              provid
                              reimb_orig
                              rvu
                              rvu_i
                              rvu_mp
                              rvu_pe
                              rvu_w
                              sex
                              spec1
                              spec2
                              spec3
                              spec_source
                              svcplace
                              unitind
                              units
                              zip_p
                              zip_s);
  set &sasDTout._6;

** check PAY_MCR & PAY_CALC - based on Odd Services Flagging **;

  if pay_calc > '0' and f_oddsvc ne '' then do;
     pay_mcr = .;
     pay_calc = '6';     * odd service -- do not use medicare payment rate;
  end;

** assign labels **;

  label betosagg    = 'BETOS Category, Aggregated'
        betosdet    = 'BETOS Category, Detailed'
        f_cpt       = 'Procedure Code Category'
        f_oddsvc    = 'Odd Service Flag'
        newbetag    = 'Aggregated BETOS, Reclassified'
        newbetdt    = 'Detailed BETOS, Reclassified'
        payertyp    = 'Payer Type(1=PRVHMO 2=PRVNHMO 3=MCRHMO 9=CAP)'
```

```
        plan        = 'Plan Type(HMOFFS/NONHMO/HMOCAP)'
        plan2       = 'Plan Type(HMOFFS/NONHMO/HMOCAP/MCRALL),All Ages'
        covoth_i    = 'Imputed COVOTHR'
        f_impoth    = 'Is Variable COVOTHR Imputed? (1/0)'
        f_cap_a     = 'Capit. Service used in Analysis (1/0)'
        f_ncap_a    = 'Non-Capit Service used in Analysis (1/0)'
        f_imprvu    = 'RVU Imputation Flag (1/0,2)'
        newspec     = 'Revised Speciality Code for Complex Algorithm'
        f_nspec     = 'Flag, Status of Specialty Code'
        pay_calc    = 'Method for Calculating Medicare Payment'
        rvu         = 'Assigned RVU'
        spec_source = 'Source of NEWSPEC specialty codes';

run;


proc contents data = inout.&sasDTout;
  title4 "Contents";
run;

proc freq data = inout.&sasDTout;
  tables f_oddsvc f_cpt / list missing;
  title4 "Frequencies of F_ODDSVC & F_CPT";
run;

proc print data = inout.&sasDTout (obs = 20);
  title4 "Samples of Records";
run;


proc freq data = &sasDTout._6;
  tables f_oddsvc * f_oddsvc_o f_cpt * f_cpt_o plan * plan_o
         plan2 * plan2_o payertyp * payertyp_o
         betosagg * betosagg_o newbetag * newbetag_o
         betosdet * betosdet_o newbetdt * newbetdt_o / list missing;
  title4 "Crossed Frequency Tables of Variables Before and After Re-Assigned";
run;


%jobend;


dm "log;    print file = I:\MCdb\datacol\&payernum\programs\&payernum._enc4.log replace";
dm "output; print file = I:\MCdb\datacol\&payernum\programs\&payernum._enc4.lst replace";

%jobstart;

proc datasets kill;
run;


*==========================================================================================
===
Program   :I:\mcdb\datacol\pxxx\reports\pxxx_enc4_data_completeness.sas.
Notes     :For recipient counts using Proc Summary, do NOT include the Missing option.
           We do not want to consider records with a missing patient ID as a known
recipient.
------------------------------------------------------------------------------------------
---
input     :pxxx_enc4.sas7bdat
output    :delivtp_pxxx_20xx.sas7bdat
           covtype_pxxx_20xx.sas7bdat
           final_pxxx_20xx.sas7bdat
           plan2_pxxx_20xx.sas7bdat
           totrecip_20xx.sas7bdat
------------------------------------------------------------------------------------------
---
Programmer:Laurie Hamilton
Date      :June 22, 2005

Modified  :Sireesha Thati
```

```
Date      :September 28, 2005

Modified  :Sireesha Thati
Date                  :October 06,2006
**=========================================================================================
===
;

options obs = max
        errors = 2
;

%let thisyear = 05;
%let lastyear = 04;
%let payer = P020;
%let drive = I;

libname in "&drive:\mcdb\datacol\&payer.\sasdata";
libname stats&thisyear "&drive:\mcdb\datacol\&payer.\sasdata\dc_enc4";
libname stats&lastyear "C:\2004_MD\MCDB_2004\datacol\&payer.\sasdata\dc_enc4";
libname inj "J:\mcdb_raw";

title1 "Data Completeness Summary Report Statistics - 20&thisyear";
title2 "Source - &payer._enc4.sas7bdat";

*****************************************************************************************

                                     Statistics

*****************************************************************************************
;
**********************
Adjust missing values

For this version, there
are no RVUs in the database
yet. Create a variable equal
to 0 in all cases.
**********************
;
data temp;
     set in.&payer._enc4 (keep = covtype delivtyp plan2 pidbdg payment);

     *** adjust missing coverage type ***;
     if covtype not in ('1', '2', '3', '4', '5', '6', '7','8','9') then covtype = 'X';

     *** adjust missing delivery system ***;
     if delivtyp not in ('1', '2', '3', '4', '5') then delivtyp = 'X';

     *** adjust missing plan 2 ***;
     if plan2 = ' ' then  plan2 = 'MISS';

     rvu_i = 0;
run;

**********************
Delivery System
**********************
;
** number of recipients **;
proc summary data = temp nway;
     class delivtyp pidbdg;
     output out = tempdr (drop = _type_);
run;

proc summary data = tempdr nway missing;
     class delivtyp;
     output out = drecip (drop = _type_ rename = (_freq_ = nrecip_&thisyear));
run;

*** number of services ***;
```

```
proc summary data = temp nway missing;
     class delivtyp;
     output out = dserv (drop = _type_ rename = (_freq_ = nserv&thisyear));
run;

*** total payments and total rvus ***;
proc summary data = temp nway missing;
     class delivtyp;
     var payment rvu_i;
     output out = dpayrvu (drop = _type_ _freq_)
                           sum   = totpay&thisyear
                                   totrvu&thisyear
;
run;

*** combine stats and output permanent dataset ***;
data stats&thisyear..delivtyp_&payer._20&thisyear;
     merge drecip
           dserv
           dpayrvu
;
     by delivtyp;
run;

title3 "Delivery System - &thisyear";
proc print data = stats&thisyear..delivtyp_&payer._20&thisyear;
run;

********************
Coverage Type
********************
;
** number of recipients **;
proc summary data = temp nway;
     class covtype pidbdg;
     output out = tempcr (drop = _type_);
run;

proc summary data = tempcr nway missing;
     class covtype;
     output out = crecip (drop = _type_ rename = (_freq_ = nrecip_&thisyear));
run;

*** number of services ***;
proc summary data = temp nway missing;
     class covtype;
     output out = cserv (drop = _type_ rename = (_freq_ = nserv&thisyear));
run;

*** total payments and total rvus ***;
proc summary data = temp nway missing;
     class covtype;
     var payment rvu_i;
     output out = cpayrvu (drop = _type_ _freq_)
                           sum   = totpay&thisyear
                                   totrvu&thisyear
;
run;

*** combine stats and output permanent dataset ***;
data stats&thisyear..covtype_&payer._20&thisyear;
     merge crecip
           cserv
           cpayrvu
;
     by covtype;
run;

title3 "Coverage Type - &thisyear";
proc print data = stats&thisyear..covtype_&payer._20&thisyear;
run;
```

```
*********************************************************
plan2

plan2 is created for all services, and includes Medicare
coverage as a separate category
*********************************************************
;
** number of recipients **;
proc summary data = temp nway;
     class plan2 pidbdg;
     output out = temppr (drop = _type_);
run;

proc summary data = temppr nway missing;
     class plan2;
     output out = precip (drop = _type_ rename = (_freq_ = nrecip_&thisyear));
run;

*** number of services ***;
proc summary data = temp nway missing;
     class plan2;
     output out = pserv (drop = _type_ rename = (_freq_ = nserv&thisyear));
run;

*** total payments and total rvus ***;
proc summary data = temp nway missing;
     class plan2;
     var payment rvu_i;
     output out = ppayrvu (drop = _type_ _freq_)
            sum  = totpay&thisyear totrvu&thisyear;
run;

*** combine stats and output permanent dataset ***;
data stats&thisyear..plan2_&payer._20&thisyear;
     merge precip
           pserv
           ppayrvu
;
     by plan2;
run;

title3 "Plan2 - &thisyear";
proc print data = stats&thisyear..plan2_&payer._20&thisyear;
run;

*****************
Total Recipients
Overall
*****************
;
proc summary data = temp nway;
     class pidbdg;
     output out = temprecip (drop = _type_);
run;

proc summary data = temprecip nway missing;
     output out = stats&thisyear..totrecip_&payer._20&thisyear (drop = _type_ rename =
(_freq_ = tnrecip&thisyear));
run;

title3 "Total Recipients - &thisyear";
proc print data = stats&thisyear..totrecip_&payer._20&thisyear;
run;

*******************************************************************************************
*********
```

Create Report

```
*****************************************************************************************
*********
;
*****************
Delivery System
*****************
;
data devs_temp1;
     merge stats&lastyear..delivtyp_&payer._20&lastyear
                                 (rename = (nrecip_&lastyear = nr_lastyr
                                            nserv&lastyear   = ns_lastyr
                                            totpay&lastyear  = tp_lastyr
                                            totrvu&lastyear  = tr_lastyr
                                           )
                                 )

           stats&thisyear..delivtyp_&payer._20&thisyear
                                 (rename = (nrecip_&thisyear  = nr_thisyr
                                            nserv&thisyear    = ns_thisyr
                                            totpay&thisyear   = tp_thisyr
                                            totrvu&thisyear   = tr_thisyr
                                           )
                                 )
;
     by delivtyp;

     nr_chg = (nr_thisyr - nr_lastyr)/nr_lastyr*100;
     ns_chg = (ns_thisyr - ns_lastyr)/ns_lastyr*100;
     tp_chg = (tp_thisyr - tp_lastyr)/tp_lastyr*100;
     *tr_chg = (tr_thisyr - tr_lastyr)/tr_lastyr*100;

     length rownum 8;
     if delivtyp = "X" then rownum = 6;
     else rownum = delivtyp;

     tablenum = 1;
run;

*proc print data=devs_temp1;
*title3 "Both Years - Delivery System";
*run;

*****calculate total for row 8***;
proc means data = devs_temp1 noprint;
     var ns_lastyr
         ns_thisyr
         tp_lastyr
         tp_thisyr
         tr_lastyr
         tr_thisyr
;
     output out = devs_tot
            sum = ns_lastyr
                  ns_thisyr
                  tp_lastyr
                  tp_thisyr
                  tr_lastyr
                  tr_thisyr
;
run;

data devs_temp2;
     merge devs_tot
           stats&lastyear..totrecip_&payer._20&lastyear
           stats&thisyear..totrecip_&payer._20&thisyear
;

     nr_lastyr = tnrecip&lastyear;
     nr_thisyr = tnrecip&thisyear;
```

```
     nr_chg = (nr_thisyr - nr_lastyr)/nr_lastyr*100;
     ns_chg = (ns_thisyr - ns_lastyr)/ns_lastyr*100;
     tp_chg = (tp_thisyr - tp_lastyr)/tp_lastyr*100;
     *tr_chg = (tr_thisyr - tr_lastyr)/tr_lastyr*100;


     tablenum = 1;
     rownum = 8;
     drop _type_ _freq_ tnrecip&lastyear tnrecip&thisyear;
run;

*proc print data=devs_temp2;
*run;

data devs_final;
     set devs_temp1
         devs_temp2
;
run;

title3 "Delivery System - Final Stats";
proc print data=devs_final;
     var tablenum rownum
         nr_lastyr nr_thisyr nr_chg
         ns_lastyr ns_thisyr ns_chg
         tp_lastyr tp_thisyr tp_chg
         tr_lastyr tr_thisyr /*tr_chg*/
;
run;

***********************
Coverage Type
***********************
;
data covtype_temp1;
     merge stats&lastyear..covtype_&payer._20&lastyear
                                (rename = (nrecip_&lastyear = nr_lastyr
                                           nserv&lastyear   = ns_lastyr
                                           totpay&lastyear  = tp_lastyr
                                           totrvu&lastyear  = tr_lastyr
                                          )
                                )

           stats&thisyear..covtype_&payer._20&thisyear
                                (rename = (nrecip_&thisyear = nr_thisyr
                                           nserv&thisyear   = ns_thisyr
                                           totpay&thisyear  = tp_thisyr
                                           totrvu&thisyear  = tr_thisyr
                                          )
                                )
;
     by covtype;

     nr_chg = (nr_thisyr - nr_lastyr)/nr_lastyr*100;
     ns_chg = (ns_thisyr - ns_lastyr)/ns_lastyr*100;
     tp_chg = (tp_thisyr - tp_lastyr)/tp_lastyr*100;
     *tr_chg = (tr_thisyr - tr_lastyr)/tr_lastyr*100;

     length rownum 8;
     if covtype = "X" then rownum = 9;
     else rownum = covtype;

     tablenum = 3;
run;

*****calculate total for row 11***;
proc means data = covtype_temp1 noprint;
     var ns_lastyr
         ns_thisyr
         tp_lastyr
         tp_thisyr
```

```
         tr_lastyr
         tr_thisyr
     ;
     output out = covtype_tot
            sum = ns_lastyr
                  ns_thisyr
                  tp_lastyr
                  tp_thisyr
                  tr_lastyr
                  tr_thisyr
     ;
run;

data covtype_temp2;
     merge covtype_tot
           stats&lastyear..totrecip_&payer._20&lastyear
           stats&thisyear..totrecip_&payer._20&thisyear;

     nr_lastyr = tnrecip&lastyear;
     nr_thisyr = tnrecip&thisyear;


     nr_chg = (nr_thisyr - nr_lastyr)/nr_lastyr*100;
     ns_chg = (ns_thisyr - ns_lastyr)/ns_lastyr*100;
     tp_chg = (tp_thisyr - tp_lastyr)/tp_lastyr*100;
     *tr_chg = (tr_thisyr - tr_lastyr)/tr_lastyr*100;


     tablenum = 3;
     rownum = 11;
     drop _type_ _freq_ tnrecip&lastyear tnrecip&thisyear;
     run;

data covtype_final;
     set covtype_temp1
         covtype_temp2
;
run;

title3 "Coverage Type - Final Stats";
proc print data=covtype_final;
     var tablenum rownum
         nr_lastyr nr_thisyr nr_chg
         ns_lastyr ns_thisyr ns_chg
         tp_lastyr tp_thisyr tp_chg
         tr_lastyr tr_thisyr /*tr_chg*/
;
run;

************************
PLAN2
************************
;
data plan2_temp1;
     merge stats&lastyear..plan2_&payer._20&lastyear
           (rename = (nrecip_&lastyear = nr_lastyr
                      nserv&lastyear   = ns_lastyr
                      totpay&lastyear  = tp_lastyr
                      totrvu&lastyear  = tr_lastyr))

           stats&thisyear..plan2_&payer._20&thisyear
           (rename = (nrecip_&thisyear = nr_thisyr
                      nserv&thisyear   = ns_thisyr
                      totpay&thisyear  = tp_thisyr
                      totrvu&thisyear  = tr_thisyr))
;
     by plan2;

     nr_chg = (nr_thisyr - nr_lastyr)/nr_lastyr*100;
     ns_chg = (ns_thisyr - ns_lastyr)/ns_lastyr*100;
     tp_chg = (tp_thisyr - tp_lastyr)/tp_lastyr*100;
```

```
     *tr_chg = (tr_thisyr - tr_lastyr)/tr_lastyr*100;

     length rownum 8;
     if plan2 = "NONHMO" then rownum = 1;
     else
     if plan2 = "HMOFFS" then rownum = 2;
     else
     if plan2 = "HMOCAP" then rownum = 3;
     else
     if plan2 in ("MCRALL", "MCRHMO") then rownum = 4;

     if plan2 = "MISS" then rownum = 5;

     tablenum = 2;
run;

*****calculate total for row 6***;
proc means data = plan2_temp1 noprint;
     var ns_lastyr
         ns_thisyr
         tp_lastyr
         tp_thisyr
         tr_lastyr
         tr_thisyr
;
     output out = plan2_tot
            sum = ns_lastyr
                  ns_thisyr
                  tp_lastyr
                  tp_thisyr
                  tr_lastyr
                  tr_thisyr
;
run;

data plan2_temp2;
     merge plan2_tot
           stats&lastyear..totrecip_&payer._20&lastyear
           stats&thisyear..totrecip_&payer._20&thisyear;

     nr_lastyr = tnrecip&lastyear;
     nr_thisyr = tnrecip&thisyear;


     nr_chg = (nr_thisyr - nr_lastyr)/nr_lastyr*100;
     ns_chg = (ns_thisyr - ns_lastyr)/ns_lastyr*100;
     tp_chg = (tp_thisyr - tp_lastyr)/tp_lastyr*100;
     *tr_chg = (tr_thisyr - tr_lastyr)/tr_lastyr*100;


     tablenum = 2;
     rownum = 6;
     drop _type_ _freq_ tnrecip&lastyear tnrecip&thisyear;
run;

data plan2_final;
     set plan2_temp1
         plan2_temp2
;
run;

title "Plan2 - Final Stats";
proc print data=plan2_final;
     var tablenum rownum
         nr_lastyr nr_thisyr nr_chg
         ns_lastyr ns_thisyr ns_chg
         tp_lastyr tp_thisyr tp_chg
         tr_lastyr tr_thisyr /*tr_chg*/
;
```

```
*****************************************************************************************
********************

                                          Create the Table

*****************************************************************************************
**********************
;
******************************
Create Dummy dataset to ensure
that the table data contains all
possible rows, regardless of the
particular payer
********************************
;
data dummy;

     tablenum=1;
     do rownum = 1 to 8;
        output;
     end;

     tablenum=2;
     do rownum = 1 to 6;
        output;
     end;

     tablenum=3;
     do rownum = 1 to 11;
        output;
     end;
run;

data all_in_all;
     set devs_final
         covtype_final
         plan2_final
;
run;

proc sort data=all_in_all;
     by tablenum rownum;
run;

data stats&thisyear..&payer._enc4_dc_crunch;
     merge dummy (in=indummy)
           all_in_all
;
     by tablenum rownum;

     if indummy then output;
run;

title3 "Data Ready for Spreadsheet";
proc print data=stats&thisyear..&payer._enc4_dc_crunch;
var delivtyp covtype plan2
    tablenum rownum

    nr_lastyr nr_thisyr nr_chg
    ns_lastyr ns_thisyr ns_chg
    tp_lastyr tp_thisyr tp_chg
    tr_lastyr tr_thisyr /*tr_chg*/
    ;

run;

%jobend;

dm "log;    print file =
I:\mcdb\datacol\&payer.\reports\&payer._enc4_data_completeness.log replace";
```

```
dm "output; print file =
I:\mcdb\datacol\&payer.\reports\&payer._enc4_data_completeness.lst replace";

%jobstart;

*******************************************************************
Program  : pxxx_enc4_DC_xls.sas
Project  : Maryland
Purpose  : For the report on the Data Completeness Summary from the Enc4 Data, create an
Excel
           spreadsheet from the crunched sas dataset

Notes    : 1. This program includes both the macro to create the spreadsheet
              and the macro call
           2. Programmer change payer, titles, filenames, and literals for
              titles in macro call at the end of the program.
-------------------------------------------------------------------

Input    : i:\mcdb\datacol\pxxx\sasdata\dc_enc4\pxxx_enc4_dc_crunch.sas7bdat
Output   : i:\mcdb\datacol\pxxx\reports\pxxx_enc4_dc_crunch.xls

--------------------------------------------------------------------
Date     : 07/06/2005
By       : Raymond Hu

Modified : 09/28/2005
           Sireesha Thati

Modified : 10/10/2006 for 2005 MCDB
                                        Sireesha Thati
*******************************************************************
;
options mprint
         mlogic
         symbolgen
;


****************************
Spreadsheet Macro
****************************
;
%macro Create_Xls_Table(This_year        =,
                        SASDataLib       =,
                        SASDataName      =,
                        Pnum             =,
                        Pname            =,
                        elserv           =,
                        subserv          =,
                        fmonday          =,
                        fyear            =,
                        ExcelOutPathName =
                        )
;

   options noxwait noxsync;

   *******************************************************************
      1.       Open Microsoft Excel
   *******************************************************************
;
   *** excel location on maryland room computer ***;
   x "C:\Progra~1\Micros~2\Office10\EXCEL.EXE ";

   *** excel location on lauries office computer ***;
   *x "C:\Progra~1\OfficeXP\Office10\EXCEL.EXE ";


   ** Sleep to allow time for Excel to come up **;
   data _null_;
     x=sleep(3);
```

```
   run;

   ******************************************************************
     2.        Open another Excel worksheet
   ******************************************************************
;
   %let cmopen2='[OPEN("i:\mcdb\datacol\include_program\enc1_dc_shell.xls")]';

   %put &cmopen2;

   filename cmds dde 'excel|system';

   data _null_;
     file cmds;
     put &cmopen2;
   run;

   data _null_;
     x=sleep(3);
   run;

   ******************************************************************
    3.        Write data to Excel by DDE
   ******************************************************************
;
   ******************************************************************
    3.1       Write header to Excel by DDE
   ******************************************************************
;


   %let
cmxcpt='Excel|i:\mcdb\datacol\include_program\[enc1_dc_shell.xls]Data_Completeness_Report
!R4C1:R9C18';
   ** Establish the DDE link **;
   filename caseman dde &cmxcpt notab;
   %let elserv = %sysfunc(putn(&elserv,comma12.));
   %let subserv = %sysfunc(putn(&subserv,comma12.));

   data _null_;
      file caseman ;
          *** concatenate payer number and payer name ***;
      pnumname = "&pnum" || ": " || "&pname";

      *** Create text for source file ***;
      text8a = "Eligible Services  : " || "&elserv";
      text8b = "Source File  : " || "&payer._enc4_dc.sas7bdat";
      text9a = "Services Submitted  : " || "&subserv";
      text9b = "File Date : "|| "&fmonday" || ', '|| "&fyear";

      put pnumname;
      put "Data After Encounter Processing";
      put "Data Completeness Summary Report";
      put ;
      put text8a '09'x '09'x '09'x '09'x '09'x '09'x  text8b  ;
      put text9a '09'x '09'x '09'x '09'x '09'x '09'x  text9b  ;
      run;


   ******************************************************************
    3.2       Write data to Excel-table1 by DDE
   ******************************************************************
;

   /*%let
cmxcpt='Excel|i:\mcdb\datacol\data_completeness_summaries\[shell_3.xls]Data_Completeness_
Report!R25C2:R30C18';*/

  %let
cmxcpt='Excel|i:\mcdb\datacol\include_program\[enc1_dc_shell.xls]Data_Completeness_Report
!R14C2:R21C18';
```

```
   ** Establish the DDE link **;
   filename caseman dde &cmxcpt notab;


   data _null_;
    file caseman ;
    set &SASDataLib..&SASDataName end=eof;

    if tablenum=1 then do;
       if nr_LastYr ne . or nr_ThisYr ne . or nr_chg ne . or
          ns_LastYr ne . or ns_ThisYr ne . or ns_chg ne . or
          tp_LastYr ne . or tp_ThisYr ne . or tp_chg ne .
          then do;

          put nr_LastYr    '09'x
              nr_ThisYr    '09'x
              nr_chg       '09'x
              ns_LastYr    '09'x
              ns_ThisYr    '09'x
              ns_chg       '09'x
              tp_LastYr    '09'x
              tp_ThisYr    '09'x
              tp_chg       '09'x
              ;
          end;

       else put;

      end;

   run;


   ******************************************************************
    3.3       Write data to Excel-table2 by DDE
   ******************************************************************
;

   /*%let
cmxcpt='Excel|i:\mcdb\datacol\data_completeness_summaries\[shell_3.xls]Data_Completeness_
Report!R25C2:R30C18';*/

  %let
cmxcpt='Excel|i:\mcdb\datacol\include_program\[enc1_dc_shell.xls]Data_Completeness_Report
!R26C2:R31C18';

   ** Establish the DDE link **;
   filename caseman dde &cmxcpt notab;


   data _null_;
    file caseman ;
    set &SASDataLib..&SASDataName end=eof;

    if tablenum=2 then do;
       if nr_LastYr ne . or nr_ThisYr ne . or nr_chg ne . or
          ns_LastYr ne . or ns_ThisYr ne . or ns_chg ne . or
          tp_LastYr ne . or tp_ThisYr ne . or tp_chg ne .
          then do;

          put nr_LastYr    '09'x
              nr_ThisYr    '09'x
              nr_chg       '09'x
              ns_LastYr    '09'x
              ns_ThisYr    '09'x
              ns_chg       '09'x
              tp_LastYr    '09'x
              tp_ThisYr    '09'x
              tp_chg       '09'x
              ;
```

```
          end;

       else put;

      end;

   run;

   ******************************************************************
    3.4       Write data to Excel-table3 by DDE
   ******************************************************************
;

   /*%let
cmxcpt='Excel|i:\mcdb\datacol\data_completeness_summaries\[shell_3.xls]Data_Completeness_
Report!R25C2:R30C18';*/

  %let
cmxcpt='Excel|i:\mcdb\datacol\include_program\[enc1_dc_shell.xls]Data_Completeness_Report
!R36C2:R48C18';

   ** Establish the DDE link **;
   filename caseman dde &cmxcpt notab;


   data _null_;
    file caseman ;
    set &SASDataLib..&SASDataName end=eof;

    if tablenum=3 then do;
       if nr_LastYr ne . or nr_ThisYr ne . or nr_chg ne . or
          ns_LastYr ne . or ns_ThisYr ne . or ns_chg ne . or
          tp_LastYr ne . or tp_ThisYr ne . or tp_chg ne .
          then do;

          put nr_LastYr    '09'x
              nr_ThisYr    '09'x
              nr_chg       '09'x
              ns_LastYr    '09'x
              ns_ThisYr    '09'x
              ns_chg       '09'x
              tp_LastYr    '09'x
              tp_ThisYr    '09'x
              tp_chg       '09'x
              ;
          end;

       else put;

      end;

   run;



   *********************************************************
     7.  Save as and close Xls
   *********************************************************
;
   %let rc3 = %sysfunc(fileexist("&exceloutPathName"));
   data _null_;
     if &rc3 = 1 then do; x " del &exceloutPathName"; end;
   run;

   data _null_;
     x=sleep(3);
   run;

   data _null_;
     file cmds;
```

```
     put "[SAVE.AS(%bquote("&exceloutPathName"))]";
     put '[FILE.CLOSE()]';
     put '[QUIT()]';
   run;

%mend;


%macro notused;

libname in2 "C:\work_HCACC\Xls2";
proc contents data=in2.p020_enc4_dc_crunch (obs=8); run;
proc print data=in2.p020_enc4_dc_crunch (obs=20); run;

%LET xxDSID=%SYSFUNC(OPEN(in2.cball,i));
%LET xxNVARS=%SYSFUNC(ATTRN(&xxDSID,nvars));
%LET xxNOBS=%SYSFUNC(ATTRN(&xxDSID,nlobs));

%put "xxNOBS=&xxNOBS";

%let payer = P020;
%let pname = Aetna Life and Health Insurance Co.;
***libname in2 "i:\mcdb\datacol\&payer.\sasdata\dc_enc4\";
libname in2 "C:\work_HCACC\Xls2\";

%let This_year = 2005;
%let SASDataLib = in2;
%let SASDataName = %str(&payer._enc4_dc_crunch);
%let Pnum = %str(&payer);
%let Pname = %str(&pname);
%let elserv  = 5479758;
%let subserv = 5481385;
%let ExceloutPathName = %str(i:\mcdb\datacol\&payer.\reports\&payer._enc4_dc_crunch.xls);
%let ExceloutPathName =
%str(C:\work_HCACC\Xls2\&payer.\reports\&payer._enc4_dc_crunch.xls);


%mend notused;


   ******************************************************************
    Macro Call
   ******************************************************************
;

%let payer = P020;
%let pname = Aetna Life and Health Insurance Co.;
%let fmonday = October 11;
%let fyear   = 2006;
libname in2 "i:\mcdb\datacol\&payer.\sasdata\dc_enc4\";
***libname in2 "C:\work_HCACC\Xls2\";
%Create_Xls_Table(This_year = 2005,
                  SASDataLib = in2,
                  SASDataName = %str(&payer._enc4_dc_crunch),
                  Pnum = %str(&payer),
                  Pname = %str(&pname),
                  elserv  = 5479758,
                  subserv = 5481385,
                  fmonday = %str(&fmonday),
                  fyear = %str(&fyear),
                  ExceloutPathName =
%str(i:\mcdb\datacol\&payer.\reports\&payer._enc4_dc_crunch.xls)
                  );



%jobend;

dm "log;    print file = I:\mcdb\datacol\&payer.\reports\&payer._enc4_DC_xls.log
replace";
```

```
dm "output; print file = I:\mcdb\datacol\&payer.\reports\&payer._enc4_DC_xls.lst
replace";

proc datasets kill;
run;
quit;


%jobstart;

**==============================================================================**;
**                                                                              **;
** Program:    P020_enc5.sas                                                    **;
** Programmer: Kasey chen/Sanee Maphungphong                                    **;
** Date      : 11/6/2007                                                        **;
**                                                                              **;
** Input:      pxxx_Enc4.sas7bdat                                               **;
** Output:     pxxx_Enc5.sas7bdat                                               **;
**                                                                              **;
** Notes:                                                                       **;
**                                                                              **;
**  1. Per MHCC, drop AGE1206.                                                  **;
**                                                                              **;
**  2. Add 11 Variables to ENC5. In past years, these vairables were created  **;
**     at later stages. However, we need to add them now because we will use  **;
**     ENC5 data to create a preliminary analysis file.                        **;
**     - Variables usually created in "Combined" file: F_AGE_NR, F_BENRULE,   **;
**       F_FINMS, F_MODF, F_PATID, F_PAYMD, F_UNCV.                            **;
**     - Variables usually created in Analysis file: FACILITY, POS, REGION_P, **;
**       RVU_2006.                                                             **:
**                                                                              **;
**  3. Include program 'fix_nonfinan_2006' to recode variables according to   **;
**     payers' responses to questions from SSS.                                **;
**     - Affacted payers: P020, P030, P070, P130, P131, P160, P350, P471,     **;
**       P480, P500, P520, P820, P830, P850, P870.                             **;
**                                                                              **;
**  4. Use format program "Enc5_dmecode.sas" which containing CPT codes       **;
**     specified by MHCC (to P510) as codes of which UNITIND should be set to **;
**     '9'.                                                                    **;
**     - Affacted payers: P320, P330 and P471.                                 **;
**                                                                              **;
**  5. Recode invalid value to blank in ZIP_S, ZIP_P and mod1.                 **;
**                                                                              **;
**  6. Format DATE variables MMDDYY8.                                           **;
**                                                                              **;
**  7. Recode UNIT and UNITIND.                                                 **;
**                                                                              **;
**  8. Re-assign PLAN, PLAN2, and PAYERTYP. This is because DELIVTYP of some  **;
**     reocrds for P620 was changed.                                            **;
**                                                                              **;
**  9. Drop 31 variables from ENC4: BEGDAY BEGMO BEGYR BIRTHMO BIRTHYR         **;
**     CLAIMPDD CLAIMPMO CLAIMPYR DXDOT1 DXDOT2 DXDOT3 DXDOT4 DXDOT5 DXDOT6   **;
**     DXDOT7 DXDOT8 DXDOT9 DXDOT10 DXVAL1 DXVAL2 DXVAL3 DXVAL4 DXVAL5        **;
**        DXVAL6 DXVAL7 DXVAL8 DXVAL9 DXVAL10 ENDDAY ENDMO ENDYR.                 **;
**                                                                              **;
** Date: 11/30/2007  Modified by Kasey Chen                                     **;
**  1. Rename 10 variables that were imputed with *_i in Enc5                  **;
**     (Exam: UNITS = UNITS_ORIG and UNITS_I = UNITS).                          **;
**  2. Rename bill charge variables (Exam: ALLOW = ALLOW_ORIG).                **;
**  3. Relabel imputed variables.                                               **;
**                                                                              **;
** Date: 12/19/2007  Modified by Kasey Chen                                     **;
**  1. Include format for $region.                                              **;
**  2. Set length for AGE1206_NR, ENROLLDT, ENROLLDT_ORIG, DISENROLLDT,        **;
**     DISENROLLDT-ORIG to 8., add format for ENROLLDT.                         **;
**  3. Delete pre-define format $REGION in the program.                         **;
**  4. Add labels for COVTYPE, MCRHMO, MCROTH, PRVHMO, PRVNHMO, MOD2, and      **;
**     MOD2_ORIG.                                                               **;
**==============================================================================**;
options obs = max errors = 2 linesize = 120 pagesize = 86;
```

```
%let drive = I;
%let pnum = P020;
%let yr = 2006;
%let year = 06;
%let nextyr = 07;

%let sasDTin = &pnum._enc4;
%let sasDTout = &pnum._enc5;

%let hmolst = 'P030','P130','P160','P480','P520','P620','P680','P870';
%let mhmolst = 'P480';

libname inout "I:\MCdb\datacol\&pnum\sasdata";


title1 "MARYLAND HEALTH CARE COMMISSION";
title2 "Year 2006 &pnum Encounter File - &pnum._enc5";

footnote1 "%sysfunc(putn(%sysfunc(time()), hhmm.)) %sysfunc(putn(%sysfunc(date()),
weekdate.))";
footnote2 "program: I:\mcdb\datacol\&pnum\programs\&pnum._enc5.sas";

options nosource2;

*** cpt procedure codes ***;

*** Include format for create PRV1 variables ***;

%include "I:\mcdb\datacol\formats\posfmt.sas";
%include "I:\mcdb\datacol\formats\Enc5_dmecode.sas";
%include "I:\mcdb\datacol\formats\mdregfmt.sas";
options source2;


***** Create Enc5 data set *****;

data &sasDTout;
        set inout.&sasDTin (drop = begday begmo begyr birthmo birthyr claimpdd claimpmo
                                   claimpyr dxdot1 dxdot2 dxdot3 dxdot4 dxdot5 dxdot6
dxdot7
                              dxdot8 dxdot9 dxdot10 dxval1 dxval2 dxval3 dxval4 dxval5
                                   dxval6 dxval7 dxval8 dxval9 dxval10 endday endmo
endyr);

*****************************************************************************;
**===> Create indicator of variables and set up length for each new variable ***;
*****************************************************************************;
length unitind_i billtype_i delivtyp_i covothr_i    $1
       units_i age1206_nr enrolldt  disenrolldt      8
       cpt_i                                        $6
       mod1_i                                       $2
       zip_s_i zip_p_i                              $5;

       unitind_i = unitind;
       units_i   = units;
       cpt_i     = cpt;

       billtype_i= billtype;
       delivtyp_i= delivtyp;
       covothr_i = covothr;
       mod1_i    = mod1;
       zip_s_i   = zip_s;
       zip_p_i   = zip_p;
       disenrolldt_i = disenrolldt;

Format disenrolldt_I MMDDYY8.;

options nosource2;
*** Include 'fix_nonfinan_2006.sas to recode ' ***;
%include "&drive:\MCdb\DataCol\Include_Program\fix_nonfinan_2006.sas";
options source2;
```

```
      format begindt birthdate clmpdate enddt enrolldt MMDDYY8.;


*=========================================================*;
* Variables usually created in "Combined" file: F_AGE_NR, *;
* F_FINMS, F_MODF, F_PATID, F_PAYMD, F_UNCV.              *;
*=========================================================*;

         ************************************************
   Create F_UNCV (1/0)
   Flag service as uncovered/covered
   In the past, we used to keep track of missing
   values by using ., .C and .M
   However, we have not done this since, at least,
   2004. Thus we take .C and .M of the code.
   ************************************************
   ;
   length f_uncv $1;
   f_uncv = '0';

   if allow_orig in (0,.) and
      reimb_orig in (0,.) then f_uncv = '1';

   ********************************************
   Create F_FINMS (1/0)
   Flag for financial services
   F_FINMS = 1 if all four financial vars = 0
   or missing.

   In the past, we used to keep track of missing
   values by using ., .C and .M
   However, we have not done this since, at least,
   2004. Thus we take .C and .M of the code.
   ********************************************
   ;
   length f_finms $1;
   f_finms = '0';

   if bill_orig    in (0,.) and
      allow_orig   in (0,.) and
      reimb_orig   in (0,.) and
      patliab_orig in (0,.) then f_finms = '1';

   ********************************************
   Create F_MODF (1/0)
   Flag for at least one MODF1 or MODF2 code =
   TC or SG
   ********************************************
   ;
   length f_modf $1;
   f_modf = '0';

   if mod1 in ('TC','SG') or
      mod2 in ('TC','SG') then f_modf = '1';

   ************************************************
   Create F_PAYMD
   Flag Payment Category:
      1=negative payment
      2=0 payment
      3=positive payment
      4=missing payment
   ************************************************
   ;
   length f_paymd $ 1;

   if payment_orig=. then f_paymd = '4';
   else if payment_orig<0 then f_paymd = '1';
   else if payment_orig=0 then f_paymd = '2';
   else if payment_orig>0 then f_paymd = '3';
```

```
  ************************************************
        Create F_AGE_NR
  Flag for Valid Age Group
  Kasey Note: Per MHCC, AGE1206 is dropped, no
                 more F_AGE is created.
  ************************************************
  ;
  length f_age_nr $1;

  if 0 <= age1206_nr <= 110 then f_age_nr = '1';
  else f_age_nr = '0';

  ************************************
  Create F_PATID
  Flag for Valid PATID
  ************************************
  ;
  length f_patid $ 1;

  if patid = '' or
     compress(patid, "-") = '' or
     compress(patid, "0") = '' or
     compress(patid, "6") = '' or
     compress(patid, "8") = '' or
   compress(patid, "9") = '' then f_patid = "0";
  else f_patid = "1";


*==========================================================*;
* Variables usually created in Analysis file: FACILITY,   *;
* POS, REGION_P, RVU-2006.                                 *;
*==========================================================*;

  ****************************************
  Create POS
  Simplified Place of Service Code
  ****************************************
  ;
  length pos $10;

  pos = put(svcplace,$posfmt.);

  ****************************************
  Create REGION_P
  Patient Region of Residence
  ****************************************
  ;
  length region_p $1;

  region_p=put(cnty_p2, $region.);

  ******************************************************************
  Create RVU_2006
  This variable is used in Practitioner Report tables).
  RVU_YYYY will also be applied to data in previous year.
  ******************************************************************
  ;
  length rvu_2006 8;

  rvu_2006 = rvu_i;

  ************************************************************
  Create FACILITY
  identify facility-based claims,
  use the same algorithm which has been used to impute RVUs
  ************************************************************
  ;
  length facility $1;

  if svcplace in ("01"," 1","1 ","02","2 "," 2","21",
```

```
                    "22","23","24","31","34","51","52") then facility = "1";
     else facility = "0";


     ************************************************************
     Create F_BENRULE
     identify Bens rule for Capitation.
     ************************************************************
     ;
     length f_benrule $2;
     f_benrule = "  ";


     ****************************************
     Recode invalid values in ZIP-P and ZIP_S
     ****************************************
     ;
     if zip_p in ("00000") then zip_p_i = " ";

     if zip_s in ("00000", "*    ", "*   ", "*  ", "* ", "*",
                  "     ", "    ", "   ", "  ", " ",
                  "20   ", "20   ", "20 ", "20 ",
                  "7537 ", "7537", "2121 ", "2121", "99999", " 0000", "L3R 9",
                  "L6T5M", "ANWAR" ) then zip_s_i = " ";

     if mod1 in ("  ", " ", "-", "- ", ".", ". " ) then mod1_i = "  ";

         output &sasDTout;
run;


*****************************************************;
* Rename/rweset values of variables to keep them     **;
* consistent across all payers' ENC5.                **;
*****************************************************;

* Rename and reset variables here;
data inout.&sasDTout;
    length enrolldt_orig disenrolldt_orig 8;
         Format enrolldt MMDDYY8.;
        set &sasDTout (rename=(billtype      = billtype_orig
                               billtype_i    = billtype
                               cdhpind       = cdhpind_orig
                          covothr       = covothr_orig
                          covtype       = covtype_orig
                          cpt           = cpt_orig
                          cpt_i         = cpt
                          delivtyp          = delivtyp_orig
                          delivtyp_i      = delivtyp
                          disenrolldt   = disenrolldt_orig
                          disenrolldt_i = disenrolldt
                          enrolldt      = enrolldt_orig
                          mod1          = mod1_orig
                          mod1_i        = mod1
                          mod2          = mod2_orig
                          partprov      = partprov_orig
                          unitind       = unitind_orig
                          unitind_i     = unitind
                          units         = units_orig
                          units_i       = units
                          zip_p         = zip_p_orig
                          zip_p_i       = zip_p
                          zip_s         = zip_s_orig
                          zip_s_i       = zip_s));

* Create/Reset variables *;
  allow    = allow_orig;
  bill     = bill_orig;
  covtype  = covtype_orig;
  patliab  = patliab_orig;
  payment  = payment_orig;
```

```
  reimb    = reimb_orig;
  cdhpind  = cdhpind_orig;
  enrolldt = enrolldt_orig;
  mod2     = mod2_orig;
  partprov = partprov_orig;

if pnum="P020" then do;
   if covoth_i in ('C','M','W') then covoth_i = '9';
end;

drop covothr_i;

**************************************
set label
**************************************;

     label clmpdate = "Claim Paid Date"
           covtype   = "Coverage Type"
           mcrhmo    = "Flag, Medicare HMO"
           mcroth    = "Flag, Medicare Other"
           prvhmo    = "Flag, HMO"
           prvnhmo   = "Flag, Non HMO"
           mod2      = "AMA Modified II"
           mod2_orig = "AMA Modified II, (Original)"
           f_imprvu = "RVU Imputation Flag (1/0,2)"
           patid    = "Scrambled PATID"
           cnty_p2  = "County to Use for Pract. Report Analysis"
           f_fac    = "Service Performed in a Facility? (1/0)"
           pay_calc = "Method for Calculating Medicare Payment"
           pidbdg   = "Unique Patient ID(PNUM/PATID/BRTHYR/BRTHMO/GENDER)"
           rvu      = "Assigned RVU"
           rvu_i    = "Assigned/Imputed RVU"
           rvu_mp   = "Medicare phys fee schedule RVU, malpractice"
           rvu_pe   = "Medicare Phys Fee Schedule RVU, Pract Expense"
           spec1    = "Practitioner Specialty 1"
           f_patid  = "Flag for Valid PATID (1/0)"
           f_uncv   = "Uncovered services(1/0)"
           f_finms  = "All finan. var. are zero or missing"
           f_modf   = "1=There is TC/SG in MODF1/MODF2"
           f_paymd  = "Payment:1=negative, 2=0, 3=+, 4=missing"
           pos      = "Simplified Place of Service Code"
           region_p = "Patient Region of Residence"
           rvu_2006 = "2006 RVU applied to 2005 file"
           facility  = "Service Location is a Facility (1/0)"
           f_benrule = "Bens Rule for Capitation"

           unitind_orig     = "Service Unit Indicator, (Original)"
           units_orig       = "Number of Units for a Service, (Original)"
           cpt_orig         = "CPT-4 /HCPCS Procedure Code, (Original)"
           billtype_orig    = "Type of Bill, (Original), (Original)"
           delivtyp_orig    = "Delivery System Type, (Original)"
           covothr_orig     = "Patient Covered by Other Insurance, (Original)"
           mod1_orig        = "AMA Modified I, (Original)"
           zip_p_orig       = "Patient Residence ZIP Code, (Original)"
           zip_s_orig       = "Service Location ZIP Code, (Original)"
           bill_orig        = "Billed Charge-BC ($), (Original)"
           disenrolldt_orig = "Disenrollment Date, (Original)"
           unitind          = "Service Unit Indicator"
           units            = "Number of Units for a Service"
           cpt              = "CPT-4 /HCPCS Procedure Code"
           billtype         = "Type of Bill"
           delivtyp         = "Delivery System Type"
           mod1             = "AMA Modified I"
           zip_p            = "Patient Residence ZIP Code"
           zip_s            = "Service Location ZIP Code"
           disenrolldt      = "Disenrollment Date"
           bill_orig        = "Billed Charge-BC ($), (Original)"
           allow_orig       = "Allowed Amount-AA ($), (Original)"
           reimb_orig       = "Reimbursement Amount-RA ($), (Original)"
           patliab_orig     = "Patient Liability-PL ($), (Original)"
           payment_orig     = "Payment=Sum(RA,PL)($), (Original)"
```

```
          bill             = "Billed Charge-BC ($)"
          allow            = "Allowed Amount-AA ($)"
          reimb            = "Reimbursement Amount-RA ($)"
          patliab          = "Patient Liability-PL ($)"
          payment          = "Payment=Sum(RA,PL)($)"
          cdhpind          = "Consumer Directed Health Plan"
          cdhpind_orig     = "Consumer Directed Health Plan, (Original)"
          enrolldt         = "Enrollment Date"
          enrolldt_orig    = "Enrollment Date, (Original)"
          partprov         = "Participating Provider Flag"
          partprov_orig    = "Participating Provider Flag, (Original)"
          covtype_orig     = "Coverage Type, (Original)"
          covoth_i         = "Imputed/Coverted COVOTHR"
          ;

run;


*****************************************
Print Contents and Sample print from ENC5.
*****************************************;

proc contents data = inout.&sasDTout;
run;

proc freq data = inout.&sasDTout;
  tables
       unitind_orig     * unitind
       units_orig       * units
       billtype_orig    * billtype
       delivtyp_orig    * delivtyp
       covothr_orig     * covoth_i
       covtype          * covtype_orig
       mod1_orig        * mod1
       mod2_orig        * mod2/list missing;

title3 "&Pnum._ENC5";
title4 "Frequency for All of Original and Recoded Variables";
run;

proc freq data = inout.&sasDTout;
  table cpt_orig * cpt/list missing;
  where cpt_orig ne cpt;

title3 "&Pnum._ENC5";
title4 "Frequency for CPT of Original and Recoded Variables";
run;

proc freq data = inout.&sasDTout;
  table zip_s_orig * zip_s/list missing;
  where zip_s_orig ne zip_s;

title3 "&Pnum._ENC5";
title4 "Frequency for ZIP_S of Original and Recoded Variables";
run;

proc freq data = inout.&sasDTout;
  table zip_p_orig * zip_p/list missing;
  where zip_p_orig ne zip_p;

title3 "&Pnum._ENC5";
title4 "Frequency for ZIP_P of Original and Recoded Variables";
run;

proc freq data = inout.&sasDTout;
  table region_p/list missing;

title3 "&Pnum._ENC5";
title4 "Frequency for Region";
run;
```

```
proc print data = inout.&sasDTout (obs=100);
format enrolldt_orig enrolldt MMDDYY10.;
  var enrolldt_orig enrolldt;
  where enrolldt_orig ne enrolldt;

title3 "&Pnum._ENC5";
title4 "ENROLLDT of Original and Recoded Variables";
run;

proc print data = inout.&sasDTout (obs=100);
format disenrolldt_orig disenrolldt MMDDYY10.;
  var disenrolldt_orig disenrolldt;
  where disenrolldt_orig ne disenrolldt;

title3 "&Pnum._ENC5";
title4 "DISENROLLDT of Original and Recoded Variables";
run;

%jobend;

dm "log;    print file = I:\MCdb\datacol\&pnum\programs\&pnum._enc5.log replace";
dm "output; print file = I:\MCdb\datacol\&pnum\programs\&pnum._enc5.lst replace";
```

Appendix C

Payer-Specific Quality Review Statements for 2006

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2006 MCDB

(Data received August 7, 2007)

P020 Aetna Life and Health Insurance Co.

Encounter Layout: 6,735,681 records read in.

	Field Name	MHCC Comments
1.	Patient ID (encrypted)	
2.	Patient Date of Birth	2 (year = 0001)
3.	Patient Sex	515 (<1%) = 3 (Unknown)
4.	Consumer Directed Health Plan (CDHP) alone or w/ HAS or HRA indicator	
5.	Patient Zip Code	
6.	Patient Covered by Other Insurance	39,995 (0.49%) = 9 (not coded)
7.	Coverage Type	
8.	Delivery System	
9.	Claim Related Condition	100% = Missing
10.	Practitioner Federal Tax ID	
11.	Participating Provider Flag	
12.	Type of Bill	
13.	Claim Control Number	
14.	Claim Paid Date	
15.	Number of Diagnosis Codes	
16.	Number of Line Items	
17.	Diagnosis Code 1	45,119 (<1%) = Invalid
18.	Diagnosis Code 2	3,245,677 (48.2%) = Missing; 19,846 (<1%) = Invalid
19.	Diagnosis Code 3	4,837,255 (71.8%) = Missing; 12,023 (<1%) = Invalid
20.	Diagnosis Code 4	Almost 100% = Missing and/or Invalid
21.	Diagnosis Code 5	Almost 100% = Missing and/or Invalid
22.	Diagnosis Code 6	Almost 100% = Missing and/or Invalid
23.	Diagnosis Code 7	Almost 100% = Missing and/or Invalid
24.	Diagnosis Code 8	Almost 100% = Missing and/or Invalid
25.	Diagnosis Code 9	Almost 100% = Missing and/or Invalid
26.	Diagnosis Code 10	Almost 100% = Missing and/or Invalid

	Field Name	MHCC Comments
27.	Service From Date	
28.	Service Thru Date	
29.	Filler	
30.	Place of Service	
31.	Service Location Zip Code	46,345 (<1%) = Missing
32.	Service Unit Indicator	
33.	Units	
34.	Procedure Code	
35.	Modifier I	
36.	Modifier II	
37.	Servicing Practitioner ID	
38.	Billed Charge	
39.	Allowed Amount	84,674 (1.26%) with BC= 0
40.	Reimbursement Amount	
41.	Patient Liability	
42.	Date of Enrollment	
43.	Date of Disenrollment	

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2006 MCDB

(Data received August 7, 2007)

P020 Aetna Life and Health Insurance Co.

Provider Layout: **126,366 records read in; 0 records after deleting exact duplicates.**

	Field Name	MHCC Comments
1.	Servicing Practitioner ID	
2.	Practitioner Federal Tax ID	
3.	Practitioner Last Name	
4.	Practitioner First Name	
5.	Practitioner Middle Initial	
6.	Practitioner Name Suffix	
7.	Practitioner Credential	
8.	Practitioner Specialty – 1	
9.	Practitioner Specialty – 2	100% = Missing
10.	Practitioner Specialty – 3	100% = Missing
11.	Practitioner DEA #	
12.	Indicator for Multi-Practitioner Health Care Organization	
Percent of Exact Duplicate Records: 0%		
Percent of Records With No Match to Encounter: 14.50%		
Encounter Services File With Provider File Link 6,280,559 services matched by FEDTAXID and PROVID, 113,681 services matched by PROVID only, 341,441 services with no match		

NOTE: Encounter, Pharmacy & Provider Directory statistics based on data reflected in a variety of analytical files and do not represent the data from original submission to SSS.

Notes on Enrollment and Disenrollment Dates:

Description	Number of Records	
Enrollment Date = Disenrollment Date	9,098	(<1%)
Service Begin Date BEFORE Enrollment Date	13,363	(<1%)
Service Begin Date AFTER Disenrollment Date	74,944	(1.11%)

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2006 MCDB

(Data received June 27, 2007)

P030 Aetna U.S. HealthCare

Encounter Layout: **4,807,625 records read in.**

	Field Name	MHCC Comments
1.	Patient ID (encrypted)	
2.	Patient Date of Birth	
3.	Patient Sex	9 (<1%) = 3 (Unknown)
4.	Consumer Directed Health Plan (CDHP) alone or w/ HAS or HRA indicator	
5.	Patient Zip Code	
6.	Patient Covered by Other Insurance	1,136,562 (23.64 %) = 9 (Not Coded)
7.	Coverage Type	
8.	Delivery System	
9.	Claim Related Condition	100% = Missing
10.	Practitioner Federal Tax ID	
11.	Participating Provider Flag	
12.	Type of Bill	
13.	Claim Control Number	
14.	Claim Paid Date	
15.	Number of Diagnosis Codes	
16.	Number of Line Items	
17.	Diagnosis Code 1	337,249 (7.01 %) = Invalid
18.	Diagnosis Code 2	2,961,958 (61.61%) = Missing; 9,104 (<1%) = Invalid
19.	Diagnosis Code 3	38,452,208 (79.98%) = Missing; 4,874 (<1%) = Invalid
20.	Diagnosis Code 4	91% = Missing and/or Invalid
21.	Diagnosis Code 5	99.52% = Missing and/or Invalid
22.	Diagnosis Code 6	Almost 100% = Missing and/or Invalid
23.	Diagnosis Code 7	Almost 100% = Missing and/or Invalid
24.	Diagnosis Code 8	Almost 100% % = Missing and/or Invalid
25.	Diagnosis Code 9	Almost 100% = Missing and/or Invalid
26.	Diagnosis Code 10	100% = Missing and/or Invalid

	Field Name	MHCC Comments
27.	Service From Date	
28.	Service Thru Date	
29.	Filler	
30.	Place of Service	
31.	Service Location Zip Code	<1% = Missing/ Invalid
32.	Service Unit Indicator	
33.	Units	About 6% Missing/Zero
34.	Procedure Code	<1% = Missing or Invalid
35.	Modifier I	4,258,960 (88.59%) = Missing
36.	Modifier II	4,774,330 (99.31%) = Missing
37.	Servicing Practitioner ID	
38.	Billed Charge	124,879 with BC = 0
39.	Allowed Amount	113,013 with AA = 0
40.	Reimbursement Amount	115,295 with RA = 0
41.	Patient Liability	
42.	Date of Enrollment	
43.	Date of Disenrollment	

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2006 MCDB

(Data received June 27, 2007)

P030 Aetna U.S. HealthCare

Provider Layout: **54,795 records read in; 54,795 records after deleting exact duplicates.**

	Field Name	MHCC Comments
1.	Servicing Practitioner ID	
2.	Practitioner Federal Tax ID	
3.	Practitioner Last Name	
4.	Practitioner First Name	
5.	Practitioner Middle Initial	
6.	Practitioner Name Suffix	
7.	Practitioner Credential	
8.	Practitioner Specialty – 1	
9.	Practitioner Specialty – 2	100% missing
10.	Practitioner Specialty – 3	100% missing
11.	Practitioner DEA #	
12.	Indicator for Multi-Practitioner Health Care Organization	
Percent of Exact Duplicate Records: 0%		
Percent of Records With No Match to Encounter: 0%		
Encounter Services File With Provider File Link: 4,806,672 services matched by FEDTAXID and PROVID, 953 services matched by PROVID only, 0 services with no match		

NOTE: Encounter, Pharmacy & Provider Directory statistics based on data reflected in a variety of analytical files and do not represent the data from original submission to SSS.

Notes on Financials:

1. Over 3% have all financial fields = 0
2. Over 8% have Billed Charged >0 and Allowed Amount = 0 and Reimbursement Amount = 0 and Patient Lib = 0.

Notes on Enrollment and Disenrollment Dates:

Description	Number of Records
Enrollment Date = Disenrollment Date	5,119 (<1%)
Service Begin Date BEFORE Enrollment Date	680 (<1%)
Service Begin Date AFTER Disenrollment Date	35,895 (<1%)

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2006 MCDB

(Data received June 27, 2007)

P070 American Republic Insurance Co.

Encounter Layout: 20470 records read in.

	Field Name	MHCC Comments
1.	Patient ID (encrypted)	
2.	Patient Date of Birth	
3.	Patient Sex	
4.	Consumer Directed Health Plan (CDHP) alone or w/ HAS or HRA indicator	
5.	Patient Zip Code	
6.	Patient Covered by Other Insurance	
7.	Coverage Type	
8.	Delivery System	
9.	Claim Related Condition	
10.	Practitioner Federal Tax ID	
11.	Participating Provider Flag	
12.	Type of Bill	
13.	Claim Control Number	
14.	Claim Paid Date	
15.	Number of Diagnosis Codes	
16.	Number of Line Items	
17.	Diagnosis Code 1	331 = Invalid (1.38 %)
18.	Diagnosis Code 2	11,610 = Missing (48.23%); 202 = Invalid (<1%);
19.	Diagnosis Code 3	16,980 = Missing (70.54%); 126 = Invalid (<1%);
20.	Diagnosis Code 4	20,200 = Missing (83.92%); 38 = Invalid (<1%);
21.	Diagnosis Code 5	23,929 = Missing (99.41%);
22.	Diagnosis Code 6	23,985 = Missing (99.65%);
23.	Diagnosis Code 7	24,007 = Missing (99.74%);
24.	Diagnosis Code 8	24,026 = Missing (99.82%);
25.	Diagnosis Code 9	Missing = (100%);

	Field Name	MHCC Comments
26.	Diagnosis Code 10	Missing = (100%);
27.	Service From Date	
28.	Service Thru Date	
29.	Filler	
30.	Place of Service	
31.	Service Location Zip Code	
32.	Service Unit Indicator	
33.	Units	
34.	Procedure Code	12.24% = Missing
35.	Modifier I	81% = Missing
36.	Modifier II	98% = Missing
37.	Servicing Practitioner ID	
38.	Billed Charge	1,160 (4.82%) with BC=0
39.	Allowed Amount	
40.	Reimbursement Amount	
41.	Patient Liability	
42.	Date of Enrollment	
43.	Date of Disenrollment	

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2006 MCDB

(Data received June 27, 2007)

P070 American Republic Insurance Co.

Provider Layout: 4,190 records read in; 4,189 records after deleting exact duplicates.

	Field Name	MHCC Comments
1.	Servicing Practitioner ID	
2.	Practitioner Federal Tax ID	
3.	Practitioner Last Name	
4.	Practitioner First Name	
5.	Practitioner Middle Initial	
6.	Practitioner Name Suffix	
7.	Practitioner Credential	
8.	Practitioner Specialty – 1	882 = Missing (19.62%)
9.	Practitioner Specialty – 2	100% = Missing
10.	Practitioner Specialty – 3	100% = Missing
11.	Practitioner DEA #	3,023 = Missing (72.17%)
12.	Indicator for Multi-Practitioner Health Care Organization	
Percent of Exact Duplicate Records: <1%		
Percent of Records With No Match to Encounter: 0%		
Encounter Services File With Provider File Link: 24,070 services matched by FEDTAXID and PROVID, 0 services matched by PROVID only, 0 services with no match		

NOTE: Encounter, Pharmacy & Provider Directory statistics based on data reflected in a variety of analytical files and do not represent the data from original submission to SSS.

Notes on Enrollment and Disenrollment Dates:

Description	Number of Records
Enrollment Date = Disenrollment Date	30 (<1%)
Service Begin Date BEFORE Enrollment Date	0 (<1%)
Service Begin Date AFTER Disenrollment Date	1381 (5.74%)

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2006 MCDB

(Data received June 27, 2007)

P130 CareFirst BlueChoice, Inc.

Encounter Layout: **18,774,307 read in.**

	Field Name	MHCC Comments
1.	Patient ID (encrypted)	Incomplete documentation provided for identification and matching of lab services and medical claims
2.	Patient Date of Birth	196 records (year= 00)/89 records (year = 1899)
3.	Patient Sex	
4.	Consumer Directed Health Plan (CDHP) alone or w/ HAS or HRA indicator	
5.	Patient Zip Code	
6.	Patient Covered by Other Insurance	12.63% = 9 (Not Coded)
7.	Coverage Type	
8.	Delivery System	
9.	Claim Related Condition	12.63%= 9 (Not Coded)
10.	Practitioner Federal Tax ID	
11.	Participating Provider Flag	26.09% = 3 (Not Coded)
12.	Type of Bill	
13.	Claim Control Number	
14.	Claim Paid Date	
15.	Number of Diagnosis Codes	
16.	Number of Line Items	
17.	Diagnosis Code 1	<1% Missing
18.	Diagnosis Code 2	75.31% Missing
19.	Diagnosis Code 3	88.21% Missing
20.	Diagnosis Code 4	96.60% Missing
21.	Diagnosis Code 5	100% Missing
22.	Diagnosis Code 6	100% Missing
23.	Diagnosis Code 7	100% Missing
24.	Diagnosis Code 8	100% Missing
25.	Diagnosis Code 9	100% Missing
26.	Diagnosis Code 10	100% Missing
27.	Service From Date	

	Field Name	MHCC Comments
28.	Service Thru Date	
29.	Filler	
30.	Place of Service	
31.	Service Location Zip Code	15.66% Missing
32.	Service Unit Indicator	
33.	Units	12.63% = 0
34.	Procedure Code	<1% Missing
35.	Modifier I	80% Missing
36.	Modifier II	80% Missing
37.	Servicing Practitioner ID	
38.	Billed Charge	<1% with BC = 0
39.	Allowed Amount	<1% with AA = 0
40.	Reimbursement Amount	<1% with RA = 0
41.	Patient Liability	
42.	Date of Enrollment	
43.	Date of Disenrollment	22.08% = '0000', 48.49% in (2008 – 2199)

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2006 MCDB

(Data received June 27, 2007)

P130 CareFirst BlueChoice, Inc.

Provider Layout: **97,749 records read in; 97,749 records after deleting exact duplicates.**
Payer provided one provider directory for both P130 and P131.

	Field Name	MHCC Comments
1.	Servicing Practitioner ID	
2.	Practitioner Federal Tax ID	
3.	Practitioner Last Name	
4.	Practitioner First Name	
5.	Practitioner Middle Initial	
6.	Practitioner Name Suffix	
7.	Practitioner Credential	
8.	Practitioner Specialty – 1	
9.	Practitioner Specialty – 2	94.43% Missing
10.	Practitioner Specialty – 3	100% Missing
11.	Practitioner DEA #	76.01% Missing
12.	Indicator for Multi-Practitioner Health Care Organization	
Percent of Exact Duplicate Records: 0%		
Percent of Records With No Match to Encounter: 46.55%		
Encounter Services File With Provider File Link: 15,826,746 services matched by FEDTAXID and PROVID, 11,792 services matched by PROVID only, 2,935,771 services with no match		

NOTE: 1) Payer submitted one Provider data file for both P130 and P131.
2) Encounter, Pharmacy & Provider Directory statistics based on data reflected in a variety of analytical files and do not represent the data from original submission to SSS.

Notes on Enrollment and Disenrollment Dates:

Description	Number of Records
Enrollment Date = Disenrollment Date	36 (0%)

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2006 MCDB

(Data received June 27, 2007)

P131 CareFirst of Maryland, Inc.

Encounter Layout: **23,547,560 records read in;**

	Field Name	MHCC Comments
1.	Patient ID (encrypted)	
2.	Patient Date of Birth	<1% = 0000, 1803 – 1899
3.	Patient Sex	
4.	Consumer Directed Health Plan (CDHP) alone or w/ HAS or HRA indicator	
5.	Patient Zip Code	<1% = Invalid
6.	Patient Covered by Other Insurance	1 record= 9 (not coded)
7.	Coverage Type	470 (<1%) = 9 (not coded)
8.	Delivery System	160 (<1%) = 9 (Not Coded)
9.	Claim Related Condition	
10.	Practitioner Federal Tax ID	226,951 (<1%) = Missing
11.	Participating Provider Flag	934,300 (3.97%) = 3 (Not Coded)
12.	Type of Bill	
13.	Claim Control Number	
14.	Claim Paid Date	
15.	Number of Diagnosis Codes	
16.	Number of Line Items	
17.	Diagnosis Code 1	1.15% = Missing or Invalid
18.	Diagnosis Code 2	About 64% = Missing or Invalid
19.	Diagnosis Code 3	About 90% = Missing or Invalid
20.	Diagnosis Code 4	About 95% = Missing or Invalid
21.	Diagnosis Code 5	100% Missing
22.	Diagnosis Code 6	100% Missing
23.	Diagnosis Code 7	100% Missing
24.	Diagnosis Code 8	100% Missing
25.	Diagnosis Code 9	100% Missing
26.	Diagnosis Code 10	100% Missing

	Field Name	MHCC Comments
27.	Service From Date	
28.	Service Thru Date	
29.	Filler	
30.	Place of Service	
31.	Service Location Zip Code	1,030,963 (4.38%) = Missing
32.	Service Unit Indicator	
33.	Units	739 (<1%) = 0
34.	Procedure Code	160,274 (1%) = Missing/Invalid
35.	Modifier I	72.88% Missing
36.	Modifier II	95.96% Missing
37.	Servicing Practitioner ID	394 (<1%) = Missing
38.	Billed Charge	
39.	Allowed Amount	
40.	Reimbursement Amount	
41.	Patient Liability	
42.	Date of Enrollment	
43	Date of Disenrollment	79.85% = 0000, 2.16% in (2008-2029)

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2006 MCDB

(Data received June 27, 2007)

P131 CareFirst of Maryland, Inc.

Provider Layout: **97,749 records read in; 97,749 records after deleting exact duplicates.**
Payer provided one provider directory for both P130 and P131.

	Field Name	MHCC Comments
1.	Servicing Practitioner ID	
2.	Practitioner Federal Tax ID	
3.	Practitioner Last Name	
4.	Practitioner First Name	
5.	Practitioner Middle Initial	
6.	Practitioner Name Suffix	
7.	Practitioner Credential	
8.	Practitioner Specialty – 1	
9.	Practitioner Specialty – 2	94.43% Missing
10.	Practitioner Specialty – 3	100% Missing
11.	Practitioner DEA #	76.01% Missing
12.	Indicator for Multi-Practitioner Health Care Organization	
Percent of Exact Duplicate Records: 0%		
Percent of Records With No Match to Encounter: 43.91%		
Encounter Services File With Provider File Link: 20,359,155 services matched by FEDTAXID and PROVID, 2,165,342 services matched by PROVID only, 1,023,063 services with no match		

NOTE: 1) Payer submitted one Provider data file for both P130 and P131.
2) Encounter, Pharmacy & Provider Directory statistics based on data reflected in a variety of analytical files and do not represent the data from original submission to SSS.

Notes on Enrollment and Disenrollment Dates:

Description	Number of Records
Enrollment Date = Disenrollment Date	485 (<1%)

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2006 MCDB

(Data received July 27, 2007)

P160 CIGNA Healthcare Mid-Atlantic, Inc.

Encounter Layout: **3,928,967 records read in.**

	Field Name	MHCC Comments
1.	Patient ID (encrypted)	
2.	Patient Date of Birth	
3.	Patient Sex	
4.	Consumer Directed Health Plan (CDHP) alone or w/ HAS or HRA indicator	
5.	Patient Zip Code	
6.	Patient Covered by Other Insurance	
7.	Coverage Type	
8.	Delivery System	
9.	Claim Related Condition	99.45% = 9 (Not coded)
10.	Practitioner Federal Tax ID	
11.	Participating Provider Flag	
12.	Type of Bill	
13.	Claim Control Number	
14.	Claim Paid Date	
15.	Number of Diagnosis Codes	
16.	Number of Line Items	
17.	Diagnosis Code 1	<1 Invalid
18.	Diagnosis Code 2	About 50% Missing or Invalid
19.	Diagnosis Code 3	About 73% Missing or Invalid
20.	Diagnosis Code 4	About 99% Missing or Invalid
21.	Diagnosis Code 5	100% Missing
22.	Diagnosis Code 6	100% Missing
23.	Diagnosis Code 7	100% Missing
24.	Diagnosis Code 8	100% Missing
25.	Diagnosis Code 9	100% Missing
26.	Diagnosis Code 10	100% Missing

	Field Name	MHCC Comments
27.	Service From Date	
28.	Service Thru Date	
29.	Filler	
30.	Place of Service	<1% Invalid
31.	Service Location Zip Code	<1% Missing/Invalid
32.	Service Unit Indicator	
33.	Units	1224 (<1%) = Missing
34.	Procedure Code	1615 (<1%) = Missing
35.	Modifier I	83.68% Missing
36.	Modifier II	99.69% Missing
37.	Servicing Practitioner ID	
38.	Billed Charge	Have data <0 for Bill type = 1
39.	Allowed Amount	Have data <0 for Bill type = 1
40.	Reimbursement Amount	Have data <0 for Bill type = 1
41.	Patient Liability	Have data <0 for Bill type = 1
42.	Date of Enrollment	
43.	Date of Disenrollment	

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2006 MCDB

(Data received July 27, 2007)

P160 CIGNA Healthcare Mid-Atlantic, Inc.

Provider Layout: **67,131 records read in; 67,131 records after deleting exact duplicates.**

	Field Name	MHCC Comments
1.	Servicing Practitioner ID	
2.	Practitioner Federal Tax ID	
3.	Practitioner Last Name	
4.	Practitioner First Name	
5.	Practitioner Middle Initial	
6.	Practitioner Name Suffix	
7.	Practitioner Credential	
8.	Practitioner Specialty – 1	
9.	Practitioner Specialty – 2	79.22% Missing
10.	Practitioner Specialty – 3	95.91% Missing
11.	Practitioner DEA #	43% Missing
12.	Indicator for Multi-Practitioner Health Care Organization	
Percent of Exact Duplicate Records: 0%		
Percent of Records With No Match to Encounter: 0%		
Encounter Services File With Provider File Link: 3,912,010 services matched by FEDTAXID and PROVID, 17,043 services matched by PROVID only, 0 services with no match		

NOTE: Encounter, Pharmacy & Provider Directory statistics based on data reflected in a variety of analytical files and do not represent the data from original submission to SSS.

Notes on Enrollment and Disenrollment Dates:

Description	Number of Records
Enrollment Date = Disenrollment Date	N/A
Service Begin Date BEFORE Enrollment Date	N/A
Service Begin Date AFTER Disenrollment Date	N/A

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2006 MCDB

(Data received June 25, 2007)

P280 Fortis Insurance Co.

Encounter Layout: **72,393 records read in.**

	Field Name	MHCC Comments
1.	Patient ID (encrypted)	
2.	Patient Date of Birth	
3.	Patient Sex	1.37% = 3 (Unknown)
4.	Consumer Directed Health Plan (CDHP) alone or w/ HAS or HRA indicator	
5.	Patient Zip Code	
6.	Patient Covered by Other Insurance	
7.	Coverage Type	
8.	Delivery System	
9.	Claim Related Condition	
10.	Practitioner Federal Tax ID	
11.	Participating Provider Flag	
12.	Type of Bill	
13.	Claim Control Number	
14.	Claim Paid Date	
15.	Number of Diagnosis Codes	
16.	Number of Line Items	
17.	Diagnosis Code 1	8.88% Invalid
18.	Diagnosis Code 2	About 42% Missing and Invalid
19.	Diagnosis Code 3	About 71% Missing and Invalid
20.	Diagnosis Code 4	About 85% Missing and Invalid
21.	Diagnosis Code 5	100% Missing
22.	Diagnosis Code 6	100% Missing
23.	Diagnosis Code 7	100% Missing
24.	Diagnosis Code 8	100% Missing
25.	Diagnosis Code 9	100% Missing
26.	Diagnosis Code 10	100% Missing

	Field Name	MHCC Comments
27.	Service From Date	
28.	Service Thru Date	
29.	Filler	
30.	Place of Service	<1% Invalid
31.	Service Location Zip Code	
32.	Service Unit Indicator	
33.	Units	7,916 (11%) = 0
34.	Procedure Code	6,142 (8.48%) = Missing/Invalid
35.	Modifier I	86% Missing
36.	Modifier II	100% Missing
37.	Servicing Practitioner ID	
38.	Billed Charge	
39.	Allowed Amount	
40.	Reimbursement Amount	
41.	Patient Liability	
42.	Date of Enrollment	
43.	Date of Disenrollment	

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2006 MCDB

(Data received June 25, 2007)

P280 Fortis Insurance Co.

Provider Layout: 7,537 records read in; 7,537 records after deleting exact duplicates.

	Field Name	MHCC Comments
1.	Servicing Practitioner ID	
2.	Practitioner Federal Tax ID	
3.	Practitioner Last Name	
4.	Practitioner First Name	
5.	Practitioner Middle Initial	
6.	Practitioner Name Suffix	
7.	Practitioner Credential	
8.	Practitioner Specialty – 1	
9.	Practitioner Specialty – 2	
10.	Practitioner Specialty – 3	
11.	Practitioner DEA #	100% Missing
12.	Indicator for Multi-Practitioner Health Care Organization	100% Missing
Percent of Exact Duplicate Records: 0%		
Percent of Records With No Match to Encounter: 0%		
Encounter Services File With Provider File Link:		72,383 services matched by FEDTAXID and PROVID, 10 services matched by PROVID only, 0 services with no match

NOTE: Encounter, Pharmacy & Provider Directory statistics based on data reflected in a variety of analytical files and do not represent the data from original submission to SSS.

Notes on Enrollment and Disenrollment Dates:

Description	Number of Records
Enrollment Date = Disenrollment Date	150 (<1%)
Service Begin Date BEFORE Enrollment Date	322 (<1%)
Service Begin Date AFTER Disenrollment Date	1127 (1.56%)

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2006 MCDB

(Data received June 28, 2007)

P320 Golden Rule Insurance Co.

Encounter Layout: **146,313 records read in.**

	Field Name	MHCC Comments
1.	Patient ID (encrypted)	
2.	Patient Date of Birth	
3.	Patient Sex	
4.	Consumer Directed Health Plan (CDHP) alone or w/ HAS or HRA indicator	
5.	Patient Zip Code	
6.	Patient Covered by Other Insurance	
7.	Coverage Type	6,830 (4.67%) = 9 (Unknown)
8.	Delivery System	
9.	Claim Related Condition	100% = 9 (not coded)
10.	Practitioner Federal Tax ID	
11.	Participating Provider Flag	
12.	Type of Bill	
13.	Claim Control Number	
14.	Claim Paid Date	
15.	Number of Diagnosis Codes	
16.	Number of Line Items	
17.	Diagnosis Code 1	20.74% Invalid
18.	Diagnosis Code 2	100% Missing
19.	Diagnosis Code 3	100% Missing
20.	Diagnosis Code 4	100% Missing
21.	Diagnosis Code 5	100% Missing
22.	Diagnosis Code 6	100% Missing
23.	Diagnosis Code 7	100% Missing
24.	Diagnosis Code 8	100% Missing
25.	Diagnosis Code 9	100% Missing
26.	Diagnosis Code 10	100% Missing

	Field Name	MHCC Comments
27.	Service From Date	
28.	Service Thru Date	
29.	Filler	
30.	Place of Service	
31.	Service Location Zip Code	<1% = Invalid
32.	Service Unit Indicator	100% = 9 (Not coded)
33.	Units	
34.	Procedure Code	
35.	Modifier I	
36.	Modifier II	100% Missing
37.	Servicing Practitioner ID	
38.	Billed Charge	
39.	Allowed Amount	
40.	Reimbursement Amount	
41.	Patient Liability	
42.	Date of Enrollment	
43.	Date of Disenrollment	

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2006 MCDB

(Data received June 28, 2007)

P320 Golden Rule Insurance Co.

Provider Layout: **6,462 records read in; 6,462 records after deleting exact duplicates.**

	Field Name	MHCC Comments
1.	Servicing Practitioner ID	
2.	Practitioner Federal Tax ID	
3.	Practitioner Last Name	
4.	Practitioner First Name	
5.	Practitioner Middle Initial	100% Missing
6.	Practitioner Name Suffix	100% Missing
7.	Practitioner Credential	100% Missing
8.	Practitioner Specialty – 1	
9.	Practitioner Specialty – 2	100% missing
10.	Practitioner Specialty – 3	100% missing
11.	Practitioner DEA #	100% Missing
12.	Indicator for Multi-Practitioner Health Care Organization	
Percent of Exact Duplicate Records: 0%		
Percent of Records With No Match to Encounter: 0%		
Encounter Services File With Provider File Link:	146,313 services matched by FEDTAXID and PROVID, 0 services matched by PROVID only, 0 services with no match	

NOTE: Encounter, Pharmacy & Provider Directory statistics based on data reflected in a variety of analytical files and do not represent the data from original submission to SSS.

Notes on Enrollment and Disenrollment Dates:

Description	Number of Records
Enrollment Date = Disenrollment Date	1,361 (<1%)
Service Begin Date BEFORE Enrollment Date	248 (<1%)
Service Begin Date AFTER Disenrollment Date	2,189 (1.5%)

MARYLAND HEALTH CARE COMMISSION Quality Review Statement for 2006 MCDB

(Last Submission Received August 15, 2007)

P325 Graphic Arts Benefit Corporation

Encounter Layout: 123,837 records read in.

	Field Name	MHCC Comments
1.	Patient ID (encrypted)	
2.	Patient Date of Birth	
3.	Patient Sex	
4.	Consumer Directed Health Plan (CDHP) alone or w/ HAS or HRA indicator	
5.	Patient Zip Code	<1% Mssing
6.	Patient Covered by Other Insurance	
7.	Coverage Type	
8.	Delivery System	
9.	Claim Related Condition	
10.	Practitioner Federal Tax ID	
11.	Participating Provider Flag	
12.	Type of Bill	
13.	Claim Control Number	
14.	Claim Paid Date	
15.	Number of Diagnosis Codes	
16.	Number of Line Items	
17.	Diagnosis Code 1	82.40% = Invalid
18.	Diagnosis Code 2	100% = Missing
19.	Diagnosis Code 3	100% = Missing
20.	Diagnosis Code 4	100% = Missing
21.	Diagnosis Code 5	100% = Missing
22.	Diagnosis Code 6	100% = Missing
23.	Diagnosis Code 7	100% = Missing
24.	Diagnosis Code 8	100% = Missing
25.	Diagnosis Code 9	100% = Missing
26.	Diagnosis Code 10	100% = Missing
27.	Service From Date	8.86% = Invalid (1908-2005)

	Field Name	MHCC Comments
28.	Service Thru Date	98.33% = Missing/Invalid [Per payer, set to equal Service From Date]
29.	Filler	
30.	Place of Service	
31.	Service Location Zip Code	8.43% Missing/and Invalid
32.	Service Unit Indicator	
33.	Units	
34.	Procedure Code	1.79% /Missing/Invalid
35.	Modifier I	100% Missing [Payer does not use this data in claim processing]
36.	Modifier II	100% Missing [Payer does not use this data in claim processing]
37.	Servicing Practitioner ID	100% Missing
38.	Billed Charge	
39.	Allowed Amount	
40.	Reimbursement Amount	
41.	Patient Liability	
42.	Date of Enrollment	
43.	Date of Disenrollment	

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2006 MCDB

(Last Submission Received August 15, 2007)

P325 Graphic Arts Benefit Corporation

Provider Layout: **4,331 records read in; 4,331 records after deleting exact duplicates.**

	Field Name	MHCC Comments
1.	Servicing Practitioner ID	100% Missing
2.	Practitioner Federal Tax ID	
3.	Practitioner Last Name	
4.	Practitioner First Name	
5.	Practitioner Middle Initial	
6.	Practitioner Name Suffix	
7.	Practitioner Credential	
8.	Practitioner Specialty – 1	100% missing
9.	Practitioner Specialty – 2	100% missing
10.	Practitioner Specialty – 3	100% missing
11.	Practitioner DEA #	100% missing
12.	Indicator for Multi-Practitioner Health Care Organization	70% missing
Percent of Exact Duplicate Records: 0%		
Percent of Records With No Match to Encounter: 100%		
Encounter Services File With Provider File Link:		0 services matched by FEDTAXID and PROVID, 0 services matched by PROVID only, 123,837 services with no match

NOTE: Encounter, Pharmacy & Provider Directory statistics based on data reflected in a variety of analytical files and do not represent the data from original submission to SSS.

Notes on Enrollment and Disenrollment Dates:

Description	Number of Records
Enrollment Date = Disenrollment Date	N/A
Service Begin Date BEFORE Enrollment Date	2,889 (2.33%)
Service Begin Date AFTER Disenrollment Date	1,337 (1.08%)

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2006 MCDB

(Last Submission Received October 19, 2007)

P330 Great-West Life and Annuity Insurance Co.

Encounter Layout: **462,464 records read in.**

	Field Name	MHCC Comments
1.	Patient ID (encrypted)	
2.	Patient Date of Birth	
3.	Patient Sex	239 (<1%) = 3 (Unknown)
4.	Consumer Directed Health Plan (CDHP) alone or w/ HAS or HRA indicator	
5.	Patient Zip Code	<1% = Invalid
6.	Patient Covered by Other Insurance	3,011 (<1%) = 9 (Not coded)
7.	Coverage Type	2,984 (<1%) = 9 (Not coded)
8.	Delivery System	3,005 (<1%) = 9 (Not coded)
9.	Claim Related Condition	3,044 (<1%) = 9 (Not coded)
10.	Practitioner Federal Tax ID	4,982 (1.08%) = Missing
11.	Participating Provider Flag	<1% = Invalid
12.	Type of Bill	
13.	Claim Control Number	
14.	Claim Paid Date	
15.	Number of Diagnosis Codes	
16.	Number of Line Items	
17.	Diagnosis Code 1	1,757 (0.4%) = Missing; 55,847 (12.08%) = Invalid
18.	Diagnosis Code 2	100% Missing and/or Invalid
19.	Diagnosis Code 3	100% Missing and/or Invalid
20.	Diagnosis Code 4	100% Missing
21.	Diagnosis Code 5	100% Missing
22.	Diagnosis Code 6	100% Missing
23.	Diagnosis Code 7	100% Missing
24.	Diagnosis Code 8	100% Missing
25.	Diagnosis Code 9	100% Missing
26.	Diagnosis Code 10	100% Missing
27.	Service From Date	

	Field Name	MHCC Comments
28.	Service Thru Date	
29.	Filler	
30.	Place of Service	25,781 (5.57%) = 00 (Invalid)
31.	Service Location Zip Code	3,073 (<1%) = Missing/Invalid
32.	Service Unit Indicator	86.44% (399,740) = 9 (Not Coded)
33.	Units	<1% = 0 or less
34.	Procedure Code	3,264 (<1%) = Missing
35.	Modifier I	95.52% = Missing
36.	Modifier II	99.89% = Missing
37.	Servicing Practitioner ID	
38.	Billed Charge	See Summary and Data Completeness Report for problems with all the financial fields.
39.	Allowed Amount	
40.	Reimbursement Amount	
41.	Patient Liability	
42.	Date of Enrollment	
43.	Date of Disenrollment	

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2006 MCDB

(Last Submission Received October 19, 2007)

P330 Great-West Life and Annuity Insurance Co.

Provider Layout: **409,277 records read in; 22,157 records after deleting exact duplicates.**

	Field Name	MHCC Comments
1.	Servicing Practitioner ID	
2.	Practitioner Federal Tax ID	
3.	Practitioner Last Name	
4.	Practitioner First Name	
5.	Practitioner Middle Initial	
6.	Practitioner Name Suffix	
7.	Practitioner Credential	
8.	Practitioner Specialty – 1	5,278 (23.82%) = Missing
9.	Practitioner Specialty – 2	20,724 (93.53%) = Missing
10.	Practitioner Specialty – 3	22,040 (99.47%) = Missing
11.	Practitioner DEA #	7,866 (35.5%) = Missing
12.	Indicator for Multi-Practitioner Health Care Organization	15,912 (71.81%) = Missing
Percent of Exact Duplicate Records: 94.59%		
Percent of Records With No Match to Encounter: <1%		
Encounter Services File With Provider File Link:		59,416 services matched by FEDTAXID and PROVID, 403,060 services matched by PROVID only, 21 services with no match

NOTE: Encounter, Pharmacy & Provider Directory statistics based on data reflected in a variety of analytical files and do not represent the data from original submission to SSS.

Notes on Enrollment and Disenrollment Dates:

Description	Number of Records
Enrollment Date = Disenrollment Date	100 (<1%)
Service Begin Date BEFORE Enrollment Date	42,160 (9.12%)
Service Begin Date AFTER Disenrollment Date	754 (<1%)

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2006 MCDB

(Data received August 31, 2007)

P350 Guardian Life Insurance Co. of America

Encounter Layout: **346,828 records read in.**

	Field Name	MHCC Comments
1.	Patient ID (encrypted)	
2.	Patient Date of Birth	
3.	Patient Sex	
4.	Consumer Directed Health Plan (CDHP) alone or w/ HAS or HRA indicator	
5.	Patient Zip Code	<1% = Invalid
6.	Patient Covered by Other Insurance	
7.	Coverage Type	
8.	Delivery System	3,049 (<1%) = 9 (Not Coded)
9.	Claim Related Condition	12,047 (3.47%) = 9 (Not Coded)
10.	Practitioner Federal Tax ID	11 (<1%) = Missing
11.	Participating Provider Flag	
12.	Type of Bill	
13.	Claim Control Number	
14.	Claim Paid Date	
15.	Number of Diagnosis Codes	
16.	Number of Line Items	
17.	Diagnosis Code 1	9,861 (2.84%) = Invalid
18.	Diagnosis Code 2	181,784 (52.41%) = Missing; 1156 (<1%) = Invalid
19.	Diagnosis Code 3	267,268 (77.06%) = Missing; 417 (<1%) = Invalid
20.	Diagnosis Code 4	100% = Missing
21.	Diagnosis Code 5	100% = Missing
22.	Diagnosis Code 6	100% = Missing
23.	Diagnosis Code 7	100% = Missing
24.	Diagnosis Code 8	100% = Missing
25.	Diagnosis Code 9	100% = Missing
26.	Diagnosis Code 10	100% = Missing

	Field Name	MHCC Comments
27.	Service From Date	
28.	Service Thru Date	
29.	Filler	
30.	Place of Service	
31.	Service Location Zip Code	<1% = Missing/Invalid
32.	Service Unit Indicator	
33.	Units	
34.	Procedure Code	6 (<1%) = Missing
35.	Modifier I	85.46% = Missing
36.	Modifier II	100% = Missing
37.	Servicing Practitioner ID	
38.	Billed Charge	
39.	Allowed Amount	
40.	Reimbursement Amount	
41.	Patient Liability	
42.	Date of Enrollment	1,701 (<1%) = Missing
43.	Date of Disenrollment	

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2006 MCDB

(Data received August 31, 2007)

P350 Guardian Life Insurance Co. of America

Provider Layout: **22,162 records read in; 22,162 records after deleting exact duplicates.**

	Field Name	MHCC Comments
1.	Servicing Practitioner ID	1 record = Missing
2.	Practitioner Federal Tax ID	29 records = Missing
3.	Practitioner Last Name	
4.	Practitioner First Name	
5.	Practitioner Middle Initial	
6.	Practitioner Name Suffix	
7.	Practitioner Credential	
8.	Practitioner Specialty – 1	
9.	Practitioner Specialty – 2	2247 (10.14%) = Missing
10.	Practitioner Specialty – 3	100% = Missing
11.	Practitioner DEA #	19,935 (90%) = Missing
12.	Indicator for Multi-Practitioner Health Care Organization	9,976 (45%) = Missing
Percent of Exact Duplicate Records: 0%		
Percent of Records With No Match to Encounter: 10%		
Encounter Services File With Provider File Link: 346,729 services by FEDTAXID and PROVID, 6 services by PROVID only, 93 services no match		

NOTE: Encounter, Pharmacy & Provider Directory statistics based on data reflected in a variety of analytical files and do not represent the data from original submission to SSS.

Notes on Enrollment and Disenrollment Dates:

Description	Number of Records
Enrollment Date = Missing	1,701 (<1%)
Enrollment Date = Disenrollment Date	2 (<1%)
Service Begin Date AFTER Disenrollment Date	20 (<1%)

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2006 MCDB

(Last Submission Received September 12, 2007)

P471 Unicare Life & Health Insurance Co.

Encounter Layout: 155,774 records read in.

	Field Name	MHCC Comments
1.	Patient ID (encrypted)	
2.	Patient Date of Birth	
3.	Patient Sex	
4.	Consumer Directed Health Plan (CDHP) alone or w/ HAS or HRA indicator	
5.	Patient Zip Code	<1% = Invalid
6.	Patient Covered by Other Insurance	140,926 (90.47%) = 9 (Unknown)
7.	Coverage Type	
8.	Delivery System	
9.	Claim Related Condition	140,924 (90.47%) = Missing; 14,850 (9.53%) = 9
10.	Practitioner Federal Tax ID	<1% = Invalid
11.	Participating Provider Flag	
12.	Type of Bill	
13.	Claim Control Number	
14.	Claim Paid Date	
15.	Number of Diagnosis Codes	
16.	Number of Line Items	
17.	Diagnosis Code 1	550 (<1%) Invalid
18.	Diagnosis Code 2	100% Missing
19.	Diagnosis Code 3	100% Missing
20.	Diagnosis Code 4	100% Missing
21.	Diagnosis Code 5	100% Missing
22.	Diagnosis Code 6	100% Missing
23.	Diagnosis Code 7	100% Missing
24.	Diagnosis Code 8	100% Missing
25.	Diagnosis Code 9	100% Missing
26.	Diagnosis Code 10	100% Missing

	Field Name	MHCC Comments
27.	Service From Date	
28.	Service Thru Date	
29.	Filler	
30.	Place of Service	
31.	Service Location Zip Code	<1% = Missing/Invalid
32.	Service Unit Indicator	14,850 (9.53%) = 9
33.	Units	<1% = 0 or less
34.	Procedure Code	565 (<1%) = Missing, 8469 (5.44%) = Invalid
35.	Modifier I	5.43% = Missing
36.	Modifier II	100% = Missing
37.	Servicing Practitioner ID	32,751 (21.02%) = Missing
38.	Billed Charge	
39.	Allowed Amount	
40.	Reimbursement Amount	
41.	Patient Liability	
42.	Date of Enrollment	
43.	Date of Disenrollment	

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2006 MCDB

(Last Submission Received September 12, 2007)

P471 Unicare Life & Health Insurance Co.

Provider Layout: **19,288 records read in; 19,258 records after deleting exact duplicates.**

	Field Name	MHCC Comments
1.	Servicing Practitioner ID	16.20% (3,125 records) missing
2.	Practitioner Federal Tax ID	
3.	Practitioner Last Name	
4.	Practitioner First Name	
5.	Practitioner Middle Initial	100% Missing
6.	Practitioner Name Suffix	100% Missing
7.	Practitioner Credential	100% Missing
8.	Practitioner Specialty – 1	5.43% = Missing/Invalid
9.	Practitioner Specialty – 2	97.64% = Missing/Invalid
10.	Practitioner Specialty – 3	99.29% = Missing/Invalid
11.	Practitioner DEA #	100% Missing
12.	Indicator for Multi-Practitioner Health Care Organization	100% Missing
Percent of Exact Duplicate Records: <1%		
Percent of Records With No Match to Encounter: 9.2%		
Encounter Services File With Provider File Link: 122,778 services by FEDTAXID and PROVID, 3 services by PROVID only, 32,993 services no match		

NOTE: Encounter, Pharmacy & Provider Directory statistics based on data reflected in a variety of analytical files and do not represent the data from original submission to SSS.

Notes on Financials:

There are 0 and <0 in all financial fields.

Notes on Enrollment and Disenrollment Dates:

Description	Number of Records
Enrollment Date = Missing	161 (<1%)
Service Begin Date BEFORE Enrollment Date	82 (<1%)
Service Begin Date AFTER Disenrollment Date	234 (<1%)

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2006 MCDB

(Last submission received November 09, 2007)

P480 Kaiser Foundation Health Plan of Mid-Atlantic

Encounter Layout: **5,087,116records read in.**

	Field Name	MHCC Comments
1.	Patient ID (encrypted)	
2.	Patient Date of Birth	4 = Year 1873
3.	Patient Sex	
4.	Consumer Directed Health Plan (CDHP) alone or w/ HAS or HRA indicator	
5.	Patient Zip Code	(<1%) = Invalid
6.	Patient Covered by Other Insurance	34,387 (<1%) = 9 (Unknown)
7.	Coverage Type	1.32% (67,330) = 9 (Unknown)
8.	Delivery System	30,173 (<1% = 9 (Unknown)
9.	Claim Related Condition	
10.	Practitioner Federal Tax ID	<1% = Missing
11.	Participating Provider Flag	
12.	Type of Bill	
13.	Claim Control Number	
14.	Claim Paid Date	
15.	Number of Diagnosis Codes	
16.	Number of Line Items	
17.	Diagnosis Code 1	71% Missing/Invalid
18.	Diagnosis Code 2	84% Missing/Invalid
19.	Diagnosis Code 3	91% Missing/Invalid
20.	Diagnosis Code 4	95% Missing/Invalid
21.	Diagnosis Code 5	98% Missing/Invalid
22.	Diagnosis Code 6	99% Missing/Invalid
23.	Diagnosis Code 7	99% Missing/Invalid
24.	Diagnosis Code 8	99% Missing/Invalid
25.	Diagnosis Code 9	99% Missing/Invalid
26.	Diagnosis Code 10	99% Missing/Invalid

	Field Name	MHCC Comments
27.	Service From Date	
28.	Service Thru Date	
29.	Filler	
30.	Place of Service	<1% = Invalid
31.	Service Location Zip Code	13.73% = Missing/Invalid
32.	Service Unit Indicator	<1% = 0
33.	Units	<1% = 0 or less
34.	Procedure Code	
35.	Modifier I	4,743,332 (93.24%) = Missing
36.	Modifier II	100% = Missing
37.	Servicing Practitioner ID	41,313 (<1%) = Missing
38.	Billed Charge	42,648 (<1%) with BC= 0
39.	Allowed Amount	
40.	Reimbursement Amount	
41.	Patient Liability	
42.	Date of Enrollment	
43.	Date of Disenrollment	

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2006 MCDB

(Last submission received November 09, 2007)

P480 Kaiser Foundation Health Plan of Mid-Atlantic

Provider Layout: 19,174 records read in; 19,174 records after deleting exact duplicates.

	Field Name	MHCC Comments
1.	Servicing Practitioner ID	
2.	Practitioner Federal Tax ID	<1% = Missing
3.	Practitioner Last Name	
4.	Practitioner First Name	
5.	Practitioner Middle Initial	
6.	Practitioner Name Suffix	
7.	Practitioner Credential	
8.	Practitioner Specialty – 1	
9.	Practitioner Specialty – 2	100% = Invalid
10.	Practitioner Specialty – 3	100% = Invalid
11.	Practitioner DEA #	15,122 (78.87%) = Missing
12.	Indicator for Multi-Practitioner Health Care Organization	
Percent of Exact Duplicate Records: 0%		
Percent of Records With No Match to Encounter: <1%		
Encounter Services File With Provider File Link: 4,916,490 services by FEDTAXID and PROVID, 102,624 services by PROVID only, 68,002 services no match		

NOTE: Encounter, Pharmacy & Provider Directory statistics based on data reflected in a variety of analytical files and do not represent the data from original submission to SSS.

Notes on Enrollment and Disenrollment Dates:

Description	Number of Records
Enrollment Date = Missing	147,202 (2.89%)

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2006 MCDB

(Data received June 21, 2007)

P500 Mamsi Life and Health Insurance Co.

Encounter Layout: **2,127,195 records read in.**

	Field Name	MHCC Comments
1.	Patient ID (encrypted)	
2.	Patient Date of Birth	
3.	Patient Sex	
4.	Consumer Directed Health Plan (CDHP) alone or w/ HAS or HRA indicator	
5.	Patient Zip Code	
6.	Patient Covered by Other Insurance	
7.	Coverage Type	
8.	Delivery System	
9.	Claim Related Condition	
10.	Practitioner Federal Tax ID	5,143 (<1%) = Missing
11.	Participating Provider Flag	
12.	Type of Bill	
13.	Claim Control Number	
14.	Claim Paid Date	
15.	Number of Diagnosis Codes	
16.	Number of Line Items	
17.	Diagnosis Code 1	<1% Missing/Invalid
18.	Diagnosis Code 2	45% = Missing/Invalid
19.	Diagnosis Code 3	69% = Missing/Invalid
20.	Diagnosis Code 4	82% = Missing/Invalid
21.	Diagnosis Code 5	95% = Missing/Invalid
22.	Diagnosis Code 6	96% = Missing/Invalid
23.	Diagnosis Code 7	97% = Missing/Invalid
24.	Diagnosis Code 8	97% = Missing/Invalid
25.	Diagnosis Code 9	98% = Missing/Invalid

	Field Name	MHCC Comments
26.	Diagnosis Code 10	100% = Missing/Invalid
27.	Service From Date	
28.	Service Thru Date	
29.	Filler	
30.	Place of Service	
31.	Service Location Zip Code	<1% = Invalid
32.	Service Unit Indicator	
33.	Units	<1% = 0
34.	Procedure Code	
35.	Modifier I	83% = Missing
36.	Modifier II	100% = Missing
37.	Servicing Practitioner ID	
38.	Billed Charge	22,134 (1.04%) BC = 0 but AA and/or RA >0
39.	Allowed Amount	
40.	Reimbursement Amount	
41.	Patient Liability	
42.	Date of Enrollment	
43.	Date of Disenrollment	

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2006 MCDB

(Data received June 21, 2007)

P500 Mamsi Life and Health Insurance Co.

Provider Layout: **25,263 records read in; 25,263 records after deleting exact duplicates.**

	Field Name	MHCC Comments
1.	Servicing Practitioner ID	
2.	Practitioner Federal Tax ID	
3.	Practitioner Last Name	
4.	Practitioner First Name	
5.	Practitioner Middle Initial	
6.	Practitioner Name Suffix	
7.	Practitioner Credential	
8.	Practitioner Specialty – 1	
9.	Practitioner Specialty – 2	100% = Missing
10.	Practitioner Specialty – 3	100% = Missing
11.	Practitioner DEA #	7,593 record (30.06%) = Missing
12.	Indicator for Multi-Practitioner Health Care Organization	
Percent of Exact Duplicate Records: 0%		
Percent of Records With No Match to Encounter: 0%		
Encounter Services File With Provider File Link: 2,122,052 services by FEDTAXID and PROVID, 5,143 services by PROVID only, 0 services no match		

NOTE: Encounter, Pharmacy & Provider Directory statistics based on data reflected in a variety of analytical files and do not represent the data from original submission to SSS.

Notes on Enrollment and Disenrollment Dates:

Description	Number of Records
Enrollment Date = Disenrollment Date	14 (<1%)
Service Begin Date BEFORE Enrollment Date	2 (<1%)
Service Begin Date AFTER Disenrollment Date	831 (<1%)

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2006 MCDB

(Data received June 26, 2007)

P510 Maryland Fidelity Insurance Co.

Encounter Layout: **46,186 records read in.**

	Field Name	MHCC Comments
1.	Patient ID (encrypted)	
2.	Patient Date of Birth	
3.	Patient Sex	49 (<1%) = 3 (Unknown)
4.	Consumer Directed Health Plan (CDHP) alone or w/ HAS or HRA indicator	100% = No
5.	Patient Zip Code	
6.	Patient Covered by Other Insurance	
7.	Coverage Type	
8.	Delivery System	
9.	Claim Related Condition	136 (<1%) = 9 (Unknown)
10.	Practitioner Federal Tax ID	
11.	Participating Provider Flag	
12.	Type of Bill	
13.	Claim Control Number	
14.	Claim Paid Date	
15.	Number of Diagnosis Codes	
16.	Number of Line Items	
17.	Diagnosis Code 1	2% = Missing/Invalid
18.	Diagnosis Code 2	61% = Missing/Invalid
19.	Diagnosis Code 3	82% = Missing/Invalid
20.	Diagnosis Code 4	91% = Missing/Invalid
21.	Diagnosis Code 5	Almost 100% = Missing
22.	Diagnosis Code 6	100% = Missing
23.	Diagnosis Code 7	100% = Missing
24.	Diagnosis Code 8	100% = Missing
25.	Diagnosis Code 9	100% = Missing
26.	Diagnosis Code 10	100% = Missing

	Field Name	MHCC Comments
27.	Service From Date	
28.	Service Thru Date	
29.	Filler	
30.	Place of Service	210 (<1%) = Missing
31.	Service Location Zip Code	<1% = Missing/Invalid
32.	Service Unit Indicator	5% = Missing/Invalid
33.	Units	<1% = 0
34.	Procedure Code	
35.	Modifier I	86% = Missing
36.	Modifier II	100% = Missing
37.	Servicing Practitioner ID	
38.	Billed Charge	
39.	Allowed Amount	
40.	Reimbursement Amount	
41.	Patient Liability	
42.	Date of Enrollment	<1% = 'NULL'
43.	Date of Disenrollment	

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2006 MCDB

(Data received June 26, 2007)

P510 Maryland Fidelity Insurance Co.

Provider Layout: **5,583 records read in; 5,583 records after deleting exact duplicates.**

	Field Name	MHCC Comments
1.	Servicing Practitioner ID	
2.	Practitioner Federal Tax ID	
3.	Practitioner Last Name	
4.	Practitioner First Name	
5.	Practitioner Middle Initial	
6.	Practitioner Name Suffix	
7.	Practitioner Credential	
8.	Practitioner Specialty – 1	
9.	Practitioner Specialty – 2	5,311 (95.13%) = Missing
10.	Practitioner Specialty – 3	5,562 (99.62%) = Missing
11.	Practitioner DEA #	2,639 (47.27%) = Missing
12.	Indicator for Multi-Practitioner Health Care Organization	
Percent of Exact Duplicate Records: 0%		
Percent of Records With No Match to Encounter: 4.03%		
Encounter Services File With Provider File Link: 45,520 services by FEDTAXID and PROVID, 0 services by PROVID only, 666 services no match		

NOTE: Encounter, Pharmacy & Provider Directory statistics based on data reflected in a variety of analytical files and do not represent the data from original submission to SSS.

Notes on Enrollment and Disenrollment Dates:

Description	Number of Records
Enrollment Date = Missing	44 (<1%)
Enrollment Date = Disenrollment Date	58 (<1%)
Service Begin Date BEFORE Enrollment Date	55 (<1%)
Service Begin Date AFTER Disenrollment Date	19,109 (41.37%)

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2004 MCDB

(Data received June 21, 2007)

P520 MD-Individual Practice Association, Inc.

Encounter Layout: **3,857,438 records read in.**

	Field Name	MHCC Comments
1.	Patient ID (encrypted)	
2.	Patient Date of Birth	
3.	Patient Sex	
4.	Consumer Directed Health Plan (CDHP) alone or w/ HAS or HRA indicator	
5.	Patient Zip Code	<1% = Invalid
6.	Patient Covered by Other Insurance	
7.	Coverage Type	
8.	Delivery System	
9.	Claim Related Condition	
10.	Practitioner Federal Tax ID	766 (<1%) = Missing
11.	Participating Provider Flag	
12.	Type of Bill	
13.	Claim Control Number	
14.	Claim Paid Date	
15.	Number of Diagnosis Codes	
16.	Number of Line Items	
17.	Diagnosis Code 1	<1% = Missing/Invalid
18.	Diagnosis Code 2	51% = Missing/Invalid
19.	Diagnosis Code 3	73% = Missing/Invalid
20.	Diagnosis Code 4	85% = Missing/Invalid
21.	Diagnosis Code 5	95% = Missing/Invalid
22.	Diagnosis Code 6	96% = Missing/Invalid
23.	Diagnosis Code 7	97% = Missing/Invalid
24.	Diagnosis Code 8	98% = Missing/Invalid
25.	Diagnosis Code 9	98% = Missing/Invalid
26.	Diagnosis Code 10	Almost 100% = Missing/Invalid

	Field Name	MHCC Comments
27.	Service From Date	
28.	Service Thru Date	
29.	Filler	
30.	Place of Service	
31.	Service Location Zip Code	
32.	Service Unit Indicator	
33.	Units	<1% = 0
34.	Procedure Code	
35.	Modifier I	85% = Missing
36.	Modifier II	100% = Missing
37.	Servicing Practitioner ID	
38.	Billed Charge	
39.	Allowed Amount	
40.	Reimbursement Amount	
41.	Patient Liability	
42.	Date of Enrollment	
43.	Date of Disenrollment	

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2004 MCDB

(Data received June 21, 2007)

P520 MD-Individual Practice Association, Inc.

Provider Layout: **25,484 records read in; 25,484 records after deleting exact duplicates.**

	Field Name	MHCC Comments
1.	Servicing Practitioner ID	
2.	Practitioner Federal Tax ID	
3.	Practitioner Last Name	
4.	Practitioner First Name	
5.	Practitioner Middle Initial	
6.	Practitioner Name Suffix	
7.	Practitioner Credential	
8.	Practitioner Specialty – 1	
9.	Practitioner Specialty – 2	100% missing
10.	Practitioner Specialty – 3	100% missing
11.	Practitioner DEA #	6,853 records (26.89%) = Missing
12.	Indicator for Multi-Practitioner Health Care Organization	
Percent of Exact Duplicate Records: 0%		
Percent of Records With No Match to Encounter: 0%		
Encounter Services File With Provider File Link: 3,856,672 services by FEDTAXID and PROVID, 766 services by PROVID only, 0 services no match		

NOTE: Encounter, Pharmacy & Provider Directory statistics based on data reflected in a variety of analytical files and do not represent the data from original submission to SSS.

Notes on Enrollment and Disenrollment Dates:

Description	Number of Records
Enrollment Date = Disenrollment Date	233 (<1%)
Service Begin Date AFTER Disenrollment Date	5.032 (<1%)

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2004 MCDB

(Data received June 20, 2007)

P530 Mega Life & Health Insurance Co.

Encounter Layout: **185,563 records read in.**

	Field Name	MHCC Comments
1.	Patient ID (encrypted)	
2.	Patient Date of Birth	
3.	Patient Sex	
4.	Consumer Directed Health Plan (CDHP) alone or w/ HAS or HRA indicator	
5.	Patient Zip Code	<1% = Invalid
6.	Patient Covered by Other Insurance	
7.	Coverage Type	
8.	Delivery System	26,954 (14.53%) = 9 (Unknown)
9.	Claim Related Condition	
10.	Practitioner Federal Tax ID	524 (<1%) = Missing
11.	Participating Provider Flag	
12.	Type of Bill	
13.	Claim Control Number	
14.	Claim Paid Date	
15.	Number of Diagnosis Codes	
16.	Number of Line Items	
17.	Diagnosis Code 1	2.62% = Missing/Invalid
18.	Diagnosis Code 2	10.35% = Missing/Invalid
19.	Diagnosis Code 3	55.94% = Missing/Invalid
20.	Diagnosis Code 4	77.80% = Missing/Invalid
21.	Diagnosis Code 5	100% = Missing
22.	Diagnosis Code 6	100% = Missing
23.	Diagnosis Code 7	100% = Missing
24.	Diagnosis Code 8	100% = Missing
25.	Diagnosis Code 9	100% = Missing
26.	Diagnosis Code 10	100% = Missing

	Field Name	MHCC Comments
27.	Service From Date	
28.	Service Thru Date	
29.	Filler	
30.	Place of Service	
31.	Service Location Zip Code	<1% = Missing/Invalid
32.	Service Unit Indicator	145,050 (78.17%) = 9 (Not Coded), 8.2% = Missing
33.	Units	
34.	Procedure Code	
35.	Modifier I	86.3% = Missing
36.	Modifier II	100% = Missing
37.	Servicing Practitioner ID	110 (<1%) = Missing
38.	Billed Charge	
39.	Allowed Amount	
40.	Reimbursement Amount	
41.	Patient Liability	
42.	Date of Enrollment	1.69% = Missing
43.	Date of Disenrollment	

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2004 MCDB

(Data received June 20, 2007)

P530 Mega Life & Health Insurance Co.

Provider Layout: **13,278 records read in; 13,278 records after deleting exact duplicates.**

	Field Name	MHCC Comments
1.	Servicing Practitioner ID	
2.	Practitioner Federal Tax ID	
3.	Practitioner Last Name	
4.	Practitioner First Name	
5.	Practitioner Middle Initial	
6.	Practitioner Name Suffix	
7.	Practitioner Credential	
8.	Practitioner Specialty – 1	
9.	Practitioner Specialty – 2	Almost 100% missing
10.	Practitioner Specialty – 3	100% missing
11.	Practitioner DEA #	100% missing
12.	Indicator for Multi-Practitioner Health Care Organization	100% Missing
Percent of Exact Duplicate Records: 0%		
Percent of Records With No Match to Encounter:<1%		
Encounter Services File With Provider File Link: 185,039 services by FEDTAXID and PROVID, 0 services by PROVID only, 524 services no match		

NOTE: Encounter, Pharmacy & Provider Directory statistics based on data reflected in a variety of analytical files and do not represent the data from original submission to SSS.

Notes on Enrollment and Disenrollment Dates:

Description	Number of Records
Enrollment Date = Missing	3,137 (1.69%)
Enrollment Date = Disenrollment Date	940 (<1%)
Service Begin Date BEFORE Enrollment Date	735 (<1%)
Service Begin Date AFTER Disenrollment Date	3,080 (1.66%)

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2006 MCDB

(Data received June 21, 2007)

P620 Optimum Choice, Inc.

Encounter Layout: 6,874,787 records read in.

	Field Name	MHCC Comments
1.	Patient ID (encrypted)	
2.	Patient Date of Birth	
3.	Patient Sex	
4.	Consumer Directed Health Plan (CDHP) alone or w/ HAS or HRA indicator	
5.	Patient Zip Code	<1% = Missing
6.	Patient Covered by Other Insurance	
7.	Coverage Type	
8.	Delivery System	22,778 (<1%) = Invalid (coded as "6")
9.	Claim Related Condition	
10.	Practitioner Federal Tax ID	<1% Missing
11.	Participating Provider Flag	
12.	Type of Bill	
13.	Claim Control Number	
14.	Claim Paid Date	
15.	Number of Diagnosis Codes	
16.	Number of Line Items	
17.	Diagnosis Code 1	<1% = Missing/Invalid
18.	Diagnosis Code 2	51.33% = Missing/Invalid
19.	Diagnosis Code 3	73.24% = Missing/Invalid
20.	Diagnosis Code 4	85.35% = Missing/Invalid
21.	Diagnosis Code 5	94.77% = Missing/Invalid
22.	Diagnosis Code 6	96.04% = Missing/Invalid
23.	Diagnosis Code 7	97.02% = Missing/Invalid
24.	Diagnosis Code 8	97.67% = Missing/Invalid
25.	Diagnosis Code 9	98.35% = Missing/Invalid
26.	Diagnosis Code 10	99.95% = Missing/Invalid

	Field Name	MHCC Comments
27.	Service From Date	
28.	Service Thru Date	
29.	Filler	
30.	Place of Service	
31.	Service Location Zip Code	<1% = Invalid
32.	Service Unit Indicator	
33.	Units	<1% (14,435) = 0 or less
34.	Procedure Code	
35.	Modifier I	85.32% Missing
36.	Modifier II	100% Missing
37.	Servicing Practitioner ID	
38.	Billed Charge	
39.	Allowed Amount	
40.	Reimbursement Amount	
41.	Patient Liability	
42.	Date of Enrollment	
43.	Date of Disenrollment	

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2006 MCDB

(Data received June 21, 2007)

P620 Optimum Choice, Inc.

Provider Layout: **53,817 records read in; 33,789 records after deleting exact duplicates.**

	Field Name	MHCC Comments
1.	Servicing Practitioner ID	
2.	Practitioner Federal Tax ID	
3.	Practitioner Last Name	
4.	Practitioner First Name	
5.	Practitioner Middle Initial	
6.	Practitioner Name Suffix	
7.	Practitioner Credential	
8.	Practitioner Specialty – 1	
9.	Practitioner Specialty – 2	100% missing
10.	Practitioner Specialty – 3	100% missing
11.	Practitioner DEA #	29.94% (10,115 records) = Missing
12.	Indicator for Multi-Practitioner Health Care Organization	
Percent of Exact Duplicate Records: 37.22%		
Percent of Records With No Match to Encounter: 0%		
Encounter Services File With Provider File Link: 6,873,846 services by FEDTAXID and PROVID, 941 services by PROVID only, 0 services no match		

NOTE: Encounter, Pharmacy & Provider Directory statistics based on data reflected in a variety of analytical files and do not represent the data from original submission to SSS.

Notes on Enrollment and Disenrollment Dates:

Description	Number of Records
Enrollment Date = Disenrollment Date	59 (<1%)
Service Begin Date BEFORE Enrollment Date	23 (<1%)
Service Begin Date AFTER Disenrollment Date	25 (<1%)

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2006 MCDB

(Data received June 27, 2007)

P680 Coventry Healthcare of Delaware. Inc.

Encounter Layout: **1,300,757 records read in.**

	Field Name	MHCC Comments
1.	Patient ID (encrypted)	
2.	Patient Date of Birth	
3.	Patient Sex	<1% = 3 (Not coded)
4.	Consumer Directed Health Plan (CDHP) alone or w/ HAS or HRA indicator	99.24% = 0
5.	Patient Zip Code	<1% = Missing/Invalid
6.	Patient Covered by Other Insurance	100% = 9 (Unknown)
7.	Coverage Type	
8.	Delivery System	
9.	Claim Related Condition	100% = 9 (Unknown)
10.	Practitioner Federal Tax ID	
11.	Participating Provider Flag	1,573 (<1%) = 3 (Unknown)
12.	Type of Bill	
13.	Claim Control Number	
14.	Claim Paid Date	400,663 (30.8%) = Missing
15.	Number of Diagnosis Codes	
16.	Number of Line Items	
17.	Diagnosis Code 1	<1% = Invalid
18.	Diagnosis Code 2	47.23% =Missing/Invalid
19.	Diagnosis Code 3	72.27% =Missing/Invalid
20.	Diagnosis Code 4	87.12% =Missing/Invalid
21.	Diagnosis Code 5	99.87% =Missing/Invalid
22.	Diagnosis Code 6	99.94% = Missing
23.	Diagnosis Code 7	100% = Missing
24.	Diagnosis Code 8	100% = Missing
25.	Diagnosis Code 9	100% = Missing
26.	Diagnosis Code 10	100% = Missing

	Field Name	MHCC Comments
27.	Service From Date	
28.	Service Thru Date	
29.	Filler	
30.	Place of Service	<1% = Invalid
31.	Service Location Zip Code	30.82% = Missing/Invalid
32.	Service Unit Indicator	
33.	Units	60,335 (4.64%) = 0 or less
34.	Procedure Code	
35.	Modifier I	84.38% Missing
36.	Modifier II	98.62% Missing
37.	Servicing Practitioner ID	
38.	Billed Charge	
39.	Allowed Amount	
40.	Reimbursement Amount	
41.	Patient Liability	
42.	Date of Enrollment	
43.	Date of Disenrollment	

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2006 MCDB

(Data received June 27, 2007)

P680 Coventry Healthcare of Delaware. Inc.

Provider Layout: **13,609 records read in; 13,609 records after deleting exact duplicates.**

	Field Name	MHCC Comments
1.	Servicing Practitioner ID	
2.	Practitioner Federal Tax ID	
3.	Practitioner Last Name	
4.	Practitioner First Name	
5.	Practitioner Middle Initial	
6.	Practitioner Name Suffix	
7.	Practitioner Credential	
8.	Practitioner Specialty – 1	
9.	Practitioner Specialty – 2	12,947 (95.13%) = Missing
10.	Practitioner Specialty – 3	100% = Missing
11.	Practitioner DEA #	8,438 (62%) = Missing
12.	Indicator for Multi-Practitioner Health Care Organization	100% = Missing
Percent of Exact Duplicate Records: 0%		
Percent of Records With No Match to Encounter: 0%		
Encounter Services File With Provider File Link: 1,300,757 services by FEDTAXID and PROVID, 0 services by PROVID only, 0 services no match		

NOTE: Encounter, Pharmacy & Provider Directory statistics based on data reflected in a variety of analytical files and do not represent the data from original submission to SSS.

Notes on Enrollment and Disenrollment Dates:

Description	Number of Records
Enrollment Date = Disenrollment Date	109,944 (8.45%)
Service Begin Date BEFORE Enrollment Date	32,697 (2.51%)
Service Begin Date AFTER Disenrollment Date	13,087 (1.01%)

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2006 MCDB

(Last Submission Data received August 27, 2007)

P760 State Farm Mutual Automobile Ins. Co.

Encounter Layout: **65,462 records read in.**

	Field Name	MHCC Comments
1.	Patient ID (encrypted)	
2.	Patient Date of Birth	
3.	Patient Sex	338 (<1%) = 3 (Unknown)
4.	Consumer Directed Health Plan (CDHP) alone or w/ HAS or HRA indicator	
5.	Patient Zip Code	
6.	Patient Covered by Other Insurance	16,135 (24.65%) = 9 (Unknown)
7.	Coverage Type	
8.	Delivery System	
9.	Claim Related Condition	
10.	Practitioner Federal Tax ID	
11.	Participating Provider Flag	
12.	Type of Bill	
13.	Claim Control Number	
14.	Claim Paid Date	
15.	Number of Diagnosis Codes	
16.	Number of Line Items	
17.	Diagnosis Code 1	34.46% = Invalid
18.	Diagnosis Code 2	65.73% = Missing/Invalid
19.	Diagnosis Code 3	73.26% = Missing/Invalid
20.	Diagnosis Code 4	100% Missing
21.	Diagnosis Code 5	100% Missing
22.	Diagnosis Code 6	100% Missing
23.	Diagnosis Code 7	100% Missing
24.	Diagnosis Code 8	100% Missing
25.	Diagnosis Code 9	100% Missing
26.	Diagnosis Code 10	100% Missing
27.	Service From Date	

	Field Name	MHCC Comments
28.	Service Thru Date	
29.	Filler	
30.	Place of Service	
31.	Service Location Zip Code	2.14% = Missing/Invalid
32.	Service Unit Indicator	283 (<1%) = 9 (Not Coded)
33.	Units	294 (<1%) = 0
34.	Procedure Code	1,337 (2.10%) = Missing
35.	Modifier I	42,265 (64.56%) = Missing
36.	Modifier II	100% Missing
37.	Servicing Practitioner ID	
38.	Billed Charge	
39.	Allowed Amount	
40.	Reimbursement Amount	
41.	Patient Liability	
42.	Date of Enrollment	
43.	Date of Disenrollment	

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2006 MCDB

(Data received July 19, 2007)

P760 State Farm Mutual Automobile Ins. Co.

Provider Layout: **5,578 records read in; 5,578 records after deleting exact duplicates.**

	Field Name	MHCC Comments
1.	Servicing Practitioner ID	
2.	Practitioner Federal Tax ID	
3.	Practitioner Last Name	
4.	Practitioner First Name	
5.	Practitioner Middle Initial	
6.	Practitioner Name Suffix	
7.	Practitioner Credential	
8.	Practitioner Specialty – 1	43.83 = Blank (Waived)
9.	Practitioner Specialty – 2	100% = Blank (Waived)
10.	Practitioner Specialty – 3	100% = Blank (Waived)
11.	Practitioner DEA #	100% = Missing (Waived)
12.	Indicator for Multi-Practitioner Health Care Organization	
Percent of Exact Duplicate Records: 0%		
Percent of Records With No Match to Encounter: N/A		
Encounter Services File With Provider File Link: N/A		

NOTES:

- Encounter, Pharmacy & Provider Directory statistics based on data reflected in a variety of analytical files and do not represent the data from original submission to SSS.
- Per documentations submitted with data, State Farm received waiver for the following format requirement:
  a) Practitioner Specialty Fields (March 23, 2001 letter to Ben Steffen.)
  b) DEA#, Practitioner Name Suffix, and Practitioner Credential fields (May 6, 2005.)

Notes on Enrollment and Disenrollment Dates:

Description	Number of Records
Service Begin Date AFTER Disenrollment Date	22,991 (35.12%)

NOTES:

- State Farm does not store "Date of Enrollment" and "Date of Disenrollment" on Group Medical Enrollees. The Group Medical Plan year is January 1 – December 31 of each year. Payer makes claim payments each year on past enrollees who are no longer on a plan but who were covered at the time of service.

As an alternative, State Farm, per discussion with Larry Monroe, provides "Policy Start Date" and "Policy End Date" for individual Medicare Supplement and Hospital/Surgical policies.

MARYLAND HEALTH CARE COMMISSION
Quality Review Statement for 2006 MCDB

(Data received June 21, 2007)

P820 United Healthcare Insurance Co.

Encounter Layout: **6,985,303 records read in.**

	Field Name	MHCC Comments
1.	Patient ID (encrypted)	
2.	Patient Date of Birth	<1% (105 with Year =1801, 1889, 1899)
3.	Patient Sex	
4.	Consumer Directed Health Plan (CDHP) alone or w/ HAS or HRA indicator	
5.	Patient Zip Code	<1% Invalid
6.	Patient Covered by Other Insurance	
7.	Coverage Type	
8.	Delivery System	<1% (773 records) = 9 (Unknown)
9.	Claim Related Condition	<1% (13 records) = 9 (Unknown)
10.	Practitioner Federal Tax ID	
11.	Participating Provider Flag	
12.	Type of Bill	
13.	Claim Control Number	
14.	Claim Paid Date	
15.	Number of Diagnosis Codes	
16.	Number of Line Items	
17.	Diagnosis Code 1	1.12% = Invalid
18.	Diagnosis Code 2	49.23% = Missing/Invalid
19.	Diagnosis Code 3	72.84% = Missing/Invalid
20.	Diagnosis Code 4	86.43% = Missing/Invalid
21.	Diagnosis Code 5	98.88% = Missing/Invalid
22.	Diagnosis Code 6	100% Missing
23.	Diagnosis Code 7	100% Missing
24.	Diagnosis Code 8	100% Missing
25.	Diagnosis Code 9	100% Missing
26.	Diagnosis Code 10	100% Missing

	Field Name	MHCC Comments
27.	Service From Date	
28.	Service Thru Date	
29.	Filler	
30.	Place of Service	<1% = Invalid
31.	Service Location Zip Code	<1% = Missing/Invalid
32.	Service Unit Indicator	
33.	Units	179,456 (2.57%) = Missing, Invalid, zero or less
34.	Procedure Code	
35.	Modifier I	91.98% Missing
36.	Modifier II	100% Missing
37.	Servicing Practitioner ID	
38.	Billed Charge	
39.	Allowed Amount	
40.	Reimbursement Amount	
41.	Patient Liability	
42.	Date of Enrollment	
43.	Date of Disenrollment	

MARYLAND HEALTH CARE COMMISSION
Quality Review Statement for 2006 MCDB

(Data received June 21, 2007)

P820 United Healthcare Insurance Co.

Provider Layout: **58,509 records read in; 58,509 records after deleting exact duplicates.**

	Field Name	MHCC Comments
1.	Servicing Practitioner ID	
2.	Practitioner Federal Tax ID	
3.	Practitioner Last Name	
4.	Practitioner First Name	
5.	Practitioner Middle Initial	
6.	Practitioner Name Suffix	
7.	Practitioner Credential	
8.	Practitioner Specialty – 1	
9.	Practitioner Specialty – 2	100% = Missing
10.	Practitioner Specialty – 3	100% = Missing
11.	Practitioner DEA #	24,669 (42.16%) = Missing
12.	Indicator for Multi-Practitioner Health Care Organization	
Percent of Exact Duplicate Records: 0%		
Percent of Records With No Match to Encounter: 0%		
Encounter Services File With Provider File Link: 6,985,303 services by FEDTAXID and PROVID, 0 services by PROVID only, 0 services no match		

NOTE: Encounter, Pharmacy & Provider Directory statistics based on data reflected in a variety of analytical files and do not represent the data from original submission to SSS.

Notes on Enrollment and Disenrollment Dates:

Description	Number of Records
Enrollment Date = Disenrollment Date	3,152 (<1%)
Service Begin Date BEFORE Enrollment Date	3,308 (<1%)
Service Begin Date AFTER Disenrollment Date	51,382 (<1%)

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2006 MCDB

(Data received July 2, 2007)

P830 Trustmark Insurance Co.

Encounter Layout: **5,877 records read in.**

	Field Name	MHCC Comments
1.	Patient ID (encrypted)	
2.	Patient Date of Birth	
3.	Patient Sex	
4.	Consumer Directed Health Plan (CDHP) alone or w/ HAS or HRA indicator	
5.	Patient Zip Code	
6.	Patient Covered by Other Insurance	
7.	Coverage Type	
8.	Delivery System	
9.	Claim Related Condition	
10.	Practitioner Federal Tax ID	12.22% (718 records) = Missing
11.	Participating Provider Flag	7.69% = 3 (Unknown)
12.	Type of Bill	
13.	Claim Control Number	
14.	Claim Paid Date	
15.	Number of Diagnosis Codes	
16.	Number of Line Items	
17.	Diagnosis Code 1	21.86% = Invalid
18.	Diagnosis Code 2	56.54% = Missing/Invalid
19.	Diagnosis Code 3	75.24% = Missing/Invalid
20.	Diagnosis Code 4	99.92% = Missing/Invalid
21.	Diagnosis Code 5	99.91% = Missing
22.	Diagnosis Code 6	100% Missing
23.	Diagnosis Code 7	100% Missing
24.	Diagnosis Code 8	100% Missing
25.	Diagnosis Code 9	100% Missing
26.	Diagnosis Code 10	100% Missing

	Field Name	MHCC Comments
27.	Service From Date	
28.	Service Thru Date	
29.	Filler	
30.	Place of Service	
31.	Service Location Zip Code	
32.	Service Unit Indicator	
33.	Units	
34.	Procedure Code	938 (15.96%) = Missing
35.	Modifier I	94.06% Missing
36.	Modifier II	100% Missing
37.	Servicing Practitioner ID	
38.	Billed Charge	
39.	Allowed Amount	
40.	Reimbursement Amount	
41.	Patient Liability	
42.	Date of Enrollment	
43.	Date of Disenrollment	

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2006 MCDB

(Data received July 2, 2007)

P830 Trustmark Insurance Co.

Provider Layout: **2,747 records read in; 585 records after deleting exact duplicates.**

	Field Name	MHCC Comments
1.	Servicing Practitioner ID	1 = Missing
2.	Practitioner Federal Tax ID	
3.	Practitioner Last Name	
4.	Practitioner First Name	
5.	Practitioner Middle Initial	100% = Missing
6.	Practitioner Name Suffix	100% = Missing
7.	Practitioner Credential	100% = Missing
8.	Practitioner Specialty – 1	
9.	Practitioner Specialty – 2	100% = Missing
10.	Practitioner Specialty – 3	100% = Missing
11.	Practitioner DEA #	100% = Missing
12.	Indicator for Multi-Practitioner Health Care Organization	100% = Missing
Percent of Exact Duplicate Records: 78.70%		
Percent of Records With No Match to Encounter: 94.61%		
Encounter Services File With Provider File Link:		277 services by FEDTAXID and PROVID, 1 services by PROVID only, 5,599 services no match

NOTE: Encounter, Pharmacy & Provider Directory statistics based on data reflected in a variety of analytical files and do not represent the data from original submission to SSS.

Notes on Enrollment and Disenrollment Dates:

Description	Number of Records
Enrollment Date = Disenrollment Date	N/A
Service Begin Date BEFORE Enrollment	N/A
Service Begin Date AFTER Disenrollment Date	N/A

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2006 MCDB

(Last Submission Data Received August 13, 2007)

P850 Union Labor Life Insurance Co.

Encounter Layout: **106,508 records read in.**

	Field Name	MHCC Comments
1.	Patient ID (encrypted)	
2.	Patient Date of Birth	
3.	Patient Sex	
4.	Consumer Directed Health Plan (CDHP) alone or w/ HAS or HRA indicator	
5.	Patient Zip Code	<1% = Invalid
6.	Patient Covered by Other Insurance	
7.	Coverage Type	
8.	Delivery System	
9.	Claim Related Condition	100% = 9
10.	Practitioner Federal Tax ID	
11.	Participating Provider Flag	
12.	Type of Bill	
13.	Claim Control Number	
14.	Claim Paid Date	
15.	Number of Diagnosis Codes	
16.	Number of Line Items	
17.	Diagnosis Code 1	16.49% = Missing/Invalid
18.	Diagnosis Code 2	56.70% = Missing/Invalid
19.	Diagnosis Code 3	77.50% = Missing/Invalid
20.	Diagnosis Code 4	88.94% = Missing/Invalid
21.	Diagnosis Code 5	100% = Missing
22.	Diagnosis Code 6	100% = Missing
23.	Diagnosis Code 7	100% = Missing
24.	Diagnosis Code 8	100% = Missing
25.	Diagnosis Code 9	100% = Missing
26.	Diagnosis Code 10	100% = Missing

	Field Name	MHCC Comments
27.	Service From Date	
28.	Service Thru Date	
29.	Filler	
30.	Place of Service	
31.	Service Location Zip Code	<1% Missing/Invalid
32.	Service Unit Indicator	
33.	Units	
34.	Procedure Code	1.26% = Missing
35.	Modifier I	83.56% = Missing
36.	Modifier II	99% = Missing
37.	Servicing Practitioner ID	
38.	Billed Charge	
39.	Allowed Amount	
40.	Reimbursement Amount	
41.	Patient Liability	
42.	Date of Enrollment	
43.	Date of Disenrollment	

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2006 MCDB

(Last Submission Data Received August 13, 2007)

P850 Union Labor Life Insurance Co.

Provider Layout: **9,731 records read in; 9,276 records after deleting exact duplicates.**

	Field Name	MHCC Comments
1.	Servicing Practitioner ID	
2.	Practitioner Federal Tax ID	
3.	Practitioner Last Name	
4.	Practitioner First Name	
5.	Practitioner Middle Initial	
6.	Practitioner Name Suffix	
7.	Practitioner Credential	
8.	Practitioner Specialty – 1	
9.	Practitioner Specialty – 2	100% = Missing
10.	Practitioner Specialty – 3	100% = Missing
11.	Practitioner DEA #	
12.	Indicator for Multi-Practitioner Health Care Organization	
Percent of Exact Duplicate Records: 4.68%		
Percent of Records With No Match to Encounter: 0%		
Encounter Services File With Provider File Link:	106,508 services by FEDTAXID and PROVID, 0 services by PROVID only, 0 services no match	

NOTE: Encounter, Pharmacy & Provider Directory statistics based on data reflected in a variety of analytical files and do not represent the data from original submission to SSS.

Notes on Enrollment and Disenrollment Dates:

Description	Number of Records
Enrollment Date = Missing	25 (<1%)
Enrollment Date = Disenrollment Date	3 (<1%)
Service Begin Date BEFORE Enrollment Date	3,197 (3%)
Service Begin Date AFTER Disenrollment Date	72 (<1%)

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2006 MCDB

(Data received June 21, 2007)

P870 United Healthcare of the Mid Atlantic, Inc.

Encounter Layout: **163,232 records read in.**

	Field Name	MHCC Comments
1.	Patient ID (encrypted)	
2.	Patient Date of Birth	
3.	Patient Sex	
4.	Consumer Directed Health Plan (CDHP) alone or w/ HAS or HRA indicator	
5.	Patient Zip Code	<1% = Invalid
6.	Patient Covered by Other Insurance	
7.	Coverage Type	
8.	Delivery System	
9.	Claim Related Condition	<1% = 9 (Unknown)
10.	Practitioner Federal Tax ID	
11.	Participating Provider Flag	
12.	Type of Bill	
13.	Claim Control Number	
14.	Claim Paid Date	
15.	Number of Diagnosis Codes	
16.	Number of Line Items	
17.	Diagnosis Code 1	<1% = Invalid
18.	Diagnosis Code 2	53.52% = Missing/Invalid
19.	Diagnosis Code 3	77.25% = Missing/Invalid
20.	Diagnosis Code 4	89.07% = Missing/Invalid
21.	Diagnosis Code 5	Almost 100% = Missing/Invalid
22.	Diagnosis Code 6	100% = Missing
23.	Diagnosis Code 7	100% = Missing
24.	Diagnosis Code 8	100% = Missing
25.	Diagnosis Code 9	100% = Missing
26.	Diagnosis Code 10	100% = Missing

	Field Name	MHCC Comments
27.	Service From Date	
28.	Service Thru Date	
29.	Filler	
30.	Place of Service	
31.	Service Location Zip Code	<1% = Missing/Invalid
32.	Service Unit Indicator	
33.	Units	2.78% = Missing, zero or less
34.	Procedure Code	
35.	Modifier I	93.34% = Missing
36.	Modifier II	100% = Missing
37.	Servicing Practitioner ID	
38.	Billed Charge	
39.	Allowed Amount	
40.	Reimbursement Amount	
41.	Patient Liability	
42.	Date of Enrollment	
43.	Date of Disenrollment	

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2006 MCDB

(Data received June 21, 2007)

P870 United Healthcare of the Mid Atlantic, Inc.

Provider Layout: **9,236 records read in; 9,236 records after deleting exact duplicates.**

	Field Name	MHCC Comments
1.	Servicing Practitioner ID	
2.	Practitioner Federal Tax ID	
3.	Practitioner Last Name	
4.	Practitioner First Name	
5.	Practitioner Middle Initial	
6.	Practitioner Name Suffix	
7.	Practitioner Credential	
8.	Practitioner Specialty – 1	
9.	Practitioner Specialty – 2	100% missing
10.	Practitioner Specialty – 3	100% missing
11.	Practitioner DEA #	2,582 (27.96%) = Missing
12.	Indicator for Multi-Practitioner Health Care Organization	
Percent of Exact Duplicate Records: 0%		
Percent of Records With No Match to Encounter: 0%		
Encounter Services File With Provider File Link:	163,232 services by FEDTAXID and PROVID, 0 services by PROVID only, 0 services no match	

NOTE: Encounter, Pharmacy & Provider Directory statistics based on data reflected in a variety of analytical files and do not represent the data from original submission to SSS.

Notes on Enrollment and Disenrollment Dates:

Description	Number of Records
Enrollment Date = Disenrollment Date	109 (<1%)
Service Begin Date BEFORE Enrollment Date	150 (<1%)
Service Begin Date AFTER Disenrollment Date	1,305 (<1%)

Appendix D

Payer-Specific Data completeness Summary Reports

MCDB Encounter Preprocessing
January 2006 - April 2007 Data

P020: Aetna Life and Health Insurance Co
Data After Initial Processing Step (enc5)
Data Completeness Summary Report

Eligible Services: 6,735,681
Services Submitted: 6,735,681

Source File: P020_enc5_dc_crunch.sas7bdat
File Date: February 27, 2008

Delivery System	Number of Recipients[1]			Number of Services[2]			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
1: HMO (Non-Medicaid, Includes Medicare)									
2: PPO-POS	145,766	173,420	19.0	3,202,214	4,063,403	26.9	217,503,839	290,070,229	33.4
3: PPO or Other Managed Care	74,747	79,220	6.0	1,640,363	1,994,596	21.6	116,073,715	144,041,698	24.1
4: Indemnity Care	16,658	16,844	1.1	637,181	672,090	5.5	24,110,346	25,194,371	4.5
5: HMO-POS Rider									
9: Payer Code=9 (Not Coded)									
Missing or Invalid Code									
Total	**230,469**	**264,118**	**14.6**	**5,479,758**	**6,730,089**	**22.8**	**357,687,900**	**459,306,298**	**28.4**

Plan[3]	Number of Recipients[1]			Number of Services			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
Non-HMO	205,372	236,092	15.0	4,503,480	5,431,606	20.6	316,935,911	395,996,035	24.9
HMO Fee for Service									
HMO Capitated	811	2,904	258.1	6,516	46,519	613.9	.	.	.
Medicare, All Types	3,538	4,347	22.9	45,812	214,483	368.2	3,896,368	17,856,175	358.3
No Plan Assigned	21,522	23,667	10.0	923,950	1,037,481	12.3	36,855,621	45,454,088	23.3
Total	**230,469**	**264,118**	**14.6**	**5,479,758**	**6,730,089**	**22.8**	**357,687,900**	**459,306,298**	**28.4**

Coverage Type	Number of Recipients[1]			Number of Services			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
1: Medicare Supplemental									
2: Individual Plan	1,197	3,112	160.0	14,720	48,312	228.2	914,057	3,209,216	251.1
3: Private Employer Sponsored Fully Self-Ins	205,032	217,054	5.9	5,017,158	5,451,630	8.7	325,090,372	369,806,134	13.8
4: Private Employer Sponsored Insured	12,060	15,939	32.2	235,933	328,966	39.4	17,033,267	24,812,386	45.7
5: Public Employee	1,539	15,860	930.5	26,125	455,293	1642.7	1,590,691	29,525,646	1756.2
6: Comprehensive Standard Health Benefit Plan	6,679	7,237	8.4	126,210	147,629	17.0	8,492,417	10,435,732	22.9
7: Medicare Provided by a Medicare HMO/CMS	4,034	5,131	27.2	59,612	298,259	400.3	4,567,096	21,517,184	371.1
8: Taft Hartley Jointly Managed Trust Fund									
9: Payer Code-9 (Unknown Coverage Type)									
Missing or Invalid Code									
Total	**230,469**	**264,118**	**14.6**	**5,479,758**	**6,730,089**	**22.8**	**357,687,900**	**459,306,298**	**28.4**

MCDB Encounter Preprocessing
January 2006 - April 2007 Data

P020: Aetna Life and Health Insurance Co
Data After Initial Processing Step (enc5)
Data Completeness Summary Report

Eligible Services: 6,735,681
Services Submitted: 6,735,681

Source File: P020_enc5_dc_crunch.sas7bdat
File Date: February 27, 2008

NOTES:
[1] Total number of recipients will be less than the sum of individual category recipient counts if some recipients receive services in more than one category.
Key to identify a unique recipient: Patient ID + Birth Year + Birth Month + Gender.

[2] The number of services passing MCDB screens. 2004 services does not include Non-MD Patients but 2005 services does include Non-MD patients.
Please see Payer-Specific MCDB Deletion Report for counts of Services failing screens.

[3] Rules for categorizing services into a PLAN:

Non-HMO
1. Payer is not an HMO provider and Coverage Type (COVTYPE) is non-Medicare (2-6) or Taft-Hartley (COVTYPE = 8).
   a. Coverage Type (COVTYPE) is non-Medicare (2-6)
   b. Coverage Type (COVTYPE) is Taft-Hartley (8).
2. Payer is an HMO provider:
   a. Delivery System (DELVTYP) is non-HMO (2-4).
   b. Coverage Type (COVTYPE) is non-Medicare (2-6)

HMO Fee for Service:
1. Payer is an HMO provider.
2. Coverage Type (COVTYPE) is non-Medicare (2-6).
3. Delivery System (DELVTYP) is HMO (1 or 5).
4. Service is not capitated (BILLTYPE = 1).

HMO Capitated:
1. Payer is an HMO provider.
2. Coverage Type (COVTYPE) is non-Medicare (2-6).
3. Delivery System (DELVTYP) is HMO (1 or 5).
4. Service is capitated (BILLTYPE = 8).

Medicare, All Types
1, All services with Coverage Type 1 or 7.

MCDB Encounter Preprocessing
January 2006 - April 2007 Data

P030: Aetna US Health Care
Data After Initial Processing Step (enc5)
Data Completeness Summary Report

Eligible Services: 4,807,625
Services Submitted: 4,807,625

Source File: P030_enc5_dc_crunch.sas7bdat
File Date: February 27, 2008

Delivery System	Number of Recipients[1]			Number of Services[2]			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
1: HMO (Non-Medicaid, Includes Medicare)	171,961	166,330	-3.3	4,009,965	4,097,030	2.2	216,790,435	244,735,102	12.9
2: PPO-POS									
3: PPO or Other Managed Care									
4: Indemnity Care									
5: HMO-POS Rider	35,978	29,631	-17.6	964,140	696,720	-27.7	55,638,960	40,251,919	-27.7
9: Payer Code=9 (Not Coded)									
Missing or Invalid Code									
Total	**207,335**	**195,431**	**-5.7**	**4,974,105**	**4,793,750**	**-3.6**	**272,429,395**	**284,987,021**	**4.6**

Plan[3]	Number of Recipients[1]			Number of Services			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
Non-HMO									
HMO Fee for Service	185,633	177,295	-4.5	3,109,698	3,166,010	1.8	238,761,132	257,002,670	7.6
HMO Capitated	137,296	122,580	-10.7	1,435,332	1,267,393	-11.7	.	.	.
Medicare, All Types	6,206	3,824	-38.4	185,243	116,714	-37.0	16,448,078	10,511,346	-36.1
No Plan Assigned	6,788	6,833	0.7	243,832	243,633	-0.1	17,220,185	17,473,005	1.5
Total	**207,335**	**195,431**	**-5.7**	**4,974,105**	**4,793,750**	**-3.6**	**272,429,395**	**284,987,021**	**4.6**

Coverage Type	Number of Recipients[1]			Number of Services			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
1: Medicare Supplemental	1	.	.	39	.	.	4,035	.	.
2: Individual Plan	148	124	-16.2	4,932	3,756	-23.8	276,034	227,786	-17.5
3: Private Employer Sponsored Fully Self-Ins	70,544	53,887	-23.6	1,586,416	1,224,901	-22.8	81,275,375	71,600,613	-11.9
4: Private Employer Sponsored Insured	47,030	36,160	-23.1	932,415	785,093	-15.8	50,691,901	46,850,120	-7.6
5: Public Employee	74,208	90,102	21.4	1,883,782	2,372,777	26.0	104,914,508	139,929,952	33.4
6: Comprehensive Standard Health Benefit Plan	14,361	10,306	-28.2	296,046	219,152	-26.0	16,258,673	13,207,770	-18.8
7: Medicare Provided by a Medicare HMO/CMS	7,130	4,986	-30.1	270,475	188,071	-30.5	19,008,869	13,170,780	-30.7
8: Taft Hartley Jointly Managed Trust Fund									
9: Payer Code-9 (Unknown Coverage Type)									
Missing or Invalid Code									
Total	**207,335**	**195,431**	**-5.7**	**4,974,105**	**4,793,750**	**-3.6**	**272,429,395**	**284,987,021**	**4.6**

MCDB Encounter Preprocessing
January 2006 - April 2007 Data

P030: Aetna US Health Care
Data After Initial Processing Step (enc5)
Data Completeness Summary Report

Eligible Services: 4,807,625 | **Source File: P030_enc5_dc_crunch.sas7bdat**
Services Submitted: 4,807,625 | **File Date: February 27, 2008**

NOTES:
[1] Total number of recipients will be less than the sum of individual category recipient counts if some recipients receive services in more than one category.
Key to identify a unique recipient: Patient ID + Birth Year + Birth Month + Gender.

[2] The number of services passing MCDB screens. 2004 services does not include Non-MD Patients but 2005 services does include Non-MD patients.
Please see Payer-Specific MCDB Deletion Report for counts of Services failing screens.

[3] Rules for categorizing services into a PLAN:

Non-HMO
1. Payer is not an HMO provider and Coverage Type (COVTYPE) is non-Medicare (2-6) or Taft-Hartley (COVTYPE = 8).
   a. Coverage Type (COVTYPE) is non-Medicare (2-6)
   b. Coverage Type (COVTYPE) is Taft-Hartley (8).
2. Payer is an HMO provider:
   a. Delivery System (DELVTYP) is non-HMO (2-4).
   b. Coverage Type (COVTYPE) is non-Medicare (2-6)

HMO Fee for Service:
1. Payer is an HMO provider.
2. Coverage Type (COVTYPE) is non-Medicare (2-6).
3. Delivery System (DELVTYP) is HMO (1 or 5).
4. Service is not capitated (BILLTYPE = 1).

HMO Capitated:
1. Payer is an HMO provider.
2. Coverage Type (COVTYPE) is non-Medicare (2-6).
3. Delivery System (DELVTYP) is HMO (1 or 5).
4. Service is capitated (BILLTYPE = 8).

Medicare, All Types
1, All services with Coverage Type 1 or 7.

MCDB Encounter Preprocessing
January 2006 - April 2007 Data

P070: American Republic Insurance Co
Data After Initial Processing Step (enc5)
Data Completeness Summary Report

Eligible Services: 24,070
Services Submitted: 24,070

Source File: P070_enc5_dc_crunch.sas7bdat
File Date: February 27, 2008

Delivery System	Number of Recipients[1]			Number of Services[2]			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
1: HMO (Non-Medicaid, Includes Medicare)									
2: PPO-POS									
3: PPO or Other Managed Care	949	562	-40.8	19,769	9,474	-52.1	1,579,464	734,806	-53.5
4: Indemnity Care	553	548	-0.9	10,200	14,523	42.4	1,360,515	2,661,032	95.6
5: HMO-POS Rider									
9: Payer Code=9 (Not Coded)									
Missing or Invalid Code									
Total	**1,269**	**903**	**-28.8**	**29,969**	**23,997**	**-19.9**	**2,939,979**	**3,395,838**	**15.5**

Plan[3]	Number of Recipients[1]			Number of Services			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
Non-HMO	1,090	748	-31.4	23,057	16,339	-29.1	2,196,787	1,940,696	-11.7
HMO Fee for Service									
HMO Capitated									
Medicare, All Types	98	92	-6.1	4,566	5,318	16.5	503,864	1,163,921	131.0
No Plan Assigned	81	63	-22.2	2,346	2,340	-0.3	239,328	291,221	21.7
Total	**1,269**	**903**	**-28.8**	**29,969**	**23,997**	**-19.9**	**2,939,979**	**3,395,838**	**15.5**

Coverage Type	Number of Recipients[1]			Number of Services			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
1: Medicare Supplemental	98	92	-6.1	4,566	5,318	16.5	503,864	1,163,921	131.0
2: Individual Plan	1,171	811	-30.7	25,403	18,679	-26.5	2,436,115	2,231,917	-8.4
3: Private Employer Sponsored Fully Self-Ins									
4: Private Employer Sponsored Insured									
5: Public Employee									
6: Comprehensive Standard Health Benefit Plan									
7: Medicare Provided by a Medicare HMO/CMS									
8: Taft Hartley Jointly Managed Trust Fund									
9: Payer Code-9 (Unknown Coverage Type)									
Missing or Invalid Code									
Total	**1,269**	**903**	**-28.8**	**29,969**	**23,997**	**-19.9**	**2,939,979**	**3,395,838**	**15.5**

MCDB Encounter Preprocessing
January 2006 - April 2007 Data

P070: American Republic Insurance Co
Data After Initial Processing Step (enc5)
Data Completeness Summary Report

Eligible Services: 24,070
Services Submitted: 24,070

Source File: P070_enc5_dc_crunch.sas7bdat
File Date: February 27, 2008

NOTES:

[1] Total number of recipients will be less than the sum of individual category recipient counts if some recipients receive services in more than one category.

Key to identify a unique recipient: Patient ID + Birth Year + Birth Month + Gender.

[2] The number of services passing MCDB screens. 2004 services does not include Non-MD Patients but 2005 services does include Non-MD patients.

Please see Payer-Specific MCDB Deletion Report for counts of Services failing screens.

[3] Rules for categorizing services into a PLAN:

Non-HMO

1. Payer is not an HMO provider and Coverage Type (COVTYPE) is non-Medicare (2-6) or Taft-Hartley (COVTYPE = 8).
   a. Coverage Type (COVTYPE) is non-Medicare (2-6)
   b. Coverage Type (COVTYPE) is Taft-Hartley (8).
2. Payer is an HMO provider:
   a. Delivery System (DELVTYP) is non-HMO (2-4).
   b. Coverage Type (COVTYPE) is non-Medicare (2-6)

HMO Fee for Service:

1. Payer is an HMO provider.
2. Coverage Type (COVTYPE) is non-Medicare (2-6).
3. Delivery System (DELVTYP) is HMO (1 or 5).
4. Service is not capitated (BILLTYPE = 1).

HMO Capitated:

1. Payer is an HMO provider.
2. Coverage Type (COVTYPE) is non-Medicare (2-6).
3. Delivery System (DELVTYP) is HMO (1 or 5).
4. Service is capitated (BILLTYPE = 8).

Medicare, All Types

1, All services with Coverage Type 1 or 7.

MCDB Encounter Preprocessing
January 2006 - April 2007 Data

P130: Carefirst Blue Choice Inc
Data After Initial Processing Step (enc5)
Data Completeness Summary Report

Eligible Services: 18,773,350
Services Submitted: 18,773,350

Source File: P130_enc5_dc_crunch.sas7bdat
File Date: March 13, 2008

Delivery System	Number of Recipients[1]			Number of Services[2]			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
1: HMO (Non-Medicaid, Includes Medicare)	585,955	439,059	-25.1	7,792,672	8,457,428	8.5	441,267,964	476,077,425	7.9
2: PPO-POS	33,681	52,695	56.5	591,214	1,080,340	82.7	40,797,132	77,442,009	89.8
3: PPO or Other Managed Care	267,354	291,515	9.0	7,189,229	7,873,771	9.5	460,441,757	505,571,865	9.8
4: Indemnity Care	10,044	6,663	-33.7	384,169	311,262	-19.0	20,127,288	18,042,005	-10.4
5: HMO-POS Rider	80,576	56,534	-29.8	1,405,570	1,050,549	-25.3	107,508,854	79,245,449	-26.3
9: Payer Code=9 (Not Coded)									
Missing or Invalid Code									
Total	**939,683**	**785,131**	**-16.4**	**17,362,854**	**18,773,350**	**8.1**	**1,070,142,995**	**1,156,378,753**	**8.1**

Plan[3]	Number of Recipients[1]			Number of Services			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
Non-HMO	254,600	295,526	16.1	5,830,392	6,916,311	18.6	431,849,430	508,585,251	17.8
HMO Fee for Service	434,474	431,526	-0.7	6,410,274	6,542,696	2.1	513,529,233	521,323,784	1.5
HMO Capitated	275,696	232,608	-15.6	2,193,018	2,370,378	8.1	.	.	.
Medicare, All Types	55,367	51,668	-6.7	2,072,443	2,093,155	1.0	54,336,633	56,212,538	3.5
No Plan Assigned	47,639	45,215	-5.1	856,727	850,810	-0.7	70,427,699	70,257,180	-0.2
Total	**939,683**	**785,131**	**-16.4**	**17,362,854**	**18,773,350**	**8.1**	**1,070,142,995**	**1,156,378,753**	**8.1**

Coverage Type	Number of Recipients[1]			Number of Services			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
1: Medicare Supplemental	332,303	283,865	-14.6	4,378,118	4,585,580	4.7	59,531,504	61,616,842	3.5
2: Individual Plan	31,240	35,003	12.0	476,811	524,913	10.1	38,746,027	41,733,329	7.7
3: Private Employer Sponsored Fully Self-Ins	148,014	168,168	13.6	2,387,181	2,886,900	20.9	179,945,392	217,460,338	20.8
4: Private Employer Sponsored Insured	68,096	65,002	-4.5	1,249,099	1,228,474	-1.7	98,717,428	93,768,115	-5.0
5: Public Employee	199,740	204,160	2.2	4,084,264	4,434,682	8.6	322,283,265	348,070,808	8.0
6: Comprehensive Standard Health Benefit Plan	281,613	292,576	3.9	4,787,381	5,112,801	6.8	370,919,379	393,729,321	6.1
7: Medicare Provided by a Medicare HMO/CMS									
8: Taft Hartley Jointly Managed Trust Fund									
9: Payer Code-9 (Unknown Coverage Type)									
Missing or Invalid Code									
Total	**939,683**	**785,131**	**-16.4**	**17,362,854**	**18,773,350**	**8.1**	**1,070,142,995**	**1,156,378,753**	**8.1**

MCDB Encounter Preprocessing
January 2006 - April 2007 Data

P130: Carefirst Blue Choice Inc
Data After Initial Processing Step (enc5)
Data Completeness Summary Report

Eligible Services: 18,773,350 | **Source File: P130_enc5_dc_crunch.sas7bdat**
Services Submitted: 18,773,350 | **File Date: March 13, 2008**

NOTES:
[1] Total number of recipients will be less than the sum of individual category recipient counts if some recipients receive services in more than one category.
Key to identify a unique recipient: Patient ID + Birth Year + Birth Month + Gender.

[2] The number of services passing MCDB screens. 2004 services does not include Non-MD Patients but 2005 services does include Non-MD patients.
Please see Payer-Specific MCDB Deletion Report for counts of Services failing screens.

[3] Rules for categorizing services into a PLAN:

Non-HMO
1. Payer is not an HMO provider and Coverage Type (COVTYPE) is non-Medicare (2-6) or Taft-Hartley (COVTYPE = 8).
   a. Coverage Type (COVTYPE) is non-Medicare (2-6)
   b. Coverage Type (COVTYPE) is Taft-Hartley (8).
2. Payer is an HMO provider:
   a. Delivery System (DELVTYP) is non-HMO (2-4).
   b. Coverage Type (COVTYPE) is non-Medicare (2-6)

HMO Fee for Service:
1. Payer is an HMO provider.
2. Coverage Type (COVTYPE) is non-Medicare (2-6).
3. Delivery System (DELVTYP) is HMO (1 or 5).
4. Service is not capitated (BILLTYPE = 1).

HMO Capitated:
1. Payer is an HMO provider.
2. Coverage Type (COVTYPE) is non-Medicare (2-6).
3. Delivery System (DELVTYP) is HMO (1 or 5).
4. Service is capitated (BILLTYPE = 8).

Medicare, All Types
1, All services with Coverage Type 1 or 7.

MCDB Encounter Preprocessing
January 2006 - April 2007 Data

P131: Carefirst of MD Inc
Data After Initial Processing Step (enc5)
Data Completeness Summary Report

Eligible Services: 23,547,560
Services Submitted: 23,547,560

Source File: P131_enc5_dc_crunch.sas7bdat
File Date: February 28, 2008

Delivery System	Number of Recipients[1]			Number of Services[2]			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
1: HMO (Non-Medicaid, Includes Medicare)									
2: PPO-POS	126,265	114,996	-8.9	3,060,586	2,873,439	-6.1	192,120,093	177,745,140	-7.5
3: PPO or Other Managed Care	513,088	534,508	4.2	13,132,348	14,057,325	7.0	732,285,700	773,630,914	5.6
4: Indemnity Care	210,639	204,630	-2.9	6,546,001	6,615,632	1.1	249,532,627	252,157,827	1.1
5: HMO-POS Rider									
9: Payer Code=9 (Not Coded)	333	47	-85.9	1,888	160	-91.5	143,212	12,859	-91.0
Missing or Invalid Code									
Total	**842,465**	**841,272**	**-0.1**	**22,740,823**	**23,546,556**	**3.5**	**1,174,081,632**	**1,203,546,740**	**2.5**

Plan[3]	Number of Recipients[1]			Number of Services			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
Non-HMO	649,061	649,758	0.1	15,032,762	15,644,308	4.1	947,188,003	975,697,729	3.0
HMO Fee for Service									
HMO Capitated									
Medicare, All Types	164,016	163,382	-0.4	6,259,552	6,496,300	3.8	141,056,869	148,148,866	5.0
No Plan Assigned	70,063	69,464	-0.9	1,448,509	1,405,948	-2.9	85,836,760	79,700,145	-7.1
Total	**842,465**	**841,272**	**-0.1**	**22,740,823**	**23,546,556**	**3.5**	**1,174,081,632**	**1,203,546,740**	**2.5**

Coverage Type	Number of Recipients[1]			Number of Services			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
1: Medicare Supplemental	169,270	169,856	0.3	6,519,000	6,789,414	4.1	147,684,667	155,816,941	5.5
2: Individual Plan	96,382	96,455	0.1	1,959,742	2,018,532	3.0	127,056,331	130,708,506	2.9
3: Private Employer Sponsored Fully Self-Ins	104,226	110,409	5.9	2,164,398	2,329,086	7.6	142,316,245	153,021,889	7.5
4: Private Employer Sponsored Insured	12,954	13,220	2.1	247,290	273,353	10.5	16,073,139	17,207,181	7.1
5: Public Employee	466,629	468,469	0.4	10,955,664	11,488,192	4.9	681,725,284	703,419,879	3.2
6: Comprehensive Standard Health Benefit Plan	35,712	25,532	-28.5	894,175	647,509	-27.6	59,089,077	43,280,781	-26.8
7: Medicare Provided by a Medicare HMO/CMS									
8: Taft Hartley Jointly Managed Trust Fund									
9: Payer Code-9 (Unknown Coverage Type)	287	235	-18.1	554	470	-15.2	136,889	91,563	-33.1
Missing or Invalid Code									
Total	**842,465**	**841,272**	**-0.1**	**22,740,823**	**23,546,556**	**3.5**	**1,174,081,632**	**1,203,546,740**	**2.5**

MCDB Encounter Preprocessing
January 2006 - April 2007 Data

P131: Carefirst of MD Inc
Data After Initial Processing Step (enc5)
Data Completeness Summary Report

Eligible Services: 23,547,560 | **Source File: P131_enc5_dc_crunch.sas7bdat**
Services Submitted: 23,547,560 | **File Date: February 28, 2008**

NOTES:
[1] Total number of recipients will be less than the sum of individual category recipient counts if some recipients receive services in more than one category.

Key to identify a unique recipient: Patient ID + Birth Year + Birth Month + Gender.

[2] The number of services passing MCDB screens. 2004 services does not include Non-MD Patients but 2005 services does include Non-MD patients.

Please see Payer-Specific MCDB Deletion Report for counts of Services failing screens.

[3] Rules for categorizing services into a PLAN:

Non-HMO

1. Payer is not an HMO provider and Coverage Type (COVTYPE) is non-Medicare (2-6) or Taft-Hartley (COVTYPE = 8).
   a. Coverage Type (COVTYPE) is non-Medicare (2-6)
   b. Coverage Type (COVTYPE) is Taft-Hartley (8).
2. Payer is an HMO provider:
   a. Delivery System (DELVTYP) is non-HMO (2-4).
   b. Coverage Type (COVTYPE) is non-Medicare (2-6)

HMO Fee for Service:

1. Payer is an HMO provider.
2. Coverage Type (COVTYPE) is non-Medicare (2-6).
3. Delivery System (DELVTYP) is HMO (1 or 5).
4. Service is not capitated (BILLTYPE = 1).

HMO Capitated:

1. Payer is an HMO provider.
2. Coverage Type (COVTYPE) is non-Medicare (2-6).
3. Delivery System (DELVTYP) is HMO (1 or 5).
4. Service is capitated (BILLTYPE = 8).

Medicare, All Types

1, All services with Coverage Type 1 or 7.

MCDB Encounter Preprocessing
January 2006 - April 2007 Data

P160: Cigna Healthcare Mid-Atlantic Inc
Data After Initial Processing Step (enc5)
Data Completeness Summary Report

Eligible Services: 3,928,967
Services Submitted: 3,928,967

Source File: P160_enc5_dc_crunch.sas7bdat
File Date: February 28, 2008

Delivery System	Number of Recipients[1]			Number of Services[2]			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
1: HMO (Non-Medicaid, Includes Medicare)	44,614	31,705	-28.9	816,613	658,321	-19.4	63,019,719	54,665,564	-13.3
2: PPO-POS	89,911	94,663	5.3	2,072,208	2,319,725	11.9	167,966,414	177,953,823	5.9
3: PPO or Other Managed Care	28,864	26,273	-9.0	563,451	594,548	5.5	47,025,466	48,808,355	3.8
4: Indemnity Care	8,530	7,262	-14.9	300,047	298,312	-0.6	11,225,929	8,946,153	-20.3
5: HMO-POS Rider	3,090	2,527	-18.2	63,391	53,623	-15.4	5,490,436	4,490,446	-18.2
9: Payer Code=9 (Not Coded)									
Missing or Invalid Code									
Total	**171,794**	**162,430**	**-5.5**	**3,815,710**	**3,924,529**	**2.9**	**294,727,964**	**294,864,341**	**0.0**

Plan[3]	Number of Recipients[1]			Number of Services			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
Non-HMO	116,334	117,876	1.3	2,482,249	2,722,417	9.7	204,827,654	215,457,503	5.2
HMO Fee for Service	43,287	33,406	-22.8	775,411	660,641	-14.8	65,163,272	56,765,828	-12.9
HMO Capitated	7,732	5,211	-32.6	70,302	58,183	-17.2	.	.	.
Medicare, All Types									
No Plan Assigned	11,934	11,033	-7.5	487,748	483,288	-0.9	24,737,038	22,641,010	-8.5
Total	**171,794**	**162,430**	**-5.5**	**3,815,710**	**3,924,529**	**2.9**	**294,727,964**	**294,864,341**	**0.0**

Coverage Type	Number of Recipients[1]			Number of Services			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
1: Medicare Supplemental									
2: Individual Plan	6	6	0.0	248	167	-32.7	29,089	18,893	-35.1
3: Private Employer Sponsored Fully Self-Ins	126,241	122,334	-3.1	2,879,503	3,013,049	4.6	218,507,599	222,071,211	1.6
4: Private Employer Sponsored Insured	45,547	40,090	-12.0	935,959	911,313	-2.6	76,191,276	72,774,237	-4.5
5: Public Employee									
6: Comprehensive Standard Health Benefit Plan									
7: Medicare Provided by a Medicare HMO/CMS									
8: Taft Hartley Jointly Managed Trust Fund									
9: Payer Code-9 (Unknown Coverage Type)									
Missing or Invalid Code									
Total	**171,794**	**162,430**	**-5.5**	**3,815,710**	**3,924,529**	**2.9**	**294,727,964**	**294,864,341**	**0.0**

MCDB Encounter Preprocessing
January 2006 - April 2007 Data

P160: Cigna Healthcare Mid-Atlantic Inc
Data After Initial Processing Step (enc5)
Data Completeness Summary Report

Eligible Services: 3,928,967 | **Source File: P160_enc5_dc_crunch.sas7bdat**
Services Submitted: 3,928,967 | **File Date: February 28, 2008**

NOTES:
[1] Total number of recipients will be less than the sum of individual category recipient counts if some recipients receive services in more than one category.

Key to identify a unique recipient: Patient ID + Birth Year + Birth Month + Gender.

[2] The number of services passing MCDB screens. 2004 services does not include Non-MD Patients but 2005 services does include Non-MD patients.

Please see Payer-Specific MCDB Deletion Report for counts of Services failing screens.

[3] Rules for categorizing services into a PLAN:

Non-HMO

1. Payer is not an HMO provider and Coverage Type (COVTYPE) is non-Medicare (2-6) or Taft-Hartley (COVTYPE = 8).
   a. Coverage Type (COVTYPE) is non-Medicare (2-6)
   b. Coverage Type (COVTYPE) is Taft-Hartley (8).
2. Payer is an HMO provider:
   a. Delivery System (DELVTYP) is non-HMO (2-4).
   b. Coverage Type (COVTYPE) is non-Medicare (2-6)

HMO Fee for Service:

1. Payer is an HMO provider.
2. Coverage Type (COVTYPE) is non-Medicare (2-6).
3. Delivery System (DELVTYP) is HMO (1 or 5).
4. Service is not capitated (BILLTYPE = 1).

HMO Capitated:

1. Payer is an HMO provider.
2. Coverage Type (COVTYPE) is non-Medicare (2-6).
3. Delivery System (DELVTYP) is HMO (1 or 5).
4. Service is capitated (BILLTYPE = 8).

Medicare, All Types

1, All services with Coverage Type 1 or 7.

MCDB Encounter Preprocessing
January 2006 - April 2007 Data

P280: Assurant or Fortis Benefits Insurance Co
Data After Initial Processing Step (enc5)
Data Completeness Summary Report

Eligible Services: 72,393
Services Submitted: 72,393

Source File: P280_enc5_dc_crunch.sas7bdat
File Date: February 27, 2008

Delivery System	Number of Recipients[1]			Number of Services[2]			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
1: HMO (Non-Medicaid, Includes Medicare)									
2: PPO-POS									
3: PPO or Other Managed Care	2,520	3,469	37.7	33,034	39,820	20.5	1,617,148	2,162,371	33.7
4: Indemnity Care	3,788	4,176	10.2	30,488	32,286	5.9	2,082,630	1,953,059	-6.2
5: HMO-POS Rider									
9: Payer Code=9 (Not Coded)									
Missing or Invalid Code									
Total	**5,179**	**6,023**	**16.3**	**63,522**	**72,106**	**13.5**	**3,699,778**	**4,115,430**	**11.2**

Plan[3]	Number of Recipients[1]			Number of Services			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
Non-HMO	5,137	5,995	16.7	62,844	71,762	14.2	3,643,959	4,092,024	12.3
HMO Fee for Service									
HMO Capitated									
Medicare, All Types									
No Plan Assigned	42	28	-33.3	678	344	-49.3	55,819	23,406	-58.1
Total	**5,179**	**6,023**	**16.3**	**63,522**	**72,106**	**13.5**	**3,699,778**	**4,115,430**	**11.2**

Coverage Type	Number of Recipients[1]			Number of Services			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
1: Medicare Supplemental									
2: Individual Plan	5,026	5,909	17.6	60,282	69,994	16.1	3,466,997	3,989,630	15.1
3: Private Employer Sponsored Fully Self-Ins									
4: Private Employer Sponsored Insured	153	114	-25.5	3,240	2,112	-34.8	232,781	125,800	-46.0
5: Public Employee									
6: Comprehensive Standard Health Benefit Plan									
7: Medicare Provided by a Medicare HMO/CMS									
8: Taft Hartley Jointly Managed Trust Fund									
9: Payer Code-9 (Unknown Coverage Type)									
Missing or Invalid Code									
Total	**5,179**	**6,023**	**16.3**	**63,522**	**72,106**	**13.5**	**3,699,778**	**4,115,430**	**11.2**

MCDB Encounter Preprocessing
January 2006 - April 2007 Data

P280: Assurant or Fortis Benefits Insurance Co
Data After Initial Processing Step (enc5)
Data Completeness Summary Report

Eligible Services: 72,393
Services Submitted: 72,393

Source File: P280_enc5_dc_crunch.sas7bdat
File Date: February 27, 2008

NOTES:
[1] Total number of recipients will be less than the sum of individual category recipient counts if some recipients receive services in more than one category.

Key to identify a unique recipient: Patient ID + Birth Year + Birth Month + Gender.

[2] The number of services passing MCDB screens. 2004 services does not include Non-MD Patients but 2005 services does include Non-MD patients.

Please see Payer-Specific MCDB Deletion Report for counts of Services failing screens.

[3] Rules for categorizing services into a PLAN:

Non-HMO

1. Payer is not an HMO provider and Coverage Type (COVTYPE) is non-Medicare (2-6) or Taft-Hartley (COVTYPE = 8).
   a. Coverage Type (COVTYPE) is non-Medicare (2-6)
   b. Coverage Type (COVTYPE) is Taft-Hartley (8).
2. Payer is an HMO provider:
   a. Delivery System (DELVTYP) is non-HMO (2-4).
   b. Coverage Type (COVTYPE) is non-Medicare (2-6)

HMO Fee for Service:

1. Payer is an HMO provider.
2. Coverage Type (COVTYPE) is non-Medicare (2-6).
3. Delivery System (DELVTYP) is HMO (1 or 5).
4. Service is not capitated (BILLTYPE = 1).

HMO Capitated:

1. Payer is an HMO provider.
2. Coverage Type (COVTYPE) is non-Medicare (2-6).
3. Delivery System (DELVTYP) is HMO (1 or 5).
4. Service is capitated (BILLTYPE = 8).

Medicare, All Types

1, All services with Coverage Type 1 or 7.

MCDB Encounter Preprocessing
January 2006 - April 2007 Data

P320: Golden Rule Insurance Co
Data After Initial Processing Step (enc5)
Data Completeness Summary Report

Eligible Services: 146,313
Services Submitted: 146,313

Source File: P320_enc5_dc_crunch.sas7bdat
File Date: February 27, 2008

Delivery System	Number of Recipients[1]			Number of Services[2]			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
1: HMO (Non-Medicaid, Includes Medicare)									
2: PPO-POS									
3: PPO or Other Managed Care	8,805	11,222	27.5	110,416	133,419	20.8	15,480,693	18,921,171	22.2
4: Indemnity Care	3,033	2,711	-10.6	15,314	12,893	-15.8	2,134,934	2,132,801	-0.1
5: HMO-POS Rider									
9: Payer Code=9 (Not Coded)									
Missing or Invalid Code									
Total	**9,358**	**11,552**	**23.4**	**125,730**	**146,312**	**16.4**	**17,615,627**	**21,053,972**	**19.5**

Plan[3]	Number of Recipients[1]			Number of Services			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
Non-HMO	8,731	10,783	23.5	119,774	137,988	15.2	16,519,793	19,454,599	17.8
HMO Fee for Service									
HMO Capitated									
Medicare, All Types	3	2	-33.3	6	5	-16.7	2,074	985	-52.5
No Plan Assigned	628	779	24.0	5,950	8,319	39.8	1,093,760	1,598,388	46.1
Total	**9,358**	**11,552**	**23.4**	**125,730**	**146,312**	**16.4**	**17,615,627**	**21,053,972**	**19.5**

Coverage Type	Number of Recipients[1]			Number of Services			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
1: Medicare Supplemental	3	2	-33.3	6	5	-16.7	2,074	985	-52.5
2: Individual Plan	8,836	10,864	23.0	121,737	139,477	14.6	16,887,192	19,791,041	17.2
3: Private Employer Sponsored Fully Self-Ins									
4: Private Employer Sponsored Insured									
5: Public Employee									
6: Comprehensive Standard Health Benefit Plan									
7: Medicare Provided by a Medicare HMO/CMS									
8: Taft Hartley Jointly Managed Trust Fund									
9: Payer Code-9 (Unknown Coverage Type)	524	698	33.2	3,987	6,830	71.3	726,361	1,261,946	73.7
Missing or Invalid Code									
Total	**9,358**	**11,552**	**23.4**	**125,730**	**146,312**	**16.4**	**17,615,627**	**21,053,972**	**19.5**

MCDB Encounter Preprocessing
January 2006 - April 2007 Data

P320: Golden Rule Insurance Co
Data After Initial Processing Step (enc5)
Data Completeness Summary Report

Eligible Services: 146,313
Services Submitted: 146,313

Source File: P320_enc5_dc_crunch.sas7bdat
File Date: February 27, 2008

NOTES:
[1] Total number of recipients will be less than the sum of individual category recipient counts if some recipients receive services in more than one category.
Key to identify a unique recipient: Patient ID + Birth Year + Birth Month + Gender.

[2] The number of services passing MCDB screens. 2004 services does not include Non-MD Patients but 2005 services does include Non-MD patients.
Please see Payer-Specific MCDB Deletion Report for counts of Services failing screens.

[3] Rules for categorizing services into a PLAN:

Non-HMO
1. Payer is not an HMO provider and Coverage Type (COVTYPE) is non-Medicare (2-6) or Taft-Hartley (COVTYPE = 8).
   a. Coverage Type (COVTYPE) is non-Medicare (2-6)
   b. Coverage Type (COVTYPE) is Taft-Hartley (8).
2. Payer is an HMO provider:
   a. Delivery System (DELVTYP) is non-HMO (2-4).
   b. Coverage Type (COVTYPE) is non-Medicare (2-6)

HMO Fee for Service:
1. Payer is an HMO provider.
2. Coverage Type (COVTYPE) is non-Medicare (2-6).
3. Delivery System (DELVTYP) is HMO (1 or 5).
4. Service is not capitated (BILLTYPE = 1).

HMO Capitated:
1. Payer is an HMO provider.
2. Coverage Type (COVTYPE) is non-Medicare (2-6).
3. Delivery System (DELVTYP) is HMO (1 or 5).
4. Service is capitated (BILLTYPE = 8).

Medicare, All Types
1, All services with Coverage Type 1 or 7.

MCDB Encounter Preprocessing
January 2006 - April 2007 Data

P325: Graphic Arts Benefit Corporation
Data After Initial Processing Step (enc5)
Data Completeness Summary Report

Eligible Services: 123,837
Services Submitted: 123,837

Source File: P325_enc5_dc_crunch.sas7bdat
File Date: February 27, 2008

Delivery System	Number of Recipients[1]			Number of Services[2]			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
1: HMO (Non-Medicaid, Includes Medicare)									
2: PPO-POS									
3: PPO or Other Managed Care	4,373	4,663	6.6	88,800	120,797	36.0	14,092,215	15,386,910	9.2
4: Indemnity Care									
5: HMO-POS Rider									
9: Payer Code=9 (Not Coded)									
Missing or Invalid Code									
Total	**4,373**	**4,663**	**6.6**	**88,800**	**120,797**	**36.0**	**14,092,215**	**15,386,910**	**9.2**

Plan[3]	Number of Recipients[1]			Number of Services			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
Non-HMO	4,216	4,509	6.9	81,470	114,489	40.5	12,596,935	14,234,211	13.0
HMO Fee for Service									
HMO Capitated									
Medicare, All Types									
No Plan Assigned	157	154	-1.9	7,330	6,308	-13.9	1,495,280	1,152,699	-22.9
Total	**4,373**	**4,663**	**6.6**	**88,800**	**120,797**	**36.0**	**14,092,215**	**15,386,910**	**9.2**

Coverage Type	Number of Recipients[1]			Number of Services			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
1: Medicare Supplemental									
2: Individual Plan									
3: Private Employer Sponsored Fully Self-Ins									
4: Private Employer Sponsored Insured	1,532	1,704	11.2	31,332	39,849	27.2	5,269,641	5,423,241	2.9
5: Public Employee									
6: Comprehensive Standard Health Benefit Plan	2,841	2,959	4.2	57,468	80,948	40.9	8,822,574	9,963,669	12.9
7: Medicare Provided by a Medicare HMO/CMS									
8: Taft Hartley Jointly Managed Trust Fund									
9: Payer Code-9 (Unknown Coverage Type)									
Missing or Invalid Code									
Total	**4,373**	**4,663**	**6.6**	**88,800**	**120,797**	**36.0**	**14,092,215**	**15,386,910**	**9.2**

MCDB Encounter Preprocessing
January 2006 - April 2007 Data

P325: Graphic Arts Benefit Corporation
Data After Initial Processing Step (enc5)
Data Completeness Summary Report

Eligible Services: 123,837
Services Submitted: 123,837

Source File: P325_enc5_dc_crunch.sas7bdat
File Date: February 27, 2008

NOTES:
[1] Total number of recipients will be less than the sum of individual category recipient counts if some recipients receive services in more than one category.
Key to identify a unique recipient: Patient ID + Birth Year + Birth Month + Gender.

[2] The number of services passing MCDB screens. 2004 services does not include Non-MD Patients but 2005 services does include Non-MD patients.
Please see Payer-Specific MCDB Deletion Report for counts of Services failing screens.

[3] Rules for categorizing services into a PLAN:

Non-HMO
1. Payer is not an HMO provider and Coverage Type (COVTYPE) is non-Medicare (2-6) or Taft-Hartley (COVTYPE = 8).
   a. Coverage Type (COVTYPE) is non-Medicare (2-6)
   b. Coverage Type (COVTYPE) is Taft-Hartley (8).
2. Payer is an HMO provider:
   a. Delivery System (DELVTYP) is non-HMO (2-4).
   b. Coverage Type (COVTYPE) is non-Medicare (2-6)

HMO Fee for Service:
1. Payer is an HMO provider.
2. Coverage Type (COVTYPE) is non-Medicare (2-6).
3. Delivery System (DELVTYP) is HMO (1 or 5).
4. Service is not capitated (BILLTYPE = 1).

HMO Capitated:
1. Payer is an HMO provider.
2. Coverage Type (COVTYPE) is non-Medicare (2-6).
3. Delivery System (DELVTYP) is HMO (1 or 5).
4. Service is capitated (BILLTYPE = 8).

Medicare, All Types
1, All services with Coverage Type 1 or 7.

MCDB Encounter Preprocessing
January 2006 - April 2007 Data

P330: Great West Life and Annuity Insurance Co
Data After Initial Processing Step (enc5)
Data Completeness Summary Report

Eligible Services: 462,464
Services Submitted: 462,464

Source File: P330_enc5_dc_crunch.sas7bdat
File Date: February 27, 2008

Delivery System	Number of Recipients[1]			Number of Services[2]			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
1: HMO (Non-Medicaid, Includes Medicare)									
2: PPO-POS	1,936	1,922	-0.7	32,917	33,112	0.6	2,934,770	3,663,998	24.8
3: PPO or Other Managed Care	21,248	18,479	-13.0	429,592	383,377	-10.8	45,092,320	49,263,544	9.3
4: Indemnity Care	6,177	5,407	-12.5	49,155	42,880	-12.8	5,071,088	6,077,447	19.8
5: HMO-POS Rider									
9: Payer Code=9 (Not Coded)	864	469	-45.7	7,412	2,989	-59.7	995,673	656,908	-34.0
Missing or Invalid Code									
Total	**25,477**	**21,986**	**-13.7**	**519,076**	**462,358**	**-10.9**	**54,093,851**	**59,661,897**	**10.3**

Plan[3]	Number of Recipients[1]			Number of Services			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
Non-HMO	23,845	20,812	-12.7	476,843	430,922	-9.6	47,908,002	52,674,235	9.9
HMO Fee for Service									
HMO Capitated									
Medicare, All Types									
No Plan Assigned	1,632	1,174	-28.1	42,233	31,436	-25.6	6,185,849	6,987,662	13.0
Total	**25,477**	**21,986**	**-13.7**	**519,076**	**462,358**	**-10.9**	**54,093,851**	**59,661,897**	**10.3**

Coverage Type	Number of Recipients[1]			Number of Services			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
1: Medicare Supplemental									
2: Individual Plan									
3: Private Employer Sponsored Fully Self-Ins	24,478	21,408	-12.5	508,881	457,727	-10.1	52,758,598	58,815,567	11.5
4: Private Employer Sponsored Insured	177	98	-44.6	2,690	1,532	-43.0	331,570	169,976	-48.7
5: Public Employee	19	30	57.9	130	133	2.3	10,764	14,220	32.1
6: Comprehensive Standard Health Benefit Plan									
7: Medicare Provided by a Medicare HMO/CMS									
8: Taft Hartley Jointly Managed Trust Fund									
9: Payer Code-9 (Unknown Coverage Type)	853	468	-45.1	7,375	2,966	-59.8	992,919	662,134	-33.3
Missing or Invalid Code									
Total	**25,477**	**21,986**	**-13.7**	**519,076**	**462,358**	**-10.9**	**54,093,851**	**59,661,897**	**10.3**

MCDB Encounter Preprocessing
January 2006 - April 2007 Data

P330: Great West Life and Annuity Insurance Co
Data After Initial Processing Step (enc5)
Data Completeness Summary Report

Eligible Services: 462,464 | **Source File: P330_enc5_dc_crunch.sas7bdat**
Services Submitted: 462,464 | **File Date: February 27, 2008**

NOTES:
[1] Total number of recipients will be less than the sum of individual category recipient counts if some recipients receive services in more than one category.
Key to identify a unique recipient: Patient ID + Birth Year + Birth Month + Gender.

[2] The number of services passing MCDB screens. 2004 services does not include Non-MD Patients but 2005 services does include Non-MD patients.
Please see Payer-Specific MCDB Deletion Report for counts of Services failing screens.

[3] Rules for categorizing services into a PLAN:

Non-HMO
1. Payer is not an HMO provider and Coverage Type (COVTYPE) is non-Medicare (2-6) or Taft-Hartley (COVTYPE = 8).
   a. Coverage Type (COVTYPE) is non-Medicare (2-6)
   b. Coverage Type (COVTYPE) is Taft-Hartley (8).
2. Payer is an HMO provider:
   a. Delivery System (DELVTYP) is non-HMO (2-4).
   b. Coverage Type (COVTYPE) is non-Medicare (2-6)

HMO Fee for Service:
1. Payer is an HMO provider.
2. Coverage Type (COVTYPE) is non-Medicare (2-6).
3. Delivery System (DELVTYP) is HMO (1 or 5).
4. Service is not capitated (BILLTYPE = 1).

HMO Capitated:
1. Payer is an HMO provider.
2. Coverage Type (COVTYPE) is non-Medicare (2-6).
3. Delivery System (DELVTYP) is HMO (1 or 5).
4. Service is capitated (BILLTYPE = 8).

Medicare, All Types
1, All services with Coverage Type 1 or 7.

MCDB Encounter Preprocessing
January 2006 - April 2007 Data

P350: Guardian Life Insurance Co of America
Data After Initial Processing Step (enc5)
Data Completeness Summary Report

Eligible Services: 346,828
Services Submitted: 346,828

Source File: P350_enc5_dc_crunch.sas7bdat
File Date: February 27, 2008

Delivery System	Number of Recipients[1]			Number of Services[2]			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
1: HMO (Non-Medicaid, Includes Medicare)	2	.	.	9	.	.	88	.	.
2: PPO-POS									
3: PPO or Other Managed Care	20,684	18,384	-11.1	416,012	334,578	-19.6	39,190,723	27,235,318	-30.5
4: Indemnity Care	3,804	466	-87.7	37,229	9,106	-75.5	4,410,764	893,464	-79.7
5: HMO-POS Rider									
9: Payer Code=9 (Not Coded)	769	623	-19.0	4,113	3,049	-25.9	67,271	54,571	-18.9
Missing or Invalid Code									
Total	**21,696**	**18,852**	**-13.1**	**457,363**	**346,733**	**-24.2**	**43,668,846**	**28,183,353**	**-35.5**

Plan[3]	Number of Recipients[1]			Number of Services			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
Non-HMO	21,161	18,433	-12.9	436,245	335,034	-23.2	41,263,183	27,227,357	-34.0
HMO Fee for Service									
HMO Capitated									
Medicare, All Types									
No Plan Assigned	680	419	-38.4	21,118	11,699	-44.6	2,405,663	955,996	-60.3
Total	**21,696**	**18,852**	**-13.1**	**457,363**	**346,733**	**-24.2**	**43,668,846**	**28,183,353**	**-35.5**

Coverage Type	Number of Recipients[1]			Number of Services			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
1: Medicare Supplemental									
2: Individual Plan									
3: Private Employer Sponsored Fully Self-Ins	5,976	5,304	-11.2	125,999	99,273	-21.2	11,242,690	7,827,828	-30.4
4: Private Employer Sponsored Insured	15,663	12,206	-22.1	326,868	212,458	-35.0	31,690,562	17,313,099	-45.4
5: Public Employee									
6: Comprehensive Standard Health Benefit Plan	3	1,504	50033.3	13	35,002	269146.2	943	3,042,426	322532.7
7: Medicare Provided by a Medicare HMO/CMS									
8: Taft Hartley Jointly Managed Trust Fund									
9: Payer Code-9 (Unknown Coverage Type)	264	.	.	4,483	.	.	734,651	.	.
Missing or Invalid Code									
Total	**21,696**	**18,852**	**-13.1**	**457,363**	**346,733**	**-24.2**	**43,668,846**	**28,183,353**	**-35.5**

MCDB Encounter Preprocessing
January 2006 - April 2007 Data

P350: Guardian Life Insurance Co of America
Data After Initial Processing Step (enc5)
Data Completeness Summary Report

Eligible Services: 346,828
Services Submitted: 346,828

Source File: P350_enc5_dc_crunch.sas7bdat
File Date: February 27, 2008

NOTES:
[1] Total number of recipients will be less than the sum of individual category recipient counts if some recipients receive services in more than one category.
Key to identify a unique recipient: Patient ID + Birth Year + Birth Month + Gender.

[2] The number of services passing MCDB screens. 2004 services does not include Non-MD Patients but 2005 services does include Non-MD patients.
Please see Payer-Specific MCDB Deletion Report for counts of Services failing screens.

[3] Rules for categorizing services into a PLAN:

Non-HMO
1. Payer is not an HMO provider and Coverage Type (COVTYPE) is non-Medicare (2-6) or Taft-Hartley (COVTYPE = 8).
   a. Coverage Type (COVTYPE) is non-Medicare (2-6)
   b. Coverage Type (COVTYPE) is Taft-Hartley (8).
2. Payer is an HMO provider:
   a. Delivery System (DELVTYP) is non-HMO (2-4).
   b. Coverage Type (COVTYPE) is non-Medicare (2-6)

HMO Fee for Service:
1. Payer is an HMO provider.
2. Coverage Type (COVTYPE) is non-Medicare (2-6).
3. Delivery System (DELVTYP) is HMO (1 or 5).
4. Service is not capitated (BILLTYPE = 1).

HMO Capitated:
1. Payer is an HMO provider.
2. Coverage Type (COVTYPE) is non-Medicare (2-6).
3. Delivery System (DELVTYP) is HMO (1 or 5).
4. Service is capitated (BILLTYPE = 8).

Medicare, All Types
1, All services with Coverage Type 1 or 7.

MCDB Encounter Preprocessing
January 2006 - April 2007 Data

P471: Unicare Life and Health Insurance Co
Data After Initial Processing Step (enc5)
Data Completeness Summary Report

Eligible Services: 155,774
Services Submitted: 155,774

Source File: P471_enc5_dc_crunch.sas7bdat
File Date: February 27, 2008

Delivery System	Number of Recipients[1]			Number of Services[2]			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
1: HMO (Non-Medicaid, Includes Medicare)									
2: PPO-POS									
3: PPO or Other Managed Care	18,862	16,542	-12.3	454,193	154,133	-66.1	44,194,010	19,911,515	-54.9
4: Indemnity Care	330	194	-41.2	3,594	1,469	-59.1	340,003	151,604	-55.4
5: HMO-POS Rider									
9: Payer Code=9 (Not Coded)									
Missing or Invalid Code									
Total	**18,979**	**16,623**	**-12.4**	**457,787**	**155,602**	**-66.0**	**44,534,013**	**20,063,119**	**-54.9**

Plan[3]	Number of Recipients[1]			Number of Services			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
Non-HMO	18,402	16,124	-12.4	427,898	142,459	-66.7	41,871,276	18,899,957	-54.9
HMO Fee for Service									
HMO Capitated									
Medicare, All Types									
No Plan Assigned	577	499	-13.5	29,889	13,143	-56.0	2,662,737	1,163,162	-56.3
Total	**18,979**	**16,623**	**-12.4**	**457,787**	**155,602**	**-66.0**	**44,534,013**	**20,063,119**	**-54.9**

Coverage Type	Number of Recipients[1]			Number of Services			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
1: Medicare Supplemental									
2: Individual Plan									
3: Private Employer Sponsored Fully Self-Ins	2,286	3,009	31.6	53,465	31,594	-40.9	5,176,064	3,786,278	-26.9
4: Private Employer Sponsored Insured	16,884	14,375	-14.9	404,315	124,007	-69.3	39,357,740	16,276,813	-58.6
5: Public Employee	2	1	-50.0	7	1	-85.7	209	28	-86.6
6: Comprehensive Standard Health Benefit Plan									
7: Medicare Provided by a Medicare HMO/CMS									
8: Taft Hartley Jointly Managed Trust Fund									
9: Payer Code-9 (Unknown Coverage Type)									
Missing or Invalid Code									
Total	**18,979**	**16,623**	**-12.4**	**457,787**	**155,602**	**-66.0**	**44,534,013**	**20,063,119**	**-54.9**

MCDB Encounter Preprocessing
January 2006 - April 2007 Data

P471: Unicare Life and Health Insurance Co
Data After Initial Processing Step (enc5)
Data Completeness Summary Report

Eligible Services: 155,774
Services Submitted: 155,774

Source File: P471_enc5_dc_crunch.sas7bdat
File Date: February 27, 2008

NOTES:
[1] Total number of recipients will be less than the sum of individual category recipient counts if some recipients receive services in more than one category.
Key to identify a unique recipient: Patient ID + Birth Year + Birth Month + Gender.

[2] The number of services passing MCDB screens. 2004 services does not include Non-MD Patients but 2005 services does include Non-MD patients.
Please see Payer-Specific MCDB Deletion Report for counts of Services failing screens.

[3] Rules for categorizing services into a PLAN:

Non-HMO
1. Payer is not an HMO provider and Coverage Type (COVTYPE) is non-Medicare (2-6) or Taft-Hartley (COVTYPE = 8).
   a. Coverage Type (COVTYPE) is non-Medicare (2-6)
   b. Coverage Type (COVTYPE) is Taft-Hartley (8).
2. Payer is an HMO provider:
   a. Delivery System (DELVTYP) is non-HMO (2-4).
   b. Coverage Type (COVTYPE) is non-Medicare (2-6)

HMO Fee for Service:
1. Payer is an HMO provider.
2. Coverage Type (COVTYPE) is non-Medicare (2-6).
3. Delivery System (DELVTYP) is HMO (1 or 5).
4. Service is not capitated (BILLTYPE = 1).

HMO Capitated:
1. Payer is an HMO provider.
2. Coverage Type (COVTYPE) is non-Medicare (2-6).
3. Delivery System (DELVTYP) is HMO (1 or 5).
4. Service is capitated (BILLTYPE = 8).

Medicare, All Types
1, All services with Coverage Type 1 or 7.

MCDB Encounter Preprocessing
January 2006 - April 2007 Data

P480: Kaiser Foundation Health Plan of Mid Atlantic
Data After Initial Processing Step (enc5)
Data Completeness Summary Report

Eligible Services: 5,087,116
Services Submitted: 5,087,116

Source File: P480_enc5_dc_crunch.sas7bdat
File Date: February 27, 2008

Delivery System	Number of Recipients[1]			Number of Services[2]			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
1: HMO (Non-Medicaid, Includes Medicare)	203,000	247,894	22.1	2,766,022	4,946,105	78.8	85,416,058	111,507,978	30.5
2: PPO-POS	2,096	.	.	34,830	.	.	1,869,101	.	.
3: PPO or Other Managed Care	17	.	.	62	.	.	0	.	.
4: Indemnity Care									
5: HMO-POS Rider	442	7,818	1668.8	720	110,740	15280.6	.	4,893,601	.
9: Payer Code=9 (Not Coded)	10,817	7,237	-33.1	130,762	30,173	-76.9	4,568,575	2,317,393	-49.3
Missing or Invalid Code									
Total	**216,024**	**255,628**	**18.3**	**2,932,396**	**5,087,018**	**73.5**	**91,853,734**	**118,718,972**	**29.2**

Plan[3]	Number of Recipients[1]			Number of Services			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
Non-HMO	1,388	.	.	21,491	.	.	1,701,159	.	.
HMO Fee for Service	73,005	82,711	13.3	852,446	834,280	-2.1	75,818,799	103,322,021	36.3
HMO Capitated	191,376	242,924	26.9	1,728,611	4,009,037	131.9	.	.	.
Medicare, All Types	12,653	13,016	2.9	264,493	223,697	-15.4	9,653,445	11,968,035	24.0
No Plan Assigned	5,726	3,878	-32.3	65,355	20,004	-69.4	4,680,331	3,428,916	-26.7
Total	**216,024**	**255,628**	**18.3**	**2,932,396**	**5,087,018**	**73.5**	**91,853,734**	**118,718,972**	**29.2**

Coverage Type	Number of Recipients[1]			Number of Services			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
1: Medicare Supplemental									
2: Individual Plan	11,602	11,701	0.9	140,275	229,876	63.9	4,590,428	4,920,334	7.2
3: Private Employer Sponsored Fully Self-Ins									
4: Private Employer Sponsored Insured	141,238	114,736	-18.8	1,489,390	1,859,041	24.8	55,880,686	53,135,232	-4.9
5: Public Employee	55,500	105,558	90.2	801,903	2,031,988	153.4	23,999,866	49,262,964	105.3
6: Comprehensive Standard Health Benefit Plan	6,632	5,777	-12.9	91,052	106,340	16.8	3,808,883	3,760,461	-1.3
7: Medicare Provided by a Medicare HMO/CMS	15,541	18,719	20.4	409,776	792,443	93.4	3,573,871	4,470,868	25.1
8: Taft Hartley Jointly Managed Trust Fund									
9: Payer Code-9 (Unknown Coverage Type)	.	9,223	.	.	67,330	.	.	3,169,113	.
Missing or Invalid Code									
Total	**216,024**	**255,628**	**18.3**	**2,932,396**	**5,087,018**	**73.5**	**91,853,734**	**118,718,972**	**29.2**

MCDB Encounter Preprocessing
January 2006 - April 2007 Data

P480: Kaiser Foundation Health Plan of Mid Atlantic
Data After Initial Processing Step (enc5)
Data Completeness Summary Report

Eligible Services: 5,087,116 | **Source File: P480_enc5_dc_crunch.sas7bdat**
Services Submitted: 5,087,116 | **File Date: February 27, 2008**

NOTES:
[1] Total number of recipients will be less than the sum of individual category recipient counts if some recipients receive services in more than one category.
Key to identify a unique recipient: Patient ID + Birth Year + Birth Month + Gender.

[2] The number of services passing MCDB screens. 2004 services does not include Non-MD Patients but 2005 services does include Non-MD patients.
Please see Payer-Specific MCDB Deletion Report for counts of Services failing screens.

[3] Rules for categorizing services into a PLAN:

Non-HMO
1. Payer is not an HMO provider and Coverage Type (COVTYPE) is non-Medicare (2-6) or Taft-Hartley (COVTYPE = 8).
   a. Coverage Type (COVTYPE) is non-Medicare (2-6)
   b. Coverage Type (COVTYPE) is Taft-Hartley (8).
2. Payer is an HMO provider:
   a. Delivery System (DELVTYP) is non-HMO (2-4).
   b. Coverage Type (COVTYPE) is non-Medicare (2-6)

HMO Fee for Service:
1. Payer is an HMO provider.
2. Coverage Type (COVTYPE) is non-Medicare (2-6).
3. Delivery System (DELVTYP) is HMO (1 or 5).
4. Service is not capitated (BILLTYPE = 1).

HMO Capitated:
1. Payer is an HMO provider.
2. Coverage Type (COVTYPE) is non-Medicare (2-6).
3. Delivery System (DELVTYP) is HMO (1 or 5).
4. Service is capitated (BILLTYPE = 8).

Medicare, All Types
1, All services with Coverage Type 1 or 7.

MCDB Encounter Preprocessing
January 2006 - April 2007 Data

P500: Mamsi Life and Health Ins Co
Data After Initial Processing Step (enc5)
Data Completeness Summary Report

Eligible Services: 2,127,195
Services Submitted: 2,127,195

Source File: P500_enc5_dc_crunch.sas7bdat
File Date: February 27, 2008

Delivery System	Number of Recipients[1]			Number of Services[2]			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
1: HMO (Non-Medicaid, Includes Medicare)									
2: PPO-POS	95,225	76,293	-19.9	2,186,131	2,018,828	-7.7	158,054,368	156,312,567	-1.1
3: PPO or Other Managed Care									
4: Indemnity Care	189	450	138.1	2,880	14,243	394.5	292,228	1,393,182	376.7
5: HMO-POS Rider									
9: Payer Code=9 (Not Coded)									
Missing or Invalid Code									
Total	**95,403**	**76,736**	**-19.6**	**2,189,011**	**2,033,071**	**-7.1**	**158,346,596**	**157,705,749**	**-0.4**

Plan[3]	Number of Recipients[1]			Number of Services			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
Non-HMO	89,069	71,017	-20.3	1,899,462	1,757,806	-7.5	144,490,652	143,830,291	-0.5
HMO Fee for Service									
HMO Capitated	.	7	.	.	38	.	.	.	.
Medicare, All Types									
No Plan Assigned	6,334	5,719	-9.7	289,549	275,227	-4.9	13,855,944	13,875,458	0.1
Total	**95,403**	**76,736**	**-19.6**	**2,189,011**	**2,033,071**	**-7.1**	**158,346,596**	**157,705,749**	**-0.4**

Coverage Type	Number of Recipients[1]			Number of Services			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
1: Medicare Supplemental									
2: Individual Plan									
3: Private Employer Sponsored Fully Self-Ins	5,009	4,310	-14.0	119,591	122,578	2.5	8,556,200	9,189,151	7.4
4: Private Employer Sponsored Insured	25,913	24,468	-5.6	562,462	601,342	6.9	42,170,073	49,049,983	16.3
5: Public Employee	10,035	9,560	-4.7	414,508	386,973	-6.6	22,367,875	21,751,552	-2.8
6: Comprehensive Standard Health Benefit Plan	54,853	38,949	-29.0	1,092,450	922,178	-15.6	85,252,448	77,715,063	-8.8
7: Medicare Provided by a Medicare HMO/CMS									
8: Taft Hartley Jointly Managed Trust Fund									
9: Payer Code-9 (Unknown Coverage Type)									
Missing or Invalid Code									
Total	**95,403**	**76,736**	**-19.6**	**2,189,011**	**2,033,071**	**-7.1**	**158,346,596**	**157,705,749**	**-0.4**

MCDB Encounter Preprocessing
January 2006 - April 2007 Data

P500: Mamsi Life and Health Ins Co
Data After Initial Processing Step (enc5)
Data Completeness Summary Report

Eligible Services: 2,127,195
Services Submitted: 2,127,195

Source File: P500_enc5_dc_crunch.sas7bdat
File Date: February 27, 2008

NOTES:
[1] Total number of recipients will be less than the sum of individual category recipient counts if some recipients receive services in more than one category.
Key to identify a unique recipient: Patient ID + Birth Year + Birth Month + Gender.

[2] The number of services passing MCDB screens. 2004 services does not include Non-MD Patients but 2005 services does include Non-MD patients.
Please see Payer-Specific MCDB Deletion Report for counts of Services failing screens.

[3] Rules for categorizing services into a PLAN:

Non-HMO
1. Payer is not an HMO provider and Coverage Type (COVTYPE) is non-Medicare (2-6) or Taft-Hartley (COVTYPE = 8).
   a. Coverage Type (COVTYPE) is non-Medicare (2-6)
   b. Coverage Type (COVTYPE) is Taft-Hartley (8).
2. Payer is an HMO provider:
   a. Delivery System (DELVTYP) is non-HMO (2-4).
   b. Coverage Type (COVTYPE) is non-Medicare (2-6)

HMO Fee for Service:
1. Payer is an HMO provider.
2. Coverage Type (COVTYPE) is non-Medicare (2-6).
3. Delivery System (DELVTYP) is HMO (1 or 5).
4. Service is not capitated (BILLTYPE = 1).

HMO Capitated:
1. Payer is an HMO provider.
2. Coverage Type (COVTYPE) is non-Medicare (2-6).
3. Delivery System (DELVTYP) is HMO (1 or 5).
4. Service is capitated (BILLTYPE = 8).

Medicare, All Types
1, All services with Coverage Type 1 or 7.

MCDB Encounter Preprocessing
January 2006 - April 2007 Data

P510: Maryland Fidelity Insurance Co
Data After Initial Processing Step (enc5)
Data Completeness Summary Report

Eligible Services: 46,186
Services Submitted: 46,186

Source File: P510_enc5_dc_crunch.sas7bdat
File Date: February 27, 2008

Delivery System	Number of Recipients[1]			Number of Services[2]			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
1: HMO (Non-Medicaid, Includes Medicare)									
2: PPO-POS									
3: PPO or Other Managed Care	30,196	8,004	-73.5	518,525	46,135	-91.1	28,930,774	3,405,878	-88.2
4: Indemnity Care									
5: HMO-POS Rider									
9: Payer Code=9 (Not Coded)									
Missing or Invalid Code									
Total	**30,196**	**8,004**	**-73.5**	**518,525**	**46,135**	**-91.1**	**28,930,774**	**3,405,878**	**-88.2**

Plan[3]	Number of Recipients[1]			Number of Services			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
Non-HMO	29,131	7,554	-74.1	488,663	42,219	-91.4	27,114,733	3,123,319	-88.5
HMO Fee for Service									
HMO Capitated									
Medicare, All Types									
No Plan Assigned	1,065	450	-57.7	29,862	3,916	-86.9	1,816,041	282,559	-84.4
Total	**30,196**	**8,004**	**-73.5**	**518,525**	**46,135**	**-91.1**	**28,930,774**	**3,405,878**	**-88.2**

Coverage Type	Number of Recipients[1]			Number of Services			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
1: Medicare Supplemental									
2: Individual Plan									
3: Private Employer Sponsored Fully Self-Ins	7,045	932	-86.8	93,980	3,875	-95.9	5,391,236	314,904	-94.2
4: Private Employer Sponsored Insured	20,933	6,690	-68.0	390,760	39,893	-89.8	21,569,367	2,914,994	-86.5
5: Public Employee									
6: Comprehensive Standard Health Benefit Plan	2,295	389	-83.1	33,785	2,367	-93.0	1,970,171	175,980	-91.1
7: Medicare Provided by a Medicare HMO/CMS									
8: Taft Hartley Jointly Managed Trust Fund									
9: Payer Code-9 (Unknown Coverage Type)									
Missing or Invalid Code									
Total	**30,196**	**8,004**	**-73.5**	**518,525**	**46,135**	**-91.1**	**28,930,774**	**3,405,878**	**-88.2**

MCDB Encounter Preprocessing
January 2006 - April 2007 Data

P510: Maryland Fidelity Insurance Co
Data After Initial Processing Step (enc5)
Data Completeness Summary Report

Eligible Services: 46,186
Services Submitted: 46,186

Source File: P510_enc5_dc_crunch.sas7bdat
File Date: February 27, 2008

NOTES:
[1] Total number of recipients will be less than the sum of individual category recipient counts if some recipients receive services in more than one category.
Key to identify a unique recipient: Patient ID + Birth Year + Birth Month + Gender.

[2] The number of services passing MCDB screens. 2004 services does not include Non-MD Patients but 2005 services does include Non-MD patients.
Please see Payer-Specific MCDB Deletion Report for counts of Services failing screens.

[3] Rules for categorizing services into a PLAN:

Non-HMO
1. Payer is not an HMO provider and Coverage Type (COVTYPE) is non-Medicare (2-6) or Taft-Hartley (COVTYPE = 8).
   a. Coverage Type (COVTYPE) is non-Medicare (2-6)
   b. Coverage Type (COVTYPE) is Taft-Hartley (8).
2. Payer is an HMO provider:
   a. Delivery System (DELVTYP) is non-HMO (2-4).
   b. Coverage Type (COVTYPE) is non-Medicare (2-6)

HMO Fee for Service:
1. Payer is an HMO provider.
2. Coverage Type (COVTYPE) is non-Medicare (2-6).
3. Delivery System (DELVTYP) is HMO (1 or 5).
4. Service is not capitated (BILLTYPE = 1).

HMO Capitated:
1. Payer is an HMO provider.
2. Coverage Type (COVTYPE) is non-Medicare (2-6).
3. Delivery System (DELVTYP) is HMO (1 or 5).
4. Service is capitated (BILLTYPE = 8).

Medicare, All Types
1, All services with Coverage Type 1 or 7.

MCDB Encounter Preprocessing
January 2006 - April 2007 Data

P520: MD Individual Practice Association Inc
Data After Initial Processing Step (enc5)
Data Completeness Summary Report

Eligible Services: 3,857,438
Services Submitted: 3,857,438

Source File: P520_enc5_dc_crunch.sas7bdat
File Date: February 27, 2008

Delivery System	Number of Recipients[1]			Number of Services[2]			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
1: HMO (Non-Medicaid, Includes Medicare)	99,227	101,304	2.1	1,862,840	2,769,420	48.7	162,926,139	185,317,340	13.7
2: PPO-POS									
3: PPO or Other Managed Care									
4: Indemnity Care									
5: HMO-POS Rider	28,514	28,525	0.0	735,836	914,030	24.2	53,249,263	58,166,633	9.2
9: Payer Code=9 (Not Coded)									
Missing or Invalid Code									
Total	**127,686**	**129,591**	**1.5**	**2,598,676**	**3,683,450**	**41.7**	**216,175,402**	**243,483,973**	**12.6**

Plan[3]	Number of Recipients[1]			Number of Services			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
Non-HMO									
HMO Fee for Service	111,224	106,672	-4.1	2,038,661	2,114,005	3.7	196,791,276	222,882,588	13.3
HMO Capitated	53,938	102,859	90.7	258,881	1,270,025	390.6	.	.	.
Medicare, All Types									
No Plan Assigned	7,366	7,553	2.5	301,134	299,420	-0.6	19,384,126	20,601,385	6.3
Total	**127,686**	**129,591**	**1.5**	**2,598,676**	**3,683,450**	**41.7**	**216,175,402**	**243,483,973**	**12.6**

Coverage Type	Number of Recipients[1]			Number of Services			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
1: Medicare Supplemental									
2: Individual Plan									
3: Private Employer Sponsored Fully Self-Ins	514	468	-8.9	8,176	8,934	9.3	707,898	615,352	-13.1
4: Private Employer Sponsored Insured	5,771	4,922	-14.7	96,340	100,931	4.8	9,195,025	7,171,107	-22.0
5: Public Employee	121,532	124,519	2.5	2,494,160	3,573,585	43.3	206,272,479	235,697,514	14.3
6: Comprehensive Standard Health Benefit Plan									
7: Medicare Provided by a Medicare HMO/CMS									
8: Taft Hartley Jointly Managed Trust Fund									
9: Payer Code-9 (Unknown Coverage Type)									
Missing or Invalid Code									
Total	**127,686**	**129,591**	**1.5**	**2,598,676**	**3,683,450**	**41.7**	**216,175,402**	**243,483,973**	**12.6**

MCDB Encounter Preprocessing
January 2006 - April 2007 Data

P520: MD Individual Practice Association Inc
Data After Initial Processing Step (enc5)
Data Completeness Summary Report

Eligible Services: 3,857,438
Services Submitted: 3,857,438

Source File: P520_enc5_dc_crunch.sas7bdat
File Date: February 27, 2008

NOTES:
[1] Total number of recipients will be less than the sum of individual category recipient counts if some recipients receive services in more than one category.
Key to identify a unique recipient: Patient ID + Birth Year + Birth Month + Gender.

[2] The number of services passing MCDB screens. 2004 services does not include Non-MD Patients but 2005 services does include Non-MD patients.
Please see Payer-Specific MCDB Deletion Report for counts of Services failing screens.

[3] Rules for categorizing services into a PLAN:

Non-HMO
1. Payer is not an HMO provider and Coverage Type (COVTYPE) is non-Medicare (2-6) or Taft-Hartley (COVTYPE = 8).
   a. Coverage Type (COVTYPE) is non-Medicare (2-6)
   b. Coverage Type (COVTYPE) is Taft-Hartley (8).
2. Payer is an HMO provider:
   a. Delivery System (DELVTYP) is non-HMO (2-4).
   b. Coverage Type (COVTYPE) is non-Medicare (2-6)

HMO Fee for Service:
1. Payer is an HMO provider.
2. Coverage Type (COVTYPE) is non-Medicare (2-6).
3. Delivery System (DELVTYP) is HMO (1 or 5).
4. Service is not capitated (BILLTYPE = 1).

HMO Capitated:
1. Payer is an HMO provider.
2. Coverage Type (COVTYPE) is non-Medicare (2-6).
3. Delivery System (DELVTYP) is HMO (1 or 5).
4. Service is capitated (BILLTYPE = 8).

Medicare, All Types
1, All services with Coverage Type 1 or 7.

MCDB Encounter Preprocessing
January 2006 - April 2007 Data

P530: Mega Life and Health Insurance Co
Data After Initial Processing Step (enc5)
Data Completeness Summary Report

Eligible Services: 185,563
Services Submitted: 185,563

Source File: P530_enc5_dc_crunch.sas7bdat
File Date: February 27, 2008

Delivery System	Number of Recipients[1]			Number of Services[2]			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
1: HMO (Non-Medicaid, Includes Medicare)									
2: PPO-POS									
3: PPO or Other Managed Care	2,041	3,453	69.2	14,307	19,804	38.4	3,608,551	4,641,539	28.6
4: Indemnity Care	14,673	14,315	-2.4	134,037	137,986	2.9	20,574,457	22,769,190	10.7
5: HMO-POS Rider									
9: Payer Code=9 (Not Coded)	5,657	6,758	19.5	19,779	26,892	36.0	2,370,429	3,062,397	29.2
Missing or Invalid Code									
Total	**16,884**	**17,364**	**2.8**	**168,123**	**184,682**	**9.8**	**26,553,437**	**30,473,126**	**14.8**

Plan[3]	Number of Recipients[1]			Number of Services			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
Non-HMO	16,715	17,249	3.2	166,447	183,397	10.2	26,283,150	30,201,206	14.9
HMO Fee for Service									
HMO Capitated									
Medicare, All Types									
No Plan Assigned	169	115	-32.0	1,676	1,285	-23.3	270,287	271,920	0.6
Total	**16,884**	**17,364**	**2.8**	**168,123**	**184,682**	**9.8**	**26,553,437**	**30,473,126**	**14.8**

Coverage Type	Number of Recipients[1]			Number of Services			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
1: Medicare Supplemental									
2: Individual Plan	16,880	17,364	2.9	167,755	184,682	10.1	26,431,783	30,473,126	15.3
3: Private Employer Sponsored Fully Self-Ins									
4: Private Employer Sponsored Insured									
5: Public Employee									
6: Comprehensive Standard Health Benefit Plan	4	.	.	368	.	.	121,654	.	.
7: Medicare Provided by a Medicare HMO/CMS									
8: Taft Hartley Jointly Managed Trust Fund									
9: Payer Code-9 (Unknown Coverage Type)									
Missing or Invalid Code									
Total	**16,884**	**17,364**	**2.8**	**168,123**	**184,682**	**9.8**	**26,553,437**	**30,473,126**	**14.8**

MCDB Encounter Preprocessing
January 2006 - April 2007 Data

P530: Mega Life and Health Insurance Co
Data After Initial Processing Step (enc5)
Data Completeness Summary Report

Eligible Services: 185,563
Services Submitted: 185,563

Source File: P530_enc5_dc_crunch.sas7bdat
File Date: February 27, 2008

NOTES:
[1] Total number of recipients will be less than the sum of individual category recipient counts if some recipients receive services in more than one category.

Key to identify a unique recipient: Patient ID + Birth Year + Birth Month + Gender.

[2] The number of services passing MCDB screens. 2004 services does not include Non-MD Patients but 2005 services does include Non-MD patients.

Please see Payer-Specific MCDB Deletion Report for counts of Services failing screens.

[3] Rules for categorizing services into a PLAN:

Non-HMO

1. Payer is not an HMO provider and Coverage Type (COVTYPE) is non-Medicare (2-6) or Taft-Hartley (COVTYPE = 8).
   a. Coverage Type (COVTYPE) is non-Medicare (2-6)
   b. Coverage Type (COVTYPE) is Taft-Hartley (8).
2. Payer is an HMO provider:
   a. Delivery System (DELVTYP) is non-HMO (2-4).
   b. Coverage Type (COVTYPE) is non-Medicare (2-6)

HMO Fee for Service:

1. Payer is an HMO provider.
2. Coverage Type (COVTYPE) is non-Medicare (2-6).
3. Delivery System (DELVTYP) is HMO (1 or 5).
4. Service is not capitated (BILLTYPE = 1).

HMO Capitated:

1. Payer is an HMO provider.
2. Coverage Type (COVTYPE) is non-Medicare (2-6).
3. Delivery System (DELVTYP) is HMO (1 or 5).
4. Service is capitated (BILLTYPE = 8).

Medicare, All Types

1, All services with Coverage Type 1 or 7.

MCDB Encounter Preprocessing
January 2006 - April 2007 Data

P620: Optimum Choice Inc
Data After Initial Processing Step (enc5)
Data Completeness Summary Report

Eligible Services: 6,874,787
Services Submitted: 6,874,787

Source File: P620_enc5_dc_crunch.sas7bdat
File Date: February 27, 2008

Delivery System	Number of Recipients[1]			Number of Services[2]			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
1: HMO (Non-Medicaid, Includes Medicare)	248,494	237,029	-4.6	4,286,547	5,731,149	33.7	369,265,531	389,378,275	5.4
2: PPO-POS									
3: PPO or Other Managed Care									
4: Indemnity Care									
5: HMO-POS Rider	35,215	33,689	-4.3	660,122	812,260	23.0	57,487,027	55,875,746	-2.8
9: Payer Code=9 (Not Coded)									
Missing or Invalid Code									
Total	**282,447**	**269,088**	**-4.7**	**4,946,669**	**6,543,409**	**32.3**	**426,752,558**	**445,254,021**	**4.3**

Plan[3]	Number of Recipients[1]			Number of Services			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
Non-HMO									
HMO Fee for Service	244,611	217,878	-10.9	4,019,394	3,870,916	-3.7	400,816,563	419,278,558	4.6
HMO Capitated	141,919	208,281	46.8	589,030	2,344,398	298.0	.	.	.
Medicare, All Types	.	6	.	.	59	.	.	3,179	.
No Plan Assigned	10,245	11,481	12.1	338,245	328,036	-3.0	25,935,995	25,972,284	0.1
Total	**282,447**	**269,088**	**-4.7**	**4,946,669**	**6,543,409**	**32.3**	**426,752,558**	**445,254,021**	**4.3**

Coverage Type	Number of Recipients[1]			Number of Services			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
1: Medicare Supplemental	.	13	.	.	215	.	.	5,756	.
2: Individual Plan	8,971	8,748	-2.5	136,921	203,654	48.7	11,363,050	12,673,922	11.5
3: Private Employer Sponsored Fully Self-Ins	48,659	41,777	-14.1	873,093	1,002,969	14.9	75,789,987	70,877,281	-6.5
4: Private Employer Sponsored Insured	91,727	89,021	-3.0	1,419,060	1,898,158	33.8	129,709,005	132,634,010	2.3
5: Public Employee	73,941	77,686	5.1	1,510,978	2,229,263	47.5	121,744,771	144,277,974	18.5
6: Comprehensive Standard Health Benefit Plan	61,177	53,996	-11.7	1,006,617	1,209,150	20.1	88,145,745	84,785,078	-3.8
7: Medicare Provided by a Medicare HMO/CMS									
8: Taft Hartley Jointly Managed Trust Fund									
9: Payer Code-9 (Unknown Coverage Type)									
Missing or Invalid Code									
Total	**282,447**	**269,088**	**-4.7**	**4,946,669**	**6,543,409**	**32.3**	**426,752,558**	**445,254,021**	**4.3**

MCDB Encounter Preprocessing
January 2006 - April 2007 Data

P620: Optimum Choice Inc
Data After Initial Processing Step (enc5)
Data Completeness Summary Report

Eligible Services: 6,874,787
Services Submitted: 6,874,787

Source File: P620_enc5_dc_crunch.sas7bdat
File Date: February 27, 2008

NOTES:
[1] Total number of recipients will be less than the sum of individual category recipient counts if some recipients receive services in more than one category.
Key to identify a unique recipient: Patient ID + Birth Year + Birth Month + Gender.

[2] The number of services passing MCDB screens. 2004 services does not include Non-MD Patients but 2005 services does include Non-MD patients.
Please see Payer-Specific MCDB Deletion Report for counts of Services failing screens.

[3] Rules for categorizing services into a PLAN:

Non-HMO
1. Payer is not an HMO provider and Coverage Type (COVTYPE) is non-Medicare (2-6) or Taft-Hartley (COVTYPE = 8).
   a. Coverage Type (COVTYPE) is non-Medicare (2-6)
   b. Coverage Type (COVTYPE) is Taft-Hartley (8).
2. Payer is an HMO provider:
   a. Delivery System (DELVTYP) is non-HMO (2-4).
   b. Coverage Type (COVTYPE) is non-Medicare (2-6)

HMO Fee for Service:
1. Payer is an HMO provider.
2. Coverage Type (COVTYPE) is non-Medicare (2-6).
3. Delivery System (DELVTYP) is HMO (1 or 5).
4. Service is not capitated (BILLTYPE = 1).

HMO Capitated:
1. Payer is an HMO provider.
2. Coverage Type (COVTYPE) is non-Medicare (2-6).
3. Delivery System (DELVTYP) is HMO (1 or 5).
4. Service is capitated (BILLTYPE = 8).

Medicare, All Types
1, All services with Coverage Type 1 or 7.

MCDB Encounter Preprocessing
January 2006 - April 2007 Data

P680: Principal Health Care of Delaware Inc
Data After Initial Processing Step (enc5)
Data Completeness Summary Report

Eligible Services: 1,300,757
Services Submitted: 1,300,757

Source File: P680_enc5_dc_crunch.sas7bdat
File Date: February 27, 2008

Delivery System	Number of Recipients[1]			Number of Services[2]			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
1: HMO (Non-Medicaid, Includes Medicare)	31,202	51,973	66.6	561,416	1,024,963	82.6	26,148,545	51,353,810	96.4
2: PPO-POS									
3: PPO or Other Managed Care	283	757	167.5	5,099	14,053	175.6	192,597	639,192	231.9
4: Indemnity Care									
5: HMO-POS Rider	7,439	12,680	70.5	140,757	261,659	85.9	6,399,461	13,001,702	103.2
9: Payer Code=9 (Not Coded)									
Missing or Invalid Code									
Total	**38,805**	**64,713**	**66.8**	**707,272**	**1,300,675**	**83.9**	**32,740,603**	**64,994,704**	**98.5**

Plan[3]	Number of Recipients[1]			Number of Services			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
Non-HMO	258	706	173.6	2,778	9,613	246.0	177,681	639,192	259.7
HMO Fee for Service	36,353	62,941	73.1	394,982	841,929	113.2	30,603,668	64,355,512	110.3
HMO Capitated	17,402	33,346	91.6	278,268	449,133	61.4	.	.	.
Medicare, All Types									
No Plan Assigned	1,417	.	.	31,244	.	.	1,959,254	.	.
Total	**38,805**	**64,713**	**66.8**	**707,272**	**1,300,675**	**83.9**	**32,740,603**	**64,994,704**	**98.5**

Coverage Type	Number of Recipients[1]			Number of Services			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
1: Medicare Supplemental									
2: Individual Plan									
3: Private Employer Sponsored Fully Self-Ins	2,982	2,999	0.6	75,074	73,363	-2.3	3,575,717	3,854,626	7.8
4: Private Employer Sponsored Insured	16,688	19,693	18.0	351,234	409,912	16.7	16,157,085	20,314,481	25.7
5: Public Employee	54	933	1627.8	717	15,423	2051.0	38,120	898,514	2257.1
6: Comprehensive Standard Health Benefit Plan	19,081	41,516	117.6	280,247	801,977	186.2	12,969,681	39,927,083	207.8
7: Medicare Provided by a Medicare HMO/CMS									
8: Taft Hartley Jointly Managed Trust Fund									
9: Payer Code-9 (Unknown Coverage Type)									
Missing or Invalid Code									
Total	**38,805**	**64,713**	**66.8**	**707,272**	**1,300,675**	**83.9**	**32,740,603**	**64,994,704**	**98.5**

MCDB Encounter Preprocessing
January 2006 - April 2007 Data

P680: Principal Health Care of Delaware Inc
Data After Initial Processing Step (enc5)
Data Completeness Summary Report

Eligible Services: 1,300,757
Services Submitted: 1,300,757

Source File: P680_enc5_dc_crunch.sas7bdat
File Date: February 27, 2008

NOTES:
[1] Total number of recipients will be less than the sum of individual category recipient counts if some recipients receive services in more than one category.
Key to identify a unique recipient: Patient ID + Birth Year + Birth Month + Gender.

[2] The number of services passing MCDB screens. 2004 services does not include Non-MD Patients but 2005 services does include Non-MD patients.
Please see Payer-Specific MCDB Deletion Report for counts of Services failing screens.

[3] Rules for categorizing services into a PLAN:

Non-HMO
1. Payer is not an HMO provider and Coverage Type (COVTYPE) is non-Medicare (2-6) or Taft-Hartley (COVTYPE = 8).
   a. Coverage Type (COVTYPE) is non-Medicare (2-6)
   b. Coverage Type (COVTYPE) is Taft-Hartley (8).
2. Payer is an HMO provider:
   a. Delivery System (DELVTYP) is non-HMO (2-4).
   b. Coverage Type (COVTYPE) is non-Medicare (2-6)

HMO Fee for Service:
1. Payer is an HMO provider.
2. Coverage Type (COVTYPE) is non-Medicare (2-6).
3. Delivery System (DELVTYP) is HMO (1 or 5).
4. Service is not capitated (BILLTYPE = 1).

HMO Capitated:
1. Payer is an HMO provider.
2. Coverage Type (COVTYPE) is non-Medicare (2-6).
3. Delivery System (DELVTYP) is HMO (1 or 5).
4. Service is capitated (BILLTYPE = 8).

Medicare, All Types
1, All services with Coverage Type 1 or 7.

MCDB Encounter Preprocessing
January 2006 - April 2007 Data

P760: State Farm Mutual Auto Insurance
Data After Initial Processing Step (enc5)
Data Completeness Summary Report

Eligible Services: 65,462
Services Submitted: 65,462

Source File: P760_enc5_dc_crunch.sas7bdat
File Date: February 27, 2008

Delivery System	Number of Recipients[1]			Number of Services[2]			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
1: HMO (Non-Medicaid, Includes Medicare)									
2: PPO-POS									
3: PPO or Other Managed Care	2,247	1,085	-51.7	45,489	24,774	-45.5	5,221,238	3,094,793	-40.7
4: Indemnity Care	943	895	-5.1	34,977	40,676	16.3	7,945,424	8,937,144	12.5
5: HMO-POS Rider									
9: Payer Code=9 (Not Coded)									
Missing or Invalid Code									
Total	**3,189**	**1,980**	**-37.9**	**80,466**	**65,450**	**-18.7**	**13,166,662**	**12,031,937**	**-8.6**

Plan[3]	Number of Recipients[1]			Number of Services			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
Non-HMO	1,822	839	-54.0	32,188	15,559	-51.7	2,903,957	1,199,661	-58.7
HMO Fee for Service									
HMO Capitated									
Medicare, All Types	800	759	-5.1	27,931	32,561	16.6	6,248,481	7,028,880	12.5
No Plan Assigned	567	382	-32.6	20,347	17,330	-14.8	4,014,224	3,803,396	-5.3
Total	**3,189**	**1,980**	**-37.9**	**80,466**	**65,450**	**-18.7**	**13,166,662**	**12,031,937**	**-8.6**

Coverage Type	Number of Recipients[1]			Number of Services			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
1: Medicare Supplemental	903	862	-4.5	34,246	40,318	17.7	7,637,034	8,859,393	16.0
2: Individual Plan	40	35	-12.5	736	380	-48.4	308,582	77,940	-74.7
3: Private Employer Sponsored Fully Self-Ins									
4: Private Employer Sponsored Insured	2,246	1,083	-51.8	45,484	24,752	-45.6	5,221,046	3,094,604	-40.7
5: Public Employee									
6: Comprehensive Standard Health Benefit Plan									
7: Medicare Provided by a Medicare HMO/CMS									
8: Taft Hartley Jointly Managed Trust Fund									
9: Payer Code-9 (Unknown Coverage Type)									
Missing or Invalid Code									
Total	**3,189**	**1,980**	**-37.9**	**80,466**	**65,450**	**-18.7**	**13,166,662**	**12,031,937**	**-8.6**

MCDB Encounter Preprocessing
January 2006 - April 2007 Data

P760: State Farm Mutual Auto Insurance
Data After Initial Processing Step (enc5)
Data Completeness Summary Report

Eligible Services: 65,462
Services Submitted: 65,462

Source File: P760_enc5_dc_crunch.sas7bdat
File Date: February 27, 2008

NOTES:
[1] Total number of recipients will be less than the sum of individual category recipient counts if some recipients receive services in more than one category.
Key to identify a unique recipient: Patient ID + Birth Year + Birth Month + Gender.

[2] The number of services passing MCDB screens. 2004 services does not include Non-MD Patients but 2005 services does include Non-MD patients.
Please see Payer-Specific MCDB Deletion Report for counts of Services failing screens.

[3] Rules for categorizing services into a PLAN:

Non-HMO
1. Payer is not an HMO provider and Coverage Type (COVTYPE) is non-Medicare (2-6) or Taft-Hartley (COVTYPE = 8).
   a. Coverage Type (COVTYPE) is non-Medicare (2-6)
   b. Coverage Type (COVTYPE) is Taft-Hartley (8).
2. Payer is an HMO provider:
   a. Delivery System (DELVTYP) is non-HMO (2-4).
   b. Coverage Type (COVTYPE) is non-Medicare (2-6)

HMO Fee for Service:
1. Payer is an HMO provider.
2. Coverage Type (COVTYPE) is non-Medicare (2-6).
3. Delivery System (DELVTYP) is HMO (1 or 5).
4. Service is not capitated (BILLTYPE = 1).

HMO Capitated:
1. Payer is an HMO provider.
2. Coverage Type (COVTYPE) is non-Medicare (2-6).
3. Delivery System (DELVTYP) is HMO (1 or 5).
4. Service is capitated (BILLTYPE = 8).

Medicare, All Types
1, All services with Coverage Type 1 or 7.

MCDB Encounter Preprocessing
January 2006 - April 2007 Data

P820: United Healthcare
Data After Initial Processing Step (enc5)
Data Completeness Summary Report

Eligible Services: 6,985,303
Services Submitted: 6,985,303

Source File: P820_enc5_dc_crunch.sas7bdat
File Date: February 27, 2008

Delivery System	Number of Recipients[1]			Number of Services[2]			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
1: HMO (Non-Medicaid, Includes Medicare)									
2: PPO-POS	132,854	177,210	33.4	2,810,695	4,052,763	44.2	230,606,869	333,712,247	44.7
3: PPO or Other Managed Care	87,877	80,319	-8.6	1,889,010	1,835,104	-2.9	144,952,417	142,802,862	-1.5
4: Indemnity Care	26,006	22,337	-14.1	1,108,329	1,091,671	-1.5	41,854,891	39,599,717	-5.4
5: HMO-POS Rider									
9: Payer Code=9 (Not Coded)	375	143	-61.9	2,418	772	-68.1	335,593	169,940	-49.4
Missing or Invalid Code									
Total	**234,296**	**272,243**	**16.2**	**5,810,452**	**6,980,310**	**20.1**	**417,749,770**	**516,284,766**	**23.6**

Plan[3]	Number of Recipients[1]			Number of Services			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
Non-HMO	199,474	235,635	18.1	4,270,356	5,258,982	23.2	351,317,397	440,914,713	25.5
HMO Fee for Service									
HMO Capitated	2,348	3,728	58.8	27,463	45,018	63.9	.	.	.
Medicare, All Types	30,482	31,247	2.5	1,308,657	1,442,943	10.3	48,376,202	54,409,334	12.5
No Plan Assigned	5,389	6,579	22.1	203,976	233,367	14.4	18,056,171	20,960,719	16.1
Total	**234,296**	**272,243**	**16.2**	**5,810,452**	**6,980,310**	**20.1**	**417,749,770**	**516,284,766**	**23.6**

Coverage Type	Number of Recipients[1]			Number of Services			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
1: Medicare Supplemental	31,705	32,852	3.6	1,362,842	1,512,742	11.0	51,668,475	59,115,355	14.4
2: Individual Plan									
3: Private Employer Sponsored Fully Self-Ins	140,835	156,177	10.9	3,082,770	3,381,516	9.7	255,674,333	284,826,945	11.4
4: Private Employer Sponsored Insured	37,693	56,294	49.3	616,005	1,130,192	83.5	46,368,585	90,864,907	96.0
5: Public Employee	25,675	28,837	12.3	743,644	955,843	28.5	63,586,756	81,476,741	28.1
6: Comprehensive Standard Health Benefit Plan	208	2	-99.0	5,191	17	-99.7	451,621	818	-99.8
7: Medicare Provided by a Medicare HMO/CMS									
8: Taft Hartley Jointly Managed Trust Fund									
9: Payer Code-9 (Unknown Coverage Type)									
Missing or Invalid Code									
Total	**234,296**	**272,243**	**16.2**	**5,810,452**	**6,980,310**	**20.1**	**417,749,770**	**516,284,766**	**23.6**

MCDB Encounter Preprocessing
January 2006 - April 2007 Data

P820: United Healthcare
Data After Initial Processing Step (enc5)
Data Completeness Summary Report

Eligible Services: 6,985,303
Services Submitted: 6,985,303

Source File: P820_enc5_dc_crunch.sas7bdat
File Date: February 27, 2008

NOTES:

[1] Total number of recipients will be less than the sum of individual category recipient counts if some recipients receive services in more than one category.

Key to identify a unique recipient: Patient ID + Birth Year + Birth Month + Gender.

[2] The number of services passing MCDB screens. 2004 services does not include Non-MD Patients but 2005 services does include Non-MD patients.

Please see Payer-Specific MCDB Deletion Report for counts of Services failing screens.

[3] Rules for categorizing services into a PLAN:

Non-HMO

1. Payer is not an HMO provider and Coverage Type (COVTYPE) is non-Medicare (2-6) or Taft-Hartley (COVTYPE = 8).
   a. Coverage Type (COVTYPE) is non-Medicare (2-6)
   b. Coverage Type (COVTYPE) is Taft-Hartley (8).
2. Payer is an HMO provider:
   a. Delivery System (DELVTYP) is non-HMO (2-4).
   b. Coverage Type (COVTYPE) is non-Medicare (2-6)

HMO Fee for Service:

1. Payer is an HMO provider.
2. Coverage Type (COVTYPE) is non-Medicare (2-6).
3. Delivery System (DELVTYP) is HMO (1 or 5).
4. Service is not capitated (BILLTYPE = 1).

HMO Capitated:

1. Payer is an HMO provider.
2. Coverage Type (COVTYPE) is non-Medicare (2-6).
3. Delivery System (DELVTYP) is HMO (1 or 5).
4. Service is capitated (BILLTYPE = 8).

Medicare, All Types

1, All services with Coverage Type 1 or 7.

MCDB Encounter Preprocessing
January 2006 - April 2007 Data

P830: Trustmark Insurance Co
Data After Initial Processing Step (enc5)
Data Completeness Summary Report

Eligible Services: 5,877
Services Submitted: 5,877

Source File: P830_enc5_dc_crunch.sas7bdat
File Date: February 27, 2008

Delivery System	Number of Recipients[1]			Number of Services[2]			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
1: HMO (Non-Medicaid, Includes Medicare)									
2: PPO-POS	254	126	-50.4	4,934	1,385	-71.9	359,723	122,820	-65.9
3: PPO or Other Managed Care	304	188	-38.2	6,413	3,534	-44.9	473,334	297,306	-37.2
4: Indemnity Care	49	50	2.0	1,460	950	-34.9	173,942	100,419	-42.3
5: HMO-POS Rider									
9: Payer Code=9 (Not Coded)									
Missing or Invalid Code									
Total	**445**	**299**	**-32.8**	**12,807**	**5,869**	**-54.2**	**1,006,999**	**520,545**	**-48.3**

Plan[3]	Number of Recipients[1]			Number of Services			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
Non-HMO	374	239	-36.1	9,184	3,907	-57.5	711,029	379,311	-46.7
HMO Fee for Service									
HMO Capitated									
Medicare, All Types									
No Plan Assigned	71	60	-15.5	3,623	1,962	-45.8	295,970	141,234	-52.3
Total	**445**	**299**	**-32.8**	**12,807**	**5,869**	**-54.2**	**1,006,999**	**520,545**	**-48.3**

Coverage Type	Number of Recipients[1]			Number of Services			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
1: Medicare Supplemental									
2: Individual Plan	71	58	-18.3	3,142	1,950	-37.9	282,676	164,774	-41.7
3: Private Employer Sponsored Fully Self-Ins	216	120	-44.4	5,127	1,176	-77.1	286,232	77,174	-73.0
4: Private Employer Sponsored Insured	162	123	-24.1	4,538	2,743	-39.6	438,091	278,597	-36.4
5: Public Employee									
6: Comprehensive Standard Health Benefit Plan									
7: Medicare Provided by a Medicare HMO/CMS									
8: Taft Hartley Jointly Managed Trust Fund									
9: Payer Code-9 (Unknown Coverage Type)									
Missing or Invalid Code									
Total	**445**	**299**	**-32.8**	**12,807**	**5,869**	**-54.2**	**1,006,999**	**520,545**	**-48.3**

MCDB Encounter Preprocessing
January 2006 - April 2007 Data

P830: Trustmark Insurance Co
Data After Initial Processing Step (enc5)
Data Completeness Summary Report

Eligible Services: 5,877
Services Submitted: 5,877

Source File: P830_enc5_dc_crunch.sas7bdat
File Date: February 27, 2008

NOTES:
[1] Total number of recipients will be less than the sum of individual category recipient counts if some recipients receive services in more than one category.

Key to identify a unique recipient: Patient ID + Birth Year + Birth Month + Gender.

[2] The number of services passing MCDB screens. 2004 services does not include Non-MD Patients but 2005 services does include Non-MD patients.

Please see Payer-Specific MCDB Deletion Report for counts of Services failing screens.

[3] Rules for categorizing services into a PLAN:

Non-HMO

1. Payer is not an HMO provider and Coverage Type (COVTYPE) is non-Medicare (2-6) or Taft-Hartley (COVTYPE = 8).
   a. Coverage Type (COVTYPE) is non-Medicare (2-6)
   b. Coverage Type (COVTYPE) is Taft-Hartley (8).
2. Payer is an HMO provider:
   a. Delivery System (DELVTYP) is non-HMO (2-4).
   b. Coverage Type (COVTYPE) is non-Medicare (2-6)

HMO Fee for Service:

1. Payer is an HMO provider.
2. Coverage Type (COVTYPE) is non-Medicare (2-6).
3. Delivery System (DELVTYP) is HMO (1 or 5).
4. Service is not capitated (BILLTYPE = 1).

HMO Capitated:

1. Payer is an HMO provider.
2. Coverage Type (COVTYPE) is non-Medicare (2-6).
3. Delivery System (DELVTYP) is HMO (1 or 5).
4. Service is capitated (BILLTYPE = 8).

Medicare, All Types

1, All services with Coverage Type 1 or 7.

MCDB Encounter Preprocessing
January 2006 - April 2007 Data

P850: Union Labor Life Insurance Co
Data After Initial Processing Step (enc5)
Data Completeness Summary Report

Eligible Services: 106,508
Services Submitted: 106,508

Source File: P850_enc5_dc_crunch.sas7bdat
File Date: February 27, 2008

Delivery System	Number of Recipients[1]			Number of Services[2]			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
1: HMO (Non-Medicaid, Includes Medicare)									
2: PPO-POS									
3: PPO or Other Managed Care	3,900	3,649	-6.4	127,021	93,894	-26.1	9,397,032	9,059,054	-3.6
4: Indemnity Care									
5: HMO-POS Rider									
9: Payer Code=9 (Not Coded)									
Missing or Invalid Code									
Total	**3,900**	**3,649**	**-6.4**	**127,021**	**93,894**	**-26.1**	**9,397,032**	**9,059,054**	**-3.6**

Plan[3]	Number of Recipients[1]			Number of Services			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
Non-HMO	3,308	3,104	-6.2	98,178	74,100	-24.5	8,118,153	7,923,015	-2.4
HMO Fee for Service									
HMO Capitated									
Medicare, All Types									
No Plan Assigned	592	545	-7.9	28,843	19,794	-31.4	1,278,879	1,136,039	-11.2
Total	**3,900**	**3,649**	**-6.4**	**127,021**	**93,894**	**-26.1**	**9,397,032**	**9,059,054**	**-3.6**

Coverage Type	Number of Recipients[1]			Number of Services			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
1: Medicare Supplemental									
2: Individual Plan									
3: Private Employer Sponsored Fully Self-Ins									
4: Private Employer Sponsored Insured									
5: Public Employee									
6: Comprehensive Standard Health Benefit Plan									
7: Medicare Provided by a Medicare HMO/CMS									
8: Taft Hartley Jointly Managed Trust Fund	3,900	3,649	-6.4	127,021	93,894	-26.1	9,397,032	9,059,054	-3.6
9: Payer Code-9 (Unknown Coverage Type)									
Missing or Invalid Code									
Total	**3,900**	**3,649**	**-6.4**	**127,021**	**93,894**	**-26.1**	**9,397,032**	**9,059,054**	**-3.6**

MCDB Encounter Preprocessing
January 2006 - April 2007 Data

P850: Union Labor Life Insurance Co
Data After Initial Processing Step (enc5)
Data Completeness Summary Report

Eligible Services: 106,508
Services Submitted: 106,508

Source File: P850_enc5_dc_crunch.sas7bdat
File Date: February 27, 2008

NOTES:
[1] Total number of recipients will be less than the sum of individual category recipient counts if some recipients receive services in more than one category.

Key to identify a unique recipient: Patient ID + Birth Year + Birth Month + Gender.

[2] The number of services passing MCDB screens. 2004 services does not include Non-MD Patients but 2005 services does include Non-MD patients.

Please see Payer-Specific MCDB Deletion Report for counts of Services failing screens.

[3] Rules for categorizing services into a PLAN:

Non-HMO

1. Payer is not an HMO provider and Coverage Type (COVTYPE) is non-Medicare (2-6) or Taft-Hartley (COVTYPE = 8).
   a. Coverage Type (COVTYPE) is non-Medicare (2-6)
   b. Coverage Type (COVTYPE) is Taft-Hartley (8).
2. Payer is an HMO provider:
   a. Delivery System (DELVTYP) is non-HMO (2-4).
   b. Coverage Type (COVTYPE) is non-Medicare (2-6)

HMO Fee for Service:

1. Payer is an HMO provider.
2. Coverage Type (COVTYPE) is non-Medicare (2-6).
3. Delivery System (DELVTYP) is HMO (1 or 5).
4. Service is not capitated (BILLTYPE = 1).

HMO Capitated:

1. Payer is an HMO provider.
2. Coverage Type (COVTYPE) is non-Medicare (2-6).
3. Delivery System (DELVTYP) is HMO (1 or 5).
4. Service is capitated (BILLTYPE = 8).

Medicare, All Types

1, All services with Coverage Type 1 or 7.

MCDB Encounter Preprocessing
January 2006 - April 2007 Data

P870: United Healthcare of Mid Atlantic Inc
Data After Initial Processing Step (enc5)
Data Completeness Summary Report

Eligible Services: 163,232
Services Submitted: 163,232

Source File: P870_enc5_dc_crunch.sas7bdat
File Date: February 27, 2008

Delivery System	Number of Recipients[1]			Number of Services[2]			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
1: HMO (Non-Medicaid, Includes Medicare)	10,808	8,425	-22.0	203,779	163,180	-19.9	16,038,769	13,470,547	-16.0
2: PPO-POS									
3: PPO or Other Managed Care									
4: Indemnity Care									
5: HMO-POS Rider									
9: Payer Code=9 (Not Coded)									
Missing or Invalid Code									
Total	**10,808**	**8,425**	**-22.0**	**203,779**	**163,180**	**-19.9**	**16,038,769**	**13,470,547**	**-16.0**

Plan[3]	Number of Recipients[1]			Number of Services			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
Non-HMO									
HMO Fee for Service	10,551	8,205	-22.2	186,856	151,232	-19.1	14,959,259	12,946,173	-13.5
HMO Capitated	640	495	-22.7	8,081	6,434	-20.4	.	.	.
Medicare, All Types	23	13	-43.5	723	169	-76.6	111,011	7,583	-93.2
No Plan Assigned	192	176	-8.3	8,119	5,345	-34.2	968,499	516,791	-46.6
Total	**10,808**	**8,425**	**-22.0**	**203,779**	**163,180**	**-19.9**	**16,038,769**	**13,470,547**	**-16.0**

Coverage Type	Number of Recipients[1]			Number of Services			Total Payment		
	2005	2006	% Change	2005	2006	% Change	2005	2006	% Change
1: Medicare Supplemental	34	29	-14.7	1,185	420	-64.6	246,293	17,599	-92.9
2: Individual Plan									
3: Private Employer Sponsored Fully Self-Ins									
4: Private Employer Sponsored Insured	10,790	8,412	-22.0	202,594	162,760	-19.7	15,792,476	13,452,948	-14.8
5: Public Employee									
6: Comprehensive Standard Health Benefit Plan									
7: Medicare Provided by a Medicare HMO/CMS									
8: Taft Hartley Jointly Managed Trust Fund									
9: Payer Code-9 (Unknown Coverage Type)									
Missing or Invalid Code									
Total	**10,808**	**8,425**	**-22.0**	**203,779**	**163,180**	**-19.9**	**16,038,769**	**13,470,547**	**-16.0**

MCDB Encounter Preprocessing
January 2006 - April 2007 Data

P870: United Healthcare of Mid Atlantic Inc
Data After Initial Processing Step (enc5)
Data Completeness Summary Report

Eligible Services: 163,232 | **Source File: P870_enc5_dc_crunch.sas7bdat**
Services Submitted: 163,232 | **File Date: February 27, 2008**

NOTES:
[1] Total number of recipients will be less than the sum of individual category recipient counts if some recipients receive services in more than one category.

Key to identify a unique recipient: Patient ID + Birth Year + Birth Month + Gender.

[2] The number of services passing MCDB screens. 2004 services does not include Non-MD Patients but 2005 services does include Non-MD patients.

Please see Payer-Specific MCDB Deletion Report for counts of Services failing screens.

[3] Rules for categorizing services into a PLAN:

Non-HMO

1. Payer is not an HMO provider and Coverage Type (COVTYPE) is non-Medicare (2-6) or Taft-Hartley (COVTYPE = 8).
   a. Coverage Type (COVTYPE) is non-Medicare (2-6)
   b. Coverage Type (COVTYPE) is Taft-Hartley (8).
2. Payer is an HMO provider:
   a. Delivery System (DELVTYP) is non-HMO (2-4).
   b. Coverage Type (COVTYPE) is non-Medicare (2-6)

HMO Fee for Service:

1. Payer is an HMO provider.
2. Coverage Type (COVTYPE) is non-Medicare (2-6).
3. Delivery System (DELVTYP) is HMO (1 or 5).
4. Service is not capitated (BILLTYPE = 1).

HMO Capitated:

1. Payer is an HMO provider.
2. Coverage Type (COVTYPE) is non-Medicare (2-6).
3. Delivery System (DELVTYP) is HMO (1 or 5).
4. Service is capitated (BILLTYPE = 8).

Medicare, All Types

1, All services with Coverage Type 1 or 7.

Appendix E

Medical Services File Contents and Codebook

2006 Maryland Medical Care Database
MCDB Encounter File

PRV06_V2.SAS7BDAT

Contents Output

Data Set Name	IN1.PRV06_V2	**Observations**	85931243
Member Type	DATA	**Variables**	101
Engine	V9	**Indexes**	0
Created	Sunday, March 16, 2008 03:32:26 PM	**Observation Length**	432
Last Modified	Sunday, March 16, 2008 03:32:26 PM	**Deleted Observations**	0
Protection		**Compressed**	BINARY
Data Set Type		**Reuse Space**	NO
Label		**Point to Observations**	YES
	WINDOWS_32	**Sorted**	NO
Encoding	wlatin1 Western (Windows)		

Engine/Host Dependent Information	
Data Set Page Size	16384
Number of Data Set Pages	1522509
Number of Data Set Repairs	0
File Name	I:\MCdb\MCDB_FN\data\prv06_v2.sas7bdat
Release Created	9.0101M3
Host Created	XP_PRO

Alphabetic List of Variables and Attributes

#	Variable	Type	Len	Format	Label
15	AGE	Num	3		Patient Age at Service
16	AGE1206_NR	Num	8		Non-Rounded Patient Age as of 12/31/2006
87	ALLOW	Num	8		Allowed Amount-AA ($)
12	BEGINDT	Num	8	MMDDYY8.	Service From Date
62	BETOSAGG	Char	1		BETOS Category, Aggregated
60	BETOSDET	Char	3		BETOS Category, Detailed
88	BILL	Num	8		Billed Charge-BC ($)
68	BILLTYPE	Char	1		Type of Bill
93	CDHPIND	Char	1		Consumer Directed Health Plan
9	CLAIMCN	Char	23		Claim Control Number
10	CLMPDATE	Num	8	MMDDYY8.	Claim Paid Date
11	CLMRELCO	Char	1		Claim Related Condition
37	CNTY_P2	Char	3		County to Use for Pract. Report Analysis
7	COUNTY_P	Char	3		Patient County of Residence-FIPS Code
8	COVOTHR	Char	1		Patient Covered by Other Insurance
46	COVOTH_I	Char	1		Imputed/Coverted COVOTHR
89	COVTYPE	Char	1		Coverage Type
71	CPT	Char	6		CPT-4 /HCPCS Procedure Code
69	DELIVTYP	Char	1		Delivery System Type
75	DISENROLLDT	Num	8	MMDDYY8.	Disenrollment Date
98	DISENROLL_P	Num	8	MMDDYY8.	Disenrollment Date at PIDBDG Level, SSS created
21	DX1	Char	5		ICD-9-CM Diagnosis Code 1
22	DX2	Char	5		ICD-9-CM Diagnosis Code 2

23	DX3	Char	5		ICD-9-CM Diagnosis Code 3
24	DX4	Char	5		ICD-9-CM Diagnosis Code 4
25	DX5	Char	5		ICD-9-CM Diagnosis Code 5
26	DX6	Char	5		ICD-9-CM Diagnosis Code 6
27	DX7	Char	5		ICD-9-CM Diagnosis Code 7
28	DX8	Char	5		ICD-9-CM Diagnosis Code 8
29	DX9	Char	5		ICD-9-CM Diagnosis Code 9
30	DX10	Char	5		ICD-9-CM Diagnosis Code 10
52	D_DAT_F	Char	1		Delete Flag - Services with Date Out of Range
48	D_DNT_F	Char	1		Delete Flag - Routine Dental Services
50	D_I7K_F	Char	1		Delete Flag - Financial Outliers
49	D_INP_F	Char	1		Delete Flag - Inpatient Services
51	D_ZIP_F	Char	1		Delete Flag - Non-Maryland Residents Services
4	ENCNT_ID	Char	8		Medical Service ID (Payer Specific)
13	ENDDT	Num	8	MMDDYY8.	Service Thru Date
1	ENROLLDT	Num	8	MMDDYY8.	Enrollment Date
97	ENROLL_P	Num	8	MMDDYY8.	Enrollment Date at PIDBDG Level, SSS created
85	FACILITY	Char	1		Service Location is a Facility (1/0)
33	FEDTAXID	Char	9		Practitioner Federal Tax ID
3	FILE_ID	Char	1		Payer Unique Input File ID
80	F_AGE_NR	Char	1		Valid Age Flag(Age1206_nr 0-110 incl) (1/0)
96	F_ANESTH	Char	1		Flag, Anesthesia Service
86	F_BENRULE	Char	2		Bens Rule for Capitation
65	F_CAP_A	Char	1		Capit. Service used in Analysis (1/0)
59	F_CPT	Char	1		Procedure Code Category
36	F_DEMOG	Char	1		Demographic Donor for Analysis File?
19	F_FAC	Char	1		Service Performed in a Facility? (1/0)
77	F_FINMS	Char	1		All finan. var. are zero or missing
47	F_IMPOTH	Char	1		Is Variable COVOTHR Imputed? (1/0)
44	F_IMPRVU	Char	1		RVU Imputation Flag (1/0,2)
35	F_INSAMP	Char	1		Service in 10% Sample?
78	F_MODF	Char	1		1=There is TC/SG in MODF1/MODF2
64	F_NCAP_A	Char	1		Non-Capit Service used in Analysis (1/0)
66	F_ODDSVC	Char	2		Odd Service Flag
81	F_PATID	Char	1		Flag for Valid PATID (1/0)
79	F_PAYMD	Char	1		Payment:1=negative, 2=0, 3=+, 4=missing
76	F_UNCV	Char	1		Uncovered services(1/0)
14	F_YR07	Char	1		Service provided in 2007+
20	INSTATE	Char	1		MD Service Location ZIP
17	LOS	Num	3		Length of Service
72	MOD1	Char	2		AMA Modified I
94	MOD2	Char	2		AMA Modified II
63	NEWBETAG	Char	1		Aggregated BETOS, Reclassified
61	NEWBETDT	Char	3		Detailed BETOS, Reclassified
32	NEWMOD	Char	2		Imputed Modifier/Last Imputation
53	NEWSPEC	Char	3		Revised Speciality Code for Complex Algorithm
31	NUMDXC	Num	3		Number of Diagnosis Codes, Calculated
95	PARTPROV	Char	1		Participating Provider Flag
5	PATID	Char	14		Scrambled PATID
90	PATLIAB	Num	8		Patient Liability-PL ($)
58	PAYERTYP	Char	1		Payer Type(1=PRVHMO 2=PRVNHMO 3=MCRHMO 9=CAP)
91	PAYMENT	Num	8		Payment=Sum(RA,PL)($)

38	PAY_CALC	Char	1		Method for Calculating Medicare Payment
45	PAY_MCR	Num	8		Prelim Calc MCR Rate-DO NOT USE AT THIS POINT
56	PLAN	Char	6		Plan Type(HMOFFS/NONHMO/HMOCAP)
57	PLAN2	Char	6		Plan Type(HMOFFS/NONHMO/HMOCAP/MCRALL),All Ages
2	PNUM	Char	4		Payer Unique ID
82	POS	Char	10		Simplified Place of Service Code
34	PROVID	Char	13		Servicing Practitioner Identifier
83	REGION_P	Char	1		Patient Region of Residence
92	REIMB	Num	8		Reimbursement Amount-RA ($)
43	RVU	Num	8		Assigned RVU
84	RVU_2006	Num	8		2006 RVU applied to 2005 file
42	RVU_I	Num	8		Assigned/Imputed RVU
40	RVU_MP	Num	4		Medicare phys fee schedule RVU, malpractice
41	RVU_PE	Num	8		Medicare Phys Fee Schedule RVU, Pract Expense
39	RVU_W	Num	4		Medicare Phys Fee Schedule RVU, Work component
6	SEX	Char	1		Patient Sex
54	SPEC1	Char	3		Practitioner Specialty 1
55	SPEC_SOURCE	Num	8		Source of NEWSPEC specialty codes
18	SVCPLACE	Char	2		Place of Service
67	UNITIND	Char	1		Service Unit Indicator
70	UNITS	Num	8		Number of Units for a Service
74	ZIP_P	Char	5		Patient Residence ZIP Code
73	ZIP_S	Char	5		Service Location ZIP Code
99	F_CAP	Char	1		Capitated Service (1/0)
100	ID_CHANGE	Char	1		PIDBDG changed using data from legacy file
101	PIDBDG	Char	25		Patient Unique Identification

2005 Maryland Medical Care Database (MCDB Private Payer)
Data Description Summary
Universe: Outpatient Medical Services
Number of Services: 85,931,243

Number of Variables: 101
File Name: PRV06_V2.SAS7BDAT

File Creation Date: March 16, 2008

VARIABLE NAME	VARIABLE DESCRIPTION	VARIABLE TYPE	DISCRETE VARIABLES			NON-DISCRETE VARIABLES			
			Variable Value	Value Description	Number of Services	Number of Missing	Minimum Value	Mean Value	Miximum Value
AGE	Patient Age at Service	Numeric			0	645	0	47	109
			Missing	Missing	645				
			0 - 17	0 - 17	10,642,166				
			18 - 64	18 - 64	58,528,681				
			65 - 110	65 - 110	16,759,751				
AGE1206_NR	Non-Rounded Patient Age as of 12/31/2006	Numeric			0	79,304		47	110
			Missing	Missing	79,304				
			0 - 17	0 - 17	9,940,522				
			18 - 64	18 - 64	57,804,312				
			65 - 110	65 - 110	18,107,105				
ALLOW	Allowed Amount-AA ($)	Numeric			0	0	-129,030	-65	375,450
			< 0	< 0	11,950,969				
			0	0	3,462,165				
			1 - 100	1 - 100	57,975,343				
			101 - 500	101 - 500	10,926,312				
			> 500	> 500	1,616,454				
BEGINDT	Service From Date	Date	Missing	Date Components Incomplete	16				
			Reported	Date Components Complete	85,931,227				
BETOSAGG	BETOS Category, Aggregated	Character	Blank	Code Not Assigned	2,254,913				
			1	Evaluation and Management	25,093,691				
			2	Procedures	16,096,980				
			3	Imaging	8,464,108				
			4	Tests	26,580,756				
			5	Durable Medical Equipment	1,649,673				

			6	Other	5,151,663				
			7	Exceptions/Unclassified	639,459				
BETOSDET	BETOS Category, Detailed	Character	Blank	Code Not Assigned	2,309,861				
			D1A	Medical/Surgical Supplies	109,941				
			D1B	Hospital Beds	29,689				
			D1C	Oxygen and Supplies	214,723				
			D1D	Wheelchairs	112,090				
			D1E	Other DME	728,131				
			D1F	Orthotic Devices	404,205				
			Invalid	Code Invalid	50,884				
			I1A	Standard Imaging - Chest	1,106,462				
			I1B	Standard Imaging - Musculoskeletal	1,691,018				
			I1C	Standard Imaging - Breast	628,231				
			I1D	Standard Imaging - Contrast Gastrointestinal	53,903				
			I1E	Standard Imaging - Nuclear Medicine	793,995				
			I1F	Standard Imaging - Other	270,870				
			I2A	Advanced Imaging - CAT: Head	252,601				
			I2B	Advanced Imaging - CAT: Other	708,293				
			I2C	Advanced Imaging - MRI: Brain	136,500				
			I2D	Advanced Imaging - MRI: Other	352,390				
			I3A	Echography - Eye	35,299				
			I3B	Echography - Abdomen/Pelvis	798,965				
			I3C	Echography - Heart	763,844				
			I3D	Echography - Carotid Arteries	75,184				
			I3E	Echography - Prostate, Transrectal	11,551				
			I3F	Echography - Other	364,033				
			I4A	Imaging/Procedure - Heart, including Cardiac Catheterization	123,654				
			I4B	Imaging/Procedure - Other	297,309				
			M1A	Office Visits - New	1,497,170				
			M1B	Office Visits - Established	14,594,443				
			M2A	Hospital Visit - Initial	238,025				
			M2B	Hospital Visit - Subsequent	1,433,460				
			M2C	Hospital Visit - Critical Care	181,456				
			M3	Emergency Room Visit	1,104,735				
			M4A	Home Visit	40,831				
			M4B	Nursing Home Visit	233,624				
			M5B	Specialist - Psychiatry	2,003,703				

			M5C	Specialist - Ophthalmology	1,484,409				
			M5D	Specialist - Other	576,288				
			M6	Consultations	1,705,547				
			O1A	Ambulance	190,250				
			O1B	Chiropractic	1,151,234				
			O1C	Enteral and Parenteral	33,611				
			O1D	Chemotherapy	125,452				
			O1E	Other Drugs	1,131,584				
			O1F	Vision, Hearing and Speech Services	5,299				
			O1G	Influenza Immunization	2,514,231				
			P0	Anesthesia	601,314				
			P1A	Major Procedure - Breast	11,364				
			P1B	Major Procedure - Colectomy	4,663				
			P1C	Major Procedure - Cholecystectomy	1,214				
			P1D	Major Procedure - TURP	2,214				
			P1E	Major Procedure - Hysterectomy	11,489				
			P1F	Major Procedure - Explor/Decompr/ExcisDisc	29,585				
			P1G	Major Procedure - Other and Major Maternal Procedures	227,536				
			P2A	Major Procedure, Cardiovascular - CABG	8,106				
			P2B	Major Procedure, Cardiovascular - Aneurysm Repair	985				
			P2C	Major Procedure, Cardiovascular - Thromboendarterectomy	1,829				
			P2D	Major Procedure, Cardiovascular - Coronary Angioplasty (PTCA)	17,197				
			P2E	Major Procedure, Cardiovascular - Pacemaker Insertion	7,382				
			P2F	Major Procedure, Cardiovascular - Other	106,004				
			P3A	Major Procedure, Orthopedic - Hip Fracture Repair	2,871				
			P3B	Major Procedure, Orthopedic - Hip Replacement	5,698				
			P3C	Major Procedure, Orthopedic - Knee Replacement	11,939				
			P3D	Major Procedure, Orthopedic - Other	77,150				
			P4A	Eye Procedures - Corneal Transplant	727				
			P4B	Eye Procedures - Cataract Removal/Lens Insertion	52,485				
			P4C	Eye Procedures - Retinal Detachment	4,132				

			P4D	Eye Procedures - Treatment of Retinal Lesions	11,268				
			P4E	Eye - Other	135,575				
			P5A	Ambulatory Procedures - Skin	815,044				
			P5B	Ambulatory Procedures - Musculoskeletal	90,033				
			P5C	Ambulatory Procedures - Inguinal Hernia Repair	8,684				
			P5D	Ambulatory Procedures - Lithotripsy	6,943				
			P5E	Ambulatory Procedures - Other	808,002				
			P6A	Minor Procedures - Skin	608,858				
			P6B	Minor Procedures - Musculoskeletal	596,645				
			P6C	Minor Procedures - Other (Medicare Fee Schedule)	10,335,008				
			P6D	Minor Procedures - Other (Non-Medicare Fee Schedule)	26,102				
			P7A	Oncology - Radiation Therapy	296,047				
			P7B	Oncology - Other	246,949				
			P8A	Endoscopy - Arthroscopy	67,375				
			P8B	Endoscopy - Upper Gastrointestinal	144,765				
			P8C	Endoscopy - Sigmoidoscopy	14,137				
			P8D	Endoscopy - Colonoscopy	280,330				
			P8E	Endoscopy - Cystoscopy	83,271				
			P8F	Endoscopy - Bronchoscopy	15,614				
			P8G	Endoscopy - Lararoscopic Cholecystectomy	16,742				
			P8H	Endoscopy - Laryngoscopy	42,784				
			P8I	Endoscopy - Other	110,709				
			P9A	Dialysis Services (Medicare Fee Schedule)	75,698				
			P9B	Dialysis Services (Non-Medicare Fee Schedule)	74,448				
			T1A	Lab Tests - Routine Venipuncture (Non-Medicare Fee Schedule)	1,746,718				
			T1B	Lab Tests - Automated General Profiles	3,037,364				
			T1C	Lab Tests - Urinalysis	1,447,910				
			T1D	Lab Tests - Blood Counts	2,262,431				
			T1E	Lab Tests - Glucose	441,706				
			T1F	Lab Tests - Bacterial Cultures	1,025,767				
			T1G	Lab Tests - Other (Medicare Fee Schedule)	1,443,743				
			T1H	Lab Test, Other (Non-Medicare Fee Schedule)	12,004,436				
			T2A	Other Tests - Electrocardiograms	1,409,074				
			T2B	Other Tests - Cardiovascular Stress Tests	262,663				
			T2C	Other Tests - EKG Monitoring	69,133				
			T2D	Other Tests - Other	1,374,946				

			Y1	Medicare Fee Schedule	268,573				
			Y2	Other - Non-Medicare Fee Schedule	48,304				
			Z2	Undefined Codes	322,582				
BILL	Billed Charge-BC ($)	Numeric			0	0	-129,030	16	7,442,325
			< 0	< 0	11,950,360				
			0	0	433,729				
			1 - 100	1 - 100	45,642,844				
			101 - 500	101 - 500	23,562,682				
			> 500	> 500	4,341,628				
BILLTYPE	Type of Bill	Character	1	Final Bill , Debit	74,062,782				
			2	Credit	92,445				
			8	Capitated	11,776,016				
CDHPIND		Character	0	No	84,637,986				
			1	Yes	1,293,257				
CDHPIND	Consumer Directed Health Plan	Character			0				
CLAIMCN	Claim Control Number	Character	Blank	Value Missing	24				
			Reported	Value Reported	85,931,219				
CLMPDATE	Claim Paid Date	Date	Missing	Date Components Incomplete	400,663				
			Reported	Date Components Complete	85,530,580				
CLMRELCO	Claim Related Condition	Character	Blank	Code Missing	11,684,230				
			0	Non-accident	63,944,614				
			1	Work Accident	1,398				
			2	Auto Accident	13,352				
			3	Other Accident	2,426,075				
			9	Payer Code=9 (Not Coded)	7,861,574				
CNTY_P2	County to Use for Pract. Report Analysis	Character	Missing	Patient Residence County Unknown	149,724				
			1	Allegany	787,624				
			3	Anne Arundel	8,328,300				
			5	Baltimore	12,653,477				
			9	Calvert	1,446,025				
			11	Caroline	417,106				
			13	Carroll	3,071,085				
			15	Cecil	1,119,584				
			17	Charles	1,927,037				
			19	Dorchester	389,698				
			21	Frederick	4,204,412				
			23	Garrett	222,399				
			25	Harford	4,211,252				

			27	Howard	4,812,440				
			29	Kent	331,127				
			31	Montgomery	16,791,430				
			33	Prince Georges	11,842,105				
			35	Queen Annes	732,760				
			37	Saint Marys	1,421,740				
			39	Somerset	218,472				
			41	Talbot	591,924				
			43	Washington	1,768,518				
			45	Wicomico	1,083,555				
			47	Worcester	756,505				
			510	Baltimore City	6,652,944				
COUNTY_P	Patient County of Residence-FIPS Code	Character	Missing	Patient Residence County Unknown	194,930				
			1	Allegany	787,593				
			3	Anne Arundel	8,329,275				
			5	Baltimore	12,652,692				
			9	Calvert	1,445,856				
			11	Caroline	417,548				
			13	Carroll	3,073,719				
			15	Cecil	1,120,226				
			17	Charles	1,930,966				
			19	Dorchester	389,871				
			21	Frederick	4,208,987				
			23	Garrett	222,647				
			25	Harford	4,214,138				
			27	Howard	4,812,607				
			29	Kent	331,348				
			31	Montgomery	16,769,626				
			33	Prince Georges	11,804,769				
			35	Queen Annes	732,911				
			37	Saint Marys	1,423,078				
			39	Somerset	218,397				
			41	Talbot	592,603				
			43	Washington	1,772,436				
			45	Wicomico	1,083,657				
			47	Worcester	756,933				
			510	Baltimore City	6,644,430				
COVOTHR	Patient Covered by Other Insurance	Character	0	No-No Other Insurance Coverage	67,184,322				

			1	Yes-Other Coverage is Primary	12,535,966				
			2	Yes-Other Coverage is Secondary	378,500				
			9	Payer Code=9 (Not Coded)	5,042,152				
			Invalid	Code Invalid	790,303				
COVOTH_I	Imputed/Coverted COVOTHR	Character	0	No-No Other Insurance Coverage	68,459,304				
			1	Yes-Other Coverage is Primary	12,535,966				
			2	Yes-Other Coverage is Secondary	782,338				
			9	Payer Code=9 (Not Coded)	4,153,635				
COVTYPE	Coverage Type	Character	1	Medicare Supplemental(Indvl,Grp,WRAP)	12,937,003				
			2	Individual Plan	3,454,811				
			3	Priv Employr Sponsord Fully Self-Insu	20,151,058				
			4	Private Employer Sponsored Insured	10,273,675				
			5	Public Employee	28,258,105				
			6	Comprehensive Std Health Benefit Plan	9,391,497				
			7	Mcare srv provided by a Mcare HMO/HCFA	1,280,972				
			8	Taft Hartley Jointly Managed Trust Fund	106,508				
			9	Payer Code=9 (Unknown)	77,614				
CPT	CPT-4 /HCPCS Procedure Code	Character	Blank	Value Missing	79,910				
			Reported	Value Reported	85,851,333				
DELIVTYP	Delivery System Type	Character	1	HMO (Non-Medicaid,Includes Medicare)	28,283,304				
			2	PPO-POS	16,543,419				
			3	PPO or Other Managed Care	27,755,381				
			4	Indemnity Care	9,300,695				
			5	HMO-POS Rider	3,984,330				
			9	Payer Code=9 (Not Coded)	64,114				
DISENROLLDT	Disenrollment Date	Date	Missing	Date Components Incomplete	66,589,611				
			Reported	Date Components Complete	19,341,632				
DISENROLL_P	Disenrollment Date at PIDBDG Level, SSS created	Date	Missing	Date Components Incomplete	70,908,987				
			Reported	Date Components Complete	15,022,256				
DX1	ICD-9-CM Diagnosis Code 1	Character	0		2,710,944				
			1		83,220,299				
DX10	ICD-9-CM Diagnosis Code 10	Character	0		85,914,848				
			1		16,395				
DX2	ICD-9-CM Diagnosis Code 2	Character	0		52,364,630				
			1		33,566,613				
DX3	ICD-9-CM Diagnosis Code 3	Character	0		70,247,665				
			1		15,683,578				

DX4	ICD-9-CM Diagnosis Code 4	Character	0		78,789,844				
			1		7,141,399				
DX5	ICD-9-CM Diagnosis Code 5	Character	0		84,784,005				
			1		1,147,238				
DX6	ICD-9-CM Diagnosis Code 6	Character	0		85,260,878				
			1		670,365				
DX7	ICD-9-CM Diagnosis Code 7	Character	0		85,461,887				
			1		469,356				
DX8	ICD-9-CM Diagnosis Code 8	Character	0		85,582,162				
			1		349,081				
DX9	ICD-9-CM Diagnosis Code 9	Character	0		85,684,244				
			1		246,999				
D_DAT_F	Delete Flag - Services with Date Out of Range	Character	1	Yes	16				
			9	No	85,931,227				
D_DNT_F	Delete Flag - Routine Dental Services	Character	1	Yes	36,278				
			9	No	85,894,965				
D_I7K_F	Delete Flag - Financial Outliers	Character	1	Yes	18,063				
			9	No	85,913,180				
D_INP_F	Delete Flag - Inpatient Services	Character	1	Yes	599,916				
			9	No	85,331,327				
D_ZIP_F	Delete Flag - Non-Maryland Residents Services	Character	1	Yes	194,930				
			9	No	85,736,313				
ENCNT_ID	Medical Service ID (Payer Specific)	Character	Assigned	Sequential ID Assigned	85,931,243				
ENDDT	Service Thru Date	Date			0				
ENROLLDT	Enrollment Date	Date	Missing	Date Components Incomplete	152,270				
			Reported	Date Components Complete	85,778,973				
ENROLL_P	Enrollment Date at PIDBDG Level, SSS created	Date	Missing	Date Components Incomplete	454,901				
			Reported	Date Components Complete	85,476,342				
FACILITY	Service Location is a Facility (1/0)	Character	0	Service Location is not a Facility	72,450,898				
			1	Service Location is a Facility	13,480,345				
FEDTAXID	Practitioner Federal Tax ID	Character	Blank	Value Missing	242,516				
			Reported	Value Reported	85,688,727				
F_AGE_NR	Valid Age Flag(Age1206_nr 0-110 incl) (1/0)	Character	0	No-AGE1203 Missing, <0, or >110	79,304				
			1	Yes-AGE1203 Between 0 and 110, Inclusive	85,851,939				
F_ANESTH	Flag, Anesthesia Service	Character	0	No - Service Not Used in Analysis	84,801,598				
			1	Yes - Service Used in Analysis	1,129,645				
F_BENRULE	Bens Rule for Capitation	Character	Blank	Not Assigned	85,931,243				
F_CAP	Capitated Service (1/0)	Character			0				

F_CAP_A	Capit. Service used in Analysis (1/0)	Character	0	No-Service Cannot Be Used in Capitated Analysis	76,743,021				
			1	Yes-Service Can Be Used in Capitated Analysis	9,188,222				
F_CPT	Procedure Code Category	Character	1	Homegrown Code	112,119				
			2	HCPCS Code	4,420,452				
			3	CPT-4 Code	79,514,550				
			4	ICD-9-CM Code	305,179				
			5	Revenue Code	1,098,008				
			6	Procedure Code Missing	76,399				
			7	Procedure Code Invalid	404,536				
F_DEMOG	Demographic Donor for Analysis File?	Character	0	No - Demographics Not Used in Analysis	82,126,537				
			1	Yes - Demographics Used in Analysis	3,804,706				
F_FAC	Service Performed in a Facility? (1/0)	Character	0	Service Not Provided in a Facility	72,758,441				
			1	Service Provided in a Facility	13,172,802				
F_FINMS	All finan. var. are zero or missing	Character	0	No - All Financial Variables are Not Equal to 0/Missing	85,608,578				
			1	Yes - All Financial Variables are Equal to 0/Missing	322,665				
F_IMPOTH	Is Variable COVOTHR Imputed? (1/0)	Character	0	No-Other Insurance Coverage is Not Imputed	84,252,423				
			1	Yes-Other Insurance Coverage is Imputed	1,678,820				
F_IMPRVU	RVU Imputation Flag (1/0,2)	Character	0	No Imputation Attempted	55,322,374				
			1	RVU Imputed	25,213,990				
			2	Attempted Imputation, Could Not Impute	5,394,879				
F_INSAMP	Service in 10% Sample?	Character	0	Not in 10% Sample	77,317,961				
			1	In 10% Sample	8,613,282				
F_MODF	1=There is TC/SG in MODF1/MODF2	Character	0	No - TC or SG Modifier Not Present	85,532,446				
			1	Yes - TC or SG Modifier Present	398,797				
F_NCAP_A	Non-Capit Service used in Analysis (1/0)	Character	0	No-Service Cannot Be Used in Non-Capitated Analysis	41,380,227				
			1	Yes-Service Can Be Used in Non-Capitated Analysis	44,551,016				
F_ODDSVC	Odd Service Flag	Character	Blank	Not Odd Services	55,557,815				
			3	Payment <= 1, Allowed Amount <= 0	1,871,064				
			4	Identified Anesthesia Outside Of Range	238,842				
			5	Identified Facility Bill	1,728,431				
			6	Minor Service Assoc With Major Service	312,411				
			8	Payer not NONHMO,HMOFFS,HMOCAP	13,598,359				
			9	Not First Dupe by Key, Descending Payment	1,610,827				
			10	Actuaries Trim, Payment, Low	265,373				
			11	Actuaries Trim, Payment, High	29,987				

			12	Actuaries Trim, RVU_I, Low	102,753				
			13	Actuaries Trim, RVU_I, High	180,642				
			14	Covoth_I = 2, Possible Secondary Payer	165,454				
			15	RVU_I <= 0	3,600,357				
			99	More Than 1 Service	6,668,928				
F_PATID	Flag for Valid PATID (1/0)	Character	0	No-Patient ID is Missing or Invalid	3,047				
			1	Yes-Patient ID is Valid	85,928,196				
F_PAYMD	Payment:1=negative, 2=0, 3=+, 4=missing	Character	1	Negative Payment	101,670				
			2	Zero Payment	3,815,201				
			3	Positive Payment	70,145,873				
			4	Missing Payment	11,868,499				
F_UNCV	Uncovered services(1/0)	Character	0	No - Service is Not Uncovered	82,536,515				
			1	Yes - Service is Uncovered	3,394,728				
F_YR07	Service provided in 2007+	Character	0	No - Service Provided in 2005	68,246,302				
			1	Yes - Service Provided in 2005+	17,684,941				
ID_CHANGE	PIDBDG changed using data from legacy file	Character			0				
INSTATE	MD Service Location ZIP	Character	Missing	Service Location Unknown (Zipcode Missing/Invalid)	4,689,792				
			0	Service Location Valid, Not MD (Based on Zipcode)	13,548,707				
			1	Service Location in MD (Based on Zipcode)	67,692,744				
LOS	Length of Service	Numeric			0	1,054	1	1	1,126
			Missing	Missing	1,054				
			1	1	84,837,385				
			> 1	> 1	1,092,804				
MOD1	AMA Modified I	Character	Blank	Missing Code	70,426,987				
			Other	All Other Codes	8,196,227				
			20	Microsurgery	119				
			21	Prolonged Evaluatn/Mgmt(E/M) Serv	984				
			22	Unusual Procedural Services	17,546				
			23	Unusual Anesthesia	121				
			24	Unrel E/M Serv by Same Phys Postop	47,722				
			25	Sep Ident. E/M by Same Phys/day	1,689,340				
			26	Professional Component	4,161,050				
			32	Mandated Services	1,397				
			47	Anesthesia by Surgeon	2,603				
			50	Bilateral Procedure	93,963				
			51	Multiple Procedures	199,867				

			52	Reduced Services	70,606				
			54	Surgical Care Only	6,815				
			55	Postoperative Management Only	882				
			56	Preoperative Management Only	363				
			57	Decision for Surgery	36,967				
			58	Staged/Rel Proc by Same Phys Postop	23,822				
			62	Two Surgeons	3,072				
			65	Surgical Team	1				
			76	Repeat Procedure by Same Physician	72,171				
			77	Repeat Procedure by Another Physician	6,330				
			78	Return to Op Rm for Rel Proc Postop	9,599				
			79	Unrelated Proc by Same Phy Postop	45,284				
			80	Assistant Surgeon	50,671				
			81	Minimum Assisting Surgeon	1,161				
			82	Assist Surgeon(qual res phy unav)	7,975				
			90	Reference (Outside) Laboratory	180,228				
			99	Multiple Modifiers	19,015				
			AN	Anesthesia(surg. codes: 10000 thru 69999)	69,120				
			QX	Cert Reg Nurse Anesth. w/o doc dir	42,671				
			QZ	Cert Reg Nurse Anesth (CRNA)	56,639				
			SG	HCFA Ambulatory Surgery	117,644				
			TC	Radiology: Technical Component	272,281				
MOD2	AMA Modified II	Character	Blank	Missing Code	84,571,806				
			Other	All Other Codes	1,196,025				
			20	Microsurgery	4				
			21	Prolonged Evaluatn/Mgmt(E/M) Serv	45				
			22	Unusual Procedural Services	931				
			23	Unusual Anesthesia	15				
			24	Unrel E/M Serv by Same Phys Postop	668				
			25	Sep Ident. E/M by Same Phys/day	10,594				
			26	Professional Component	4,050				
			32	Mandated Services	8				
			47	Anesthesia by Surgeon	14				
			50	Bilateral Procedure	6,333				
			51	Multiple Procedures	59,184				
			52	Reduced Services	10,473				
			54	Surgical Care Only	471				
			55	Postoperative Management Only	45				

			56	Preoperative Management Only	16				
			57	Decision for Surgery	2,552				
			58	Staged/Rel Proc by Same Phys Postop	3,070				
			62	Two Surgeons	200				
			76	Repeat Procedure by Same Physician	29,461				
			77	Repeat Procedure by Another Physician	14,341				
			78	Return to Op Rm for Rel Proc Postop	859				
			79	Unrelated Proc by Same Phy Postop	7,743				
			80	Assistant Surgeon	1,470				
			81	Minimum Assisting Surgeon	63				
			82	Assist Surgeon(qual res phy unav)	626				
			90	Reference (Outside) Laboratory	586				
			99	Multiple Modifiers	116				
			AN	Anesthesia(surg. codes: 10000 thru 69999)	4				
			QX	Cert Reg Nurse Anesth. w/o doc dir	85				
			QZ	Cert Reg Nurse Anesth (CRNA)	274				
			SG	HCFA Ambulatory Surgery	3,685				
			TC	Radiology: Technical Component	5,426				
NEWBETAG	Aggregated BETOS, Reclassified	Character	Blank	Code Not Assigned	2,119,435				
			1	Evaluation and Management	24,994,993				
			2	Procedures	16,426,686				
			3	Imaging	8,413,157				
			4	Tests	26,550,872				
			5	Durable Medical Equipment	1,648,733				
			6	Other	5,140,100				
			7	Exceptions/Unclassified	637,267				
NEWBETDT	Detailed BETOS, Reclassified	Character	Blank	Code Not Assigned	2,119,435				
			D1A	Medical/Surgical Supplies	109,776				
			D1B	Hospital Beds	29,688				
			D1C	Oxygen and Supplies	214,493				
			D1D	Wheelchairs	112,092				
			D1E	Other DME	727,796				
			D1F	Orthotic Devices	404,048				
			Invalid	Code Invalid	50,840				
			I1A	Standard Imaging - Chest	1,105,868				
			I1B	Standard Imaging - Musculoskeletal	1,690,469				
			I1C	Standard Imaging - Breast	628,130				
			I1D	Standard Imaging - Contrast Gastrointestinal	53,826				

			I1E	Standard Imaging - Nuclear Medicine	792,661				
			I1F	Standard Imaging - Other	266,076				
			I2A	Advanced Imaging - CAT: Head	252,371				
			I2B	Advanced Imaging - CAT: Other	707,747				
			I2C	Advanced Imaging - MRI: Brain	136,358				
			I2D	Advanced Imaging - MRI: Other	352,311				
			I3A	Echography - Eye	35,271				
			I3B	Echography - Abdomen/Pelvis	798,843				
			I3C	Echography - Heart	760,480				
			I3D	Echography - Carotid Arteries	75,138				
			I3E	Echography - Prostate, Transrectal	11,544				
			I3F	Echography - Other	363,529				
			I4A	Imaging/Procedure - Heart, including Cardiac Catheterization	122,234				
			I4B	Imaging/Procedure - Other	260,301				
			M1A	Office Visits - New	1,491,389				
			M1B	Office Visits - Established	14,540,713				
			M2A	Hospital Visit - Initial	237,657				
			M2B	Hospital Visit - Subsequent	1,417,696				
			M2C	Hospital Visit - Critical Care	176,340				
			M3	Emergency Room Visit	1,102,905				
			M4A	Home Visit	40,807				
			M4B	Nursing Home Visit	233,169				
			M5B	Specialist - Psychiatry	2,002,261				
			M5C	Specialist - Ophthalmology	1,483,584				
			M5D	Specialist - Other	575,477				
			M6	Consultations	1,692,995				
			O1A	Ambulance	189,709				
			O1B	Chiropractic	1,150,128				
			O1C	Enteral and Parenteral	33,024				
			O1D	Chemotherapy	125,451				
			O1E	Other Drugs	1,122,401				
			O1F	Vision, Hearing and Speech Services	5,295				
			O1G	Influenza Immunization	2,514,092				
			P0	Anesthesia	1,129,625				
			P1A	Major Procedure - Breast	10,878				
			P1B	Major Procedure - Colectomy	4,464				
			P1C	Major Procedure - Cholecystectomy	1,169				

			P1D	Major Procedure - TURP	2,082				
			P1E	Major Procedure - Hysterectomy	10,613				
			P1F	Major Procedure - Explor/Decompr/ExcisDisc	29,058				
			P1G	Major Procedure - Other and Major Maternal Procedures	215,859				
			P2A	Major Procedure, Cardiovascular - CABG	7,900				
			P2B	Major Procedure, Cardiovascular - Aneurysm Repair	971				
			P2C	Major Procedure, Cardiovascular - Thromboendarterectomy	1,758				
			P2D	Major Procedure, Cardiovascular - Coronary Angioplasty (PTCA)	17,183				
			P2E	Major Procedure, Cardiovascular - Pacemaker Insertion	7,264				
			P2F	Major Procedure, Cardiovascular - Other	91,318				
			P3A	Major Procedure, Orthopedic - Hip Fracture Repair	2,782				
			P3B	Major Procedure, Orthopedic - Hip Replacement	5,460				
			P3C	Major Procedure, Orthopedic - Knee Replacement	11,360				
			P3D	Major Procedure, Orthopedic - Other	74,754				
			P4A	Eye Procedures - Corneal Transplant	687				
			P4B	Eye Procedures - Cataract Removal/Lens Insertion	49,946				
			P4C	Eye Procedures - Retinal Detachment	4,020				
			P4D	Eye Procedures - Treatment of Retinal Lesions	11,238				
			P4E	Eye - Other	134,712				
			P5A	Ambulatory Procedures - Skin	812,981				
			P5B	Ambulatory Procedures - Musculoskeletal	86,106				
			P5C	Ambulatory Procedures - Inguinal Hernia Repair	8,166				
			P5D	Ambulatory Procedures - Lithotripsy	6,421				
			P5E	Ambulatory Procedures - Other	782,733				
			P6A	Minor Procedures - Skin	607,294				
			P6B	Minor Procedures - Musculoskeletal	524,320				
			P6C	Minor Procedures - Other (Medicare Fee Schedule)	10,302,016				
			P6D	Minor Procedures - Other (Non-Medicare Fee Schedule)	26,036				
			P7A	Oncology - Radiation Therapy	295,034				
			P7B	Oncology - Other	246,911				

			P8A	Endoscopy - Arthroscopy	63,985				
			P8B	Endoscopy - Upper Gastrointestinal	140,136				
			P8C	Endoscopy - Sigmoidoscopy	14,069				
			P8D	Endoscopy - Colonoscopy	270,919				
			P8E	Endoscopy - Cystoscopy	81,521				
			P8F	Endoscopy - Bronchoscopy	15,123				
			P8G	Endoscopy - Lararoscopic Cholecystectomy	15,511				
			P8H	Endoscopy - Laryngoscopy	42,517				
			P8I	Endoscopy - Other	109,804				
			P9A	Dialysis Services (Medicare Fee Schedule)	75,607				
			P9B	Dialysis Services (Non-Medicare Fee Schedule)	74,375				
			T1A	Lab Tests - Routine Venipuncture (Non-Medicare Fee Schedule)	1,738,865				
			T1B	Lab Tests - Automated General Profiles	3,037,348				
			T1C	Lab Tests - Urinalysis	1,447,821				
			T1D	Lab Tests - Blood Counts	2,274,083				
			T1E	Lab Tests - Glucose	441,503				
			T1F	Lab Tests - Bacterial Cultures	1,024,691				
			T1G	Lab Tests - Other (Medicare Fee Schedule)	1,442,296				
			T1H	Lab Test, Other (Non-Medicare Fee Schedule)	12,035,921				
			T2A	Other Tests - Electrocardiograms	1,406,396				
			T2B	Other Tests - Cardiovascular Stress Tests	262,629				
			T2C	Other Tests - EKG Monitoring	69,117				
			T2D	Other Tests - Other	1,370,202				
			Y1	Medicare Fee Schedule	267,159				
			Y2	Other - Non-Medicare Fee Schedule	48,301				
			Z2	Undefined Codes	321,807				
NEWMOD	Imputed Modifier/Last Imputation	Character	Blank	Missing Code	81,356,442				
			26	Professional Component	4,307,801				
			TC	Radiology: Technical Component	267,000				
NEWSPEC	Revised Speciality Code for Complex Algorithm	Character	1	General Practice	894,308				
			2	General Surgery	969,365				
			3	Allergy & Immunology	717,737				
			4	Anesthesiology	861,722				
			5	Cardiology	2,755,990				
			6	Dermatology	1,535,577				
			7	Emergency Medicine	1,479,485				
			8	Endocrinology Medicine	513,054				

			9	Family Practice	4,812,591				
			10	Gastroenterology	824,338				
			11	Geriatrics	38,325				
			12	Hand Surgery	47,682				
			13	Hematology	486,567				
			14	Internal Medicine	9,561,696				
			15	Infectious Disease	138,625				
			16	Nephrology	311,500				
			17	Neurology	614,116				
			18	Nuclear Medicine	74,281				
			19	Oncology	1,498,979				
			20	Ophthalmology	1,645,881				
			21	Orthopedic Surgery	2,209,020				
			22	Osteopathy (include manipulations)	23,147				
			23	Otology/Laryngo/Rhino/Otolaryngo	866,280				
			24	Pathology	2,497,018				
			25	Pediatrics	5,598,047				
			26	Peripheral Vascular Disease/Surgery	52,415				
			27	Plastic Surgery	144,704				
			28	Physical Medicine & Rehabilitation	634,083				
			29	Proctology	25,394				
			30	Psychiatry	833,505				
			31	Pulmonary Disease	458,361				
			32	Radiology	4,949,004				
			33	Rheumatology	474,462				
			34	Surgical Specialty Not Listed Here	233,139				
			35	Thoracic Surgery	90,696				
			36	Urology	823,567				
			37	Misc Special (pub health/indusl med)	128,051				
			38	Physician w/o Identfd Spec/Not Listd	1,819,928				
			39	Obstetrics/Gynecology	2,953,145				
			40	Acupuncturist	214,179				
			41	Alcohol/Drug Detox Services	32,102				
			42	Ambulance Services	199,173				
			43	Audiologist/Speech Pathologist	190,198				
			44	Chiropractor	3,719,169				
			45	Freestanding Clinic-Not a Govt Agncy	101,938				
			46	Day Care FCLTY-Medical Mental Health	6,632				

			47	Dietitian/Licensed Nutritionist	42,311				
			48	Home Health Provider	144,677				
			49	Advanced Practice Nurse: Anesthetist	64,159				
			50	Advanced Practice Nurse: Midwife	57,031				
			51	Advancd Practice Nurse: Nurse Practitioner	201,225				
			52	Advd Practice Nurse: Psychotherapist	16,927				
			53	Nurse Other than Advanced Practice	111,461				
			54	Occupational Therapist	306,618				
			55	Optometrist	399,686				
			56	Podiatrist	1,209,334				
			57	Physical Therapist	5,947,665				
			58	Psychologist	543,158				
			59	Clinical Social Worker	534,308				
			60	Pub Health/Welf Agy-fed & local govt	66,663				
			61	Voluntary Health Agency	1				
			62	Other Specialty Not Listed Above	693,488				
			63	Respiratory Therapist	2,068				
			70	General Dentist	76,152				
			71	Endodontist	6,686				
			72	Orthodontist	2,551				
			73	Oral Surgeon	50,405				
			74	Pedodontist	5,271				
			75	Periodontist	2,570				
			76	Prosthodontist	823				
			80	Freestanding Pharmacy incl grocery	91,815				
			81	Mail Order Pharmacy	11,820				
			82	Independent Laboratory	10,988,995				
			83	Independent Medical Supply Company	1,318,245				
			84	Optician/Optometrist	40,632				
			85	All Other Supplies	3,350				
			90	Freestanding Medical Facility	488,756				
			91	Freestanding Surgical Facility	312,134				
			92	Freestanding Imaging Center	403,149				
			93	Other Facility	2,534,083				
			100	Neonatology	85,806				
			101	Multi-Specialty Medical Practice	59,169				
			102	Mental Health Clinic	48,875				
NUMDXC	Number of Diagnosis Codes, Calculated	Numeric			0			2	1

			0	0	1,682,383				
			1	1	50,461,352				
			>1	>1	33,787,508				
PARTPROV	Participating Provider Flag	Character	Invalid	Invalid Code	5				
			1	Yes-Participating Provider	72,572,115				
			2	No-Not a Participating Provider	7,525,249				
			3	Payer Code=3 (Not Coded)	5,833,874				
PATID	Scrambled PATID	Character	Reported	Value Reported	85,931,243				
PATLIAB	Patient Liability-PL ($)	Numeric			0	0	-7,129	-94	375,450
			< 0	< 0	9,090,671				
			0	0	45,158,245				
			1 - 100	1 - 100	30,290,991				
			101 - 500	101 - 500	1,257,085				
			> 500	> 500	134,251				
PAYERTYP	Payer Type(1=PRVHMO 2=PRVNHMO 3=MCRHMO 9=CAP)	Character	1	Private HMO FFS	18,638,399				
			2	Private Non-HMO	39,407,105				
			3	Medicare HMO FFS	223,713				
			5	Not Classified	15,793,527				
			9	Capitated Services	11,868,499				
PAYMENT	Payment=Sum(RA,PL)($)	Numeric			0	11,868,499	-28,958	76	375,45
			Missing	Missing	11,868,499				
			< 0	< 0	101,670				
			0	0	3,815,201				
			1 - 100	1 - 100	59,180,503				
			101 - 500	101 - 500	9,608,369				
			> 500	> 500	1,357,001				
PAY_CALC	Method for Calculating Medicare Payment	Character	0	Do not calculate payment	2,307,878				
			2	Calculate Using Facility RVUs	4,594,016				
			3	Calculate Using Non-Facility RVUs	30,274,531				
			4	Procedure Code is a Lab Fee Schedule Code	17,086,289				
			5	Procedure Code with an Imputed Total RVU but no Components	2,614,163				
			6	Flagged as Odd Service - Do not Use MCR Rate	29,054,366				
PAY_MCR	Prelim Calc MCR Rate-DO NOT USE AT THIS POINT	Numeric			0	31,362,244	1	60	6,107
			Missing	Missing	31,362,244				
			1 - 25	1 - 25	20,486,738				
			26 - 75	26 - 75	20,031,083				

			> 75	> 75	14,051,178				
PIDBDG		Character	Reported	Value Not Missing	85,931,243				
PLAN	Plan Type(HMOFFS/NONHMO/HMOCAP)	Character	Blank	Not Assigned	17,186,181				
			HMOCAP	HMO Capitated Plan	10,699,558				
			HMOFFS	HMO Fee-For-Services Plan	18,638,399				
			NONHMO	Non-HMO Plan	39,407,105				
PLAN2	Plan Type(HMOFFS/NONHMO/HMOCAP/MCRALL),All Ages	Character	Blank	Not Assigned	5,388,340				
			HMOCAP	HMO Capitated Plan	11,868,499				
			HMOFFS	HMO Fee-For-Services Plan	18,638,399				
			MCRALL	Medicare Plan	10,628,900				
			NONHMO	Non-HMO Plan	39,407,105				
PNUM	Payer Unique ID	Character	P020	Aetna Life & Health Insurance Co.	6,735,681				
			P030	Aetna U.S. Healthcare	4,807,625				
			P070	American Republic Insurance Co.	24,070				
			P130	CareFirst BlueChoice, Inc.	18,774,307				
			P131	CareFirst of Maryland, Inc.	23,547,560				
			P160	CIGNA Healthcare Mid-Atlantic, Inc.	3,928,967				
			P280	Assurant/Fortis Insurance Co.	72,393				
			P320	Golden Rule Insurance Co.	146,313				
			P325	Graphic Arts Benefit Corporation	123,837				
			P330	Great-West Life & Annuity Insurance Co.	462,464				
			P350	Guardian Life Insurance Co. of America	346,828				
			P471	Unicare Life & Health Insurance Co.	155,774				
			P480	Kaiser Foundation Health Plan of Mid-Atlantic	5,087,116				
			P500	MAMSI Life and Health Insurance Co.	2,127,195				
			P510	Maryland Fidelity Insurance Co.	46,186				
			P520	MD-Individual Practice Association, Inc.	3,857,438				
			P530	MEGA Life & Health Insurance Co.	185,563				
			P620	Optimum Choice, Inc.	6,874,787				
			P680	Coventry Healthcare of Delaware, Inc.	1,300,757				
			P760	State Farm Mutual Automobile Insurance Co.	65,462				
			P820	United Healthcare Insurance Co.	6,985,303				
			P830	Trustmark Insurance Co.	5,877				
			P850	Union Labor Life Insurance Co.	106,508				
			P870	United Healthcare of the Mid-Atlantic, Inc.	163,232				
POS	Simplified Place of Service Code	Character	Inpatient	Inpatient Facility	5,235,401				

			Office	Provider Office	55,024,694				
			Other	Other Facility	18,315,957				
			Outpatient	Outpatient Facility	7,355,191				
PROVID	Servicing Practitioner Identifier	Character	Blank	Value Missing	198,406				
			Reported	Value Reported	85,732,837				
REGION_P	Patient Region of Residence	Character	Blank	Value Missing	149,724				
			1	National Capital Area	28,633,535				
			2	Baltimore Metropolitan Area	39,729,498				
			3	Eastern Shore	5,640,731				
			4	Southern Maryland	4,794,802				
			5	Western Maryland	6,982,953				
REIMB	Reimbursement Amount-RA ($)	Numeric			0	0	-28,858	-83	254,038
			< 0	< 0	11,932,708				
			0	0	10,508,633				
			1 - 100	1 - 100	54,987,078				
			101 - 500	101 - 500	7,313,375				
			> 500	> 500	1,189,449				
RVU	Assigned RVU	Numeric			0	32,916,747	0	2	139
			Missing	Missing	32,916,747				
			0.01 - 1.00	0.01 - 1.00	19,188,782				
			1.01 - 5.00	1.01 - 5.00	30,819,959				
			> 5	> 5	3,005,755				
RVU_2006	2006 RVU applied to 2005 file	Numeric			0	7,647,820	0	2	139
			Missing	Missing	7,647,820				
			0.01 - 1.00	0.01 - 1.00	41,653,022				
			1.01 - 5.00	1.01 - 5.00	33,399,182				
			> 5	> 5	3,231,219				
RVU_I	Assigned/Imputed RVU	Numeric			0	7,647,820	0	2	139
			Missing	Missing	7,647,820				
			0.01 - 1.00	0.01 - 1.00	41,653,022				
			1.01 - 5.00	1.01 - 5.00	33,399,182				
			> 5	> 5	3,231,219				
RVU_MP	Medicare phys fee schedule RVU, malpractice	Numeric			0	7,702,757	0	0	17
			Missing	Missing	7,702,757				
			0	0	21,521,453				
			0.01 - 1.00	0.01 - 1.00	56,177,454				
			1.01 - 5.00	1.01 - 5.00	482,150				
			> 5	> 5	47,429				

RVU_PE	Medicare Phys Fee Schedule RVU, Pract Expense	Numeric			0	7,702,757	0	1	126
			Missing	Missing	7,702,757				
			0.01 - 1.00	0.01 - 1.00	59,430,047				
			1.01 - 5.00	1.01 - 5.00	17,177,905				
			> 5	> 5	1,620,534				
RVU_W	Medicare Phys Fee Schedule RVU, Work component	Numeric			0	7,702,757	0	1	95
			Missing	Missing	7,702,757				
			0	0	23,287,107				
			0.01 - 1.00	0.01 - 1.00	36,002,462				
			1.01 - 5.00	1.01 - 5.00	18,053,395				
			> 5	> 5	885,522				
SEX	Patient Sex	Character	1	Male	33,611,773				
			2	Female	52,317,287				
			3	Payer Code=3 (Not Coded)	2,183				
SPEC1	Practitioner Specialty 1	Character	Blank	Code Not Reported	12,399,019				
			Invalid	Code Invalid	281				
			1	General Practice	658,572				
			2	General Surgery	729,711				
			3	Allergy & Immunology	566,873				
			4	Anesthesiology	772,864				
			5	Cardiology	2,082,133				
			6	Dermatology	1,417,496				
			7	Emergency Medicine	1,327,589				
			8	Endocrinology Medicine	366,442				
			9	Family Practice	4,233,078				
			10	Gastroenterology	571,040				
			11	Geriatrics	17,739				
			12	Hand Surgery	36,661				
			13	Hematology	305,655				
			14	Internal Medicine	7,848,930				
			15	Infectious Disease	90,658				
			16	Nephrology	232,691				
			17	Neurology	520,499				
			18	Nuclear Medicine	48,339				
			19	Oncology	1,050,033				
			20	Ophthalmology	1,528,521				

			21	Orthopedic Surgery	1,983,838				
			22	Osteopathy (include manipulations)	4,241				
			23	Otology/Laryngo/Rhino/Otolaryngo	776,421				
			24	Pathology	2,232,854				
			25	Pediatrics	5,206,753				
			26	Peripheral Vascular Disease/Surgery	31,706				
			27	Plastic Surgery	102,661				
			28	Physical Medicine & Rehabilitation	539,223				
			29	Proctology	17,013				
			30	Psychiatry	699,842				
			31	Pulmonary Disease	349,872				
			32	Radiology	4,527,894				
			33	Rheumatology	354,065				
			34	Surgical Specialty Not Listed Here	186,496				
			35	Thoracic Surgery	64,468				
			36	Urology	750,147				
			37	Misc Special (pub health/indusl med)	110,485				
			38	Physician w/o Identfd Spec/Not Listd	1,550,447				
			39	Obstetrics/Gynecology	2,607,775				
			40	Acupuncturist	174,888				
			41	Alcohol/Drug Detox Services	24,324				
			42	Ambulance Services	164,194				
			43	Audiologist/Speech Pathologist	165,103				
			44	Chiropractor	2,547,499				
			45	Freestanding Clinic-Not a Govt Agncy	69,244				
			46	Day Care FCLTY-Medical Mental Health	4,622				
			47	Dietitian/Licensed Nutritionist	23,442				
			48	Home Health Provider	108,274				
			49	Advanced Practice Nurse: Anesthetist	56,638				
			50	Advanced Practice Nurse: Midwife	22,237				
			51	Advancd Practice Nurse: Nurse Practitioner	130,083				
			52	Advd Practice Nurse: Psychotherapist	13,695				
			53	Nurse Other than Advanced Practice	85,073				
			54	Occupational Therapist	273,779				
			55	Optometrist	363,563				
			56	Podiatrist	1,137,096				
			57	Physical Therapist	5,582,751				
			58	Psychologist	486,033				

			59	Clinical Social Worker	488,511				
			60	Pub Health/Welf Agy-fed & local govt	27,205				
			61	Voluntary Health Agency	1				
			62	Other Specialty Not Listed Above	585,574				
			63	Respiratory Therapist	745				
			70	General Dentist	36,550				
			71	Endodontist	1,023				
			72	Orthodontist	946				
			73	Oral Surgeon	32,216				
			74	Pedodontist	1,431				
			75	Periodontist	1,162				
			76	Prosthodontist	131				
			80	Freestanding Pharmacy incl grocery	58,297				
			81	Mail Order Pharmacy	10,778				
			82	Independent Laboratory	9,643,445				
			83	Independent Medical Supply Company	1,218,698				
			84	Optician/Optometrist	31,111				
			85	All Other Supplies	1,382				
			90	Freestanding Medical Facility	395,684				
			91	Freestanding Surgical Facility	264,767				
			92	Freestanding Imaging Center	331,682				
			93	Other Facility	2,347,513				
			100	Neonatology	62,084				
			101	Multi-Specialty Medical Practice	18,761				
			102	Mental Health Clinic	37,983				
SPEC_SOURCE	Source of NEWSPEC specialty codes	Numeric	1	Provider with Single Specialty Code	76,275,264				
			2	Provider with Multiple Specialty Codes	4,605,907				
			3	No Provider - Random	5,050,072				
SVCPLACE	Place of Service	Character	Other	Invalid Code	379,861				
			11	Provider's Office	55,024,694				
			12	Patient's Home	1,734,290				
			21	Inpatient Hospital	5,235,401				
			22	Outpatient Hospital	5,163,567				
			23	Emergency Room (Hospital Portion)	2,191,624				
			24	Ambulatory Surgical Center	852,792				
			25	Birthing Center	790				
			26	Military Treatment Facility	2,565				
			32	Nursing Facility	319,552				

			33	Custodial Care Facility	6,060				
			34	Hospice	4,167				
			41	Ambulance-Land	183,271				
			42	Ambulance-Air or Water	2,733				
			50	Federally Qualified Health Center	2				
			51	Inpatient Psychiatric Facility	21,922				
			52	Psychiatric Facility Partial Hospitaln	10,872				
			53	Community Mental Health Center	4,994				
			54	Intermedi Care FCLTY/Mentally Retarded	41				
			55	Residential Subst Abuse Treatment Fcty	3,878				
			56	Psychiatric Residential Treatment Ctr	575				
			60	Mass Immunization Center	776				
			61	Comprehensive Inpat Rehabilitatn Fclty	1,900				
			62	Comprehensive Outpat Rehabilitatn Fcty	151,702				
			65	End Stage Renal Disease Treatment Fcty	53,061				
			71	State or Local Public Health Clinic	2,873				
			81	Independent Laboratory	13,870,154				
			99	Not Coded or Other Unlisted Facility	707,126				
UNITIND	Service Unit Indicator	Character	Blank	Missing	85,676				
			1	Transportation Miles	56,444				
			2	Anesthesia Time Units	299,856				
			3	Visits	82,561,984				
			4	Oxygen Units	90,196				
			5	Blood Units	4,096				
			6	Allergy Tests	82,559				
			Other	Invalid Code	2,370,378				
			8	Anesthesia Time Minutes	366,781				
			9	Not Coded	13,273				
UNITS	Number of Units for a Service	Numeric			0	13,197	-99	2	999
			Missing	Missing	13,197				
			< 1	< 1	8,128,365				
			1	1	72,150,277				
			1.029882813		2				
			1.03984375		2				
			1.05		2				
			1.069921875		1				
			1.099853516		575				
			1.1		2,606				

			1.109960938		1				
			1.119921875		1				
			1.13984375		2				
			1.15		3				
			1.199951172		839				
			1.2		1,919				
			1.219921875		1				
			1.229882813		1				
			1.269921875		2				
			1.299804688		452				
			1.3		3,744				
			1.309960938		1				
			1.329882813		1				
			1.33984375		2				
			1.35		2				
			1.399902344		478				
			1.4		1,794				
			1.409960938		2				
			1.43984375		2				
			1.459960938		1				
			1.469921875		1				
			1.48984375		1				
			1.5		3,813				
			1.519921875		1				
			1.55		2				
			1.58984375		1				
			1.599853516		348				
			1.6		1,562				
			1.609765625		2				
			1.619921875		1				
			1.6296875		2				
			1.65		1				
			1.669921875		3				
			1.699951172		189				
			1.7		3,686				
			1.719921875		1				
			1.7296875		1				
			1.769921875		1				

			1.799804688		191				
			1.8		1,516				
			1.819921875		1				
			1.85		1				
			1.899902344		96				
			1.9		2,598				
			1.959765625		1				
			> 1	> 1	5,612,948				
ZIP_P	Patient Residence ZIP Code	Character	Blank	Value Missing	30,303				
			Reported	Value Reported	85,900,940				
ZIP_S	Service Location ZIP Code	Character	Blank	Value Missing	4,618,241				
			Reported	Value Reported	81,313,002				

Appendix F

Detailed Specifications and SAS Code Used to Impute RVUs

MCDB RVU Imputation Methodology

Year 2006 MCDB Private Payer and MCDB Medicare Services

Section 1. Imputation Input

The imputation program uses data from many external source files. These files are converted to SAS® datasets which are then used as the source for various formats created automatically using the dynamic formatting feature of SAS® (CNTLIN option of PROC FORMAT). The data sources, created SAS® datasets and derived formats are listed below.

1.1 *Dataset PxxxEnc1*.

- The annual MCDB Private Payer dataset contains service-level records from all reporting private payers. Each service-level record in this dataset has an assigned BETOS code. SSS creates a data file containing records from the *PxxxEnc1* data set of the following payers:
  - P020
  - P030
  - P130
  - P131
  - P160
  - P500
  - P510
  - P520
  - P620
  - P680
  - P820
  - P870.

Note: Although only private-payer data are used as the input file for imputation, the imputed values are used for both private payer and Medicare data.

1.2 Dataset PPRRVU06.

The revised 2006 National Physician Fee Schedule Relative Value File issued by CMS in August 2006. Derived formats:

- **$RVU06F :** 2006 Fully-Implemented Facility RVUs for each procedure.
- **$RVU06N :** 2006 Fully-Implemented Non-Facility RVUs for each procedure.
- **$Y06STAT :** 2006 Fee Schedule Status Code for each procedure.
- **$Y06_26M :** All CPT/HCPCS with Modifier=26 (professional component).
- **$Y06_TCM :** All CPT/HCPCS with Modifier=TC (technical component).

1.3 Dataset PPRRVU05.

The 2005 National Physician Fee Schedule Relative Value File issued by CMS on October 12, 2005. Derived formats:

- **$Y05STAT :** 2005 Fee Schedule Status Code for each procedure.

- **$Y05_26M :** All CPT/HCPCS with Modifier=26 (professional component).
- **$Y05_TCM :** All CPT/HCPCS with Modifier=TC (technical component).

1.4 *Dataset ASCPUF06.*

The 2006 Ambulatory Surgical Center (ASC) Base Eligibility Public Use File (PUF) from CMS. Derived formats:

- **$Y06_ASC:** Ambulatory Surgical Center group for each ASC procedure.

1.5 *Dataset CLAB2006.*

The 2006 Clinical Diagnostic Laboratory Fee Schedule (CLAB) Public Use File (PUF). Derived formats:

- **$LABFEE :** The Clinical Diagnostic Laboratory Fee Schedule for the state of Maryland.

Section 2. The Imputation Process

2.1 Construction of the Imputation Dataset

The imputation of total RVUs is done on the entire MCDB.

2.1.1 Keep Only Needed Variables

To reduce processing time the imputation program keeps only the variables needed for computation (these variables are further defined in the MCDB Data Dictionary, Appendix D). For 2006, these are:

ALLOW_ORIG	MOD1
BEGINDT	MOD2
BETOSDET	PATID
BILL_ORIG	PATLIAB_ORIG
BILLTYPE	PAYMENT_ORIG
CPT	PNUM
ENDDT	REIMB_ORIG
F_CAP_ORIG	SVCPLACE
F_CPT	UNITIND
MCRSTAT	UNITS

2.1.2 Apply Major Screens

Only MCDB service records which pass the screens for the non-capitated portion of the annual Practitioner Report Pricing Analysis (F_CAP_ORIG='1') are used in the RVU imputation. Please note that these records automatically have valid CPT/HCPCS procedure codes (F_CPT = '2' or '3').

2.1.3 Apply Additional Deletions

The following services are not used in imputation of total RVUs.

- Dental/Orthodontic services with CPT Procedure Codes = D0001-D7420 and D8001-D8999.
- Bills with allowed amount = 0 or payment <= 1.
- Anesthesia procedures, identified as follows:
  - CPT Procedure Code 00100-01999,99100,99116,99135,99140
  - Anesthesia Time Units (Unit Indicator = 2)
  - Anesthesia-related AMA modifiers (AN,47,P1,P2,P3,P4,P5,30,QX,QZ,23)
- Minor surgery/split surgery bills (AMA Modifier Codes 52, 53, 54, 55, 56, 62, 57, 62 and 65.
- Assistance at surgery (AMA Modifier Codes 80,81 and 82).
- Records containing bilateral or other multiple procedures (AMA Modifier Codes 50 and 51).
- Bills with all financial variables (BILL_ORIG, ALLOW_ORIG, REIMB_ORIG, PATLIAB_ORIG, PAYMENT_ORIG) = 0.
- Credit/Debit adjustment records. For 2006 these are all services with BILLTYPE not equal to 1.

2.2 Steps in the Imputation Methodology

Once the imputation dataset is constructed, the following steps are used to impute total RVUs.

1. Identify facility-based services, using CMS Place of Service Code. These are Inpatient Hospital (21), Outpatient Hospital (22), Hospital portion emergency room (23), ambulatory surgical center (24), Skilled Nursing Facility (31), Hospice, Impatient Psychiatric facilities (51), and Partial Hospitalization psychiatric facilities (53). All other services are classified as non-facility based.

2. Create a new modifier (NEWMOD) code with reference to services with a TC or 26 modifier (Note: These rules were first used for the Year 2000 imputations and are based on inspection of the 1999 data). Specifically:

   - For all services with an AMA modifier code (MOD1 or MOD2) of TC or SG

     - ❑ Set the new modifier to 26 for all 93544 (aortography) and 93545 (selective coronary angiography) and whenever the current modifier is 26.

     - ❑ Set the new modifier to TC for all current modifiers.

     - ❑ Set the new modifier to 26 to all facility-based claims not assigned above.

   - Set the new modifier to blank for all remaining services.

3. Assign a Medicare Ambulatory Surgical Center group for all ASC related claims (service place=24). A record with a service place=24 that is not part of Medicare ASC procedures list (assigned ASC group=missing) is labeled as a non-facility service.

4. Using facility and non-facility based RVUs, assign RVU depending on the modifier. Some exceptions:

   - For clinical lab related claims, RVU is the lab fee divided by the Medicare conversion factor (37.8975 in 2006).

   - For therapy codes (97001-97546), use non-facility fully implemented RVU.

   - If payment per RVU is high and the number of units is reported, then payment should be divided by units (the claims has multiple units that make the procedure much more expensive).

   Please note that in 2001, a special rule for screening mammography (CPT 76092) was used. The rate was based on Medicare maximum allowed charge for procedure divided by overall physician services conversion factor for 2001(38.2581). Since 2002, this CPT code is on the regular Medicare fee schedule file.

5. Find out the status code using 2005 and 2006 RVU data sets.

6. Keep only one record per unique recipient-service by sorting using the key PIDBDG, BEGINDT, CPT, NEWMOD, and PAYMENT_ORIG (descending) and dropping all but the service record with the highest recorded payment

7. Compute the mean payment for each CPT/HCPCS code (with the detailed BETOS carried along).

8. Use the mean payment to delete outliers using the following rule:

   With the exception of procedures with status code =C (Carriers price the code), delete any service with payment below 1/3 of the mean or above 6 times the mean is deleted. Using RVU, delete any service with payment below ¼ of the RVU times the conversion factor or above 8 times the RVU multiplied by the conversion factor is deleted.

9. Summarize RVU and Payment by BETOSDET (detailed BETOS code) and calculate Payment per RVU (or Conversion Factor).

10. Compute Mean Payment by CPT and New Modifier (BETOSDET is carried along).

11. Compute imputed RVU by multiplying the Mean Payment by the inverse of the Conversion Factor.

12. Any imputed RVU greater than 200 is considered as outlier and therefore set to missing.

13. RVU for drug procedures was not imputed.

14. RVU for durable medical equipment not imputed.

15. RVU for less common procedure (less than 10 services in the database) was not imputed.

Section 3. Imputation Output

3.1 Dataset IMPRVU06.

Contains imputed total RVUs for each HCPCS/CPT code available in the database but not available in the 2005-2006 National Physician Fee Schedule Relative Value File.

```
%jobstart;

 ******************************************************;
 * Program: Compute_Imputed_RVU_&currentyear.sas       *;
 * Programmer: Adrien Ndikumwami/Chris Hogan           *;
 * Date: December 05, 2001                             *;
 *-----------------------------------------------------*;
 *  This program imputes RVUs for procedures that do   *;
 *  not have an assigned RVU by CMS                    *;
 *  The following format is needed to run the program: *;
 *  Format_RVU_Imputed_&currentyear..sas               *;
 *  ADAPTED TO 2001 ON NOVEMBER 14, 2002               *;
 *  ADAPTED TO 2002 ON SEPTEMBER 05, 2003              *;
 *  ADAPTED TO 2003 ON JULY 17, 2004                   *;
 *  ADAPTED TO 2004 ON JULY 19, 2005                   *;
 *  ADAPTED TO 2005 ON JULY 11, 2006                   *;
 *  ADAPTED TO 2006 ON AUGUST 6, 2007                  *;
 ******************************************************;

libname RVU "I:\MCdb\DataCol\Formats\RVUs";
libname IN  "I:\MCdb\Datacol\Formats\RVUs\imputation";

%let sasDTin = rvu_mcdb;

%let lastyear = 2005;
%let lyr = 05;
%let currentyear = 2006;
%let cyr = 06;

* GENERATE FOUR FORMATS TO FLAG CODES FOR POSSIBLE TC OR 26 MODIFIER ;
* THESE ARE JUST CODES IN &lastyear. AND &currentyear. MFS THAT WERE LISTED WITH ;
* 26 OR TC MODIFIER ;
* $Y&lyr._26M, $Y&lyr._TCM, $Y&cyr._26M, $Y&cyr._TCM ;

* OTHER FORMATS INCLUDE MEDICARE PHYSICIAN FEE SCHEDULE RVUS ;
* MEDICARE LAB FEES -- USE THEM TO GAP-FILL THE LAB DATA ;
* ASC DATA TO CHECK ON HOW ASC FACILITY FEES COMPARE TO PHYSICIAN FEES ;
* AND THE &lastyear. AND &currentyear. STATUS CODE ON THE MEDICARE FEE SCHEDULE, TO ;
* FLAG CODES THAT ARE PRICED BY CARRIER REPORT AND THEREFORE SHOULD NOT ;
* BE SUBJECT TO THE ACTUARY TRIM FOR OUTLIERS ;

options nosource2;
%INCLUDE "I:\mcdb\datacol\Formats\Format_RVU_&currentyear._ForImputation.sas";
%INCLUDE "I:\mcdb\datacol\Formats\Format_cpt_description.sas";
%include "I:\MCdb\datacol\Formats\RVUs\Programs\Format_Drug_Proc.sas";
options source2;

proc format;
  value $f_rvu
        "0" = "0-Imputed"
        "1" = "1-Imputed RVU > 200"
        "2" = "2-Drug Procedure"
        "3" = "3-DME"
        "4" = "4-Small Sample";
run;

************************************************************** ;
* PROCESS&currentyear. 10% SAMPLE PRIVATE PAYER DATA ;
************************************************************** ;

DATA ONE (keep=BETOSDET  CPT          ENDDT        MCR_STAT     UNITS
               NEWMOD    PAYMENT_ORIG RVU          PIDBDG);
  SET IN.&sasDTin (Keep= ALLOW_ORIG   BETOSDET     BILL_ORIG    BILLTYPE
                         CPT          ENDDT        F_CAP_ORIG
                         F_CPT        MOD1         MOD2         PATID
                         PATLIAB_ORIG PAYMENT_ORIG PNUM         REIMB_ORIG
                         SVCPLACE     UNITIND      UNITS        PIDBDG
                         PATID        BIRTHYR      BIRTHMO      SEX
                   obs=max);
  where f_cap_orig ne '1' and f_cpt in ('2','3');
```

```
* ********************************************************** ;
* ********************************************************** ;
* DROP RECORDS NOT USED IN STUDY ;


  **Drop CPT for Dental/Orthodontic**;
  IF      (CPT GE 'D0001' AND CPT LE 'D7420') OR
          (CPT GE 'D8001' AND CPT LE 'D8999') THEN DELETE;

  ** Drop ALLOW_ORIG amount=0 or PAYMENT_ORIG LE 1**;
  IF (ALLOW_ORIG LE 0) or (PAYMENT_ORIG LE 1)  THEN DELETE;

  **Drop Anesthesia procedures**;

  IF  (CPT GE '00100' AND CPT LE '01999') OR
       CPT IN ('99100','99116','99135','99140') OR
       UNITIND IN ('2','8') or
       /*
       spec1 in ('004','049') or
       spec2 in ('004','049') or
       spec3 in ('004','049') or
       */
       mod1 in ('AN','47','P1','P2','P3','P4','P5','30','QX','QZ','23') OR
       mod2 in ('AN','47','P1','P2','P3','P4','P5','30','QX','QZ','23')  THEN  DELETE;

  /*
  **Drop easility identified facility bills**;
  IF  SPEC1 IN ('090','091','092','093') OR
      MOD1 = 'SG' OR MOD2 = 'SG'       THEN DELETE;
  */

  **Drop identified minor surgery/split surgery bills**;
  IF    MOD1 IN ('52','53','54','55','56','62','57','62','65') OR
        MOD2 IN ('52','53','54','55','56','62','57','62','65')
        THEN DELETE;

  **Drop Assistance at surgery**;
  IF    MOD1 IN ('80','81','82') OR
        MOD2 IN ('80','81','82')
        THEN DELETE;

  **Drop records with bilateral and multiple procedures ;
  IF MOD1 = '51' OR MOD2 = '51'  THEN DELETE;
  IF MOD1 = '50' OR MOD2 = '50'  THEN DELETE;

  **Drop records with all financial variables=0**;
  IF BILL_ORIG NE 0 AND (ALLOW_ORIG = 0 AND REIMB_ORIG = 0 AND PATLIAB_ORIG = 0 AND PAYMENT_ORIG
= 0)
        THEN DELETE;

  **Keep only full debit or final bill records**;
   if billtype ='1';

  *******************************************************
  * WORK ON MODIFIERS FOR 26 AND BLANK FOR RADIOLOGY    *
  * IMPUTE THEM BY SITE OF SERVICE                      *
  *******************************************************;

   LENGTH ID_26_&lyr. ID_TC_&lyr. ID_26_&cyr. ID_TC_&cyr. $ 2 NEWMOD $ 2 ;
   LABEL   ID_26_&lyr. = 'CPT FOR WHICH MOD OF 26 APPEARS MFS &lastyear.'
           ID_TC_&lyr. = 'CPT FOR WHICH MOD OF TC APPEARS MFS &lastyear.'
           ID_26_&cyr. = 'CPT FOR WHICH MOD OF 26 APPEARS MFS &currentyear.'
           ID_TC_&cyr. = 'CPT FOR WHICH MOD OF TC APPEARS MFS &currentyear.';

   ID_26_&lyr. = PUT(CPT,$Y&lyr._26M.) ;
   ID_TC_&lyr. = PUT(CPT,$Y&lyr._TCM.) ;
   ID_26_&cyr. = PUT(CPT,$Y&cyr._26M.) ;
   ID_TC_&cyr. = PUT(CPT,$Y&cyr._TCM.) ;
```

```
   NEWMOD = 'XX' ;

  *Identify facility-based claims(IP,OP,ASC,SNF);
  IF SVCPLACE IN ('01',' 1','1 ',
                  '02','2 ',' 2',
                  '21','22','23','24',
                  '31','34',
                  '51','52')
    THEN FACILITY = 1 ;
    ELSE FACILITY = 0 ;

  * ONLY PROCESS THE CLAIMS WHERE 26 OR TC MODIFIER MIGHT APPLY ;
  * special exceptions noted by inspection of the data;
  * 93544, 93545 are out of line -- what is the solution to that ;
  * ASSIGN BASED ON EXISTING MODIFIER OR, BARRING THAT, ON PLACE OF SVC ;
  * MENTION OF INDIVIDUAL CPTS BASED ON INSPECTION OF &lastyear. DATA ;
  * NOT SURE IF THIS RUNS CORRECTLY (CPT VERSUS CPT) ;

   IF ID_26_&lyr. = '26' OR ID_26_&cyr. = '26' OR
      ID_TC_&lyr. = 'TC' OR ID_TC_&cyr. = 'TC' THEN DO ;
        if cpt in ('93555','93556') then newmod = '26' ;
        else if mod1 = '26' or mod2 = '26' then newmod = '26' ;
        ELSE IF   MOD1 = 'TC' OR MOD2 = 'TC' THEN NEWMOD = 'TC' ;
        else if FACILITY = 1 THEN NEWMOD = '26' ;
        ELSE NEWMOD = '  ' ;
   END ;
   IF NEWMOD NOT IN ('26','TC') THEN NEWMOD = '  ' ;



     ***DETERMINE Y&currentyear. MEDICARE ASC based on the file ASCPUF&cyr. from CMS(HCFA)*;
     LENGTH ASCGRP $ 1 ;
     IF SVCPLACE = '24' THEN  ASCGRP = PUT(CPT,$Y&cyr._ASC.) ;

   * ADD &currentyear. FULLY IMPLEMENTED RVUS, AND PUT ON LAB FEES AS WELL ;
    LENGTH CPT_4_RVU $ 7 ;

    CPT_4_RVU = LEFT(TRIM(CPT))|| LEFT(TRIM(NEWMOD));    **CPT || PRICING MOD FOR RVU MERGE**;
    RVU_FAC = INPUT(put(CPT_4_RVU,$RVU&cyr.F.),6.2); *Y&currentyear. FULLY IMPLEMENTED FACILITY
RVU*;
    IF RVU_FAC LE 0 THEN RVU_FAC = . ;
    RVU_NFAC = INPUT(put(CPT_4_RVU,$RVU&cyr.N.),6.2); *Y&currentyear. FULLY IMPLEMENTED
NONFACILITY RVU*;
    IF RVU_NFAC LE 0 THEN RVU_NFAC = . ;
    RVU_LAB = INPUT(PUT(CPT,$L00901B.),6.2) ;   *L00901B is for MD fees*;
    IF RVU_LAB NE . THEN RVU_LAB = RVU_LAB/37.8975; *APPROX  RVU FOR LAB BASED ON MEDICARE LFS*;
    IF RVU_LAB LE 0 THEN RVU_LAB = . ;


  * IMPLEMENT RULE FOR MEDICARE RVUS BY SITE ;
  IF FACILITY = 1 THEN DO ;
      IF SVCPLACE = '24' THEN DO ;
            IF ASCGRP NE '' THEN RVU = RVU_FAC ;
            ELSE                 RVU = RVU_NFAC ;
      END;
      ELSE DO ;
                                 RVU = RVU_FAC ;
      END ;
  END ;
  ELSE DO ;
                                 RVU = RVU_NFAC ;
  END ;

  * SPECIAL MEDICARE RULE FOR THERAPY CODES ;
  * THIS IS MY BEST GUESS AT THERAPY BILLED UNDER MEDICARE FEE SCHEDULE;
  IF CPT GT '97001' AND CPT LE '97456'
                        THEN    RVU = RVU_NFAC ;

  **Impute LAB RVUS using Medicare LAB fee data**;
  IF RVU = . AND RVU_LAB NE . THEN RVU = RVU_LAB ;
```

```
  * SPECIAL CHECK ON BILLS WITH MULTIPLE UNITS ;
  * if dividing by n of units gives a more reasonable answer, then do it ;
  IF PAYMENT_ORIG>0 AND RVU>0 AND UNITS>1 THEN DO;
     IF PAYMENT_ORIG/RVU > 60 AND (30 < (PAYMENT_ORIG/UNITS)/RVU < 60) THEN
                        PAYMENT_ORIG  = PAYMENT_ORIG/UNITS ;
  END ;
  * ADD MEDICARE PAYMENT STATUS CODE, SO THAT YOU LATER CAN ID THE CODES ;
  * THAT ARE MISCELLANEOUS CODES PRICED BY CARRIER REPORT ;
  * (THAT IS STATUS CODE C ON THE MEDICARE FEE SCHEDULE FILE) ;
  LENGTH MCR_STAT $ 1 ;
  MCR_STAT = PUT(CPT,$RVU&cyr.ST.) ;
  IF MCR_STAT = '' THEN MCR_STAT = PUT(CPT,$RVU&lyr.ST.) ;

RUN ;


  ******************************************************************;
  * NOW CHECK FOR MULTIPLE RECORDS FOR SAME PERSON DATE CPT          *;
  * IF FOUND, TAKE MOST EXPENSIVE ONE                                *;
  ******************************************************************;

PROC SORT DATA = ONE OUT=TWO ;
  BY PIDBDG ENDDT CPT NEWMOD DESCENDING PAYMENT_ORIG ;
RUN ;

DATA THREE (DROP=PIDBDG ENDDT) ;
  SET TWO ;
  BY PIDBDG ENDDT CPT NEWMOD DESCENDING PAYMENT_ORIG ;
  IF FIRST.NEWMOD;
RUN;

  * IMPOSE THE HCFA ACTUARIES' TRIM ;
  * ACTUARIES' SCREEN STEP GOES HERE ;
  * SCREEN OUT BILL WITH ALLOWED LESS THAN ONE THIRD OR MORE THAN THREE ;
  * TIMES THE MEAN ALLOWED CHARGE FOR THE ENTIRE DATABASE ;
  * AGAIN, FOR PRODUCTION, RESTRICT TO ONLY GOOD RECORDS ;

PROC SORT DATA = THREE ;
    BY CPT NEWMOD ;
RUN ;

PROC MEANS NOPRINT MEAN DATA = THREE ;
   BY CPT NEWMOD ;
   VAR PAYMENT_ORIG ;
   ID BETOSDET ;
   OUTPUT OUT = CPTSUM MEAN = PAYMENT_ORIG_ ;
RUN ;

* note asymmetric actuaries trim ;
DATA FOUR ;
  MERGE THREE  CPTSUM ;
  BY CPT NEWMOD ;

* NOTE THAT DUE TO DROPPER RESTRICTION, THERE MAY BE RECORDS HERE ;
* WITH MISSING MEAN ALLOWED CHARGE AND SUCH ;
* BE SURE TO TAKE THAT INTO ACCOUNT IN THE CODE ;

* SOME CODES ARE MISCELLANEOUS CATEGORIES (TYPICALLY XXX99 OR XXX49) ;
* AND YOU CANNOT PROPERLY APPLY THE ACTUARIES' TRIM TO THOSE ;
* ALSO, NO NEED TO APPLY TO RECORDS TO BE DROPPED FROM PRICE ANALYSIS ;

  IF MCR_STAT NE 'C' THEN DO ;
    IF PAYMENT_ORIG < .33*PAYMENT_ORIG_ then DELETE;            **< .33*MEAN PAYMENT_ORIG**;
    IF PAYMENT_ORIG >   6*PAYMENT_ORIG_ THEN DELETE;            **> 6*MEAN PAYMENT_ORIGED**;
    IF RVU NE . AND PAYMENT_ORIG < .25*37.8975*RVU THEN DELETE;
    IF RVU NE . AND PAYMENT_ORIG >   8*37.8975*RVU THEN DELETE;
  END ;
RUN ;

PROC SUMMARY DATA = FOUR;
WHERE RVU >0 AND PAYMENT_ORIG >0;
```

```
CLASS BETOSDET ;
VAR RVU PAYMENT_ORIG ;
OUTPUT OUT = TEMP1 SUM = RVU PAYMENT_ORIG ;
RUN ;

DATA TEMP1;
  SET TEMP1;
  CF = PAYMENT_ORIG/RVU ;   ***PMT PER RVU, BY BETOS DET***;
  INVCF = 1/CF ;       ***RVU PER PAYMENT_ORIG, BY BETOS DET***;
RUN ;

PROC PRINT DATA=TEMP1 UNIFORM ;
title "Summary stats used for imputation";
RUN ;

* FIND CODES WITHOUT RVUS ;
PROC SUMMARY MISSING NWAY DATA = FOUR;
   WHERE PAYMENT_ORIG NE . AND RVU LE 0;
   CLASS CPT NEWMOD ;
   VAR PAYMENT_ORIG UNITS;
   ID BETOSDET ;
   OUTPUT OUT = MEANPAY (DROP=_TYPE_ RENAME=(_FREQ_=SERVICES)) MEAN = MEANPAY SUM=TPAY TUNITS ;
RUN ;

PROC SORT  DATA = TEMP1 ;
  BY BETOSDET ;
RUN;

DATA TEMP2;
 SET TEMP1 ;
 IF _TYPE_ = 0 ;
 KEEP INVCF ; * AGGREGATE ;
RUN;

PROC SORT  DATA = MEANPAY ;
  BY BETOSDET ;
run;
DATA MEANPAY2 ;
  SET MEANPAY ;
  RETAIN AGGINVCF ;
  LABEL AGGINVCF = 'AGGREGATE INVERSE CONVERSION FACTOR' ;
  IF _N_ = 1 THEN SET TEMP2 (RENAME = (INVCF = AGGINVCF)) ;
RUN ;


DATA rvu.IMPRVU&cyr.;
  LENGTH F_RVOUT $1;
  MERGE MEANPAY2(IN = IN1) TEMP1 (IN = IN2 KEEP = BETOSDET INVCF) ;
  BY BETOSDET ;
  IF IN1 ;
  * USE AGGREGATE IF THERE IS NO OTHER INFORMATION ;
  IF INVCF = . THEN INVCF = AGGINVCF ;
  RVU = MEANPAY*INVCF ;

/** Decision by Curt Mueller to set any RVU>200 to missing--outlier 9/3/2004 **/

/** Also, per Curt Mueller's e-mail dated 07/23/2005, we re-set RVU to missing **/
/** for Drugs codes (Procedure codes starting with 'J' and some codes starting **/
/** with 'C', 'Q', or 'S') and DME codes                                       **/

  f_rvout = '0';
  if substr(betosdet, 1, 1) = "D" then do;
     rvu = .;
     f_rvout = '3';
  end;
  else if substr(cpt, 1, 1) = "J" or put(cpt, $d_oth.) = '1' or betosdet = "O1E" then do;
     rvu = .;
     f_rvout = '2';
  end;
  else if rvu > 200 then do;
     rvu = .;
```

```
     f_rvout = '1';
  end;
  else if services < 10 then do;
     rvu = .;
     f_rvout = '4';
  end;
  cptdesc=put(cpt,$cptdesc.);

RUN;

PROC PRINT DATA=RVU.IMPRVU&cyr. UNIFORM ;
SUM SERVICES  TUNITS;
format f_rvout $f_rvu.;
title "Imputed RVUs";
RUN ;

PROC PRINT DATA=RVU.IMPRVU&cyr. UNIFORM ;
where f_rvout='1';
SUM SERVICES TUNITS;
title "Outlier Imputed RVUs set to missing";
RUN ;


proc freq data = rvu.imprvu&cyr.;
  tables f_rvout / list missing;
  format f_rvout $f_rvu.;
  title "Frequency of RVU Flag";
run;

proc freq data = rvu.imprvu&cyr.;
  tables f_rvout / list missing;
  weight services;
  format f_rvout $f_rvu.;
  title "Frequency of RVU Flag - Weight by SERVICES";
run;

proc means data = rvu.imprvu&cyr. mean p25 p50 p75 p90 min max std;
  class f_rvout;
  var meanpay;
  weight services;
  title "Mean of RVU Flag - Weight by SERVICES";
run;


%jobend;

dm "log;    print file=I:\MCdb\DataCol\Formats\RVUs\log_out\Compute_Imputed_RVU_&currentyear..log
replace";
dm "output; print file=I:\MCdb\DataCol\Formats\RVUs\log_out\Compute_Imputed_RVU_&currentyear..lst
replace";
```

Process of Assignment and Imputation of Relative Value Units

Program / Program Description	Input Dataset Used	Formats Used	Output Dataset Created
I:\MCdb\datacol\Formats\RVUs\imputation**rvu_imputation_subset.sas**	I:\MCdb\datacol\Pxxx\sasdata\ **Pxxx_enc1.sas7bdat**		I:\MCdb\datacol\Formats\RVUs\imputation\ **rvu_mcdb.sas7bdat**
•► This program combines records from ENC1 datasets of large payers to create a dataset, rvu_mcdb.sas7bdat, which is used as the input file for RVU Imputation. (For Year 2006 the Enc1 datasets of payers P020, P030, P130, P131, P160, P500, P510, P520, P620, P680, P820, & P870 were used).			
I:\MCdb\datacol\Formats\RVUs\Programs**Compute_Imputed_RVU_2006.sas**	I:\MCdb\datacol\Formats\RVUs\imputation\ **rvu_mcdb.sas7bdat**	I:\mcdb\datacol\Formats\ **Format_RVU_06_ForImputation.sas**	I:\MCdb\DataCol\Formats\RVUs\ **IMPRVU06.sas7bdat**
•► This Program imputes RVUs for procedures in rvu_mcdb.sas7bdat that do not have CMS assigned RVUs.		I:\mcdb\datacol\Formats\ **Format_cpt_description.sas ***	
•► Conversion Factor (CF) must be updated before running the program. (Source of CF factor is pprvuxx.sas7bdat).		I:\MCdb\datacol\Formats\RVUs\Programs\ **Format_Drug_Proc.sas**	
I:\MCdb\datacol\Formats\RVUs\Programs**Compute_Impute_RVU_Components_2006.sas**	I:\MCdb\datacol\Formats\RVUs\imputation\ **rvu_mcdb.sas7bdat**	I:\mcdb\datacol\Formats\ **Format_RVU_2006_COMP.SAS**	I:\MCdb\DataCol\Formats\RVUs\ **RVUCOMP06.sas7bdat**
•► This Program assigns RVU Components (Work, Practice, and Malpractice) to valid procedures in dataset rvu_mcdb.sas7bdat.			
•► Compute Medicare Payment Rates.			
•► Impute RVU components if missing.			
•► Conversion Factor (CF) must be updated before running the program. (Source of CF factor is pprvuxx.sas7bdat).			
•► This program is run twice. • 1st run - Use previous year's GPCI values, to get the average GPCI values to be used for the current year. • 2nd run - Using the average GPCI from the first run as the gap fill values, impute the RVU Components.			
I:\MCdb\datacol\Pxxx\Programs**rvu_imputation_subset_ck.sas**	I:\MCdb\datacol\Pxxx\sasdata\		I:\MCdb\datacol\Pxxx\sasdata\
•► This program combines records from ENC5 datasets of large payers to create a dataset, rvu_mcdb_ck.sas7bdat, which is used as the input file for RVU Imputation checking up only. (For Year 2006 the Enc5datasets of payers P020, P030, P130, P131, P160, P500, P510, P520, P620, P680, P820, & P870 were used).	**Pxxx_Enc5.sas7bdat**		**rvu_mcdb_ck.sas7bdat**

I:\MCdb\datacol\Formats\RVUs\Programs**RVU_Assigned_Check_2006.sas**	I:\MCdb\datacol\Formats\RVUs\imputation\ **rvu_mcdb_ck.sas7bdat**	I:\MCdb\datacol\Formats\PERMANENT\RVU_formats**Format_RVU_2006.sas**	I:\MCdb\datacol\Formats\RVUs\imputation\ **rvu_mcdb_imp.sas7bdat**
•► RVUs (Work, Practice, and Malpractice) are assigned to valid procedures.			
•► Medicare Payment Rates are computed.			
•► The components are estimated for procedures with imputed total RVUs.			
•► Conversion Factor (CF) must be updated before running the program. (Source of CF factor is pprvuxx.sas7bdat).			
•► Facility ratios must be updated before the program is run. (Source of facility ratios is the output from Compute_Impute_RVU_components_2006.sas).			
Note:			
•► RVU_Assigned_Check_2006.sas Program is just a Test program for the RVU part of Pxxx_Enc5.sas			
•► It will not affect Pxxx_enc5.sas even if this program is not run) .			
•► The description for RVU_Assigned_Check_2006.sas is same as the description of Pxxx_Enc5.sas			

```
%jobstart;

 ******************************************************;
 * Program: Compute_Imputed_RVU_&currentyear.sas       *;
 * Programmer: Adrien Ndikumwami/Chris Hogan           *;
 * Date: December 05, 2001                             *;
 *-----------------------------------------------------*;
 *  This program imputes RVUs for procedures that do   *;
 *  not have an assigned RVU by CMS                    *;
 *  The following format is needed to run the program: *;
 *  Format_RVU_Imputed_&currentyear..sas               *;
 *  ADAPTED TO 2001 ON NOVEMBER 14, 2002               *;
 *  ADAPTED TO 2002 ON SEPTEMBER 05, 2003              *;
 *  ADAPTED TO 2003 ON JULY 17, 2004                   *;
 *  ADAPTED TO 2004 ON JULY 19, 2005                   *;
 *  ADAPTED TO 2005 ON JULY 11, 2006                   *;
 *  ADAPTED TO 2006 ON AUGUST 6, 2007                  *;
 ******************************************************;

libname RVU "I:\MCdb\DataCol\Formats\RVUs";
libname IN  "I:\MCdb\Datacol\Formats\RVUs\imputation";

%let sasDTin = rvu_mcdb;

%let lastyear = 2005;
%let lyr = 05;
%let currentyear = 2006;
%let cyr = 06;

* GENERATE FOUR FORMATS TO FLAG CODES FOR POSSIBLE TC OR 26 MODIFIER ;
* THESE ARE JUST CODES IN &lastyear. AND &currentyear. MFS THAT WERE LISTED WITH ;
* 26 OR TC MODIFIER ;
* $Y&lyr._26M, $Y&lyr._TCM, $Y&cyr._26M, $Y&cyr._TCM ;

* OTHER FORMATS INCLUDE MEDICARE PHYSICIAN FEE SCHEDULE RVUS ;
* MEDICARE LAB FEES -- USE THEM TO GAP-FILL THE LAB DATA ;
* ASC DATA TO CHECK ON HOW ASC FACILITY FEES COMPARE TO PHYSICIAN FEES ;
* AND THE &lastyear. AND &currentyear. STATUS CODE ON THE MEDICARE FEE SCHEDULE, TO ;
* FLAG CODES THAT ARE PRICED BY CARRIER REPORT AND THEREFORE SHOULD NOT ;
* BE SUBJECT TO THE ACTUARY TRIM FOR OUTLIERS ;

options nosource2;
%INCLUDE "I:\mcdb\datacol\Formats\Format_RVU_&currentyear._ForImputation.sas";
%INCLUDE "I:\mcdb\datacol\Formats\Format_cpt_description.sas";
%include "I:\MCdb\datacol\Formats\RVUs\Programs\Format_Drug_Proc.sas";
options source2;

proc format;
  value $f_rvu
        "0" = "0-Imputed"
        "1" = "1-Imputed RVU > 200"
        "2" = "2-Drug Procedure"
        "3" = "3-DME"
        "4" = "4-Small Sample";
run;

**************************************************************** ;
* PROCESS&currentyear. 10% SAMPLE PRIVATE PAYER DATA ;
**************************************************************** ;

DATA ONE (keep=BETOSDET  CPT          ENDDT        MCR_STAT     UNITS
               NEWMOD    PAYMENT_ORIG RVU          PIDBDG);
  SET IN.&sasDTin (Keep= ALLOW_ORIG   BETOSDET     BILL_ORIG    BILLTYPE
                         CPT          ENDDT        F_CAP_ORIG
                         F_CPT        MOD1         MOD2         PATID
                         PATLIAB_ORIG PAYMENT_ORIG PNUM         REIMB_ORIG
                         SVCPLACE     UNITIND      UNITS        PIDBDG
                         PATID        BIRTHYR      BIRTHMO      SEX
                    obs=max);
  where f_cap_orig ne '1' and f_cpt in ('2','3');
```

```
* ********************************************************** ;
* ********************************************************** ;
* DROP RECORDS NOT USED IN STUDY ;


  **Drop CPT for Dental/Orthodontic**;
  IF      (CPT GE 'D0001' AND CPT LE 'D7420') OR
          (CPT GE 'D8001' AND CPT LE 'D8999') THEN DELETE;

  ** Drop ALLOW_ORIG amount=0 or PAYMENT_ORIG LE 1**;
  IF (ALLOW_ORIG LE 0) or (PAYMENT_ORIG LE 1)  THEN DELETE;

  **Drop Anesthesia procedures**;

  IF  (CPT GE '00100' AND CPT LE '01999') OR
       CPT IN ('99100','99116','99135','99140') OR
       UNITIND IN ('2','8') or
       /*
       spec1 in ('004','049') or
       spec2 in ('004','049') or
       spec3 in ('004','049') or
       */
       mod1 in ('AN','47','P1','P2','P3','P4','P5','30','QX','QZ','23') OR
       mod2 in ('AN','47','P1','P2','P3','P4','P5','30','QX','QZ','23')  THEN  DELETE;

  /*
  **Drop easility identified facility bills**;
  IF  SPEC1 IN ('090','091','092','093') OR
      MOD1 = 'SG' OR MOD2 = 'SG'       THEN DELETE;
  */

  **Drop identified minor surgery/split surgery bills**;
  IF    MOD1 IN ('52','53','54','55','56','62','57','62','65') OR
        MOD2 IN ('52','53','54','55','56','62','57','62','65')
        THEN DELETE;

  **Drop Assistance at surgery**;
  IF    MOD1 IN ('80','81','82') OR
        MOD2 IN ('80','81','82')
        THEN DELETE;

  **Drop records with bilateral and multiple procedures ;
  IF MOD1 = '51' OR MOD2 = '51'  THEN DELETE;
  IF MOD1 = '50' OR MOD2 = '50'  THEN DELETE;

  **Drop records with all financial variables=0**;
  IF BILL_ORIG NE 0 AND (ALLOW_ORIG = 0 AND REIMB_ORIG = 0 AND PATLIAB_ORIG = 0 AND PAYMENT_ORIG
= 0)
        THEN DELETE;

  **Keep only full debit or final bill records**;
   if billtype ='1';

  *****************************************************
  * WORK ON MODIFIERS FOR 26 AND BLANK FOR RADIOLOGY    *
  * IMPUTE THEM BY SITE OF SERVICE                      *
  *****************************************************;

   LENGTH ID_26_&lyr. ID_TC_&lyr. ID_26_&cyr. ID_TC_&cyr. $ 2 NEWMOD $ 2 ;
   LABEL   ID_26_&lyr. = 'CPT FOR WHICH MOD OF 26 APPEARS MFS &lastyear.'
           ID_TC_&lyr. = 'CPT FOR WHICH MOD OF TC APPEARS MFS &lastyear.'
           ID_26_&cyr. = 'CPT FOR WHICH MOD OF 26 APPEARS MFS &currentyear.'
           ID_TC_&cyr. = 'CPT FOR WHICH MOD OF TC APPEARS MFS &currentyear.';

   ID_26_&lyr. = PUT(CPT,$Y&lyr._26M.) ;
   ID_TC_&lyr. = PUT(CPT,$Y&lyr._TCM.) ;
   ID_26_&cyr. = PUT(CPT,$Y&cyr._26M.) ;
   ID_TC_&cyr. = PUT(CPT,$Y&cyr._TCM.) ;

   NEWMOD = 'XX' ;
```

```
  *Identify facility-based claims(IP,OP,ASC,SNF);
  IF SVCPLACE IN ('01',' 1','1 ',
                  '02','2 ',' 2',
                  '21','22','23','24',
                  '31','34',
                  '51','52')
    THEN FACILITY = 1 ;
    ELSE FACILITY = 0 ;

  * ONLY PROCESS THE CLAIMS WHERE 26 OR TC MODIFIER MIGHT APPLY ;
  * special exceptions noted by inspection of the data;
  * 93544, 93545 are out of line -- what is the solution to that ;
  * ASSIGN BASED ON EXISTING MODIFIER OR, BARRING THAT, ON PLACE OF SVC ;
  * MENTION OF INDIVIDUAL CPTS BASED ON INSPECTION OF &lastyear. DATA ;
  * NOT SURE IF THIS RUNS CORRECTLY (CPT VERSUS CPT) ;

   IF ID_26_&lyr. = '26' OR ID_26_&cyr. = '26' OR
      ID_TC_&lyr. = 'TC' OR ID_TC_&cyr. = 'TC' THEN DO ;
        if cpt in ('93555','93556') then newmod = '26' ;
        else if mod1 = '26' or mod2 = '26' then newmod = '26' ;
        ELSE IF   MOD1 = 'TC' OR MOD2 = 'TC' THEN NEWMOD = 'TC' ;
        else if FACILITY = 1 THEN NEWMOD = '26' ;
        ELSE NEWMOD = '  ' ;
   END ;
   IF NEWMOD NOT IN ('26','TC') THEN NEWMOD = '  ' ;



     ***DETERMINE Y&currentyear. MEDICARE ASC based on the file ASCPUF&cyr. from CMS(HCFA)*;
     LENGTH ASCGRP $ 1 ;
     IF SVCPLACE = '24' THEN  ASCGRP = PUT(CPT,$Y&cyr._ASC.) ;

   * ADD &currentyear. FULLY IMPLEMENTED RVUS, AND PUT ON LAB FEES AS WELL ;
    LENGTH CPT_4_RVU $ 7 ;

    CPT_4_RVU = LEFT(TRIM(CPT))|| LEFT(TRIM(NEWMOD));    **CPT || PRICING MOD FOR RVU MERGE**;
    RVU_FAC = INPUT(put(CPT_4_RVU,$RVU&cyr.F.),6.2); *Y&currentyear. FULLY IMPLEMENTED FACILITY
RVU*;
    IF RVU_FAC LE 0 THEN RVU_FAC = . ;
    RVU_NFAC = INPUT(put(CPT_4_RVU,$RVU&cyr.N.),6.2); *Y&currentyear. FULLY IMPLEMENTED
NONFACILITY RVU*;
    IF RVU_NFAC LE 0 THEN RVU_NFAC = . ;
    RVU_LAB = INPUT(PUT(CPT,$L00901B.),6.2) ;   *L00901B is for MD fees*;
    IF RVU_LAB NE . THEN RVU_LAB = RVU_LAB/37.8975; *APPROX  RVU FOR LAB BASED ON MEDICARE LFS*;
    IF RVU_LAB LE 0 THEN RVU_LAB = . ;


  * IMPLEMENT RULE FOR MEDICARE RVUS BY SITE ;
  IF FACILITY = 1 THEN DO ;
      IF SVCPLACE = '24' THEN DO ;
            IF ASCGRP NE '' THEN RVU = RVU_FAC ;
            ELSE                 RVU = RVU_NFAC ;
      END;
      ELSE DO ;
                                 RVU = RVU_FAC ;
      END ;
  END ;
  ELSE DO ;
                                 RVU = RVU_NFAC ;
  END ;

  * SPECIAL MEDICARE RULE FOR THERAPY CODES ;
  * THIS IS MY BEST GUESS AT THERAPY BILLED UNDER MEDICARE FEE SCHEDULE;
  IF CPT GT '97001' AND CPT LE '97456'
                        THEN    RVU = RVU_NFAC ;

  **Impute LAB RVUS using Medicare LAB fee data**;
  IF RVU = . AND RVU_LAB NE . THEN RVU = RVU_LAB ;

  * SPECIAL CHECK ON BILLS WITH MULTIPLE UNITS ;
```

```
  * if dividing by n of units gives a more reasonable answer, then do it ;
  IF PAYMENT_ORIG>0 AND RVU>0 AND UNITS>1 THEN DO;
     IF PAYMENT_ORIG/RVU > 60 AND (30 < (PAYMENT_ORIG/UNITS)/RVU < 60) THEN
                        PAYMENT_ORIG  = PAYMENT_ORIG/UNITS ;
  END ;
  * ADD MEDICARE PAYMENT STATUS CODE, SO THAT YOU LATER CAN ID THE CODES ;
  * THAT ARE MISCELLANEOUS CODES PRICED BY CARRIER REPORT ;
  * (THAT IS STATUS CODE C ON THE MEDICARE FEE SCHEDULE FILE) ;
  LENGTH MCR_STAT $ 1 ;
  MCR_STAT = PUT(CPT,$RVU&cyr.ST.) ;
  IF MCR_STAT = '' THEN MCR_STAT = PUT(CPT,$RVU&lyr.ST.) ;

RUN ;


  ******************************************************************;
  * NOW CHECK FOR MULTIPLE RECORDS FOR SAME PERSON DATE CPT          *;
  * IF FOUND, TAKE MOST EXPENSIVE ONE                                *;
  ******************************************************************;

PROC SORT DATA = ONE OUT=TWO ;
  BY PIDBDG ENDDT CPT NEWMOD DESCENDING PAYMENT_ORIG ;
RUN ;

DATA THREE (DROP=PIDBDG ENDDT) ;
  SET TWO ;
  BY PIDBDG ENDDT CPT NEWMOD DESCENDING PAYMENT_ORIG ;
  IF FIRST.NEWMOD;
RUN;

  * IMPOSE THE HCFA ACTUARIES' TRIM ;
  * ACTUARIES' SCREEN STEP GOES HERE ;
  * SCREEN OUT BILL WITH ALLOWED LESS THAN ONE THIRD OR MORE THAN THREE ;
  * TIMES THE MEAN ALLOWED CHARGE FOR THE ENTIRE DATABASE ;
  * AGAIN, FOR PRODUCTION, RESTRICT TO ONLY GOOD RECORDS ;

PROC SORT DATA = THREE ;
    BY CPT NEWMOD ;
RUN ;

PROC MEANS NOPRINT MEAN DATA = THREE ;
   BY CPT NEWMOD ;
   VAR PAYMENT_ORIG ;
   ID BETOSDET ;
   OUTPUT OUT = CPTSUM MEAN = PAYMENT_ORIG_ ;
RUN ;

* note asymmetric actuaries trim ;
DATA FOUR ;
  MERGE THREE  CPTSUM ;
  BY CPT NEWMOD ;

* NOTE THAT DUE TO DROPPER RESTRICTION, THERE MAY BE RECORDS HERE ;
* WITH MISSING MEAN ALLOWED CHARGE AND SUCH ;
* BE SURE TO TAKE THAT INTO ACCOUNT IN THE CODE ;

* SOME CODES ARE MISCELLANEOUS CATEGORIES (TYPICALLY XXX99 OR XXX49) ;
* AND YOU CANNOT PROPERLY APPLY THE ACTUARIES' TRIM TO THOSE ;
* ALSO, NO NEED TO APPLY TO RECORDS TO BE DROPPED FROM PRICE ANALYSIS ;

  IF MCR_STAT NE 'C' THEN DO ;
    IF PAYMENT_ORIG < .33*PAYMENT_ORIG_ then DELETE;            **< .33*MEAN PAYMENT_ORIG**;
    IF PAYMENT_ORIG >   6*PAYMENT_ORIG_ THEN DELETE;            **> 6*MEAN PAYMENT_ORIGED**;
    IF RVU NE . AND PAYMENT_ORIG < .25*37.8975*RVU THEN DELETE;
    IF RVU NE . AND PAYMENT_ORIG >   8*37.8975*RVU THEN DELETE;
  END ;
RUN ;

PROC SUMMARY DATA = FOUR;
WHERE RVU >0 AND PAYMENT_ORIG >0;
CLASS BETOSDET ;
```

```
VAR RVU PAYMENT_ORIG ;
OUTPUT OUT = TEMP1 SUM = RVU PAYMENT_ORIG ;
RUN ;

DATA TEMP1;
  SET TEMP1;
  CF = PAYMENT_ORIG/RVU ;   ***PMT PER RVU, BY BETOS DET***;
  INVCF = 1/CF ;       ***RVU PER PAYMENT_ORIG, BY BETOS DET***;
RUN ;

PROC PRINT DATA=TEMP1 UNIFORM ;
title "Summary stats used for imputation";
RUN ;

* FIND CODES WITHOUT RVUS ;
PROC SUMMARY MISSING NWAY DATA = FOUR;
   WHERE PAYMENT_ORIG NE . AND RVU LE 0;
   CLASS CPT NEWMOD ;
   VAR PAYMENT_ORIG UNITS;
   ID BETOSDET ;
   OUTPUT OUT = MEANPAY (DROP=_TYPE_ RENAME=(_FREQ_=SERVICES)) MEAN = MEANPAY SUM=TPAY TUNITS ;
RUN ;

PROC SORT  DATA = TEMP1 ;
  BY BETOSDET ;
RUN;

DATA TEMP2;
 SET TEMP1 ;
 IF _TYPE_ = 0 ;
 KEEP INVCF ; * AGGREGATE ;
RUN;

PROC SORT  DATA = MEANPAY ;
  BY BETOSDET ;
run;
DATA MEANPAY2 ;
  SET MEANPAY ;
  RETAIN AGGINVCF ;
  LABEL AGGINVCF = 'AGGREGATE INVERSE CONVERSION FACTOR' ;
  IF _N_ = 1 THEN SET TEMP2 (RENAME = (INVCF = AGGINVCF)) ;
RUN ;


DATA rvu.IMPRVU&cyr.;
  LENGTH F_RVOUT $1;
  MERGE MEANPAY2(IN = IN1) TEMP1 (IN = IN2 KEEP = BETOSDET INVCF) ;
  BY BETOSDET ;
  IF IN1 ;
  * USE AGGREGATE IF THERE IS NO OTHER INFORMATION ;
  IF INVCF = . THEN INVCF = AGGINVCF ;
  RVU = MEANPAY*INVCF ;

/** Decision by Curt Mueller to set any RVU>200 to missing--outlier 9/3/2004 **/

/** Also, per Curt Mueller's e-mail dated 07/23/2005, we re-set RVU to missing **/
/** for Drugs codes (Procedure codes starting with 'J' and some codes starting **/
/** with 'C', 'Q', or 'S') and DME codes                                       **/

  f_rvout = '0';
  if substr(betosdet, 1, 1) = "D" then do;
     rvu = .;
     f_rvout = '3';
  end;
  else if substr(cpt, 1, 1) = "J" or put(cpt, $d_oth.) = '1' or betosdet = "O1E" then do;
     rvu = .;
     f_rvout = '2';
  end;
  else if rvu > 200 then do;
     rvu = .;
     f_rvout = '1';
```

```
  end;
  else if services < 10 then do;
     rvu = .;
     f_rvout = '4';
  end;
  cptdesc=put(cpt,$cptdesc.);

RUN;

PROC PRINT DATA=RVU.IMPRVU&cyr. UNIFORM ;
SUM SERVICES  TUNITS;
format f_rvout $f_rvu.;
title "Imputed RVUs";
RUN ;

PROC PRINT DATA=RVU.IMPRVU&cyr. UNIFORM ;
where f_rvout='1';
SUM SERVICES TUNITS;
title "Outlier Imputed RVUs set to missing";
RUN ;


proc freq data = rvu.imprvu&cyr.;
  tables f_rvout / list missing;
  format f_rvout $f_rvu.;
  title "Frequency of RVU Flag";
run;

proc freq data = rvu.imprvu&cyr.;
  tables f_rvout / list missing;
  weight services;
  format f_rvout $f_rvu.;
  title "Frequency of RVU Flag - Weight by SERVICES";
run;

proc means data = rvu.imprvu&cyr. mean p25 p50 p75 p90 min max std;
  class f_rvout;
  var meanpay;
  weight services;
  title "Mean of RVU Flag - Weight by SERVICES";
run;


%jobend;

dm "log;    print file=I:\MCdb\DataCol\Formats\RVUs\log_out\Compute_Imputed_RVU_&currentyear..log
replace";
dm "output; print file=I:\MCdb\DataCol\Formats\RVUs\log_out\Compute_Imputed_RVU_&currentyear..lst
replace";
```

Appendix G

Practitioner Report Files Contents and SAS Code

Year 2006 Analysis File
11. Sample print of inout.&sasDTout

The CONTENTS Procedure

Data Set Name	INOUT.ANAL06_F_V2	Observations	47915217
Member Type	DATA	Variables	81
Engine	V9	Indexes	0
Created	Saturday, March 15, 2008 12:30:33 AM	Observation Length	376
Last Modified	Saturday, March 15, 2008 12:30:33 AM	Deleted Observations	0
Protection		Compressed	NO
Data Set Type		Sorted	NO
Label			
Data Representation	WINDOWS_32		
Encoding	wlatin1 Western (Windows)		

Engine/Host Dependent Information

Data Set Page Size	16384
Number of Data Set Pages	1114309
First Data Page	1
Max Obs per Page	43
Obs in First Data Page	14
Number of Data Set Repairs	0
File Name	I:\Analysis\Practrpt06\Data\anal06_f_v2.sas7bdat
Release Created	9.0101M3
Host Created	XP_PRO

Alphabetic List of Variables and Attributes

#	Variable	Type	Len	Format	Label
10	AGE1206_NR	Num	8		Non-Rounded Patient Age as of 12/31/2006
60	ALLOW	Num	8		Allowed Amount-AA ($)
7	BEGINDT	Num	8	MMDDYY8.	Service From Date
41	BETOSAGG	Char	1		BETOS Category, Aggregated
39	BETOSDET	Char	3		BETOS Category, Detailed
61	BILL	Num	8		Billed Charge-BC ($)
47	BILLTYPE	Char	1		Type of Bill
66	CDHPIND	Char	1		Consumer Directed Health Plan
6	CLAIMCN	Char	23		Claim Control Number
26	CNTY_P2	Char	3		County to Use for Pract. Report Analysis
4	COUNTY_P	Char	3		Patient County of Residence-FIPS Code
5	COVOTHR	Char	1		Patient Covered by Other Insurance
34	COVOTH_I	Char	1		Imputed/Coverted COVOTHR
62	COVTYPE	Char	1		Coverage Type
50	CPT	Char	6		CPT-4 /HCPCS Procedure Code
48	DELIVTYP	Char	1		Delivery System Type
71	DISENROLL_P	Num	8	MMDDYY8.	Disenrollment Date at PIDBDG Level, SSS created
12	DX1	Char	5		ICD-9-CM Diagnosis Code 1
13	DX2	Char	5		ICD-9-CM Diagnosis Code 2
14	DX3	Char	5		ICD-9-CM Diagnosis Code 3
15	DX4	Char	5		ICD-9-CM Diagnosis Code 4
16	DX5	Char	5		ICD-9-CM Diagnosis Code 5
17	DX6	Char	5		ICD-9-CM Diagnosis Code 6
18	DX7	Char	5		ICD-9-CM Diagnosis Code 7
19	DX8	Char	5		ICD-9-CM Diagnosis Code 8
20	DX9	Char	5		ICD-9-CM Diagnosis Code 9
21	DX10	Char	5		ICD-9-CM Diagnosis Code 10
2	ENCNT_ID	Char	8		Medical Service ID (Payer Specific)
8	ENDDT	Num	8	MMDDYY8.	Service Thru Date
70	ENROLL_P	Num	8	MMDDYY8.	Enrollment Date at PIDBDG Level, SSS created
59	FACILITY	Char	1		Service Location is a Facility (1/0)
23	FEDTAXID	Char	9		Practitioner Federal Tax ID
69	F_ANESTH	Char	1		Flag, Anesthesia Service
72	F_CAP	Char	1		Capitated Service (1/0)
44	F_CAP_A	Char	1		Capit. Service used in Analysis (1/0)
38	F_CPT	Char	1		Procedure Code Category
25	F_DEMOG	Char	1		Demographic Donor for Analysis File?
32	F_IMPRVU	Char	1		RVU Imputation Flag (1/0,2)
55	F_MODF	Char	1		1=There is TC/SG in MODF1/MODF2

23:42 Friday, March 14, 2008
I:\Analysis\Practrpt06\Programs\Analysis_Detect_Outliers.sas

Year 2006 Analysis File
11. Sample print of inout.&sasDTout

The CONTENTS Procedure

Alphabetic List of Variables and Attributes

#	Variable	Type	Len	Format	Label
43	F_NCAP_A	Char	1		Non-Capit Service used in Analysis (1/0)
75	F_OANL	Char	1		Flag for records to be used for outlier identification
45	F_ODDSVC	Char	2		Odd Service Flag
54	F_UNCV	Char	1		Uncovered services(1/0)
9	F_YR07	Char	1		Service provided in 2007+
81	F_odd11	Char	1		Odd service: Payment>3 times mean payment
78	GPNum	Char	4		Group Payer ID
51	MOD1	Char	2		AMA Modified I
67	MOD2	Char	2		AMA Modified II
42	NEWBETAG	Char	1		Aggregated BETOS, Reclassified
40	NEWBETDT	Char	3		Detailed BETOS, Reclassified
22	NEWMOD	Char	2		Imputed Modifier/Last Imputation
35	NEWSPEC	Char	3		Revised Speciality Code for Complex Algorithm
76	Nunits	Num	8		Redefined unit
68	PARTPROV	Char	1		Participating Provider Flag
63	PATLIAB	Num	8		Patient Liability-PL ($)
37	PAYERTYP	Char	1		Payer Type(1=PRVHMO 2=PRVNHMO 3=MCRHMO 9=CAP)
64	PAYMENT	Num	8		Payment=Sum(RA,PL)($)
27	PAY_CALC	Char	1		Method for Calculating Medicare Payment
33	PAY_MCR	Num	8		Prelim Calc MCR Rate-DO NOT USE AT THIS POINT
36	PLAN	Char	6		Plan Type(HMOFFS/NONHMO/HMOCAP)
1	PNUM	Char	4		Payer Unique ID
56	POS	Char	10		Simplified Place of Service Code
77	PPU	Num	8		Payment per Unit
24	PROVID	Char	13		Servicing Practitioner Identifier
57	REGION_P	Char	1		Patient Region of Residence
65	REIMB	Num	8		Reimbursement Amount-RA ($)
58	RVU_2006	Num	8		2006 RVU applied to 2005 file
31	RVU_I	Num	8		Assigned/Imputed RVU
29	RVU_MP	Num	4		Medicare phys fee schedule RVU, malpractice
30	RVU_PE	Num	8		Medicare Phys Fee Schedule RVU, Pract Expense
28	RVU_W	Num	4		Medicare Phys Fee Schedule RVU, Work component
3	SEX	Char	1		Patient Sex
11	SVCPLACE	Char	2		Place of Service
46	UNITIND	Char	1		Service Unit Indicator
49	UNITS	Num	8		Number of Units for a Service
53	ZIP_P	Char	5		Patient Residence ZIP Code
52	ZIP_S	Char	5		Service Location ZIP Code
73	id_change	Char	1		PIDBDG changed using data from legacy file
79	mean_ppu	Num	8		
80	oddsvc	Char	2		
74	pidbdg	Char	25		

```
proc datasets kill;
run;
quit;


%jobstart;

**=========================================================================**
** Program:     makeset_analysis_2006_v1.sas
** Project:     Maryland Practitioner Report
** Purpose:     Create 2004 analysis data set including all Diagnosis Codes
** Input:       prv04s_1.sas7bdat
** Output:      anal04_f.sas7bdat
**              anal05_jan_apr.sas7bdat
**
** Programmer:  Po-Lun Chou/Sanee Maphungphong
** Date:        October 18, 2005
**
** Modified by: Po-Lun Chou
** Date:        December 23, 2005
** Input:       prv04s_2.sas7bdat
** Output:      anal04_f.sas7bdat
**              anal05_jan_apr.sas7bdat
**
** Modified by: Sireesha Thati
** Date:        November 16, 2005
** Input:       Prv05_V3.sas7bdat
** Output:      anal05_f.sas7bdat
**              anal06_jan_apr.sas7bdat
** Note:        For this Year Analysis file is created from FULL MCDB
**              dataset and not from the 10% sample.
**              Added variables f_insamp, disenroll_p, enroll_p, age1205_nr
**
** Modified by: Kasey Chen / Po-Lun Chou
** Date:        March 4, 2008
** Input:       prv06_v2.sas7bdat
** Output:      anal06_f.sas7bdat
**              anal07_jan_apr.sas7bdat
** Note:        1. Changed service year to 2006,
**              2. Changed input data file from PRV05_V3 to PRV06_V1,
**              3. Do not keep following varibales in 2006 analysis data
**                 file:
**                 (1) AGE1206 --  MHCC prefer to use not-rounded variable
**                                 AGE1206_NR instead of this one,
**                 (2) F_INSAMP -- We don't generate the 10% sample data
**                                 file anymore, so we don't have to keep
**                                 this variable.
**=========================================================================**;
options obs = max center errors = 2 linesize = 123 pagesize = 90;

libname in06  "I:\MCdb\MCDB_FN\data";
libname out06 "I:\Analysis\Practrpt06\Data";

%let inset    = /* prv06_v2 */  prv06_v1;
%let outset1  = anal06_f;
%let outset2  = anal07_jan_apr;
%let progname = makeset_analysis_2006_v1;
%let hmolst   = 'P030','P130','P131','P160','P480','P520','P620','P660','P680','P870';
%let mhmolst  = 'P480';

title1 "Year 2006 Analysis Set - for Year 2006 Practitioner Report";
title2 "with Diagnosis Codes DX1 - DX10";

footnote1;
footnote2;
footnote3 "Program: I:\Analysis\Practrpt06\Programs\&progname..sas";
footnote4 "%sysfunc(putn(%sysfunc(time()), hhmm.)) %sysfunc(putn(%sysfunc(date()), weekdate.))";

options nosource2;
%include "I:\mcdb\datacol\formats\posfmt.sas";
%include "I:\mcdb\datacol\formats\mdregfmt.sas";
options source2;


data out06.&outset1 (compress = binary
                     drop = d_dnt_f d_inp_f d_i7k_f d_zip_f d_dat_f)
     out06.&outset2 (compress = binary
                     drop = d_dnt_f d_inp_f d_i7k_f d_zip_f d_dat_f);
     set in06.&inset (keep = age1206_nr
                             allow
                             begindt
                             betosagg
                             betosdet
                             bill
                             billtype
                             cdhpind
                             claimcn
                             cnty_p2
                             county_p
                             covothr
                             covoth_i
                             covtype
                             cpt
                             delivtyp
                             disenroll_p
                             dx1-dx10
                             encnt_id
```

```
                            enddt
                            enroll_p
                            facility
                            fedtaxid
                            f_anesth
                         /* f_cap   */
                            f_cap_a
                            f_cpt
                            f_demog
                            f_imprvu
                            f_modf
                            f_ncap_a
                            f_oddsvc
                            f_uncv
                            f_yr07
                            mod1

                            mod2
                            newbetag
                            newbetdt
                            newmod
                            newspec
                            partprov
                            patliab
                            payertyp
                            payment
                            pay_calc
                            pay_mcr
                            pidbdg
                            plan
                            pnum
                            pos
                            provid
                            region_p
                            reimb
                            rvu_2006
                            rvu_i
                            rvu_mp
                            rvu_pe
                            rvu_w
                            sex
                            svcplace
                            unitind
                            units
                            zip_p
                            zip_s

                            d_dnt_f
                            d_inp_f
                            d_i7k_f
                            d_zip_f
                            d_dat_f);

*==============================================*;
* Quick check and modify --                    *;
* Re-set deletion flag D_DAT_F due to error    *;
* in PXXX_Enc1.sas.                            *;
*==============================================*;

     if begindt = . and
        enddt   = . then d_dat_f = '1';
     else if begindt ne . then do;
        if year(begindt) < 2006 then d_dat_f = '1';
     end;
     else if year(enddt) < 2006 then d_dat_f = '1';

*==============================================*;
* Apply the global screens (D_XXX_F variables)  *;
*==============================================*;

     if (d_dnt_f = "1") or (d_inp_f = "1") or (d_i7k_f = "1") or
        (d_zip_f = "1") or (d_dat_f = "1") then delete;

*================================*;
* modify Anesthesia records      *;
*================================*;

     if newspec in ("004", "049") then do;
        if f_oddsvc = " " then f_oddsvc = "04";
     end;

*==================================================*;
* combine missing/misc aggregate betos categories  *;
*==================================================*;

     if newbetag in (' ','6','7','N') then newbetag = 'X';
     if newbetdt in (' ','N/A')       then newbetdt = 'XXX';

*==================================================*;
* financials for HMOs                              *;
* If payer is on list of HMO payers,               *;
* and BC, AA and RA = -999,                        *;
* and delivery system = 1 or 5 then                *;
* set financial variables = .C and set             *;
* capitation flag to 1                             *;
*==================================================*;
```

```
     length F_CAP $1;
     label f_cap = "Capitated Service (1/0)";
     f_cap = '0';
     if pnum in (&HMOlst.)    and
        delivtyp in ('1','5') and
        bill  = -999          and
        allow = -999          and
        reimb = -999          then do;
          f_cap = '1';
          bill  = .;
          allow = .;
          reimb = .;
          if patliab = -999 then patliab = .;
     end;

*==================================================*;
* set all payment data on capitated records to     *;
* missing                                          *;
*==================================================*;

     if f_cap_a = '1' then do ;
        payment = . ;
        bill    = . ;
        allow   = . ;
        patliab = . ;
        reimb   = . ;
     end ;

*================================*;
* create final data sets         *;
*================================*;

     if (f_yr07 = "0") and
        (f_cap_a = "1" or f_ncap_a = "1") and
        (0 le age1206_nr le 64) then output out06.&outset1;
     else if (f_yr07 = "1") and
             (0 le age1206_nr le 64) then output out06.&outset2;

run;


*** analysis data file anal06_f ***;

proc contents data = out06.&outset1;
     title3 "Contents of %upcase(&outset1)";
run;

proc freq data = out06.&outset1;
     tables f_yr07 f_oddsvc billtype region_p / list missing;
     tables f_cap_a * f_ncap_a / list missing;
     tables (f_anesth f_oddsvc) * unitind / list missing;
     tables pnum * (plan billtype) / list missing;
     title3 "Diagnostics on Variables from %upcase(&outset1)";
run;

proc means data = out06.&outset1 mean min max;
     var bill allow reimb patliab payment rvu_i rvu_2006;
     where billtype = "1";
     title3 "Means on Financial Variables with BILLTYPE = 1 from %upcase(&outset1)";
run;

proc means data = out06.&outset1 mean min max;
     var bill allow reimb patliab payment rvu_i rvu_2006;
     where billtype = "8";
     title3 "Means on Financial Variables with BILLTYPE = 8 from %upcase(&outset1)";
run;

proc means data = out06.&outset1 mean min max;
     var bill allow reimb patliab payment rvu_i rvu_2006;
     where billtype = "2";
     title3 "Means on Financial Variables with BILLTYPE = 2 from %upcase(&outset1)";
run;

proc print data = out06.&outset1 (obs = 10);
     title3 "10 Observations from %upcase(&outset1)";
run;


*** further check on ANAL06_F ***;

data anal06;
  set out06.anal06_f;
  length denry denrm denrd 3;

/* applying the global screens */
  if (f_oddsvc ne " ") or (region_p = " ") then delete;

  denry = year(disenroll_p);
  denrm = month(disenroll_p);
  denrd = day(disenroll_p);
run;

proc freq data = anal06;
  tables denry denrm denrd / list missing;
  tables denry * denrm / list missing;
  title3 "Information of DISENROLL_P in %upcase(&outset1)";
run;
```

```
proc freq data = anal06;
  tables denrd / list missing;
  where denry = 2006 and denrm = 12;
  title3 "Date Information while Year = 2006 and Month = 12 in DISENROLL_P in %upcase(&outset1)";
run;


*** analysis data file anal07_jan_apr ***;

proc contents data = out06.&outset2;
     title3 "Contents of %upcase(&outset2)";
run;

proc freq data = out06.&outset2;
     tables f_yr07 / list missing;
     title3 "Diagnostics on Variables from %upcase(&outset2)";
run;

proc means data = out06.&outset2 mean min max;
     var bill allow reimb patliab payment pay_mcr rvu_i rvu_2006;
     where billtype = "1";
     title3 "Means on Financial Variables with BILLTYPE = 1 from %upcase(&outset2)";
run;

proc means data = out06.&outset2 mean min max;
     var bill allow reimb patliab payment pay_mcr rvu_i rvu_2006;
     where billtype = "8";
     title3 "Means on Financial Variables with BILLTYPE = 8 from %upcase(&outset2)";
run;

proc means data = out06.&outset2 mean min max;
     var bill allow reimb patliab payment pay_mcr rvu_i rvu_2006;
     where billtype = "2";
     title3 "Means on Financial Variables with BILLTYPE = 2 from %upcase(&outset2)";
run;

proc print data=out06.&outset2 (obs = 10);
     title3 "10 Observations from %upcase(&outset2)";
run;


%jobend;

dm "log;    print file = I:\Analysis\Practrpt06\Programs\makeset_analysis_2006_v1.log replace";
dm "log;    print file = I:\Analysis\Practrpt06\Programs\makeset_analysis_2006_v1.log replace";
dm "output; print file = I:\Analysis\Practrpt06\Programs\makeset_analysis_2006_v1.lst replace";
dm "output; print file = I:\Analysis\Practrpt06\Programs\makeset_analysis_2006_v1.lst replace";
```

```
proc datasets kill;
run;
quit;


%jobstart;

**=============================================================================================**
** Program:     makeset_analysis_2006_v2.sas
** Project:     Maryland Practitioner Report
** Purpose:     Create 2004 analysis data set including all Diagnosis Codes
** Input:       prv04s_1.sas7bdat
** Output:      anal04_f.sas7bdat
**              anal05_jan_apr.sas7bdat
**
** Programmer:  Po-Lun Chou/Sanee Maphungphong
** Date:        October 18, 2005
**
** Modified by: Po-Lun Chou
** Date:        December 23, 2005
** Input:       prv04s_2.sas7bdat
** Output:      anal04_f.sas7bdat
**              anal05_jan_apr.sas7bdat
**
** Modified by: Sireesha Thati
** Date:        November 16, 2005
** Input:       Prv05_V3.sas7bdat
** Output:      anal05_f.sas7bdat
**              anal06_jan_apr.sas7bdat
** Note:        For this Year Analysis file is created from FULL MCDB
**              dataset and not from the 10% sample.
**              Added variables f_insamp, disenroll_p, enroll_p, age1205_nr
**
** Modified by: Kasey Chen / Po-Lun Chou
** Date:        March 4, 2008/run date: 03-06-2008
** Input:       prv06_v2.sas7bdat
** Output:      anal06_f.sas7bdat
**              anal07_jan_apr.sas7bdat
** Note:        1. Changed service year to 2006,
**              2. Changed input data file from PRV05_V3 to PRV06_V1,
**              3. Do not keep following varibales in 2006 analysis data
**                 file:
**                 (1) AGE1206 --  MHCC prefer to use not-rounded variable
**                                 AGE1206_NR instead of this one,
**                 (2) F_INSAMP -- We don't generate the 10% sample data
**                                 file anymore, so we don't have to keep
**                                 this variable.
**
** Modified by: Sanee Maphungphong
** Date:        March 13, 2008
**
** Input:       anal06_f_03062008.sas7bdat (rename file "anal06_f" from 03-06-2008 run)
**              anal07_jan_apr_03062008.sas7bdat (rename file "anal0anal07_jan_apr" from 03-06-2008 run)
**
** Output:      anal06_f.sas7bdat
**              anal07_jan_apr.sas7bdat
**
** Purpose:     To recreate ANAL06_F.
**              Apply format created based on Legacy ID Xwalk to change PIDBDG of P130 data.
**              (The same changes will be appplied to PRV06_V1).
**=============================================================================================**;
options obs = max center errors = 2 linesize = 123 pagesize = 90;

libname inds  "I:\MCdb\datacol\P130\sasdata";
libname in06  "I:\MCdb\MCDB_FN\data";
libname out06 "I:\Analysis\Practrpt06\Data";

*-------------------------------------------------------*
* Assign Data set name here. Dataset name must be reviewed *
* and changed accordingly                               *
*-------------------------------------------------------*;

%let inset1    = anal06_f_03062008;
%let inset2    = anal07_jan_apr_03062008;
%let inset3    = P130_Enc1 /* prv06_v1 */;
%let outset1   = anal06_f;
%let outset2   = anal07_jan_apr;

%let progname  = makeset_analysis_2006_v2;
%let hmolst    = 'P030','P130','P131','P160','P480','P520','P620','P660','P680','P870';
%let mhmolst   = 'P480';

title1 "Maryland Medical Care Commission";
title2 "2006 Practitioner Report Analysis Dataset";

footnote1;
footnote2;
footnote3 "Program: I:\Analysis\Practrpt06\Programs\&progname..sas";
footnote4 "%sysfunc(putn(%sysfunc(time()), hhmm.)) %sysfunc(putn(%sysfunc(date()), weekdate.))";

options nosource2;
%include "I:\mcdb\datacol\formats\posfmt.sas";
%include "I:\mcdb\datacol\formats\mdregfmt.sas";
options source2;

proc format;
  value $delvsys
```

```
                      '1' = '1: HMO (Non-Medicaid)'
                      '2' = '2: PPO-POS'
                      '3' = '3: PPO/Other Managed Care'
                      '4' = '4: Indemnity Care'
                      '5' = '5: HMO-POS Rider'
                      '9' = '9: Not Coded'
                      ' ' = 'Blank';
  value $covtyp
                      '1' = '1: Medicare Supp. (Indvl,Grp,WRAP)'
                      '2' = '2: Individual Plan'
                      '3' = '3: Private Employer, Fully Self-Ins.'
                      '4' = '4: Private Employer'
                      '5' = '5: Public Employee'
                      '6' = '6: Comp Std Health Benefit Plan'
                      '7' = '7: Med. Serv provided by Med. HMO/HCFA'
                      '8' = '8: Taft Hartley Trust'
                      '9' = '9: Not Coded'
                      'Y' = 'Y: Blank/Missing'
                    other = 'Code Invalid';
run;


*-------------------------------------------------------*
* Make changes to PIDBDG in P130 (Carefirst BlueChoice) *
* records based on the "Legacy Id" Xwalk data from the  *
* payer.                                                *
*-------------------------------------------------------*;

*==> Create the format for "old" and "new" PIDBDG;
data xx (Keep=start label fmtname type);
  set inds.p130_legacy_id end=eof;
  length start $25;
  start=trim(pidbdg_bt8);
  label=put(pidbdg_new,$25.);
  fmtname = '$PIDBDG';
  type='c';
  output;
  if eof then do;
     fmtname = '$PIDBDG';
     start   = 'other';
     label   = " ";
     type    = 'c';
     output;
  end;
run;

proc format cntlin=xx /*fmtlib */;
run;

*==> Apply format to ANAL06_F (renamed to ANAL06_F_03062008);
data year06 (keep=pidbdg_new pidbdg_orig id_change)
     out06.&outset1 (compress = binary
                       rename = (pidbdg_new=pidbdg)
                       drop   = pidbdg_orig);
  set out06.&inset1;

id_change='0';
pidbdg_new=pidbdg;

if pnum='P130' then do;
   if put(pidbdg, $pidbdg.) ne " " then do;
      pidbdg_new=put(pidbdg, $pidbdg.);
      id_change='1';
   end;
end;
if id_change='1' then output year06;
output out06.&outset1;

rename pidbdg=pidbdg_orig;
label id_change="PIDBDG changed using data from legacy file";
run;

proc print data=year06 (obs =50);
 where id_change='1';
 var pidbdg_new pidbdg_orig;
title3 "&outset1 -- Sample of original and new PIDBDGs";
run;

proc freq data=out06.&outset1;
  table billtype*id_change / list missing;
  tables delivtyp covtype partprov billtype plan region_p cdhpind /list missing;
  format delivtyp $delvsys. covtype $covtyp.;
title3 "&outset1 -- Variables After Replace PIDBDG";
run;

*==> Apply format to ANAL07_JAN_APR (renamed to ANAL07_JAN_APR_03062008);
data year07 (keep=pidbdg_new pidbdg_orig id_change)
     out06.&outset2 (compress = binary
                       rename = (pidbdg_new=pidbdg)
                       drop   = pidbdg_orig);
  set out06.&inset2;

id_change='0';
pidbdg_new=pidbdg;


if pnum='P130' then do;
```

```
   if put(pidbdg, $pidbdg.) ne " " then do;
      pidbdg_new=put(pidbdg, $pidbdg.);
      id_change='1';
   end;
end;
if id_change='1' then output year07;
output out06.&outset2;

rename pidbdg=pidbdg_orig;
label id_change="PIDBDG changed using data from legacy file";
run;

proc print data=year07 (obs =50);
 where id_change='1';
 var pidbdg_new pidbdg_orig;
title3 "&outset2 -- Sample of original and new PIDBDGs";
run;

proc freq data=out06.&outset2;
  table billtype*id_change / list missing;
  tables delivtyp covtype partprov billtype plan region_p cdhpind /list missing;
  format delivtyp $delvsys. covtype $covtyp.;
title3 "&outset2 -- Variables After Replace PIDBDG";
run;

*----------------------------------*
* Contents and diagnostics on the  *
* analysis data file: ANAL06_F     *
*----------------------------------*;

title3 "Contents of %upcase(&outset1)";
proc contents data = out06.&outset1;
run;

title3 "Diagnostics on Variables from %upcase(&outset1)";
proc freq data = out06.&outset1;
  tables f_yr07 f_oddsvc billtype region_p / list missing;
  tables f_cap_a * f_ncap_a / list missing;
  tables (f_anesth f_oddsvc) * unitind / list missing;
  tables pnum * (plan billtype) / list missing;

run;

title3 "Means on Financial Variables with BILLTYPE = 1 from %upcase(&outset1)";
proc means data = out06.&outset1 mean min max;
  var bill allow reimb patliab payment rvu_i rvu_2006;
  where billtype = "1";

run;

title3 "Means on Financial Variables with BILLTYPE = 8 from %upcase(&outset1)";
proc means data = out06.&outset1 mean min max;
  var bill allow reimb patliab payment rvu_i rvu_2006;
  where billtype = "8";
run;

title3 "Means on Financial Variables with BILLTYPE = 2 from %upcase(&outset1)";
proc means data = out06.&outset1 mean min max;
   var bill allow reimb patliab payment rvu_i rvu_2006;
   where billtype = "2";
run;

title3 "10 Observations from %upcase(&outset1)";
proc print data = out06.&outset1 (obs = 10);
run;


*==> Additional check on ANAL06_F;

data anal06;
  set out06.anal06_f;
  length denry denrm denrd 3;

/* applying the analysis screens */
  if (f_oddsvc ne " ") or (region_p = " ") then delete;
  denry = year(disenroll_p);
  denrm = month(disenroll_p);
  denrd = day(disenroll_p);
run;

title3 "Frequency of Year, Month, and Day of DISENROLL_P in %upcase(&outset1)";
title4 "Data from Analysis file with Odd Service Screen applied";
proc freq data = anal06;
  tables denry denrm denrd / list missing;
  tables denry * denrm / list missing;
run;

title3 "Day of DISENROLL_P when Month=December Year=2006 in %upcase(&outset1)";
proc freq data = anal06;
  tables denrd / list missing;
  where denry = 2006 and denrm = 12;
run;


*-------------------------------------*
* Contents and diagnostics on the    *
* analysis data file: ANAL07_Jan_Apr *
```

```
*-----------------------------------*;

title3 "Contents of %upcase(&outset2)";
proc contents data = out06.&outset2;
run;

title3 "Diagnostics on Variables from %upcase(&outset2)";
proc freq data = out06.&outset2;
  tables f_yr07 / list missing;
run;

title3 "Means on Financial Variables with BILLTYPE = 1 from %upcase(&outset2)";
proc means data = out06.&outset2 mean min max;
  var bill allow reimb patliab payment pay_mcr rvu_i rvu_2006;
  where billtype = "1";
run;

title3 "Means on Financial Variables with BILLTYPE = 8 from %upcase(&outset2)";
proc means data = out06.&outset2 mean min max;
  var bill allow reimb patliab payment pay_mcr rvu_i rvu_2006;
  where billtype = "8";
run;

title3 "Means on Financial Variables with BILLTYPE = 2 from %upcase(&outset2)";
proc means data = out06.&outset2 mean min max;
  var bill allow reimb patliab payment pay_mcr rvu_i rvu_2006;
  where billtype = "2";
run;

title3 "10 Observations from %upcase(&outset2)";
proc print data=out06.&outset2 (obs = 10);
run;


%jobend;

dm "log;    print file = I:\Analysis\Practrpt06\Programs\makeset_analysis_2006_v2.log replace";
dm "log;    print file = I:\Analysis\Practrpt06\Programs\makeset_analysis_2006_v2.log replace";
dm "output; print file = I:\Analysis\Practrpt06\Programs\makeset_analysis_2006_v2.lst replace";
dm "output; print file = I:\Analysis\Practrpt06\Programs\makeset_analysis_2006_v2.lst replace";
```

```
proc datasets kill;
run;
quit;


%jobstart;


**=========================================================================**;
**                                                                         **;
** Program:  Analysis_Detect_Outliers.sas                                  **;
** Programmer: Adrien Ndikumwami                                           **;
** Date      : 2/22/2007                                                   **;
**                                                                         **;
** Input:     I:\Analysis\Practrpt06\Data\anal06_f.sas7bdat               **;
** Output:    I:\Analysis\Practrpt06\Data\anal06_f_v2.sas7bdat             **;
**                                                                         **;
** Notes:                                                                  **;
**                                                                         **;
**     This program reads the analysis file (with odd services still       **;
**     included) and re-assign odd service=11  by reducing the upper       **;
**    threshold to payment+3* Mean.                                        **;
**     The output has the same number of observation as the input file.    **;
**  Steps:                                                                 **;
**                                                                         **;
**  A) Compute Mean and Standard Deviation for each PNUM-PLAN-CPT-MOD      **;
**     combination.                                                        **;
**   Screens:                                                              **;
**    1. Exclude anesthesia services (CPT=10000x)                          **;
**    2. Exclude non-MD residents-already excluded in anal06_f             **;
**    3. Exclude dental services- already excluded in anal06_f             **;
**    4. Exclude inpatient services -already excluded in anal06_f          **;
**    5. Exclude out-of-range services-already excluded in anal06_f        **;
**    6. Exclude HMO capitated and unknown plan                            **;
**    7. Exclude known odd services                                        **;
**                                                                         **;
**  B. Create a dynamic format: $pamean for the mean.                      **;
**          standard deviation.                                                **;
**                                                                         **;
**  C. Use the two format to assign a mean  for each                       **;
**     occurence of PNUM-PLAN-CPT-MOD.  The screened variables are not     **;
**     assigned an outlier.                                                **;
**    Outlier: Any payment beyond the 3 times the Mean is flagged oddsvc=11  **;
**=========================================================================**;
options obs = max errors = 2 linesize = 120 pagesize = 86;

%let sasDTin  = anal06_f;
%let sasDTout = anal06_f_V2;

libname inout "I:\Analysis\Practrpt06\Data";

title1 "MARYLAND HEALTH CARE COMMISSION";
title2 "Year 2006 Analysis File";

footnote1 "%sysfunc(putn(%sysfunc(time()), hhmm.)) %sysfunc(putn(%sysfunc(date()), weekdate.))";
footnote2 "I:\Analysis\Practrpt06\Programs\Analysis_Detect_Outliers.sas";

options nosource2;
%include "I:\MCdb\datacol\Formats\Format_CPT_description.sas";
%include "I:\Analysis\Practrpt06\Programs\Template_ODS_Outliers.sas";
options source2;

proc format;
value $betosa
           '1'   = '1-Evaluation and Management'
           '2'   = '2-Procedures'
           '3'   = '3-Imaging'
           '4'   = '4-Tests'
           '5'   = '5-Durable Medical Equipment'
           '6'   = '6-Other'
           '7'   = '7-Exceptions/Unclassified'
           other = 'Unknown';
run;


*---------------------------- --------------------------*
* Use all screens used for analysis data to compute       *
* payment per unit. Set all nunits <=0 to 1.               *
*----------------------------------------------------------*;


Data one;
  set inout.&sasDTin;

  if    F_Anesth="0"  and
        plan not in ("HMOCAP","") and
        F_Oddsvc= " "
        then F_OANL='1' ;
                  Nunits=Units;

if F_OANL='1' then do;
     if Nunits <=0 then NUnits=1;
     PPU=payment/Nunits;
end;
GPNum=Pnum;
if Pnum in ("P130","P131") then GPnum="P13C";
if Pnum in ("P500","P520","P620") then GPnum="P50C";
```

```
label   GPnum='Group Payer ID'
        PPU='Payment per Unit'
        F_OANL='Flag for records to be used for outlier identification'
        Nunits='Redefined unit';
run;

proc print uniform noobs data=one (obs=25);
where F_OANL='1';
  var GPnum plan cpt newmod payment nunits PPU F_OANL;
title3 '1. Sample print After PPU calculation';
run;

proc freq data=one;
Tables plan * F_OANL/missing list;
title3 '2. Freq of PLAN by F_OANL';
run;

proc freq data=one;
where payment<=0 and F_OANL='1';
Tables F_OANL/missing list;
title3 '3. Records in Analysis file where payment<=0 - check if there are records';
run;
*----------------------------------------------------*
* compute mean and standard deviation of payment    *
* by GPnum plan cpt and newmod.                      *
*----------------------------------------------------*;
 proc means data=one missing nway noprint;
 where F_OANL='1';
  class GPnum Plan cpt newmod;
  var PPU;
  output out=PPUSTAT (rename=(_freq_=services)) mean=meanpa  std=stdpa;
 run;

proc print uniform noobs data=PPUSTAT (obs=25);
title3 '4. Sample print of output of mean and STD calculation';
run;

*-------------------------------*
* Create a format for the mean. *
*-------------------------------*;

data pamean(keep=start label fmtname);
 length label $12 start $19;
 set ppustat end=eof;
 where services>30;
 start=Gpnum||Plan||cpt||newmod;
 label=put(meanpa,12.5);
 fmtname='$pamean';
 output;
 if eof then do;
  fmtname='$pamean';
  start='other';
  label=' ';
  output;
 end;

proc sort data=pamean nodupkey;
by start; run;

proc format cntlin=pamean;
run;

proc print uniform noobs data=pamean(obs=10);
title3 '4. sample print of format PAMEAN';
run;

*----------------------------------------------------*
* Flag records with outlier payment and produce    *
* statistics.                                       *
*----------------------------------------------------*;

data inout.&sasDTout (drop=Key );
 set one ;
 if F_OANL='1'  then do;
    Key=Gpnum||plan||cpt||newmod;
    mean_ppu=input(put(Key,$pamean.),12.5);
    if not (ppu = . or mean_ppu =.) and ppu > (3 * mean_ppu) then do;
             oddsvc='11';
                  F_odd11='1';
         end;
end;
label    F_odd11="Odd service: Payment>3 times mean payment";
run;

proc freq data=inout.&sasDTout;
  table GPnum*f_Oanl*Plan*f_odd11 Pnum*F_odd11  / list missing;
  format newbetag $betosa.;
title3 "5. Key statistics after applying outliers to the entire data set";
run;


options orientation=landscape ;
ODS tagsets.MSOffice2K  style=CustOut BODY="I:\Analysis\Practrpt06\Programs\2006analysis_oddservices.xls";
ODS ESCAPECHAR='^';

proc univariate data=inout.&sasDTout noprint;
output out=StatBO_HMO  KURTOSIS= Kurt_Pay Kurt_PPU
```

```
                  MAX     = Max_Pay Max_PPU
                  MEAN    = Mean_Pay Mean_PPU
                                  MEDIAN  = Med_Pay Med_PPU
                  MIN     = Min_Pay Min_PPU
                  N       = N_Pay N_PPU
                  SKEWNESS= SKEW_Pay SKEW_PPU
                  STD     = STD_Pay STD_PPU payment ppu;

where  F_OANL='1'   and Plan="HMOFFS" and
((cpt ="99213" and newbetag='1') or
(cpt ="97110" and newbetag='2') or
(cpt="78465" and newbetag='3') or
(cpt="88305" and newbetag='4') or
(cpt="E0601" and newbetag='5') or
(cpt="98941" and newbetag='6') or
(cpt="S9083" and newbetag='7'));
class GPnum cpt;
var payment ppu;
format cpt $cptdesc. newbetag $betosa.;
run;

Proc print data=StatBO_HMO noobs;
var GPnum cpt N_PPU  Mean_PPU  Med_PPU  Min_PPU  Max_PPU  Kurt_PPU SKEW_PPU STD_PPU ;
format  N_PPU comma10. Mean_PPU  MED_PPU Min_PPU Max_PPU  comma7.2 Kurt_PPU SKEW_PPU STD_PPU comma7.4;
title3 "6. Univariate stats BEFORE removing outliers-HMO";
run;

proc univariate data=inout.&sasDTout noprint;
output out=StatBO_NHMO  KURTOSIS= Kurt_Pay Kurt_PPU
                  MAX     = Max_Pay Max_PPU
                  MEAN    = Mean_Pay Mean_PPU
                                  MEDIAN  = Med_Pay Med_PPU
                  MIN     = Min_Pay Min_PPU
                  N       = N_Pay N_PPU
                  SKEWNESS= SKEW_Pay SKEW_PPU
                  STD     = STD_Pay STD_PPU payment ppu;

where  F_OANL='1'   and Plan="NONHMO" and
((cpt ="99213" and newbetag='1') or
(cpt ="97110" and newbetag='2') or
(cpt="78465" and newbetag='3') or
(cpt="88305" and newbetag='4') or
(cpt="E0601" and newbetag='5') or
(cpt="98941" and newbetag='6') or
(cpt="S9083" and newbetag='7'));
class GPnum cpt;
var payment ppu;
format cpt $cptdesc. newbetag $betosa.;
run;

Proc print data=StatBO_NHMO noobs;
var GPnum cpt N_PPU  Mean_PPU  Med_PPU  Min_PPU  Max_PPU  Kurt_PPU SKEW_PPU STD_PPU ;
format  N_PPU comma10. Mean_PPU  MED_PPU Min_PPU Max_PPU  comma7.2 Kurt_PPU SKEW_PPU STD_PPU comma7.4;
title3 "7. Univariate stats BEFORE removing outliers-NON-HMO";
run;



proc univariate data=inout.&sasDTout noprint;
output out=StatAO_HMO  KURTOSIS= Kurt_Pay Kurt_PPU
                  MAX     = Max_Pay Max_PPU
                  MEAN    = Mean_Pay Mean_PPU
                                  MEDIAN  = Med_Pay Med_PPU
                  MIN     = Min_Pay Min_PPU
                  N       = N_Pay N_PPU
                  SKEWNESS= SKEW_Pay SKEW_PPU
                  STD     = STD_Pay STD_PPU payment ppu;

where  F_OANL='1'   and Plan="HMOFFS" and (F_odd11 ne '1') and
((cpt ="99213" and newbetag='1') or
(cpt ="97110" and newbetag='2') or
(cpt="78465" and newbetag='3') or
(cpt="88305" and newbetag='4') or
(cpt="E0601" and newbetag='5') or
(cpt="98941" and newbetag='6') or
(cpt="S9083" and newbetag='7'));
class GPnum cpt;
var payment ppu;
format cpt $cptdesc. newbetag $betosa.;
run;

Proc print data=StatAO_HMO noobs;
var GPnum cpt N_PPU  Mean_PPU  Med_PPU  Min_PPU  Max_PPU  Kurt_PPU SKEW_PPU STD_PPU ;
format  N_PPU comma10. Mean_PPU  MED_PPU Min_PPU Max_PPU  comma7.2 Kurt_PPU SKEW_PPU STD_PPU comma7.4;
title3 "8. Univariate stats AFTER removing outliers-HMO";
run;

proc univariate data=inout.&sasDTout noprint;
output out=StatAO_NHMO  KURTOSIS= Kurt_Pay Kurt_PPU
                  MAX     = Max_Pay Max_PPU
                  MEAN    = Mean_Pay Mean_PPU
                                  MEDIAN  = Med_Pay Med_PPU
                  MIN     = Min_Pay Min_PPU
                  N       = N_Pay N_PPU
                  SKEWNESS= SKEW_Pay SKEW_PPU
                  STD     = STD_Pay STD_PPU payment ppu;
```

```
where  F_OANL='1'   and Plan="NONHMO" and (F_odd11 ne '1') and
((cpt ="99213" and newbetag='1') or
(cpt ="97110" and newbetag='2') or
(cpt="78465" and newbetag='3') or
(cpt="88305" and newbetag='4') or
(cpt="E0601" and newbetag='5') or
(cpt="98941" and newbetag='6') or
(cpt="S9083" and newbetag='7'));
class GPnum cpt;
var payment ppu;
format cpt $cptdesc. newbetag $betosa.;
run;


Proc print data=StatAO_NHMO noobs;
var GPnum cpt N_PPU  Mean_PPU  Med_PPU  Min_PPU  Max_PPU  Kurt_PPU SKEW_PPU STD_PPU ;
format  N_PPU comma10. Mean_PPU  MED_PPU Min_PPU Max_PPU  comma7.2 Kurt_PPU SKEW_PPU STD_PPU comma7.4;
title3 "9. Univariate stats AFTER removing outliers-NON-HMO";
run;


ODS tagsets.MSOffice2K  close;


*****************************************
Print Contents and Sample print
*****************************************;

proc print uniform noobs data=inout.&sasDTout (obs=25);
where F_OANL='1';
  var GPnum pnum plan cpt newmod payment nunits PPU F_OANL mean_ppu;
title3 '10. Sample print of  inout.&sasDTout';
run;

proc contents data = inout.&sasDTout;
title3 '11. Sample print of  inout.&sasDTout';
run;


%jobend;

dm "log;    print file = I:\Analysis\Practrpt06\Programs\Analysis_Detect_Outliers.log replace";
dm "output; print file = I:\Analysis\Practrpt06\Programs\Analysis_Detect_Outliers.lst replace";
```

Appendix H – Medicare File Contents, Codebook, and SAS Programs Used in Medicare File Development

2006 Maryland Medical Care Database
Medicare File

MCR06_V2.SAS7BDAT

Contents Output

Data Set Name	IN3.MCR06_V2	**Observations**	30255855
Member Type	DATA	**Variables**	75
Engine	V9	**Indexes**	0
Created	Tuesday, June 03, 2008 12:47:44 PM	**Observation Length**	272
Last Modified	Tuesday, June 03, 2008 12:47:44 PM	**Deleted Observations**	0
Protection		**Compressed**	NO
Data Set Type		**Sorted**	NO
Label			
Data Representation	WINDOWS_32		
Encoding	wlatin1 Western (Windows)		
Engine/Host Dependent Information			
Data Set Page Size	16384		
Number of Data Set Pages	504265		
First Data Page	1		
Max Obs per Page	60		
Obs in First Data Page	22		
Number of Data Set Repairs	0		
File Name	I:\MCdb\mcr_fn\sasdata\mcr06_v2.sas7bdat		
Release Created	9.0101M3		
Host Created	XP_PRO		

Alphabetic List of Variables and Attributes

#	Variable	Type	Len	Format	Label
1	age1206_nr	Num	3		Patient Age as of 12/31/2006, not rounded
22	allow	Num	8		Allowed Amount-AA ($)
3	begday	Char	2		Service From Day
52	begindt	Num	8	MMDDYY10.	Service From Date
4	begmo	Char	2		Service From Month
10	begyr	Char	4		Service From Year
50	betosagg	Char	1		BETOS Category, Aggregated
49	betosdet	Char	3		BETOS Category, Detailed
21	bill	Num	8		Billed Charge-BC ($)
5	birthmo	Char	2		Patient Birth Month
12	birthyr	Char	4		Patient Birth Year
6	claimpdd	Char	2		Claim Paid Day
7	claimpmo	Char	2		Claim Paid Month
13	claimpyr	Char	4		Claim Paid Year
51	clmpdate	Num	8	MMDDYY10.	Claim Paid Date
72	cnty_p2	Char	3		County to Use for Pract. Report Analysis
44	county_p	Char	3		Patient County of Residence FIPS Code

32	cpt	Char	5		CPT-4/HCPCS/Other Procedure Code
14	dx1	Char	5		ICD-9-CM Diagnosis Code 1
15	dx2	Char	5		ICD-9-CM Diagnosis Code 2
16	dx3	Char	5		ICD-9-CM Diagnosis Code 3
17	dx4	Char	5		ICD-9-CM Diagnosis Code 4
73	encnt_id	Char	8		Sequential Record Number
8	endday	Char	2		Service Thru Day
53	enddt	Num	8	MMDDYY10.	Service Thru Date
9	endmo	Char	2		Service Thru Month
11	endyr	Char	4		Service Thru Year
56	f_age	Char	1		Valid Age Flag(age1206_nr 65-110) (1/0)
75	f_agecat	Char	1		Age Category Code
54	f_baddt	Char	1		Service Date Imputation Flag
45	f_county	Char	1		Patient Residence County Known (1/0)
41	f_cpt	Char	1		Procedure Code Category (1-7)
71	f_demog	Char	1		Demographic Donor for Analysis File? (1/0)
74	f_fac	Char	1		Service Performed in a Facility? (1/0)
40	f_finms	Char	1		All 4 Financial Vars=0/Missing (1/0)
63	f_imprvu	Char	1		RVU Imputation Flag (1/0,2)
36	f_mcr2nd	Char	1		Medicare is Secondary Payer (1/0)
35	f_modf	Char	1		TC or SG in Modifiers I or II (1/0)
70	f_oddsvc	Char	2		Odd Service Flag
57	f_patid	Char	1		Valid Patient ID (1/0)
46	f_paymd	Char	1		Payment Category (1=Neg,2=0,3=>0,4=Miss)
55	f_serv_a	Char	1		Service Used in Analysis (1/0)
38	f_uncv	Char	1		Uncovered (AA&RA=0/Miss) Service (1/0)
39	f_yr07	Char	1		Service Provided in 2006+ (1/0)
18	fedtaxid	Char	10		Practitioner Federal Tax ID
47	instate	Char	1		MD Service Location ZIP (1/0/blank)
42	los	Num	3		Length of Service(1 for Same&Next Day)
34	mcrstat	Char	2		Medicare Status Code
19	mod1	Char	2		AMA Modifier I
20	mod2	Char	2		AMA Modifier II
69	newbetag	Char	1		Aggregated BETOS, Reclassified
68	newbetdt	Char	3		Detailed BETOS, Reclassified
67	newmod	Char	2		Imputed Modifier/Last Imputation
43	numdxc	Num	3		Number of Diagnosis Codes, Calculated
31	patid	Char	9		Patient ID, Encrypted
24	patliab	Num	8		Patient Liability-PL ($)
58	pay_calc	Char	1		Method for Calculating Medicare Payment
65	pay_mcr	Num	8		Theoretical Medicare Payment
48	payertyp	Char	1		Payer Type (4=MCRNHMO)
37	payment	Num	8		Payment=Sum(RA,PL) ($)
66	plan	Char	6		Plan Type
30	provid	Char	6		Servicing Practitioner Identifier
23	reimb	Num	8		Reimbursement Amount-RA ($)
61	rvu	Num	8		Assigned RVU
64	rvu_i	Num	8		Assigned/Imputed RVU

60	rvu_mp	Num	4		Medicare Phys Fee Schedule RVU, Malpractice
62	rvu_pe	Num	8		Medicare Phys Fee Schedule RVU, Pract Expense
59	rvu_w	Num	4		Medicare Phys Fee Schedule RVU, Work Component
28	sex	Char	1		Patient Sex
25	spec	Char	3		Practitioner Specialty
27	svcplace	Char	2		Place of Service
33	unitind	Char	1		Service Unit Indicator
2	units	Num	3		Number of Units for a Service
26	zip_p	Char	5		Patient Residence ZIP Code
29	zip_s	Char	5		Service Location ZIP Code

2005 Maryland Medical Care Database
Data Description Summary
Universe: Outpatient Medical Services
Number of Services: 30,255,855

Number of Variables: 75
File Name: MCR06_V2.SAS7BDAT

File Creation Date: May 05, 2008

VARIABLE NAME	VARIABLE DESCRIPTION	VARIABLE TYPE	DISCRETE VARIABLES			NON-DISCRETE VARIABLES			
			Variable Value	Value Description	Number of Services	Number of Missing Value	Minimum Value	Mean Value	Miximum Value
AGE1206_NR	Patient Age as of 12/31/2006, not rounded	Numeric				80	3	74	109
ALLOW	Allowed Amount-AA ($)	Numeric				0	0	67	44,325
BEGDAY	Service From Day	Character	01-31	Day = 01-31	30,255,855				
BEGINDT	Service From Date	Date	Reported	Date Complete	30,255,855				
BEGMO	Service From Month	Character	01-12	Month = 01-12	30,255,855				
BEGYR	Service From Year	Character	2006	Year = 2006	30,255,855				
BETOSAGG	BETOS Category, Aggregated	Character	Missing	Code Not Assigned	393,684				
			1	Evaluation and Management	9,557,316				
			2	Procedures	4,925,610				
			3	Imaging	3,598,489				
			4	Tests	9,792,998				
			5	Durable Medical Equipment	31,505				
			6	Other	1,848,791				
			7	Exceptions/Unclassified	107,462				
BETOSDET	BETOS Category, Detailed	Character	Missing	Code Not Assigned	393,684				
			D1A	Medical/Surgical Supplies	9,228				
			D1C	Oxygen and Supplies	153				
			D1D	Wheelchairs	10				
			D1E	Other DME	1,931				
			D1F	Orthotic Devices	17,962				
			D1G	Drugs Administered Through DME	2,221				
			I1A	Standard Imaging - Chest	690,954				
			I1B	Standard Imaging - Musculoskeletal	559,336				
			I1C	Standard Imaging - Breast	24,666				

			I1D	Standard Imaging - Contrast Gastrointestinal	35,590				
			I1E	Standard Imaging - Nuclear Medicine	407,339				
			I1F	Standard Imaging - Other	194,809				
			I2A	Advanced Imaging - CAT: Head	141,676				
			I2B	Advanced Imaging - CAT: Other	341,377				
			I2C	Advanced Imaging - MRI: Brain	62,657				
			I2D	Advanced Imaging - MRI: Other	116,127				
			I3A	Echography - Eye	38,698				
			I3B	Echography - Abdomen/Pelvis	109,703				
			I3C	Echography - Heart	469,756				
			I3D	Echography - Carotid Arteries	68,886				
			I3E	Echography - Prostate, Transrectal	6,795				
			I3F	Echography - Other	163,562				
			I4A	Imaging/Procedure - Heart, including Cardiac Catheterization	87,266				
			I4B	Imaging/Procedure - Other	79,292				
			M1A	Office Visits - New	272,363				
			M1B	Office Visits - Established	4,263,032				
			M2A	Hospital Visit - Initial	205,353				
			M2B	Hospital Visit - Subsequent	1,617,588				
			M2C	Hospital Visit - Critical Care	122,156				
			M3	Emergency Room Visit	368,146				
			M4A	Home Visit	29,429				
			M4B	Nursing Home Visit	499,709				
			M5B	Specialist - Psychiatry	517,201				
			M5C	Specialist - Ophthalmology	891,658				
			M5D	Specialist - Other	83,451				
			M6	Consultations	687,230				
			O1A	Ambulance	315,331				
			O1B	Chiropractic	238,567				
			O1C	Enteral and Parenteral	1,013				
			O1D	Chemotherapy	73,041				
			O1E	Other Drugs	546,243				
			O1F	Vision, Hearing and Speech Services	2,020				
			O1G	Influenza Immunization	672,576				
			P0	Anesthesia	393,038				
			P1A	Major Procedure - Breast	960				
			P1B	Major Procedure - Colectomy	3,573				
			P1C	Major Procedure - Cholecystectomy	891				

			P1D	Major Procedure - TURP	1,955				
			P1E	Major Procedure - Hysterectomy	1,226				
			P1F	Major Procedure - Explor/Decompr/ExcisDisc	12,960				
			P1G	Major Procedure - Other and Major Maternal Procedures	67,331				
			P2A	Major Procedure, Cardiovascular - CABG	6,976				
			P2B	Major Procedure, Cardiovascular - Aneurysm Repair	1,206				
			P2C	Major Procedure, Cardiovascular - Thromboendarterectomy	1,919				
			P2D	Major Procedure, Cardiovascular - Coronary Angioplasty (PTCA)	14,140				
			P2E	Major Procedure, Cardiovascular - Pacemaker Insertion	8,903				
			P2F	Major Procedure, Cardiovascular - Other	75,585				
			P3A	Major Procedure, Orthopedic - Hip Fracture Repair	4,197				
			P3B	Major Procedure, Orthopedic - Hip Replacement	4,019				
			P3C	Major Procedure, Orthopedic - Knee Replacement	8,456				
			P3D	Major Procedure, Orthopedic - Other	32,342				
			P4A	Eye Procedures - Corneal Transplant	666				
			P4B	Eye Procedures - Cataract Removal/Lens Insertion	57,764				
			P4C	Eye Procedures - Retinal Detachment	1,443				
			P4D	Eye Procedures - Treatment of Retinal Lesions	8,053				
			P4E	Eye - Other	58,301				
			P5A	Ambulatory Procedures - Skin	493,708				
			P5B	Ambulatory Procedures - Musculoskeletal	25,964				
			P5C	Ambulatory Procedures - Inguinal Hernia Repair	2,574				
			P5D	Ambulatory Procedures - Lithotripsy	1,095				
			P5E	Ambulatory Procedures - Other	278,792				
			P6A	Minor Procedures - Skin	369,369				
			P6B	Minor Procedures - Musculoskeletal	234,720				
			P6C	Minor Procedures - Other (Medicare Fee Schedule)	1,933,902				
			P6D	Minor Procedures - Other (Non-Medicare Fee Schedule)	5,987				
			P7A	Oncology - Radiation Therapy	155,281				
			P7B	Oncology - Other	237,700				
			P8A	Endoscopy - Arthroscopy	12,832				
			P8B	Endoscopy - Upper Gastrointestinal	67,334				
			P8C	Endoscopy - Sigmoidoscopy	4,784				
			P8D	Endoscopy - Colonoscopy	104,915				
			P8E	Endoscopy - Cystoscopy	56,663				
			P8F	Endoscopy - Bronchoscopy	10,304				
			P8G	Endoscopy - Lararoscopic Cholecystectomy	4,192				

			P8H	Endoscopy - Laryngoscopy	15,514				
			P8I	Endoscopy - Other	19,212				
			P9A	Dialysis Services (Medicare Fee Schedule)	56,667				
			P9B	Dialysis Services (Non-Medicare Fee Schedule)	68,197				
			T1A	Lab Tests - Routine Venipuncture (Non-Medicare Fee Schedule)	1,246,332				
			T1B	Lab Tests - Automated General Profiles	1,380,644				
			T1C	Lab Tests - Urinalysis	391,982				
			T1D	Lab Tests - Blood Counts	804,383				
			T1E	Lab Tests - Glucose	110,999				
			T1F	Lab Tests - Bacterial Cultures	217,054				
			T1G	Lab Tests - Other (Medicare Fee Schedule)	436,685				
			T1H	Lab Test, Other (Non-Medicare Fee Schedule)	3,709,300				
			T2A	Other Tests - Electrocardiograms	873,639				
			T2B	Other Tests - Cardiovascular Stress Tests	136,071				
			T2C	Other Tests - EKG Monitoring	36,692				
			T2D	Other Tests - Other	449,217				
			Y1	Medicare Fee Schedule	45,919				
			Y2	Other - Non-Medicare Fee Schedule	60,429				
			Z2	Undefined Codes	1,114				
BILL	Billed Charge-BC ($)	Numeric			0	0	0	167	99,242
BIRTHMO	Patient Birth Month	Character	01-12	Month = 01-12	30,255,855				
BIRTHYR	Patient Birth Year	Character	1880-1899	Year = 1880-1899	589				
			1900-1919	Year = 1900-1919	2,945,212				
			1920-1939	Year = 1920-1939	21,684,528				
			1940-1959	Year = 1940-1959	4,558,184				
			1960-1979	Year = 1960-1979	1,003,070				
			1980-1999	Year = 1980-1999	63,573				
			2000-2005	Year = 2000-2005	699				
CLAIMPDD	Claim Paid Day	Character	01-31	Day = 01-31	30,255,855				
CLAIMPMO	Claim Paid Month	Character	01-12	Month = 01-12	30,255,855				
CLAIMPYR	Claim Paid Year	Character	2006	Year = 2006	27,491,815				
			2007	Year = 2007	2,764,040				
CLMPDATE	Claim Paid Date	Date	Reported	Date Complete	30,255,855				
CNTY_P2	County to Use for Pract. Report Analysis	Character	Missing	Patient Residence County Unknown	47,954				

			1	Allegany	577,016				
			3	Anne Arundel	2,834,383				
			5	Baltimore	5,361,804				
			9	Calvert	439,690				
			11	Caroline	171,734				
			13	Carroll	1,024,988				
			15	Cecil	524,180				
			17	Charles	569,694				
			19	Dorchester	209,579				
			21	Frederick	1,071,214				
			23	Garrett	164,070				
			25	Harford	1,287,713				
			27	Howard	921,660				
			29	Kent	194,305				
			31	Montgomery	4,909,567				
			33	Prince Georges	3,328,211				
			35	Queen Annes	235,234				
			37	Saint Marys	476,934				
			39	Somerset	158,675				
			41	Talbot	297,649				
			43	Washington	976,748				
			45	Wicomico	575,811				
			47	Worcester	453,116				
			510	Baltimore City	3,443,926				
COUNTY_P	Patient County of Residence FIPS Code	Character	Missing	Patient Residence County Unknown	27,460				
			1	Allegany	576,987				
			3	Anne Arundel	2,839,971				
			5	Baltimore	5,365,364				
			9	Calvert	440,888				
			11	Caroline	173,009				
			13	Carroll	1,032,649				
			15	Cecil	525,322				
			17	Charles	569,352				
			19	Dorchester	208,982				
			21	Frederick	1,075,531				
			23	Garrett	164,576				
			25	Harford	1,294,264				
			27	Howard	928,437				

			29	Kent	194,009				
			31	Montgomery	4,911,223				
			33	Prince Georges	3,314,201				
			35	Queen Annes	236,033				
			37	Saint Marys	480,382				
			39	Somerset	158,291				
			41	Talbot	298,027				
			43	Washington	978,130				
			45	Wicomico	578,674				
			47	Worcester	453,016				
			510	Baltimore City	3,431,077				
CPT	CPT-4/HCPCS/Other Procedure Code	Character	Reported	Value Reported	30,255,855				
DX1	ICD-9-CM Diagnosis Code 1	Character	0		3,954				
			1		30,251,901				
DX2	ICD-9-CM Diagnosis Code 2	Character	0		11,196,675				
			1		19,059,180				
DX3	ICD-9-CM Diagnosis Code 3	Character	0		18,561,225				
			1		11,694,630				
DX4	ICD-9-CM Diagnosis Code 4	Character	0		23,846,594				
			1		6,409,261				
ENCNT_ID	Sequential Record Number	Character	Assigned	Sequential ID Assigned	30,255,855				
ENDDAY	Service Thru Day	Character	01-31	Day = 01-31	30,255,855				
ENDDT	Service Thru Date	Date	Reported	Date Complete	30,255,855				
ENDMO	Service Thru Month	Character	01-12	Month = 01-12	30,255,855				
ENDYR	Service Thru Year	Character	2006	Year = 2006	30,255,855				
FEDTAXID	Practitioner Federal Tax ID	Character	Blank	Value Missing	1				
			Reported	Value Reported	30,255,854				
F_AGE	Valid Age Flag(age1206_nr 65-110) (1/0)	Character	0	AGE1206 Invalid - Missing, <65, >110	3,879,008				
			1	AGE1206 Valid (65-110 inclusive)	26,376,847				
F_AGECAT	Age Category Code	Character	1	AGE12006_nr Missing	80				
			2	0-64 Years Old	3,878,928				
			3	65-110 Years Old	26,376,847				
F_BADDT	Service Date Imputation Flag	Character	4	Both dates good	30,255,855				
F_COUNTY	Patient Residence County Known (1/0)	Character	0	No - County of Residence is Not Known	27,460				
			1	Yes - County of Residence is Known	30,228,395				
F_CPT	Procedure Code Category (1-7)	Character	2	HCPCS Code	2,266,091				
			3	CPT-4 Code	27,978,855				
			7	Procedure Code Invalid	10,909				

F_DEMOG	Demographic Donor for Analysis File? (1/0)	Character	0	No - Demographics Not Used in Analysis	29,641,201				
			1	Yes - Demographics Used in Analysis	614,654				
F_FAC	Service Performed in a Facility? (1/0)	Character	0	Service Not Provided in a Facility	23,229,895				
			1	Service Provided in a Facility	7,025,960				
F_FINMS	All 4 Financial Vars=0/Missing (1/0)	Character	0	No - Not All Financial Variables are Equal to 0/Missing	30,255,068				
			1	Yes - All Financial Variables are Equal to 0/Missing	787				
F_IMPRVU	RVU Imputation Flag (1/0,2)	Character	0	No - RVU Not Imputed	19,965,995				
			1	Yes - RVU Imputed	9,291,378				
			2	No - RVU Not Imputed, No Donor RVU	998,482				
F_MCR2ND	Medicare is Secondary Payer (1/0)	Character	0	No - Medicare Not Secondary Coverage	30,072,163				
			1	Yes - Medicare is Secondary Coverage	183,692				
F_MODF	TC or SG in Modifiers I or II (1/0)	Character	0	No - TC or SG Modifier Not Present	29,916,022				
			1	Yes - TC or SG Modifier Present	339,833				
F_ODDSVC	Odd Service Flag	Character	Blank	Not Odd Services	20,684,277				
			3	Payment <= 1, Allowed Amount <= 0	2,026,703				
			4	Identified Anesthesia Outside Of Range	83,858				
			5	Identified Facility Bill	437,812				
			6	Minor Service Assoc With Major Service	229,924				
			8	Payer not NONHMO,HMOFFS,HMOCAP	2,941,462				
			9	Not First Dupe by Key, Descending Payment	1,238,440				
			10	Actuaries Trim, Payment, Low	67,896				
			11	Actuaries Trim, Payment, High	1,671				
			12	Actuaries Trim, RVU_I, Low	70,931				
			13	Actuaries Trim, RVU_I, High	22,383				
			15	RVU_I <= 0	570,754				
			99	More Than 1 Service	1,879,744				
F_PATID	Valid Patient ID (1/0)	Character	1	Yes-Patient ID is Valid	30,255,855				
F_PAYMD	Payment Category (1=Neg,2=0,3=>0,4=Miss)	Character	2	Zero payment	2,887,856				
			3	Positive payment	27,367,999				
F_SERV_A	Service Used in Analysis (1/0)	Character	0	No - Service Not Used in Analysis	3,357,819				
			1	Yes - Service Used in Analysis	26,898,036				
F_UNCV	Uncovered (AA&RA=0/Miss) Service (1/0)	Character	0	No - Service is Not Uncovered	27,392,802				
			1	Yes - Service is Uncovered	2,863,053				
F_YR07	Service Provided in 2006+ (1/0)	Character	0	No - Service Provided in 2006	30,255,855				
INSTATE	MD Service Location ZIP (1/0/blank)	Character	0	Service Location Valid, Not MD (Based on Zipcode)	5,531				
			1	Service Location in MD (Based on Zipcode)	30,250,324				
LOS	Length of Service(1 for Same&Next Day)	Numeric			0	0	1	2	364
MCRSTAT	Medicare Status Code	Character	10	Aged Without ESRD	25,602,489				

			11	Aged With ESRD	687,220				
			20	Disabled Without ESRD	3,282,294				
			21	Disabled With ESRD	421,430				
			31	ESRD Only	262,422				
MOD1	AMA Modifier I	Character	Blank	Missing Code	20,455,282				
			Other	All Other Codes	5,450,459				
			21	Prolonged Evaluatn/Mgmt(E/M) Serv	684				
			22	Unusual Procedural Services	1,259				
			23	Unusual Anesthesia	4				
			24	Unrel E/M Serv by Same Phys Postop	42,308				
			25	Sep Ident. E/M by Same Phys/day	812,670				
			26	Professional Component	2,586,120				
			32	Mandated Services	1,356				
			47	Anesthesia by Surgeon	112				
			50	Bilateral Procedure	62,080				
			51	Multiple Procedures	135,040				
			52	Reduced Services	10,662				
			54	Surgical Care Only	2,881				
			55	Postoperative Management Only	1,256				
			56	Preoperative Management Only	356				
			57	Decision for Surgery	26,296				
			58	Staged/Rel Proc by Same Phys Postop	21,727				
			62	Two Surgeons	2,254				
			76	Repeat Procedure by Same Physician	45,961				
			77	Repeat Procedure by Another Physician	5,757				
			78	Return to Op Rm for Rel Proc Postop	10,866				
			79	Unrelated Proc by Same Phy Postop	40,659				
			80	Assistant Surgeon	14,975				
			81	Minimum Assisting Surgeon	756				
			82	Assist Surgeon(qual res phy unav)	3,504				
			90	Reference (Outside) Laboratory	113,199				
			99	Multiple Modifiers	29				
			QX	Cert Reg Nurse Anesth. w/o doc dir	36,853				
			QZ	Cert Reg Nurse Anesth (CRNA)	35,405				
			SG	HCFA Ambulatory Surgery	164,506				
			TC	Radiology: Technical Component	170,579				
MOD2	AMA Modifier II	Character	Blank	Missing Code	28,466,594				
			Other	All Other Codes	1,550,205				

			21	Prolonged Evaluatn/Mgmt(E/M) Serv	99				
			22	Unusual Procedural Services	275				
			23	Unusual Anesthesia	1				
			24	Unrel E/M Serv by Same Phys Postop	934				
			25	Sep Ident. E/M by Same Phys/day	20,392				
			26	Professional Component	12				
			32	Mandated Services	3				
			47	Anesthesia by Surgeon	3				
			50	Bilateral Procedure	8,579				
			51	Multiple Procedures	122,290				
			52	Reduced Services	6,182				
			54	Surgical Care Only	686				
			55	Postoperative Management Only	113				
			56	Preoperative Management Only	17				
			57	Decision for Surgery	4,218				
			58	Staged/Rel Proc by Same Phys Postop	2,892				
			62	Two Surgeons	128				
			76	Repeat Procedure by Same Physician	32,643				
			77	Repeat Procedure by Another Physician	19,830				
			78	Return to Op Rm for Rel Proc Postop	1,232				
			79	Unrelated Proc by Same Phy Postop	10,872				
			80	Assistant Surgeon	1,157				
			81	Minimum Assisting Surgeon	52				
			82	Assist Surgeon(qual res phy unav)	648				
			90	Reference (Outside) Laboratory	385				
			99	Multiple Modifiers	14				
			QX	Cert Reg Nurse Anesth. w/o doc dir	5				
			QZ	Cert Reg Nurse Anesth (CRNA)	1				
			SG	HCFA Ambulatory Surgery	5,392				
			TC	Radiology: Technical Component	1				
NEWBETAG	Aggregated BETOS, Reclassified	Character	1	Evaluation and Management	9,557,316				
			2	Procedures	4,925,607				
			3	Imaging	3,598,489				
			4	Tests	9,793,001				
			5	Durable Medical Equipment	31,505				
			X	Missing/Not Assigned/Other/Exceptions	2,349,937				
NEWBETDT	Detailed BETOS, Reclassified	Character	D1A	Medical/Surgical Supplies	9,228				
			D1C	Oxygen and Supplies	153				

			D1D	Wheelchairs	10				
			D1E	Other DME	1,931				
			D1F	Orthotic Devices	17,962				
			D1G	Drugs Administered Through DME	2,221				
			I1A	Standard Imaging - Chest	690,954				
			I1B	Standard Imaging - Musculoskeletal	559,336				
			I1C	Standard Imaging - Breast	24,666				
			I1D	Standard Imaging - Contrast Gastrointestinal	35,590				
			I1E	Standard Imaging - Nuclear Medicine	407,339				
			I1F	Standard Imaging - Other	194,809				
			I2A	Advanced Imaging - CAT: Head	141,676				
			I2B	Advanced Imaging - CAT: Other	341,377				
			I2C	Advanced Imaging - MRI: Brain	62,657				
			I2D	Advanced Imaging - MRI: Other	116,127				
			I3A	Echography - Eye	38,698				
			I3B	Echography - Abdomen/Pelvis	109,703				
			I3C	Echography - Heart	469,756				
			I3D	Echography - Carotid Arteries	68,886				
			I3E	Echography - Prostate, Transrectal	6,795				
			I3F	Echography - Other	163,562				
			I4A	Imaging/Procedure - Heart, including Cardiac Catheterization	87,266				
			I4B	Imaging/Procedure - Other	79,292				
			M1A	Office Visits - New	272,363				
			M1B	Office Visits - Established	4,263,032				
			M2A	Hospital Visit - Initial	205,353				
			M2B	Hospital Visit - Subsequent	1,617,588				
			M2C	Hospital Visit - Critical Care	122,156				
			M3	Emergency Room Visit	368,146				
			M4A	Home Visit	29,429				
			M4B	Nursing Home Visit	499,709				
			M5B	Specialist - Psychiatry	517,201				
			M5C	Specialist - Ophthalmology	891,658				
			M5D	Specialist - Other	83,451				
			M6	Consultations	687,230				
			O1A	Ambulance	315,331				
			O1B	Chiropractic	238,567				
			O1C	Enteral and Parenteral	1,013				
			O1D	Chemotherapy	73,041				

			O1E	Other Drugs	546,243				
			O1F	Vision, Hearing and Speech Services	2,020				
			O1G	Influenza Immunization	672,577				
			P0	Anesthesia	393,034				
			P1A	Major Procedure - Breast	960				
			P1B	Major Procedure - Colectomy	3,573				
			P1C	Major Procedure - Cholecystectomy	891				
			P1D	Major Procedure - TURP	1,955				
			P1E	Major Procedure - Hysterectomy	1,226				
			P1F	Major Procedure - Explor/Decompr/ExcisDisc	12,960				
			P1G	Major Procedure - Other and Major Maternal Procedures	67,331				
			P2A	Major Procedure, Cardiovascular - CABG	6,976				
			P2B	Major Procedure, Cardiovascular - Aneurysm Repair	1,206				
			P2C	Major Procedure, Cardiovascular - Thromboendarterectomy	1,919				
			P2D	Major Procedure, Cardiovascular - Coronary Angioplasty (PTCA)	14,140				
			P2E	Major Procedure, Cardiovascular - Pacemaker Insertion	8,903				
			P2F	Major Procedure, Cardiovascular - Other	75,585				
			P3A	Major Procedure, Orthopedic - Hip Fracture Repair	4,197				
			P3B	Major Procedure, Orthopedic - Hip Replacement	4,019				
			P3C	Major Procedure, Orthopedic - Knee Replacement	8,456				
			P3D	Major Procedure, Orthopedic - Other	32,342				
			P4A	Eye Procedures - Corneal Transplant	666				
			P4B	Eye Procedures - Cataract Removal/Lens Insertion	57,764				
			P4C	Eye Procedures - Retinal Detachment	1,443				
			P4D	Eye Procedures - Treatment of Retinal Lesions	8,053				
			P4E	Eye - Other	58,301				
			P5A	Ambulatory Procedures - Skin	493,708				
			P5B	Ambulatory Procedures - Musculoskeletal	25,964				
			P5C	Ambulatory Procedures - Inguinal Hernia Repair	2,574				
			P5D	Ambulatory Procedures - Lithotripsy	1,095				
			P5E	Ambulatory Procedures - Other	278,792				
			P6A	Minor Procedures - Skin	369,369				
			P6B	Minor Procedures - Musculoskeletal	234,720				
			P6C	Minor Procedures - Other (Medicare Fee Schedule)	1,933,902				
			P6D	Minor Procedures - Other (Non-Medicare Fee Schedule)	5,987				
			P7A	Oncology - Radiation Therapy	155,281				
			P7B	Oncology - Other	237,700				

			P8A	Endoscopy - Arthroscopy	12,832				
			P8B	Endoscopy - Upper Gastrointestinal	67,334				
			P8C	Endoscopy - Sigmoidoscopy	4,784				
			P8D	Endoscopy - Colonoscopy	104,915				
			P8E	Endoscopy - Cystoscopy	56,663				
			P8F	Endoscopy - Bronchoscopy	10,304				
			P8G	Endoscopy - Lararoscopic Cholecystectomy	4,193				
			P8H	Endoscopy - Laryngoscopy	15,514				
			P8I	Endoscopy - Other	19,212				
			P9A	Dialysis Services (Medicare Fee Schedule)	56,667				
			P9B	Dialysis Services (Non-Medicare Fee Schedule)	68,197				
			T1A	Lab Tests - Routine Venipuncture (Non-Medicare Fee Schedule)	1,246,332				
			T1B	Lab Tests - Automated General Profiles	1,380,647				
			T1C	Lab Tests - Urinalysis	391,982				
			T1D	Lab Tests - Blood Counts	804,383				
			T1E	Lab Tests - Glucose	110,999				
			T1F	Lab Tests - Bacterial Cultures	217,054				
			T1G	Lab Tests - Other (Medicare Fee Schedule)	436,685				
			T1H	Lab Test, Other (Non-Medicare Fee Schedule)	3,709,300				
			T2A	Other Tests - Electrocardiograms	873,639				
			T2B	Other Tests - Cardiovascular Stress Tests	136,071				
			T2C	Other Tests - EKG Monitoring	36,692				
			T2D	Other Tests - Other	449,217				
			XXX	Missing/Code Not Assigned	393,683				
			Y1	Medicare Fee Schedule	45,919				
			Y2	Other - Non-Medicare Fee Schedule	60,429				
			Z2	Undefined Codes	1,114				
NEWMOD	Imputed Modifier/Last Imputation	Character	Blank	Missing Code	27,474,518				
			26	Professional Component	2,611,158				
			TC	Radiology: Technical Component	170,179				
NUMDXC	Number of Diagnosis Codes, Calculated	Numeric			0	0	1	2	4
PATID	Patient ID, Encrypted	Character	Reported	Value Reported	30,255,855				
PATLIAB	Patient Liability-PL ($)	Numeric			0	0	0	14	8,865
PAYERTYP	Payer Type (4=MCRNHMO)	Character	Missing	Group Not Assigned - Age=0-64	3,879,008				
			4	Medicare Non-HMO	26,376,847				
PAYMENT	Payment=Sum(RA,PL) ($)	Numeric			0	0	0	66	44,325
PAY_CALC	Method for Calculating Medicare Payment	Character	0	Do not calculate payment	10,909				

			2	Calculate Using Facility RVUs	4,129,922				
			3	Calculate Using Non-Facility RVUs	9,710,617				
			4	Procedure Code is a Lab Fee Schedule Code	5,965,666				
			5	Procedure Code with an Imputed Total RVU but no Components	878,072				
			6	Flagged as Odd Service - Do not Use MCR Rate	9,560,669				
PAY_MCR	Theoretical Medicare Payment	Numeric			0	9,571,578	2	63	5,779
PLAN	Plan Type	Character	Blank	Not Assigned	3,879,008				
			MCR	Medicare Non-HMO Plan	26,376,847				
PROVID	Servicing Practitioner Identifier	Character	Blank	Value Missing	2,735,416				
			Reported	Value Reported	27,520,439				
REIMB	Reimbursement Amount-RA ($)	Numeric			0	0	0	52	35,460
RVU	Assigned RVU	Numeric			0	10,300,769	0	2	130
RVU_I	Assigned/Imputed RVU	Numeric			0	1,009,391	0	2	133
RVU_MP	Medicare Phys Fee Schedule RVU, Malpractice	Numeric			0	1,009,391	0	0	17
RVU_PE	Medicare Phys Fee Schedule RVU, Pract Expense	Numeric			0	1,009,391	0	1	126
RVU_W	Medicare Phys Fee Schedule RVU, Work Component	Numeric			0	1,009,391	0	1	81
SEX	Patient Sex	Character	1	Male	12,276,646				
			2	Female	17,979,209				
SPEC	Practitioner Specialty	Character	1	General Practice	103,830				
			2	General Surgery	360,853				
			3	Allergy & Immunology	55,877				
			4	Anesthesiology	311,630				
			5	Cardiology	2,337,112				
			6	Dermatology	541,688				
			7	Emergency Medicine	432,183				
			8	Endocrinology Medicine	194,743				
			9	Family Practice	1,141,820				
			10	Gastroenterology	330,669				
			11	Geriatrics	79,710				
			12	Hand Surgery	10,770				
			13	Hematology	83,898				
			14	Internal Medicine	4,604,541				
			15	Infectious Disease	121,658				
			16	Nephrology	374,334				
			17	Neurology	262,141				
			18	Nuclear Medicine	68,241				
			19	Oncology	1,523,525				

			20	Ophthalmology	1,085,858				
			21	Orthopedic Surgery	817,403				
			22	Osteopathy (include manipulations)	6,428				
			23	Otology/Laryngo/Rhino/Otolaryngo	229,014				
			24	Pathology	280,937				
			25	Pediatrics	28,324				
			26	Peripheral Vascular Disease/Surgery	113,085				
			27	Plastic Surgery	55,495				
			28	Physical Medicine & Rehabilitation	208,782				
			29	Proctology	13,311				
			30	Psychiatry	390,338				
			31	Pulmonary Disease	478,206				
			32	Radiology	2,322,979				
			33	Rheumatology	293,933				
			34	Surgical Specialty Not Listed Here	57,049				
			35	Thoracic Surgery	55,416				
			36	Urology	520,088				
			37	Misc Special (pub health/indusl med)	70,438				
			38	Physician w/o Identfd Spec/Not Listd	84,078				
			39	Obstetrics/Gynecology	210,963				
			42	Ambulance Services	315,610				
			43	Audiologist/Speech Pathologist	31,288				
			44	Chiropractor	457,849				
			45	Freestanding Clinic-Not a Govt Agncy	3,050				
			47	Dietitian/Licensed Nutritionist	2,411				
			49	Advanced Practice Nurse: Anesthetist	74,103				
			50	Advanced Practice Nurse: Midwife	1,469				
			51	Advancd Practice Nurse: Nurse Practitioner	296,639				
			53	Nurse Other than Advanced Practice	12				
			54	Occupational Therapist	59,146				
			55	Optometrist	131,432				
			56	Podiatrist	737,107				
			57	Physical Therapist	1,198,741				
			58	Psychologist	99,220				
			59	Clinical Social Worker	124,703				
			60	Pub Health/Welf Agy-fed & local govt	51,714				
			61	Voluntary Health Agency	2				
			62	Other Specialty Not Listed Above	124,378				

			80	Freestanding Pharmacy incl grocery	87				
			82	Independent Laboratory	5,713,340				
			83	Independent Medical Supply Company	2,543				
			85	All Other Supplies	162,961				
			91	Freestanding Surgical Facility	183,082				
			92	Freestanding Imaging Center	1				
			93	Other Facility	223,619				
SVCPLACE	Place of Service	Character	Other	Invalid Code	150,534				
			11	Provider's Office	15,867,664				
			12	Patient's Home	196,580				
			21	Inpatient Hospital	4,301,307				
			22	Outpatient Hospital	1,389,070				
			23	Emergency Room (Hospital Portion)	863,539				
			24	Ambulatory Surgical Center	444,896				
			25	Birthing Center	10				
			26	Military Treatment Facility	34				
			32	Nursing Facility	832,405				
			33	Custodial Care Facility	47,906				
			34	Hospice	362				
			41	Ambulance-Land	312,822				
			42	Ambulance-Air or Water	2,762				
			51	Inpatient Psychiatric Facility	19,140				
			52	Psychiatric Facility Partial Hospitaln	7,425				
			53	Community Mental Health Center	52,054				
			54	Intermedi Care FCLTY/Mentally Retarded	405				
			55	Residential Subst Abuse Treatment Fcty	20				
			56	Psychiatric Residential Treatment Ctr	47				
			61	Comprehensive Inpat Rehabilitatn Fclty	6,616				
			62	Comprehensive Outpat Rehabilitatn Fcty	111				
			65	End Stage Renal Disease Treatment Fcty	58,175				
			71	State or Local Public Health Clinic	34,685				
			81	Independent Laboratory	5,602,356				
			99	Not Coded or Other Unlisted Facility	64,930				
UNITIND	Service Unit Indicator	Character	Other	Invalid Code	8				
			1	Transportation Miles	229				
			2	Anesthesia Time Units	244,891				
			3	Visits	30,010,726				
			4	Oxygen Units	1				

UNITS	Number of Units for a Service	Numeric			0	0	0	2	999
ZIP_P	Patient Residence ZIP Code	Character	Reported	Value Reported	30,255,855				
ZIP_S	Service Location ZIP Code	Character	Blank	Value Missing	65				
			Reported	Value Reported	30,255,790				

Program	Formats Used	Input Dataset	Output Dataset	Purpose
PHYbene_load.sas	I:\MCdb\datacol\Formats\dental.sas	ps_bene_line_1.sas7bdat	psbene2006.sas7bdat	1. Convert data from HCFA Physician/Supplier format inta
	I:\MCdb\datacol\Formats\stnamefmt.sas	ps_bene_line_2.sas7bdat	psbene2006x.sas7bdat	a format similar to Maryland private payer files.
		ps_bene_line_3.sas7bdat		2. Provide FREQs and UNIVARIATEs for all variables.
		ps_bene_line_4.sas7bdat		
		ps_bene_line_5.sas7bdat		
		ps_bene_line_6.sas7bdat		
		ps_bene_line_7.sas7bdat		
		ps_bene_line_8.sas7bdat		
		ps_bene_line_9.sas7bdat		
mcr_1.sas	I:\mcdb\datacol\formats\uszipfmt.sas	psbene2006x.sas7bdat	mcr_1.sas7bdat	1. Create intermediate Medicare dataset from processed
	I:\mcdb\datacol\formats\mdstzipfmt.sas			HCFA file.
	I:\mcdb\datacol\formats\mdcyzipfmt.sas			2. Created, imputed or modified several variables comparable to
	I:\mcdb\datacol\formats\cptfmt.sas			the ones in the Encounter.
	I:\mcdb\datacol\formats\hcpcsfmt.sas			
	I:\mcdb\datacol\formats\revcode.sas			
	I:\mcdb\datacol\formats\icd9pr_05.sas			
	Format Library in			
	I:\MCdb\datacol\Formats\PERMANENT\BETOS_formats			
	I:\MCdb\datacol\Formats\PERMANENT\RVU_formats			
mcr_2.sas	Format Library in	mcr_1.sas7bdat	mcr_2.sas7bdat	Assign RVUs (Work,Practice, and Malpractice) to valid
	I:\MCdb\datacol\Formats\PERMANENT\RVU_formats			procedures. Medicare Payment Rates are also computed.
				The components are estimated for procedures with imputed
				RVUs
mcr_3.sas	N/A	mcr_2.sas7bdat	mcr_3.sas7bdat	1. Identify debit/credit pairs based on duplication of all financials
				in pnum/patient/service begin/service end/procedure code/provid
				2. Create F_INANAL flag identifying records passing global
				analysis screens
				3. Subset only records passing global screens and
				0 le age12XX le 64 screen

mcr_4.sas	Format Library in	mcr_3.sas7bdat	mcr_4.sas7bdat	1. Flag Odd Services
	I:\MCdb\datacol\Formats\PERMANENT\RVU_formats			2. Finalize medicare payment (PAY_MCR) by setting to missing
				of service is flagged "ODD"
				3. Modified 2 intermediate variables COND04 & COND05,
				which were used to generate variable F_ODDSVC
				(Odd Service Flag/)
mcr_5.sas	I:\mcdb\datacol\formats\mdregfmt.sas	mcr_4.sas7bdat	mcr_5.sas7bdat	1. Assign f_demog flag, create new county var to be used in
				analysis
				2. Use SEX as part of patient id, the only demographic variable
				distributed will be patient residence county.
				3. Flag demographic donor record.
				4. Distribute residence info
				5. Output is an interim dataset analysis rather than final
mcr_f.sas		mcr_5.sas7bdat	mcr06_V1.sas7bdat	Create final deliverable dataset
codebook_mcr_all_06.sas	I:\MCdb\mcr_fn\programs_2006\mcrfmt.sas	mcr06_V2.sas7bdat	mcrcball06_doc.sas7bdat	Create contents and codebook for the MCDB Documenation.
	I:\MCdb\mcr_fn\programs_2006\dxfmt.sas		mcr06_V2.sas7bdat	This is part of the annual deliverable.
				We re-created variable F_AGECAT and creaetd final Medicare
				file (MCD06_V2.) This is done for this year only.

```
%jobstart;

/* PROGRAM NAME: PHYSUPP_BENE.SAS                                          */
/***************************************************************************/
/* Programmer:   Jeff McCartney  9/28/2000                                 */
/* Convert data from HCFA Physician/Supplier format inta a format similar  */
/* to Maryland private payer files.                                        */
/* Also provide FREQs and UNIVARIATEs for all variables.                   */
/* Modified by Raymond Hu on 01/19/2000                                    */
/*       1) Add SEX,                                                       */
/* Modified by Raymond Hu on 02/01/2000                                    */
/*       1) use SUM function to compute PAYMENT                            */
/*       2) Donote delete PAYMENT=0 in this program                        */
/* Modified by Jeff McCartney on 02/03/2000                                */
/*       Error correction: previously, I defined PATID = CAN || BIC        */
/*       but it should be PATID = CAN || EQ_BIC !!!!!                      */
/* Modified by JOHN May on 10/15/2001                                      */
/*       Modified to take in version I.                                    */
/*       To calculate _aged, update procdate macro variable to begin date  */
/*       processing period.                                                */
/* Modified by JOHN May on 10/10/2003                                      */
/*       Updated the place of service format, map 31 to 32, and 72 to 71.  */
/*       Change call to dental format to dentexc.                          */
/***************************************************************************/
/***************************************************************************/
/* Program Name: i:\mcdb\datacol\psbs001\programs\physupxtr_beneficiary.sas */
/* Programmer:   Jeff McCartney  10/28/1999                                */
/* Convert the raw file of 1998 SAF Physician/Supplier Claims (100% of     */
/* all benes where SSS State of beneficiary indicates "Maryland")          */
/* into SAS.                                                               */
/* REVISED 2/10/2000: A bug that wrote out duplicates (for records passing */
/* the PRCGND and LLWCHG filter and singly for those not passing this      */
/* filter) has been corrected.  We expect the output file to have          */
/* 17,319,502 obs rather than 37,389,170 obs.                              */
/***************************************************************************/

*** modified 2/1/2005 to update crosswalk from CMS specialty codes to
    MHCC specialty codes - lsh ***;
*********************************************************************
Modified : Sireesha Thati -- 02/14/2006
Note: This year we got data from CMS in 9 SAS datasets
      (Before this we used to get data in the form of text files)
      Hence, we create PSBENE2005.Sas7bdat by setting all these
      9 SAS datasets that we got from CMS.All the 9 files contain header
      and line items..All the line items are processed using arrays.
      From year 2005 the fields CAN and EQ_BIC which are used to construct
             patid are set to blank by RDDC. Hence we assigned patid = BID_4573_1
             to uniquely identify the beneficiaries
             (see email from RDDC Support dated February 13, 2007)
*********************************************************************
Modified : Kasey Chen -- 01/08/2008
Note:  1. Input data sets are named as pb_md_2006_*.sas7bdat (* = 1 - 9).
       2. Changed service year to 2006.
       3. Revised one of included format file (stnamefmt.sas).
       4. Variable BID_4573_1 in 2005 has been changed to BID_6434_1 in 2006.
*********************************************************************;



%MACRO AGE (DATE1, DATE2); /* CALCULATES BENE AGE AS OF PROCESSING DATE */
 INT((INTCK('MONTH',&DATE1,&DATE2)-(DAY(&DATE1)>DAY(&DATE2)))/12);
%MEND AGE;

 /*  SET PROCESSING DATE SYMBOLIC PARAMETERS  */
%let procdate='01jan06';  %let yy=06; %let yyyy=2006;
run;

libname outdd  "I:\MCdb\mcr_fn\sasdata";
libname indd   "D:\SHEA 2006\data";

options nosource2;

%include "I:\MCdb\datacol\Formats\dental.sas";
%include "I:\MCdb\datacol\Formats\stnamefmt.sas";

options source2;

options obs=max;
data outdd.psbene&YYYY
    (drop = ANSHNT01 - ANSHNT13 ASTTCD01 - ASTTCD13 BETOS01 - BETOS13
                                                    CNMT01   - CNMT13   CRRPCL01 - CRRPCL13 DCMNCD01- DCMNCD13
                                                    DED_SW01 - DED_SW13 DMPRC01  - DMPRC13        DMSTDT01- DMSTDT13
                                                    EXPDT101 - EXPDT113 EXPDT201 - EXPDT213 HCFPCL01- HCFPCL13
                                                    HCPSCD01 - HCPSCD13 LABAMT01 - LABAMT13 LBNPMT01- LBNPMT13
                                                    LCLYCD01 - LCLYCD13 LDDMT01  - LDDMT13        LLMTMT01- LLMTMT13
                                                    LLWCHG01 - LLWCHG13 LNDGNS01 - LNDGNS13 LNPMT01 - LNPMT13
                                                    LPRDMT01 - LPRDMT13 LPRPMT01 - LPRPMT13 LPRYCD01- LPRYCD13
                                                    LSBCHG01 - LSBCHG13 MDFCD101 - MDFCD113 MDFCD201- MDFCD213
                                                    MTSCNT01 - MTSCNT13 MTSIND01 - MTSIND13 NCSTYM01- NCSTYM13
                                                    PLCRVC01 - PLCRVC13 PMTDCD01 - PMTDCD13 PMTDSW01- PMTDSW13
                                                    PRCGND01 - PRCGND13 PRFUPN01 - PRFUPN13 PRPYLW01- PRPYLW13
                                                    PRTPTG01 - PRTPTG13 PRVSTT01 - PRVSTT13 PRVTYP01- PRVTYP13

                                                    PRVZP01  - PRVZP13  PTYSRV01 - PTYSRV13 SRVCNT01- SRVCNT13
                                                    TAXNUM01 - TAXNUM13 TYPVCB01 - TYPVCB13 J
                  compress = binary);
```

```
set indd.pb_md_2006_1(
                keep = ANSHNT01 - ANSHNT13 ASTTCD01 - ASTTCD13          BETOS01 - BETOS13
                                           CNMT01   - CNMT13   CRRPCL01 - CRRPCL13   DCMNCD01- DCMNCD13
                                           DED_SW01 - DED_SW13 DEMOIND1 - DEMOIND5   DEMONUM1- DEMONUM5
                                           DEMOTXT1 - DEMOTXT5 DGNSIND1 - DGNSIND4   DGNS_CD1- DGNS_CD4
                                           DMPRC01  - DMPRC13 D MSTDT01 - DMSTDT13    EXPDT101- EXPDT113
                                           EXPDT201 - EXPDT213 HCFPCL01 - HCFPCL13   HCPSCD01- HCPSCD13
                                           LABAMT01 - LABAMT13 LBNPMT01 - LBNPMT13   LCLYCD01- LCLYCD13
                                           LDDMT01  - LDDMT13 LLMTMT01 - LLMTMT13    LLWCHG01- LLWCHG13
                                           LNDGNS01 - LNDGNS13 LNPMT01  - LNPMT13    LPRDMT01- LPRDMT13
                                           LPRPMT01 - LPRPMT13 LPRYCD01 - LPRYCD13   LSBCHG01- LSBCHG13
                                           MDFCD101 - MDFCD113 MDFCD201 - MDFCD213   MTSCNT01- MTSCNT13
                                           MTSIND01 - MTSIND13 NCSTYM01 - NCSTYM13   PLCRVC01- PLCRVC13
                                           PMTDCD01 - PMTDCD13 PMTDSW01 - PMTDSW13   PRCGND01- PRCGND13
                                           PRFUPN01 - PRFUPN13 PRPYLW01 - PRPYLW13   PRTPTG01- PRTPTG13
                                           PRVSTT01 - PRVSTT13 PRVTYP01 - PRVTYP13   PRVZP01 - PRVZP13
                                           PTYSRV01 - PTYSRV13 SRVCNT01 - SRVCNT13   TAXNUM01- TAXNUM13
                                           TYPVCB01 - TYPVCB13 ALOWCHRG   ASGMNTCD   BENEPAID  BENE_DOB
                                           BENE_PMT   BENE_ZIP BIC        BID_6434_1 CAN       CARRCNTL
                                           CARR_NUM   CDEMCNT  CDGNCNT    CEDCNT     CLINECNT  CMCOCNT
                                           CNTY_CD    CPATCNT  CPO_PROV   DEDAPPLY   EQ_BIC    FROM_DT
                                           HCPCS_YR   MS_CD    PDGNS_CD   PMTDNLCD   PMT_AMT   PROV_PMT
                                           PRPAYAMT   RACE     RCPT_DT    RFR_NPI    RFR_UPIN  SBMTCHRG
                                           SCHLD_DT   SEX      STATE_CD   THRU_DT    WKLY_DT)

        indd.pb_md_2006_2(keep = ANSHNT01 - ANSHNT13  ASTTCD01 - ASTTCD13    BETOS01 - BETOS13
                                 CNMT01   - CNMT13    CRRPCL01 - CRRPCL13   DCMNCD01- DCMNCD13
                                 DED_SW01 - DED_SW13  DEMOIND1 - DEMOIND5   DEMONUM1- DEMONUM5
                                 DEMOTXT1 - DEMOTXT5  DGNSIND1 - DGNSIND4   DGNS_CD1- DGNS_CD4
                                 DMPRC01  - DMPRC13   DMSTDT01 - DMSTDT13   EXPDT101- EXPDT113
                                 EXPDT201 - EXPDT213  HCFPCL01 - HCFPCL13   HCPSCD01- HCPSCD13
                                 LABAMT01 - LABAMT13  LBNPMT01 - LBNPMT13   LCLYCD01- LCLYCD13
                                 LDDMT01  - LDDMT13   LLMTMT01 - LLMTMT13   LLWCHG01- LLWCHG13
                                 LNDGNS01 - LNDGNS13  LNPMT01  - LNPMT13    LPRDMT01- LPRDMT13
                                 LPRPMT01 - LPRPMT13  LPRYCD01 - LPRYCD13   LSBCHG01- LSBCHG13
                                 MDFCD101 - MDFCD113  MDFCD201 - MDFCD213   MTSCNT01- MTSCNT13
                                 MTSIND01 - MTSIND13  NCSTYM01 - NCSTYM13   PLCRVC01- PLCRVC13
                                 PMTDCD01 - PMTDCD13  PMTDSW01 - PMTDSW13   PRCGND01- PRCGND13
                                 PRFUPN01 - PRFUPN13  PRPYLW01 - PRPYLW13   PRTPTG01- PRTPTG13
                                 PRVSTT01 - PRVSTT13  PRVTYP01 - PRVTYP13   PRVZP01 - PRVZP13
                                 PTYSRV01 - PTYSRV13  SRVCNT01 - SRVCNT13   TAXNUM01- TAXNUM13
                                 TYPVCB01 - TYPVCB13  ALOWCHRG   ASGMNTCD   BENEPAID  BENE_DOB
                                 BENE_PMT   BENE_ZIP  BIC        BID_6434_1 CAN       CARRCNTL
                                 CARR_NUM   CDEMCNT   CDGNCNT    CEDCNT     CLINECNT  CMCOCNT
                                 CNTY_CD    CPATCNT   CPO_PROV   DEDAPPLY   EQ_BIC    FROM_DT
                                 HCPCS_YR   MS_CD     PDGNS_CD   PMTDNLCD   PMT_AMT   PROV_PMT
                                 PRPAYAMT   RACE      RCPT_DT    RFR_NPI    RFR_UPIN  SBMTCHRG
                                 SCHLD_DT   SEX       STATE_CD   THRU_DT    WKLY_DT  )

        indd.pb_md_2006_3(keep = ANSHNT01 - ANSHNT13  ASTTCD01 - ASTTCD13   BETOS01 - BETOS13
                                 CNMT01   - CNMT13    CRRPCL01 - CRRPCL13   DCMNCD01- DCMNCD13
                                 DED_SW01 - DED_SW13  DEMOIND1 - DEMOIND5   DEMONUM1- DEMONUM5
                                 DEMOTXT1 - DEMOTXT5  DGNSIND1 - DGNSIND4   DGNS_CD1- DGNS_CD4
                                 DMPRC01  - DMPRC13   DMSTDT01 - DMSTDT13   EXPDT101- EXPDT113
                                 EXPDT201 - EXPDT213  HCFPCL01 - HCFPCL13   HCPSCD01- HCPSCD13
                                 LABAMT01 - LABAMT13  LBNPMT01 - LBNPMT13   LCLYCD01- LCLYCD13
                                 LDDMT01  - LDDMT13   LLMTMT01 - LLMTMT13   LLWCHG01- LLWCHG13
                                 LNDGNS01 - LNDGNS13  LNPMT01  - LNPMT13    LPRDMT01- LPRDMT13
                                 LPRPMT01 - LPRPMT13  LPRYCD01 - LPRYCD13   LSBCHG01- LSBCHG13
                                 MDFCD101 - MDFCD113  MDFCD201 - MDFCD213   MTSCNT01- MTSCNT13
                                 MTSIND01 - MTSIND13  NCSTYM01 - NCSTYM13   PLCRVC01- PLCRVC13
                                 PMTDCD01 - PMTDCD13  PMTDSW01 - PMTDSW13   PRCGND01- PRCGND13
                                 PRFUPN01 - PRFUPN13  PRPYLW01 - PRPYLW13   PRTPTG01- PRTPTG13
                                 PRVSTT01 - PRVSTT13  PRVTYP01 - PRVTYP13   PRVZP01 - PRVZP13
                                 PTYSRV01 - PTYSRV13  SRVCNT01 - SRVCNT13   TAXNUM01- TAXNUM13
                                 TYPVCB01 - TYPVCB13  ALOWCHRG   ASGMNTCD   BENEPAID  BENE_DOB
                                 BENE_PMT   BENE_ZIP  BIC        BID_6434_1 CAN       CARRCNTL
                                 CARR_NUM   CDEMCNT   CDGNCNT    CEDCNT     CLINECNT  CMCOCNT
                                 CNTY_CD    CPATCNT   CPO_PROV   DEDAPPLY   EQ_BIC    FROM_DT
                                 HCPCS_YR   MS_CD     GNS_CD     PMTDNLCD   PMT_AMT   PROV_PMT
                                 PRPAYAMT   RACE      RCPT_DT    RFR_NPI    RFR_UPIN  SBMTCHRG
                                 SCHLD_DT   SEX       STATE_CD   THRU_DT    WKLY_DT  )

        indd.pb_md_2006_4(keep = ANSHNT01 - ANSHNT13  ASTTCD01 - ASTTCD13    BETOS01 - BETOS13
                                 CNMT01   - CNMT13    CRRPCL01 - CRRPCL13   DCMNCD01- DCMNCD13
                                 DED_SW01 - DED_SW13  DEMOIND1 - DEMOIND5   DEMONUM1- DEMONUM5
                                 DEMOTXT1 - DEMOTXT5  DGNSIND1 - DGNSIND4   DGNS_CD1- DGNS_CD4
                                 DMPRC01  - DMPRC13   DMSTDT01 - DMSTDT13   EXPDT101- EXPDT113
                                 EXPDT201 - EXPDT213  HCFPCL01 - HCFPCL13   HCPSCD01- HCPSCD13
                               LABAMT01 - LABAMT13  LBNPMT01 - LBNPMT13   LCLYCD01- LCLYCD13
                               LDDMT01  - LDDMT13  LLMTMT01 - LLMTMT13     LLWCHG01- LLWCHG13
                               LNDGNS01 - LNDGNS13  LNPMT01  - LNPMT13    LPRDMT01- LPRDMT13
                               LPRPMT01 - LPRPMT13  LPRYCD01 - LPRYCD13   LSBCHG01- LSBCHG13
                               MDFCD101 - MDFCD113  MDFCD201 - MDFCD213   MTSCNT01- MTSCNT13
                               MTSIND01 - MTSIND13  NCSTYM01 - NCSTYM13   PLCRVC01- PLCRVC13
                               PMTDCD01 - PMTDCD13  PMTDSW01 - PMTDSW13   PRCGND01- PRCGND13
                               PRFUPN01 - PRFUPN13  PRPYLW01 - PRPYLW13   PRTPTG01- PRTPTG13
                               PRVSTT01 - PRVSTT13  PRVTYP01 - PRVTYP13   PRVZP01 - PRVZP13
                               PTYSRV01 - PTYSRV13  SRVCNT01 - SRVCNT13   TAXNUM01- TAXNUM13
                               TYPVCB01 - TYPVCB13  ALOWCHRG   ASGMNTCD   BENEPAID  BENE_DOB
                               BENE_PMT   BENE_ZIP  BIC        BID_6434_1 CAN       CARRCNTL
                               CARR_NUM   CDEMCNT   CDGNCNT    CEDCNT     CLINECNT  CMCOCNT
                               CNTY_CD    CPATCNT  CPO_PROV     DEDAPPLY  EQ_BIC     FROM_DT
                               HCPCS_YR   MS_CD     PDGNS_CD   PMTDNLCD   PMT_AMT   PROV_PMT
                               PRPAYAMT   RACE      RCPT_DT    RFR_NPI    RFR_UPIN  SBMTCHRG
                               SCHLD_DT   SEX       STATE_CD   THRU_DT    WKLY_DT  )
```

```
indd.pb_md_2006_5(keep = ANSHNT01 - ANSHNT13    ASTTCD01 - ASTTCD13 BETOS01 - BETOS13
                         CNMT01   - CNMT13     CRRPCL01 - CRRPCL13   DCMNCD01- DCMNCD13
                         DED_SW01 - DED_SW13   DEMOIND1 - DEMOIND5   DEMONUM1- DEMONUM5
                         DEMOTXT1 - DEMOTXT5   DGNSIND1 - DGNSIND4   DGNS_CD1- DGNS_CD4
                         DMPRC01  - DMPRC13    DMSTDT01 - DMSTDT13   EXPDT101- EXPDT113
                         EXPDT201 - EXPDT213   HCFPCL01 - HCFPCL13   HCPSCD01- HCPSCD13
                         LABAMT01 - LABAMT13   LBNPMT01 - LBNPMT13   LCLYCD01- LCLYCD13
                         LDDMT01  - LDDMT13    LLMTMT01 - LLMTMT13   LLWCHG01- LLWCHG13
                         LNDGNS01 - LNDGNS13   LNPMT01  - LNPMT13    LPRDMT01- LPRDMT13
                         LPRPMT01 - LPRPMT13   LPRYCD01 - LPRYCD13   LSBCHG01- LSBCHG13
                         MDFCD101 - MDFCD113   MDFCD201 - MDFCD213   MTSCNT01- MTSCNT13
                         MTSIND01 - MTSIND13   NCSTYM01 - NCSTYM13   PLCRVC01- PLCRVC13
                         PMTDCD01 - PMTDCD13   PMTDSW01 - PMTDSW13   PRCGND01- PRCGND13
                         PRFUPN01 - PRFUPN13   PRPYLW01 - PRPYLW13   PRTPTG01- PRTPTG13
                         PRVSTT01 - PRVSTT13   PRVTYP01 - PRVTYP13   PRVZP01 - PRVZP13
                         PTYSRV01 - PTYSRV13   SRVCNT01 - SRVCNT13   TAXNUM01- TAXNUM13
                         TYPVCB01 - TYPVCB13   ALOWCHRG   ASGMNTCD   BENEPAID   BENE_DOB
                         BENE_PMT   BENE_ZIP   BIC        BID_6434_1 CAN        CARRCNTL
                         CARR_NUM   CDEMCNT    CDGNCNT    CEDCNT     CLINECNT   CMCOCNT
                         CNTY_CD    CPATCNT    PO_PROV    DEDAPPLY   EQ_BIC     FROM_DT
                         HCPCS_YR   MS_CDP     DGNS_CD    PMTDNLCD   PMT_AMT    PROV_PMT
                         PRPAYAMT   RACE       RCPT_DT    RFR_NPI    RFR_UPIN   SBMTCHRG
                         SCHLD_DT   SEX        STATE_CD   THRU_DT    WKLY_DT )

indd.pb_md_2006_6(keep = ANSHNT01 - ANSHNT13    ASTTCD01 - ASTTCD13   BETOS01 - BETOS13
                         CNMT01   - CNMT13     CRRPCL01 - CRRPCL13   DCMNCD01- DCMNCD13
                         DED_SW01 - DED_SW13   DEMOIND1 - DEMOIND5   DEMONUM1- DEMONUM5
                         DEMOTXT1 - DEMOTXT5   DGNSIND1 - DGNSIND4   DGNS_CD1- DGNS_CD4
                         DMPRC01  - DMPRC13    DMSTDT01 - DMSTDT13   EXPDT101- EXPDT113
                         EXPDT201 - EXPDT213   HCFPCL01 - HCFPCL13   HCPSCD01- HCPSCD13
                         LABAMT01 - LABAMT13   LBNPMT01 - LBNPMT13   LCLYCD01- LCLYCD13
                         LDDMT01  - LDDMT13    LLMTMT01 - LLMTMT13   LLWCHG01- LLWCHG13
                         LNDGNS01 - LNDGNS13   LNPMT01  - LNPMT13    LPRDMT01- LPRDMT13
                         LPRPMT01 - LPRPMT13   LPRYCD01 - LPRYCD13   LSBCHG01- LSBCHG13
                         MDFCD101 - MDFCD113   MDFCD201 - MDFCD213   MTSCNT01- MTSCNT13
                         MTSIND01 - MTSIND13   NCSTYM01 - NCSTYM13   PLCRVC01- PLCRVC13
                         PMTDCD01 - PMTDCD13   PMTDSW01 - PMTDSW13   PRCGND01- PRCGND13
                         PRFUPN01 - PRFUPN13   PRPYLW01 - PRPYLW13   PRTPTG01- PRTPTG13
                         PRVSTT01 - PRVSTT13   PRVTYP01 - PRVTYP13   PRVZP01 - PRVZP13
                         PTYSRV01 - PTYSRV13   SRVCNT01 - SRVCNT13   TAXNUM01- TAXNUM13
                         TYPVCB01 - TYPVCB13   ALOWCHRG   ASGMNTCD   BENEPAID   BENE_DOB
                         BENE_PMT   BENE_ZIP   BIC        BID_6434_1 CAN        CARRCNTL
                         CARR_NUM   CDEMCNT    CDGNCNT    CEDCNT     CLINECNT   CMCOCNT
                         CNTY_CD    CPATCNT    CPO_PROV   DEDAPPLY   EQ_BIC     FROM_DT
                         HCPCS_YR   MS_CD      PDGNS_CD   PMTDNLCD   PMT_AMT    PROV_PMT
                         PRPAYAMT   RACE       RCPT_DT    RFR_NPI    RFR_UPIN   SBMTCHRG
                         SCHLD_DT   SEX        STATE_CD   THRU_DT    WKLY_DT  )

indd.pb_md_2006_7(keep = ANSHNT01 - ANSHNT13  ASTTCD01 - ASTTCD13 BETOS01 - BETOS13
                       CNMT01   - CNMT13    CRRPCL01 - CRRPCL13    DCMNCD01- DCMNCD13
                       DED_SW01 - DED_SW13  DEMOIND1 - DEMOIND5    DEMONUM1- DEMONUM5
                       DEMOTXT1 - DEMOTXT5  DGNSIND1 - DGNSIND4    DGNS_CD1- DGNS_CD4
                       DMPRC01  - DMPRC13    DMSTDT01 - DMSTDT13    EXPDT101- EXPDT113
                       EXPDT201 - EXPDT213  HCFPCL01 - HCFPCL13    HCPSCD01- HCPSCD13
                       LABAMT01 - LABAMT13  LBNPMT01 - LBNPMT13    LCLYCD01- LCLYCD13
                       LDDMT01  - LDDMT13    LLMTMT01 - LLMTMT13    LLWCHG01- LLWCHG13
                       LNDGNS01 - LNDGNS13  LNPMT01  - LNPMT13     LPRDMT01- LPRDMT13
                       LPRPMT01 - LPRPMT13  LPRYCD01 - LPRYCD13    LSBCHG01- LSBCHG13
                       MDFCD101 - MDFCD113  MDFCD201 - MDFCD213    MTSCNT01- MTSCNT13
                       MTSIND01 - MTSIND13  NCSTYM01 - NCSTYM13    PLCRVC01- PLCRVC13
                       PMTDCD01 - PMTDCD13  PMTDSW01 - PMTDSW13    PRCGND01- PRCGND13
                       PRFUPN01 - PRFUPN13  PRPYLW01 - PRPYLW13    PRTPTG01- PRTPTG13
                       PRVSTT01 - PRVSTT13  PRVTYP01 - PRVTYP13    PRVZP01 - PRVZP13
                       PTYSRV01 - PTYSRV13  SRVCNT01 - SRVCNT13    TAXNUM01- TAXNUM13
                       TYPVCB01 - TYPVCB13  ALOWCHRG  ASGMNTCD     BENEPAID  BENE_DOB
                       BENE_PMT   BENE_ZIP  BIC        BID_6434_1  CAN       CARRCNTL
                       CARR_NUM   CDEMCNT   CDGNCNT    CEDCNT      CLINECNT  CMCOCNT
                       CNTY_CD    CPATCNT   CPO_PROV   DEDAPPLY    EQ_BIC    FROM_DT
                       HCPCS_YR   MS_CD     PDGNS_CD   PMTDNLCD    PMT_AMT   PROV_PMT
                       PRPAYAMT   RACE      RCPT_DT   RFR_NPI      RFR_UPIN  SBMTCHRG
                     SCHLD_DT   SEX       STATE_CD  THRU_DT     WKLY_DT  )

indd.pb_md_2006_8(keep = ANSHNT01 - ANSHNT13 ASTTCD01 - ASTTCD13    BETOS01 - BETOS13
                       CNMT01   - CNMT13     CRRPCL01 - CRRPCL13   DCMNCD01- DCMNCD13
                       DED_SW01 - DED_SW13   DEMOIND1 - DEMOIND5   DEMONUM1- DEMONUM5
                       DEMOTXT1 - DEMOTXT5   DGNSIND1 - DGNSIND4   DGNS_CD1- DGNS_CD4
                       DMPRC01  - DMPRC13     DMSTDT01 - DMSTDT13   EXPDT101- EXPDT113
                       EXPDT201 - EXPDT213   HCFPCL01 - HCFPCL13   HCPSCD01- HCPSCD13
                       LABAMT01 - LABAMT13   LBNPMT01 - LBNPMT13   LCLYCD01- LCLYCD13
                       LDDMT01  - LDDMT13     LLMTMT01 - LLMTMT13   LLWCHG01- LLWCHG13
                       LNDGNS01 - LNDGNS13   LNPMT01  - LNPMT13    LPRDMT01- LPRDMT13
                       LPRPMT01 - LPRPMT13   LPRYCD01 - LPRYCD13   LSBCHG01- LSBCHG13
                       MDFCD101 - MDFCD113   MDFCD201 - MDFCD213   MTSCNT01- MTSCNT13
                       MTSIND01 - MTSIND13   NCSTYM01 - NCSTYM13   PLCRVC01- PLCRVC13
                       PMTDCD01 - PMTDCD13   PMTDSW01 - PMTDSW13   PRCGND01- PRCGND13
                       PRFUPN01 - PRFUPN13   PRPYLW01 - PRPYLW13   PRTPTG01- PRTPTG13
                       PRVSTT01 - PRVSTT13   PRVTYP01 - PRVTYP13   PRVZP01 - PRVZP13
                       PTYSRV01 - PTYSRV13   SRVCNT01 - SRVCNT13   TAXNUM01- TAXNUM13
                       TYPVCB01 - TYPVCB13   ALOWCHRG   ASGMNTCD   BENEPAID  BENE_DOB
                       BENE_PMT   BENE_ZIP   BIC        BID_6434_1 CAN        CARRCNTL
                       CARR_NUM   CDEMCNT    CDGNCNT    CEDCNT     CLINECNT   MCOCNT
                       CNTY_CD    CPATCNT     CPO_PROV   DEDAPPLY   EQ_BIC     FROM_DT
                       HCPCS_YR   MS_CD      PDGNS_CD   PMTDNLCD   PMT_AMT     PROV_PMT
                       PRPAYAMT   RACE       RCPT_DT    RFR_NPI    RFR_UPIN   SBMTCHRG
                       SCHLD_DT   SEX        STATE_CD   THRU_DT     WKLY_DT  )
```

```
indd.pb_md_2006_9(keep = ANSHNT01 - ANSHNT13  ASTTCD01 - ASTTCD13 BETOS01 - BETOS13
                         CNMT01   - CNMT13    CRRPCL01 - CRRPCL13   DCMNCD01- DCMNCD13
                         DED_SW01 - DED_SW13  DEMOIND1 - DEMOIND5   DEMONUM1- DEMONUM5
                         DEMOTXT1 - DEMOTXT5  DGNSIND1 - DGNSIND4   DGNS_CD1- DGNS_CD4
                         DMPRC01  - DMPRC13    DMSTDT01 - DMSTDT13   EXPDT101- EXPDT113
                         EXPDT201 - EXPDT213  HCFPCL01 - HCFPCL13   HCPSCD01- HCPSCD13
                         LABAMT01 - LABAMT13  LBNPMT01 - LBNPMT13   LCLYCD01- LCLYCD13
                         LDDMT01  - LDDMT13    LLMTMT01 - LLMTMT13   LLWCHG01- LLWCHG13
                         LNDGNS01 - LNDGNS13  LNPMT01  - LNPMT13    LPRDMT01- LPRDMT13
                         LPRPMT01 - LPRPMT13  LPRYCD01 - LPRYCD13   LSBCHG01- LSBCHG13
                         MDFCD101 - MDFCD113  MDFCD201 - MDFCD213   MTSCNT01- MTSCNT13
                         MTSIND01 - MTSIND13  NCSTYM01 - NCSTYM13   PLCRVC01- PLCRVC13
                         PMTDCD01 - PMTDCD13  PMTDSW01 - PMTDSW13   PRCGND01- PRCGND13
                         PRFUPN01 - PRFUPN13  PRPYLW01 - PRPYLW13   PRTPTG01- PRTPTG13
                         PRVSTT01 - PRVSTT13  PRVTYP01 - PRVTYP13   PRVZP01 - PRVZP13
                         PTYSRV01 - PTYSRV13  SRVCNT01 - SRVCNT13   TAXNUM01- TAXNUM13
                         TYPVCB01 - TYPVCB13  ALOWCHRG   ASGMNTCD   BENEPAID  BENE_DOB
                         BENE_PMT   BENE_ZIP  BIC        BID_6434_1 CAN       CARRCNTL
                         CARR_NUM   CDEMCNT   CDGNCNT    CEDCNT     CLINECNT  CMCOCNT
                         CNTY_CD    CPATCNT  CPO_PROV   DEDAPPLY   EQ_BIC    FROM_DT
                         HCPCS_YR   MS_CD     PDGNS_CD   PMTDNLCD   PMT_AMT   PROV_PMT
                         PRPAYAMT   RACE      RCPT_DT    RFR_NPI    RFR_UPIN  SBMTCHRG
                         SCHLD_DT   SEX       STATE_CD   THRU_DT    WKLY_DT  );

ARRAY TEMP_ANSHNT(13)    ANSHNT01 - ANSHNT13;
ARRAY TEMP_ASTTCD(13)    ASTTCD01 - ASTTCD13;
ARRAY TEMP_BETOS(13)             BETOS01  - BETOS13;
ARRAY TEMP_CNMT(13)              CNMT01    - CNMT13;
ARRAY TEMP_CRRPCL(13)    CRRPCL01 - CRRPCL13;
ARRAY TEMP_DCMNCD(13)    DCMNCD01 - DCMNCD13;
ARRAY TEMP_DED_SW(13)    DED_SW01 - DED_SW13;
ARRAY TEMP_DMPRC(13)     DMPRC01  - DMPRC13;
ARRAY TEMP_DMSTDT(13)    DMSTDT01 - DMSTDT13;
ARRAY TEMP_EXPDT1(13)    EXPDT101 - EXPDT113;
ARRAY TEMP_EXPDT2(13)    EXPDT201 - EXPDT213;
ARRAY TEMP_HCFPCL(13)    HCFPCL01 - HCFPCL13;
ARRAY TEMP_HCPSCD(13)    HCPSCD01 - HCPSCD13;
ARRAY TEMP_LABAMT(13)    LABAMT01 - LABAMT13;
ARRAY TEMP_LBNPMT(13)    LBNPMT01 - LBNPMT13;
ARRAY TEMP_LCLYCD(13)    LCLYCD01 - LCLYCD13;
ARRAY TEMP_LDDMT(13)     LDDMT01  - LDDMT13;
ARRAY TEMP_LLMTMT(13)    LLMTMT01 - LLMTMT13;
ARRAY TEMP_LLWCHG(13)    LLWCHG01 - LLWCHG13;
ARRAY TEMP_LNDGNS(13)    LNDGNS01 - LNDGNS13;
ARRAY TEMP_LNPMT(13)     LNPMT01  - LNPMT13;
ARRAY TEMP_LPRDMT(13)    LPRDMT01 - LPRDMT13;
ARRAY TEMP_LPRPMT(13)    LPRPMT01 - LPRPMT13;
ARRAY TEMP_LPRYCD(13)    LPRYCD01 - LPRYCD13;
ARRAY TEMP_LSBCHG(13)    LSBCHG01 - LSBCHG13;
ARRAY TEMP_MDFCD1(13)    MDFCD101 - MDFCD113;
ARRAY TEMP_MDFCD2(13)    MDFCD201 - MDFCD213;
ARRAY TEMP_MTSCNT(13)  MTSCNT01 - MTSCNT13;
ARRAY TEMP_MTSIND(13)    MTSIND01 - MTSIND13;
ARRAY TEMP_NCSTYM(13)    NCSTYM01 - NCSTYM13;
ARRAY TEMP_PLCRVC(13)    PLCRVC01 - PLCRVC13;
ARRAY TEMP_PMTDCD(13)    PMTDCD01 - PMTDCD13;
ARRAY TEMP_PMTDSW(13)    PMTDSW01 - PMTDSW13;
ARRAY TEMP_PRCGND(13)    PRCGND01 - PRCGND13;
ARRAY TEMP_PRFUPN(13)    PRFUPN01 - PRFUPN13;
ARRAY TEMP_PRPYLW(13)    PRPYLW01 - PRPYLW13;
ARRAY TEMP_PRTPTG(13)    PRTPTG01 - PRTPTG13;
ARRAY TEMP_PRVSTT(13)    PRVSTT01 - PRVSTT13;
ARRAY TEMP_PRVTYP(13)    PRVTYP01 - PRVTYP13;
ARRAY TEMP_PRVZP(13)       PRVZP01  - PRVZP13;
ARRAY TEMP_PTYSRV(13)    PTYSRV01 - PTYSRV13;
ARRAY TEMP_SRVCNT(13)    SRVCNT01 - SRVCNT13;
ARRAY TEMP_TAXNUM(13)    TAXNUM01 - TAXNUM13;
ARRAY TEMP_TYPVCB(13)    TYPVCB01 - TYPVCB13;

DO J = 1 TO CLINECNT;
                         ANSHNT = TEMP_ANSHNT(J);
      ASTTCD = TEMP_ASTTCD(J);
      BETOS  = TEMP_BETOS(J);
      CNMT       = TEMP_CNMT(J);
      CRRPCL = TEMP_CRRPCL(J);
      DCMNCD = TEMP_DCMNCD(J);
      DED_SW = TEMP_DED_SW(J);
      DMPRC  = TEMP_DMPRC(J);
      DMSTDT = TEMP_DMSTDT(J);
      EXPDT1 = TEMP_EXPDT1(J);
      EXPDT2 = TEMP_EXPDT2(J);
      HCFPCL = TEMP_HCFPCL(J);
      HCPSCD = TEMP_HCPSCD(J);
      LABAMT = TEMP_LABAMT(J);
      LBNPMT = TEMP_LBNPMT(J);
      LCLYCD = TEMP_LCLYCD(J);
      LDDMT  = TEMP_LDDMT(J);
      LLMTMT = TEMP_LLMTMT(J);
      LLWCHG = TEMP_LLWCHG(J);
      LNDGNS = TEMP_LNDGNS(J);
      LNPMT  = TEMP_LNPMT(J);
      LPRDMT = TEMP_LPRDMT(J);
      LPRPMT = TEMP_LPRPMT(J);
      LPRYCD = TEMP_LPRYCD(J);
      LSBCHG = TEMP_LSBCHG(J);
      MDFCD1 = TEMP_MDFCD1(J);
      MDFCD2 = TEMP_MDFCD2(J);
```

```
      MTSCNT = TEMP_MTSCNT(J);
      MTSIND = TEMP_MTSIND(J);
      NCSTYM = TEMP_NCSTYM(J);
      PLCRVC = TEMP_PLCRVC(J);
      PMTDCD = TEMP_PMTDCD(J);
      PMTDSW = TEMP_PMTDSW(J);
      PRCGND = TEMP_PRCGND(J);
      PRFUPN = TEMP_PRFUPN(J);
      PRPYLW = TEMP_PRPYLW(J);
      PRTPTG = TEMP_PRTPTG(J);
      PRVSTT = TEMP_PRVSTT(J);
      PRVTYP = TEMP_PRVTYP(J);
      PRVZP  = TEMP_PRVZP(J);
      PTYSRV = TEMP_PTYSRV(J);
      SRVCNT = TEMP_SRVCNT(J);
      TAXNUM = TEMP_TAXNUM(J);
      TYPVCB = TEMP_TYPVCB(J);
   OUTPUT;
END;

LABEL
     CAN      = 'BENE_CLM_ACNT_NUM'
     EQ_BIC   = 'NCH_CTGRY_EQTBL_BIC_CD'
     BIC      = 'BENE_IDENT_CD'
     STATE_CD = 'BENE_RSDNC_SSA_STD_STATE_CD'
     CNTY_CD  = 'BENE_RSDNC_SSA_STD_CNTY_CD'
     FROM_DT  = 'CLM_FROM_DT'
     THRU_DT  = 'CLM_THRU_DT'
     RCPT_DT  = 'CARR_CLM_RCPT_DT'
     SCHLD_DT = 'CARR_CLM_SCHLD_PMT_DT'
     WKLY_DT  = 'NCH_WKLY_PROC_DT'
     CARR_NUM = 'CARR_NUM'
     BENE_ZIP = 'BENE_MLG_CNTCT_ZIP_CD'
     SEX      = 'BENE_SEX_IDENT_CD'
     RACE     = 'BENE_RACE_CD'
     BENE_DOB = 'BENE_BIRTH_DT'
     MS_CD    = 'CWF_BENE_MDCR_STUS_CD'
     PDGNS_CD = 'CLM_PRNCPAL_DGNS_CD'
     PMT_AMT  = 'CLM_PMT_AMT'
     PRPAYAMT = 'CARR_CLM_PRMRY_PYR_PD_AMT'
     CARRCNTL = 'CARR_CLM_CNTL_NUM'
     PMTDNLCD = 'CARR_CLM_PMT_DNL_CD'
     RFR_UPIN = 'CARR_CLM_RFRG_UPIN_NUM'
     RFR_NPI  = 'CARR_CLM_RFRG_PHYSN_NPI_NUM'
     ASGMNTCD = 'CARR_CLM_PRVDR_ASGNMT_IND_SW'
     PROV_PMT = 'NCH_CLM_PRVDR_PMT_AMT'
     BENE_PMT = 'NCH_CLM_BENE_PMT_AMT'
     BENEPAID = 'CARR_CLM_BENE_PD_AMT'
     SBMTCHRG = 'NCH_CARR_SBMT_CHRG_AMT'
     ALOWCHRG = 'NCH_CARR_ALOW_CHRG_AMT'
     DEDAPPLY = 'CARR_CLM_CASH_DDCTBL_APPLY_AMT'
     HCPCS_YR = 'CARR_CLM_HCPCS_YR_CD'
     CPO_PROV = 'CPO_PRVDR_NUM'
     CEDCNT   = 'CARR_NCH_EDIT_CD_CNT'
     CPATCNT  = 'CARR_NCH_PATCH_CD_CNT'
     CMCOCNT  = 'CARR_MCO_PRD_CNT'
     CDEMCNT  = 'CARR_CLM_DEMO_ID_CNT'
     CDGNCNT  = 'CARR_CLM_DGNS_CD_CNT'
     CLINECNT = 'CARR_CLM_LINE_CNT'

     DEMOIND1 = 'NCH_DEMO_TRLR_IND_CD(1)'
     DEMOIND2 = 'NCH_DEMO_TRLR_IND_CD(2)'
     DEMOIND3 = 'NCH_DEMO_TRLR_IND_CD(3)'
     DEMOIND4 = 'NCH_DEMO_TRLR_IND_CD(4)'
     DEMOIND5 = 'NCH_DEMO_TRLR_IND_CD(5)'
     DEMONUM1 = 'CLM_DEMO_ID_NUM(1)'
     DEMONUM2 = 'CLM_DEMO_ID_NUM(2)'
     DEMONUM3 = 'CLM_DEMO_ID_NUM(3)'
     DEMONUM4 = 'CLM_DEMO_ID_NUM(4)'
     DEMONUM5 = 'CLM_DEMO_ID_NUM(5)'
     DEMOTXT1 = 'CLM_DEMO_INFO_TXT(1)'
     DEMOTXT2 = 'CLM_DEMO_INFO_TXT(2)'
     DEMOTXT3 = 'CLM_DEMO_INFO_TXT(3)'
     DEMOTXT4 = 'CLM_DEMO_INFO_TXT(4)'
     DEMOTXT5 = 'CLM_DEMO_INFO_TXT(5)'

     DGNSIND1 = 'NCH_DGNS_TRLR_IND_CD(1)'
     DGNSIND2 = 'NCH_DGNS_TRLR_IND_CD(2)'
     DGNSIND3 = 'NCH_DGNS_TRLR_IND_CD(3)'
     DGNSIND4 = 'NCH_DGNS_TRLR_IND_CD(4)'
     DGNS_CD1 = 'CLM_DGNS_CD(1)'
     DGNS_CD2 = 'CLM_DGNS_CD(2)'
     DGNS_CD3 = 'CLM_DGNS_CD(3)'
     DGNS_CD4 = 'CLM_DGNS_CD(4)'

     PRFUPN   = 'CARR_LINE_PRFRMG_UPIN_NUM'
     PRVTYP   = 'CARR_LINE_PRVDR_TYPE_CD'
     TAXNUM   = 'LINE_PRVDR_TAX_NUM'
     PRVSTT   = 'LINE_NCH_PRVDR_STATE_CD'
     PRVZP    = 'CARR_LINE_PRFRMG_PRVDR_ZIP_CD'
     HCFPCL   = 'LINE_HCFA_PRVDR_SPCLTY_CD'
     CRRPCL   = 'CARR_LINE_PRVDR_SPCLTY_CD'
     PRTPTG   = 'LINE_PRVDR_PRTCPTG_IND_CD'
     ASTTCD   = 'CARR_LINE_RDCD_PHYSN_ASTNT_CD'
     SRVCNT   = 'LINE_SRVC_CNT'
     TYPVCB   = 'LINE_HCFA_TYPE_SRVC_CD'
     PTYSRV   = 'CARR_LINE_TYPE_SRVC_CD'
```

```
        PLCRVC   = 'LINE_PLC_SRVC_CD'
        LCLYCD   = 'CARR_LINE_PRCNG_LCLTY_CD'
        EXPDT1   = 'LINE_1ST_EXPNS_DT'
        EXPDT2   = 'LINE_LAST_EXPNS_DT'
        HCPSCD   = 'LINE_HCPCS_CD'
        MDFCD1   = 'LINE_HCPCS_INITL_MDFR_CD'
        MDFCD2   = 'LINE_HCPCS_2ND_MDFR_CD'
        BETOS    = 'LINE_NCH_BETOS_CD'
        LNPMT    = 'LINE_NCH_PMT_AMT'
        LBNPMT   = 'LINE_BENE_PMT_AMT'
        LPRPMT   = 'LINE_PRVDR_PMT_AMT'
        LDDMT    = 'LINE_BENE_PTB_DDCTBL_AMT'
        LPRYCD   = 'LINE_BENE_PRMRY_PYR_CD'
        LPRDMT   = 'LINE_BENE_PRMRY_PYR_PD_AMT'
        CNMT     = 'LINE_COINSRNC_AMT'
        LLMTMT   = 'CARR_LINE_PSYCH_OT_PT_LMT_AMT'
        PRPYLW   = 'LINE_PRMRY_PYR_ALOW_CHRG_AMT'
        LSBCHG   = 'LINE_SBMT_CHRG_AMT'
        LLWCHG   = 'LINE_ALOW_CHRG_AMT'
        LABAMT   = 'CARR_LINE_CLNCL_LAB_CHRG_AMT'
        PRCGND   = 'LINE_PRCSG_IND_CD'
        PMTDSW   = 'LINE_PMT_80_100_CD'
        DED_SW   = 'LINE_SRVC_DDCTBL_IND_SW'
        PMTDCD   = 'LINE_PMT_IND_CD'
        MTSCNT   = 'CARR_LINE_MTUS_CNT'
        MTSIND   = 'CARR_LINE_MTUS_IND_CD'
        LNDGNS   = 'LINE_DGNS_CD'
        ANSHNT   = 'CARR_LINE_ANSTHSA_UNIT_CNT'
        DCMNCD   = 'LINE_ADDTNL_CLM_DCMTN_IND_CD'
        DMSTDT   = 'CARR_LINE_DME_CVRG_PRD_STRT_DT'
        DMPRC    = 'LINE_DME_PURC_PRICE_AMT'
        NCSTYM   = 'CARR_LINE_DME_NCSTY_MO_CNT'
        ;

   FORMAT
        DMSTDT               9.
        FROM_DT              9.
        THRU_DT              9.
        RCPT_DT              9.
        SCHLD_DT             9.
        WKLY_DT              9.
        BENE_DOB             9.
        PMT_AMT              COMMA14.2
        PRPAYAMT             COMMA14.2

        PROV_PMT             COMMA14.2
        BENE_PMT             COMMA14.2
        BENEPAID             COMMA14.2
        SBMTCHRG             COMMA14.2
        ALOWCHRG             COMMA14.2
        DEDAPPLY             COMMA14.2
        EXPDT1               9.
        EXPDT2               9.
        LNPMT                COMMA14.2
        LBNPMT               COMMA14.2
        LPRPMT               COMMA14.2
        LDDMT                COMMA14.2
        LPRDMT               COMMA14.2
        CNMT                 COMMA14.2
        LLMTMT               COMMA14.2
        PRPYLW               COMMA14.2
        LSBCHG               COMMA14.2
        LLWCHG               COMMA14.2
        LABAMT               COMMA14.2
        DMSTDT               9.
        DMPRC                COMMA14.2
        ;

run;

proc contents data=outdd.psbene&YYYY;
title2 "&yyyy Physician/Supplier Data: From Beneficiary Version";
run;

/* Step 2 */

data outdd.psbene&YYYY.x (label="Reformatted HCFA &yyyy 5% Phys/Supp File"
               keep=patid
               begday begmo begyr
               birthdd birthmo birthyr claimpdd claimpmo
               claimpyr diag1 diag2 diag3 diag4 proc endday
               endmo endyr fedtaxid modf1 modf2 ndiag noline
               patzip provid srvlzip nunits srvunit
               bchrge alloamt reimamt patliab
               typvcb srvtype hcfpcl specelty1 srvplac ms_cd state_cd payment lprdmt
               _modtcsg _aged _med2nd
               sex cnty_cd prtptg
               );


label begday   = 'CLM_FRM_DT: Day'
      begmo    = 'CLM_FRM_DT: Month'
      begyr    = 'CLM_FRM_DT: Year'
      birthdd  = 'BENE_BIRTH_DT: Day'
      birthmo  = 'BENE_BIRTH_DT: Month'
      birthyr  = 'BENE_BIRTH_YR: Year'
      claimpdd = 'CARR_CLM_SCHLD_PMT_DT: Day'
```

```
      claimpmo = 'CARR_CLM_SCHLD_PMT_DT: Month'
      claimpyr = 'CARR_CLM_SCHLD_PMT_DT: Year'
      diag1    = 'CLM_DGNS_CD (1)'
      diag2    = 'CLM_DGNS_CD (2)'
      diag3    = 'CLM_DGNS_CD (3)'
      diag4    = 'CLM_DGNS_CD (4)'
      endday   = 'CLM_THRU_DT: Day'
      endmo    = 'CLM_THRU_DT: Month'
      endyr    = 'CLM_THRU_DT: Year'
      fedtaxid = 'LINE_PRVDR_TAX_NUM'
      modf1    = 'LINE_HCPCS_INITL_MDFR_CD'
      modf2    = 'LINE_HCPCS_2ND_MDFR_CD'
      ndiag    = 'CARR_CLM_DGNS_CD_CNT'
      bchrge   = 'LINE_SBMT_CHRG_AMT'
      alloamt  = 'LINE_ALOW_CHRG_AMT'
      reimamt  = 'LINE_NCH_PMT_AMT'
      patliab  = 'LINE_PTB_DED_AMT + LINE_COINSRNC_AMT'
      srvtype  = 'Remapped LINE_HCFA_TYPE_SRVC_CD'
      specltyl = 'Remapped LINE_HCFA_PRVDR_SPCLTY_CD'
      patzip   = 'BENE_MLG_CNTCT_ZIP'
      noline   = 'CARR_CLM_LINE_CNT'
      srvplac  = 'LINE_PLC_SRVC_CD'
      sex      = 'BENE_SEX_IDENT_CD'
      srvlzip  = 'CARR_LINE_PRFRMG_PRVDR_ZIP_CD'
      provid   = 'CARR_LINE_PRFRMG_UPIN_NUM'
      patid    = 'BENEFICIARY IDENTIFICATION CODE'
      proc     = 'LINE_HCPCS_CD'
      nunits   = 'CARR_LINE_MTUS_CNT'
      srvunit  = 'CARR_LINE_MTUS_IND_CD'
      ms_cd    = 'BENE_MDCR_STUS_CD'
      state_cd = 'BENE_RSDNC_SSA_STD_STATE_CD'
      _modtcsg = 'FLAG: 1 if MODF1 or MODF2=TC or SG'
      _aged    = "FLAG: 1 if bene is >=65 as of 1/1/&yyyy"
      _med2nd  = 'Medicare is Secondary Payer: 1=yes 0=no'
      lprdmt   = 'HCFA: LINE_BENE_PRMRY_PYR_PD_AMT'
      payment  = 'sum(REIMAMT,PATLIAB)'
      prtptg   = 'Line Provider Participating Indicator'
;

set outdd.psbene&YYYY (rename=(
   clinecnt=noline
   dgns_cd1=diag1
   dgns_cd2=diag2
   dgns_cd3=diag3
   dgns_cd4=diag4
   cdgncnt =ndiag
   mdfcd1  =modf1
   mdfcd2  =modf2
   mtscnt  =nunits
   mtsind  =srvunit
   taxnum  =fedtaxid
   plcrvc  =srvplac
   lnpmt   =reimamt
   lsbchg  =bchrge
   llwchg  =alloamt
   prfupn  =provid
   hcpscd  =proc
   )
   ) end=endfile;


retain _notmd _dental _not&yy
       _convah _convaj _convak _convan _convav _convqx 0;

_cfromdt = put(from_dt,8.);

begday  = substr(_cfromdt,7,2);
begmo   = substr(_cfromdt,5,2);
begyr   = substr(_cfromdt,1,4);

/* Screen per meeting with Sophie on 12/17/1999 - IF any of the following
   are true, update counters and bypass the record. */

if (state_cd ne '21') or (put(proc,$dentexc.) ne '0') or (begyr ne &yyyy) then do;

if (state_cd ne '21') then _notmd = _notmd + 1; /* not a Maryland resident */

if (put(proc,$dentexc.) ne '0') then _dental = _dental + 1; /* Dental HCPCS */

if (begyr ne &yyyy) then _not&yy = _not&yy + 1; /* Invalid Beginning Date */

  goto endcheck;

end;

length srvlzip patzip $ 5 _med2nd _aged _modtcsg $ 1 patid $ 9;

srvlzip = substr(prvzp,1,5);
patzip  = substr(bene_zip,1,5);


/* Format data to look more closely like private data */

patid = BID_6434_1;

_cdob = put(bene_dob,8.);
```

```
birthdd = substr(_cdob,7,2);
birthmo = substr(_cdob,5,2);
birthyr = substr(_cdob,1,4);

_cschdt = put(schld_dt,8.);

claimpdd = substr(_cschdt,7,2);
claimpmo = substr(_cschdt,5,2);
claimpyr = substr(_cschdt,1,4);

_cthrudt = put(thru_dt,8.);

endday   = substr(_cthrudt,7,2);
endmo    = substr(_cthrudt,5,2);
endyr    = substr(_cthrudt,1,4);

patliab = max(lddmt,0) + max(cnmt,0);

payment = sum(reimamt,patliab);
*****if (payment le 0) then goto endcheck;


/* Flag if Medicare is not the primary payer */
if (lprdmt gt 0) then _med2nd = '1';
else _med2nd = '0';

/* Re-Map SRVPLAC */
if (SRVPLAC eq '31') then SRVPLAC = '32';
ELSE if (SRVPLAC eq '72') then SRVPLAC = '71';

*** specialty ***;
length speclty1 $ 3;
  *** set speclty1 ***;
  if hcfpcl ne " " then do;
    if hcfpcl = "01" then speclty1 = "001";
    else if hcfpcl = "02" then speclty1 = "002";
    else if hcfpcl = "03" then speclty1 = "003";
    else if hcfpcl = "05" then speclty1 = "004";
    else if hcfpcl = "06" then speclty1 = "005";
    else if hcfpcl = "07" then speclty1 = "006";
    else if hcfpcl = "93" then speclty1 = "007";
    else if hcfpcl = "46" then speclty1 = "008";
    else if hcfpcl = "08" then speclty1 = "009";
    else if hcfpcl = "10" then speclty1 = "010";
    else if hcfpcl = "38" then speclty1 = "011";
    else if hcfpcl = "40" then speclty1 = "012";
    else if hcfpcl = "82" then speclty1 = "013";
    else if hcfpcl = "11" then speclty1 = "014";
    else if hcfpcl = "44" then speclty1 = "015";
    else if hcfpcl = "39" then speclty1 = "016";
    else if hcfpcl = "13" then speclty1 = "017";
    else if hcfpcl = "36" then speclty1 = "018";
    else if hcfpcl in ("83", "90", "91", "92", "98") then speclty1 = "019";
    else if hcfpcl = "18" then speclty1 = "020";
    else if hcfpcl = "20" then speclty1 = "021";
    else if hcfpcl = "12" then speclty1 = "022";
    else if hcfpcl = "04" then speclty1 = "023";
    else if hcfpcl = "22" then speclty1 = "024";
    else if hcfpcl = "37" then speclty1 = "025";
    else if hcfpcl in ("76", "77") then speclty1 = "026";
    else if hcfpcl = "24" then speclty1 = "027";
    else if hcfpcl = "25" then speclty1 = "028";
    else if hcfpcl = "28" then speclty1 = "029";
    else if hcfpcl in ("26", "27", "86") then speclty1 = "030";
    else if hcfpcl = "29" then speclty1 = "031";
    else if hcfpcl in ("30", "94") then speclty1 = "032";
    else if hcfpcl = "66" then speclty1 = "033";
    else if hcfpcl in ("14", "85") then speclty1 = "034";
    else if hcfpcl in ("33", "78") then speclty1 = "035";
    else if hcfpcl = "34" then speclty1 = "036";
    else if hcfpcl in ("19", "09", "72") then speclty1 = "037";
    else if hcfpcl in ("79", "84", "99", "81") then speclty1 = "038";
    else if hcfpcl = "16" then speclty1 = "039";
    else if hcfpcl = "59" then speclty1 = "042";
    else if hcfpcl = "64" then speclty1 = "043";
    else if hcfpcl = "35" then speclty1 = "044";
    else if hcfpcl = "70" then speclty1 = "045";
    else if hcfpcl = "71" then speclty1 = "047";
    else if hcfpcl = "43" then speclty1 = "049";
    else if hcfpcl = "42" then speclty1 = "050";
    else if hcfpcl in ("50", "89") then speclty1 = "051";
    else if hcfpcl = "67" then speclty1 = "054";
    else if hcfpcl = "41" then speclty1 = "055";
    else if hcfpcl = "48" then speclty1 = "056";
    else if hcfpcl = "65" then speclty1 = "057";
    else if hcfpcl in ("62", "68") then speclty1 = "058";
    else if hcfpcl = "80" then speclty1 = "059";
    else if hcfpcl = "60" then speclty1 = "060";
    else if hcfpcl = "61" then speclty1 = "061";
    else if hcfpcl in ("32", "55", "56", "57", "58", "97") then speclty1 = "062";
    else if hcfpcl in ("A5", "A7", "A8") then speclty1 = "080";
    else if hcfpcl in ("69", "95", "75") then speclty1 = "082";
    else if hcfpcl in ("52", "54") then speclty1 = "083";
    else if hcfpcl = "96" then speclty1 = "084";
    else if hcfpcl in ("63", "87") then speclty1 = "085";
    else if hcfpcl = "49" then speclty1 = "091";
    else if hcfpcl = "45" then speclty1 = "092";
```

```
    else if hcfpcl in ("88", "47", "51", "53", "73", "74", "A0", "A1", "A2", "A3", "A4", "A6")
         then speclty1 = "093";
  end;



/* Type of service */

if (typvcb eq '1') then srvtype = '01';
else
if (typvcb eq '2') then srvtype = '02';
else
if (typvcb eq '3') then srvtype = '03';
else
if (typvcb eq '4') then srvtype = '04';
else
if (typvcb eq '5') then srvtype = '05';
else
if (typvcb eq '6') then srvtype = '06';
else
if (typvcb eq '7') then srvtype = '07';
else
if (typvcb eq '8') then srvtype = '08';
else
if (typvcb eq '9') then srvtype = '09';
else
if (typvcb eq '0') then srvtype = '10';
else
if (typvcb eq 'A') then srvtype = '11';
else
if (typvcb eq 'B') then srvtype = '04';
else
if (typvcb eq 'C') then srvtype = '04';
else
if (typvcb eq 'D') then srvtype = '99';
else
if (typvcb eq 'E') then srvtype = '09';
else
if (typvcb eq 'F') then srvtype = '13';
else
if (typvcb eq 'G') then srvtype = '09';
else
if (typvcb eq 'H') then srvtype = '09';
else
if (typvcb eq 'I') then srvtype = '12';
else
if (typvcb eq 'J') then srvtype = '12';
else
if (typvcb eq 'K') then srvtype = '12';
else
if (typvcb eq 'L') then srvtype = '14';
else
if (typvcb eq 'M') then srvtype = '15';
else
if (typvcb eq 'N') then srvtype = '02';
else
if (typvcb eq 'P') then srvtype = '12';
else
if (typvcb eq 'Q') then srvtype = '99';
else
if (typvcb eq 'R') then srvtype = '18';
else
if (typvcb eq 'S') then srvtype = '09';
else
if (typvcb eq 'T') then srvtype = '09';
else
if (typvcb eq 'U') then srvtype = '09';
else
if (typvcb eq 'V') then srvtype = '19';
else
if (typvcb eq 'W') then srvtype = '09';
else
if (typvcb eq 'Y') then srvtype = '20';
else
if (typvcb eq 'Z') then srvtype = '21';

/* Recode of practioner specialty per Ben Steffen email of 12/17/1999  */

if ((modf1 eq 'AH') or (modf2 eq 'AH')) and
  (speclty1 ne '058') then do;
  _convah = _convah + 1;
  speclty1 = '058';
end;
else
if ((modf1 eq 'AJ') or (modf2 eq 'AJ')) and
  (speclty1 ne '059') then do;
  _convaj = _convaj + 1;
  speclty1 = '059';
end;
else
if ((modf1 in ('AK','AL')) or (modf2 in ('AK','AL'))) and
  (speclty1 ne '051') then do;
  _convak = _convak + 1;
  speclty1 = '051';
end;
else
if ((modf1 in ('AN','AS','AU')) or (modf2 in ('AN','AS','AU'))) and
```

```
  (speclty1 ne '062') then do;
  _convan = _convan + 1;
  speclty1 = '062';
end;
else
if ((modf1 in ('AV','AW','AY')) or (modf2 in ('AV','AW','AY'))) and
  (speclty1 ne '053') then do;
  _convav = _convav + 1;
  speclty1 = '053';
end;
else
if ((modf1 in ('QX','QZ')) or (modf2 in ('QX','QZ'))) and
  (speclty1 ne '049') then do;
  _convqx = _convqx + 1;
  speclty1 = '049';
end;

/* Flag per Ben Steffen memo of 12/21/1999 to Sophie, item 1a (as revised  */
/* per meeting with Sophie on 12/23/1999):                                 */

if (modf1 in ('TC','SG')) or (modf2 in ('TC','SG')) then _modtcsg = '1';
else _modtcsg = '0';

/* Flag per Ben Steffen memo of 12/21/1999 to Sophie, item 1a (as revised  */
/* per meeting with Sophie on 12/23/1999):                                 */

/* Define bene as "aged" if age 65 or older AS OF PROCDATE                 */
/* ***** NOTE: REMEMBER TO UPDATE PROCDATE VARIABLE YEARLY                 */

bene_dt=input(put(bene_DOB,8.),YYMMDD8.);
AGE = %AGE(BENE_DT, &PROCDATE.d);
IF AGE GE 65 THEN _AGED = '1';
ELSE _AGED = '0';

output;

endcheck:

if (endfile) then do;
  put '##### End of File Reached: ';
  put "# records excluded because begin date before &yyyy = " _not&yy 6.;
  put '# records excluded because not a MD resident = ' _notmd 6.;
  put '# records excluded because of dental HCPCS = ' _dental 6.;
  put '# records where specialty changed because MODF1 or MODF2 was AH = ' _convah 6.;
  put '# records where specialty changed because MODF1 or MODF2 was AJ = ' _convaj 6.;
  put '# records where specialty changed because MODF1 or MODF2 was AK/AL = ' _convak 6.;
  put '# records where specialty changed because MODF1 or MODF2 was AN/AS/AU = ' _convan 6.;
  put '# records where specialty changed because MODF1 or MODF2 was AV/AW/AY = ' _convav 6.;
  put '# records where specialty changed because MODF1 or MODF2 was QX/QZ = ' _convqx 6.;
end;

return;

run;



************;

proc format;

value $statfmt
    '01' = 'AL'
    '02' = 'AK'
    '03' = 'AZ'
    '04' = 'AR'
    '05' = 'CA'
    '06' = 'CO'
    '07' = 'CT'
    '08' = 'DE'
    '09' = 'DC'
    '10' = 'FL'
    '11' = 'GA'
    '12' = 'HI'
    '13' = 'ID'
    '14' = 'IL'
    '15' = 'IN'
    '16' = 'IA'
    '17' = 'KS'
    '18' = 'KY'
    '19' = 'LA'
    '20' = 'ME'
    '21' = 'MD'
    '22' = 'MA'
    '23' = 'MI'
    '24' = 'MN'
    '25' = 'MS'
    '26' = 'MO'
    '27' = 'MT'
    '28' = 'NE'
    '29' = 'NV'
    '30' = 'NH'
    '31' = 'NJ'
    '32' = 'NM'
    '33' = 'NY'
    '34' = 'NC'
    '35' = 'ND'
```

```
    '36' = 'OH'
    '37' = 'OK'
    '38' = 'OR'
    '39' = 'PA'
    '40' = 'PR'
    '41' = 'RI'
    '42' = 'SC'
    '43' = 'SD'
    '44' = 'TN'
    '45' = 'TX'
    '46' = 'UT'
    '47' = 'VT'
    '48' = 'VI'
    '49' = 'VA'
    '50' = 'WA'
    '51' = 'WV'
    '52' = 'WI'
    '53' = 'WY'
;

value $carrfmt
   '00580' = 'XACT/DC'
   '00860' = 'XACT/NJ, Highmark-NJ'
   '00865' = 'XACT/PA, Highmark-PA'
   '00903' = 'Texas BS - DC'
   '10071' = 'Travelers (RR), RRB United-Healthcare'
   '10490' = 'VA - United Healthcare'
;

value $sexfmt
   '0' = 'Unknown'
   '1' = 'Male'
   '2' = 'Female'
;

value $racefmt
   '0' = 'Unknown'
   '1' = 'White'
   '2' = 'Black'
   '3' = 'Other'
   '4' = 'Asian'
   '5' = 'Hispanic'
   '6' = 'N. Amer. Native'
;

value $mstafmt
   '10' = 'Aged w/o ESRD'
   '11' = 'Aged w/ESRD'
   '20' = 'Disabled w/o ESRD'
   '21' = 'Disabled w/ESRD'
   '31' = 'ESRD only'
;

value $denyfmt
   '0' = 'Denied'
   '1' = 'Physician/supplier'
   '2' = 'Beneficiary'
   '3' = 'Both phy/supp and beneficiary'
   '4' = 'Hospital (hospital based physicians)'
   '5' = 'Both hospital and beneficiary'
   '6' = 'Group practice'
   '7' = 'Other entries'
   '8' = 'Federally funded'
   '9' = 'PA service'
   'A' = 'Bene under limitation of liability'
   'B' = 'Phy/supp under limitation of liability'
   'D' = 'Denied du to demonstration involvement'
   'P' = 'Physician ownership denial'
   'T' = 'MSP cost avoided-IEQ contractor'
   'U' = 'MSP cost avoided-HMO rate cell adj.'
   'V' = 'MSP cost avoided-litigation settlement'
   'X' = 'MSP cost avoided-generic'
   'Y' = 'MSP cost avoided-IRS/SSA data match proj'
;

value $asgnfmt
   'A' = 'Assigned claim'
   'N' = 'Non-assigned claim'
;

value $dmtrfmt
   'D' = 'Demo trailer present'
;

value $dmidfmt
   '01' = '01:NHCCMQ (RUGS) Demo'
   '02' = '02:Natl. HHA Prosp. Payment Demo'
   '03' = '03:Telemedicine Demo'
   '04' = '04:UMWA Managed Care Demo'
   '05' = '05:MCO Encounter Data Demo'
   '06' = '06:CABG Demo'
   '07' = '07:PCOE Demo'
   '08' = '08:Provider Partnership Demo'
   '15' = '15:ESRD Managed Care (MCO Enc. Data)'
   '30' = '30:LVRS or NETT Clinical Study'
   '31' = '31:VA Pricing Special Processing'
   '32' = '32:DOD Medicare Subvention Demo'
```

```
;

value $dtrlfmt
   'Y' = 'Diagnosis code trailer present'
;

value $ptypfmt
   '0' = '0:Clinics, groups, assns, partnerships'
   '1' = '1:Phys or supps reporting as solo pract'
   '2' = '2:Suppliers (other than solo pract)'
   '3' = '3:Institutional provider'
   '4' = '4:Independent laboratories'
   '5' = '5:Clinics (multiple specialties)'
   '6' = '6:Groups (single specialty)'
   '7' = '7:Other entities'
;

value $placfmt
           '11'='11:Provider''s Office'
           '12'='12:Patient''s Home'
           '21'='21:Inpatient Hospital'
           '22'='22:Outpatient Hospital'
           '23'='23:Emergency Room (Hospital Portion)'
           '24'='24:Ambulatory Surgical Center'
           '25'='25:Birthing Center'
           '26'='26:Military Treatment Facility'
           '32'='32:Nursing Facility/Skilled Nursing Facility'
           '33'='33:Custodial Care Facility'
           '34'='34:Hospice'
           '41'='41:Ambulance-Land'
           '42'='42:Ambulance-Air or Water'
           '50'='50:Federally Qualified Health Center'
           '51'='51:Inpatient Psychiatric Facility'
           '52'='52:Psychiatric Facility Partial Hospitaln'
           '53'='53:Community Mental Health Center'
           '54'='54:Intermedi Care FCLTY/Mentally Retarded'
           '55'='55:Residential Subst Abuse Treatment Fcty'
           '56'='56:Psychiatric Residential Treatment Ctr'
           '60'='60:Mass Immunization Center'
           '61'='61:Comprehensive Inpat Rehabilitatn Fclty'
           '62'='62:Comprehensive Outpat Rehabilitatn Fcty'
           '65'='65:End Stage Renal Disease Treatment Fcty'
           '71'='71:State,Local or Rural Public Health Clinic/'
           '81'='81:Independent Laboratory'
           '99'='99:Not Coded or Other Unlisted Facility'
;


value $specfmt
   '00' = '00:Carrier wide'
   '01' = '01:General practice'
   '02' = '02:General surgery'
   '03' = '03:Allergy/immunology'
   '04' = '04:Otolaryngology'
   '05' = '05:Anesthesiology'
   '06' = '06:Cardiology'
   '07' = '07:Dermatology'
   '08' = '08:Family practice'
   '09' = '09:Gynecology (osteopaths only)'
   '10' = '10:Gastroenterology'
   '11' = '11:Internal medicine'
   '12' = '12:Osteopathic manipulative therapy'
   '13' = '13:Neurology'
   '14' = '14:Neurosurgery'
   '15' = '15:Obstetrics'
   '16' = '16:Obstetrics/gynecology'
   '17' = '17:Ophth., otology, laryngology, rhinology'
   '18' = '18:Ophthalmology'
   '19' = '19:Oral surgery (dentists only)'
   '20' = '20:Orthopedic surgery'
   '21' = '21:Pathologic anatomy, clinical pathology'
   '22' = '22:Pathology'
   '23' = '23:Peripheral vascular disease, med. or surg.'
   '24' = '24:Plastic and reconstructive surgery'
   '25' = '25:Physical medicine and rehabilitation'
   '26' = '26:Psychiatry'
   '27' = '27:Psychiatry, neurology'
   '28' = '28:Colorectal surgery'
   '29' = '29:Pulmonary disease'
   '30' = '30:Diagnostic radiology'
   '31' = '31:Roentgenology, radiology'
   '32' = '32:Radiation therapy'
   '33' = '33:Thoracic surgery'
   '34' = '34:Urology'
   '35' = '35:Chropractic'

   '36' = '36:Nuclear medicine'
   '37' = '37:Pediatric medicine'
   '38' = '38:Geriatric medicine'
   '39' = '39:Nephrology'
   '40' = '40:Hand surgery'
   '41' = '41:Optometry'
   '42' = '42:Certified nurse midwife'
   '43' = '43:Crna, anesthesia assistant'
   '44' = '44:Infectious disease'
   '45' = '45:Mammography screening center'
   '46' = '46:Endocrinology'
```

```
   '48' = '48:Podiatry'
   '47' = 'Independent Diagnostic Testing Facility'
   '49' = '49:Ambulatory surgical center'
   '50' = '50:Nurse practitioner'
   '51' = '51:Medical supply company w/cert. orthotist'
   '52' = '52:Medical supply co. w/cert.  prosthetist'
   '53' = '53:Med. supply co. w/cert. prosth.-orthotist'
   '54' = '54:Medical supply company not incl. in 51,52,53'
   '55' = '55:Individual certified orthotist'
   '56' = '56:Individual certified prosthetist'
   '57' = '57:Individual certified prosth.-orthotist'
   '58' = '58:Individuals not incl. in 55, 56, lor 57'
   '59' = '59:Ambulance service supplier'
   '60' = '60:Public health or welfare agencies'
   '61' = '61:Voluntary health or charitable agencies'
   '62' = '62:Psychologist'
   '63' = '63:Portable X-ray supplier'
   '64' = '64:Audiologist'
   '65' = '65:Physical therapist'
   '66' = '66:Rheumatology'
   '67' = '67:Occupational therapist'
   '68' = '68:Clinical psychologist'
   '69' = '69:Clinical laboratory'
   '70' = '70:Multispecialty clinic or group practice'
   '71' = '71:Diagnostic X-ray (GPPP)'
   '72' = '72:Diagnostice laboratory (GPPP)'
   '73' = '73:Physiotherapy (GPPP)'
   '74' = '74:Occupational therapy (GPPP)'
   '75' = '75:Other medical care (GPPP)'
   '76' = '76:Peripheral vascular disease'
   '77' = '77:Vascular surger'
   '78' = '78:Cardiac surgery'
   '79' = '79:Addiction medicine'
   '80' = '80:Licensed clinical social worker'
   '81' = '81:Critical care (intensivists)'
   '82' = '82:Hematology'
   '83' = '83:Hematology/oncology'
   '84' = '84:Preventive medicine'
   '85' = '85:Maxillofacial surgery'
   '86' = '86:Neuropsychiatry'
   '87' = '87:All other suppliers'
   '88' = '88:Unknown supplier/provider specialty'
   '89' = '89:Certified clinical nurse specialist'
   '90' = '90:Medical oncology'
   '91' = '91:Surgical oncology'
   '92' = '92:Radiation oncology'
   '93' = '93:Emergency medicine'
   '94' = '94:Interventional radiology'
   '95' = '95:Independent physiological laboratory'
   '96' = '96:Optician'
   '97' = '97:Physician assistant'
   '98' = '98:Gynecologist/oncologist'
   '99' = '99:Unknown physician specialty'
   'A0' = 'A0:Hospital'
   'A1' = 'A1:SNF'
   'A2' = 'A2:Intermediate care nursing facility'
   'A3' = 'A3:Nursing facility, other'
   'A4' = 'A4:HHA'
   'A5' = 'A5:Pharmacy'
   'A6' = 'A6:Medical supply company w/resp. ther.'

   'A7' = 'A7:Department store'
   'A8' = 'A8:Grocery store'
;

value $rdpyfmt
   '0' = 'N/A'
   '1' = '65%'
   '2' = '75%'
   '3' = '85%'
;

value $tpsvfmt
   '1' = '1:Medical care'
   '2' = '2:Surgery'
   '3' = '3:Consultation'
   '4' = '4:Diagnostic radiology'
   '5' = '5:Diagnostic laboratory'
   '6' = '6:Therapeutic radiology'
   '7' = '7:Anesthesia'
   '8' = '8:Assistant at surgery'
   '9' = '9:0ther medical items or services'
   '0' = '0:Whole blood'
   'A' = 'A:Used DME'
   'B' = 'B:High risk screening mammography'
   'C' = 'C:Low risk screening mammography'
   'D' = 'D:Ambulance'
   'E' = 'E:Enteral/parenteral/nutrients/supplies'
   'F' = 'F:Ambulatory surgical center'
   'G' = 'G:Immunosuppressive drugs'
   'H' = 'H:Hospice services'
   'I' = 'I:Purchase of DME'
   'J' = 'J:Diabetic shoes'
   'K' = 'K:Hearing items and services'
   'L' = 'L:ESRD supplies'
   'M' = 'M:Monthly capitation payment for dialysis'
   'N' = 'N:Kidney donor'
```

```
   'P' = 'P:Lump sum purhcase of DME, prosth., orth.'
   'Q' = 'Q:Vision items or services'
   'R' = 'R:Rental of DME'
   'S' = 'S:Surgical dressings or other med. supplies'
   'T' = 'T:Psychological therapy'
   'U' = 'U:Occupational therapy'
   'V' = 'V:Pneumococcal/flu/vaccine'
   'W' = 'W:Physical therapy'
   'Y' = 'Y:Second opinion on elective surgery'
   'Z' = 'Z:Third opinion on elective surgery'
;

value $prtpfmt
   '1' = '1:Participating'
   '2' = '2:All/some $s appld to deduc Partic'
   '3' = '3:Asgnmnt accepted/non-participating'
   '4' = '4:Asgnmnt not accepted/not-participating'
   '5' = '5:Accept-some $s appld to deduc non-Part'
   '6' = '6:Not accp-all $s appld to deduc non-Part'
   '7' = '7:Partic provider not accepting assgnmnt'
;

value $betfmt
   'M1A' = 'Office visits - new'
   'M1B' = 'Office visits - established'
   'M2A' = 'Hospital visit - initial'
   'M2B' = 'Hospital visit - subsequent'
   'M2C' = 'Hospital visit - critical care'
   'M3 ' = 'Emergency room visit'
   'M4A' = 'Home visit'
   'M4B' = 'Nursing home visit'
   'M5A' = 'Specialist - pathology'
   'M5B' = 'Specialist - psychiatry'
   'M5C' = 'Specialist - opthamology'
   'M5D' = 'Specialist - other'
   'M6 ' = 'Consultations'
   'P0 ' = 'Anesthesia'
   'P1A' = 'Major proc. - breast'
   'P1B' = 'Major proc. - colectomy'
   'P1C' = 'Major proc. - cholecystectomy'
   'P1D' = 'Major proc. - turp'
   'P1E' = 'Major proc. - hysterctomy'
   'P1F' = 'Major proc. - explor/decom/excisdisc'
   'P1G' = 'Major proc. - other'
   'P2A' = 'Major proc. - cardiovasc:CABG'
   'P2B' = 'Major proc. - cardiovasc:aneurysm rep'
   'P2C' = 'Major proc. - cardiovasc:thromboend.'
   'P2D' = 'Major proc. - cardiovasc:coron angiop'
   'P2E' = 'Major proc. - cardiovasc:pacemaker ins'
   'P2F' = 'Major proc. - cardiovasc:other'
   'P3A' = 'Major proc. - orthop:hip fract repair'
   'P3B' = 'Major proc. - orthop:hip replacement'
   'P3C' = 'Major proc. - orthop:knee replacement'
   'P3D' = 'Major proc. - orthop:other'
   'P4A' = 'Eye proc. - corneal transplant'
   'P4B' = 'Eye proc. - cataract removal/lens ins.'
   'P4C' = 'Eye proc. - retinal detachment'
   'P4D' = 'Eye proc. - treatment'
   'P4E' = 'Eye proc. - other'
   'P5A' = 'Ambulatory proc. - skin'
   'P5B' = 'Ambulatory proc. - musculoskeletal'
   'P5C' = 'Ambulatory proc. - inguinal hernia rep'
   'P5D' = 'Ambulatory proc. - lithotripsy'
   'P5E' = 'Ambulatory proc. - other'
   'P6A' = 'Minor proc. - skin'
   'P6B' = 'Minor proc. - musculoskeletal'
   'P6C' = 'Minor proc. - other (Medicare fee sch.)'
   'P6D' = 'Minor proc. - other (non-Med. fee sch.)'
   'P7A' = 'Oncology - radiation therapy'
   'P7B' = 'Oncology - other'
   'P8A' = 'Endoscopy - arthroscopy'
   'P8B' = 'Endoscopy - upper gastrointestinal'
   'P8C' = 'Endoscopy - sigmoidoscopy'
   'P8D' = 'Endoscopy - colonoscopy'
   'P8E' = 'Endoscopy - cystoscopy'
   'P8F' = 'Endoscopy - bronchoscopy'
   'P8G' = 'Endoscopy - laparoscopic cholecystecomy'
   'P8H' = 'Endoscopy - laryngoscopy'
   'P8I' = 'Endoscopy - other'
   'P9A' = 'Dialysis services'
   'I1A' = 'Standard imaging - chest'
   'I1B' = 'Standard imaging - musculoskeletal'
   'I1C' = 'Standard imaging - breast'
   'I1D' = 'Standard imaging - contrast gastroint.'
   'I1E' = 'Standard imaging - nuclear medicine'
   'I1F' = 'Standard imaging - other'
   'I2A' = 'Advanced imaging - CAT: head'
   'I2B' = 'Advanced imaging - CAT: other'
   'I2C' = 'Advanced imaging - MRI: brain'
   'I2D' = 'Advanced imaging - MRI: other'
   'I3A' = 'Echography - eye'
   'I3B' = 'Echography - abdomen/pelvis'
   'I3C' = 'Echography - heart'
   'I3D' = 'Echography - carotid arteries'
   'I3E' = 'Echography - prostate, transrectal'
   'I3F' = 'Echography - other'
   'I4A' = 'Imaging/proc. - heart incl. cardiac cath'
```

```
   'I4B' = 'Imaging/proc. - other'
   'T1A' = 'Lab tests-routine venipun (non-Med. fee)'
   'T1B' = 'Lab tests-automated general profiles'
   'T1C' = 'Lab tests-urinalysis'
   'T1D' = 'Lab tests-blood counts'
   'T1E' = 'Lab tests-glucose'
   'T1F' = 'Lab tests-bacterial cultures'
   'T1G' = 'Lab tests-other (Med. fee schedule)'
   'T1H' = 'Lab tests-other (non-Med. fee schedule)'
   'T2A' = 'Other tests-electrocardiograms'
   'T2B' = 'Other tests-cardiovasc. stress tests'
   'T2C' = 'Other tests-EKG monitoring'
   'T2D' = 'Other tests-other'
   'D1A' = 'Medical/surgical supplies'
   'D1B' = 'Hospital beds'
   'D1C' = 'Oxygen and supplies'
   'D1D' = 'Wheelchairs'
   'D1E' = 'Other DME'
   'D1F' = 'Orthotic devices'
   'O1A' = 'Ambulance'
   'O1B' = 'Chiropractic'
   'O1C' = 'Enteral and parenteral'
   'O1D' = 'Chemotherapy'
   'O1E' = 'Other drugs'
   'O1F' = 'Vision, hearing and speech services'
   'O1G' = 'Influenze immunization'
   'Y1 ' = 'Other - Medicare fee schedule'
   'Y2 ' = 'Other - non-Medicare fee schedule'
   'Z1 ' = 'Local codes'
   'Z2 ' = 'Undefined codes'
;

value $pyrfmt
   'A' = 'A:Working aged bene/spouse w/eghp'
   'B' = 'B:ESRD bene in 18 mo coord period w/eghp'
   'C' = 'C:Cond. paymt by Medicare-reimb expected'
   'D' = 'D:Automobile no-fault'
   'E' = 'E:Workers compensation'
   'F' = 'F;Pub Hlth Srvc or other federal agency'
   'G' = 'G:Working disables bene'
   'H' = 'H:Black lung'
   'I' = 'I:Dept. of Veterans Affairs'
   'J' = 'J:Any liability insur. (eff. 3/94-3/97)'
   'L' = 'L:Any liability insur. (eff. 4/97)'
   'M' = 'M:Override code: eghp services involved'
   'N' = 'N:Override code: non-eghp srvcs involved'
   ' ' = '(Blank):Medicare is primary payer'
   'T' = 'T:MSP cost avoided-IEQ contractor'
   'U' = 'U:MSP cost avoided-HMO rate cell adj cont'
   'V' = 'V:MSP cost avoided-litigatin setlmt cont'
   'X' = 'X:MSP cost avoided override code'
   'Y' = 'Y:Other secondary payer investigation'
   'Z' = 'Z:Medicare is primary payer'
;

value $prcfmt
   'A' = 'Allowed'
   'B' = 'Benefits exhausted'
   'C' = 'Noncovered care'
   'D' = 'Denied'
   'I' = 'Invalid data'
   'L' = 'CLIA'
   'M' = 'Multiple submittal-duplicate line item'
   'N' = 'Medically unnecesary'
   'O' = 'Other'
   'P' = 'Physician ownership denial'
   'R' = 'Reprocessed-adj. based on subseq procsng'
   'S' = 'Secondary payer'
   'T' = 'MSP cost avoided-IEQ contractor'
   'U' = 'MSP cost avoided-HMO rate cell adjust.'
   'V' = 'MSP cost avoided-litigation'
   'X' = 'MSP cost avoided-generic'
   'Y' = 'MSP cost avoided-IRS/SSA data match proj'
   'Z' = 'Zero payment- allowed tests'
;

value $pdswfmt
   '0' = '80%'
   '1' = '100%'
   '3' = '100% limitation of liability only'
;


value $dedifmt
   '0' = '0:Service subject to deductible'
   '1' = '1:Service not subject to deductible'
;

value $pindfmt
   '1' = '1:Actual charge'
   '2' = '2:Customary charge'
   '3' = '3:Prevailing charge'
   '4' = '4:Other (ASC fees, rad-OP limits, denial)'
   '5' = '5:Lab fee schedule'
   '6' = '6:Physician fee schedule - full fee amt.'
   '7' = '7:Physician fee schedule - transition'
   '8' = '8:Clinical psychologist fee schedule'
```

```
   '9' = '9:DME and prosth/ortho fee schedules'
;

value $mtsfmt
   '0' = '0:Values reported as zero'
   '1' = '1:Transportation (ambulance) miles'
   '2' = '2:Anesthesia time units'
   '3' = '3:Services'
   '4' = '4:Oxygen units'
   '5' = '5:Units of blood'
   '6' = '6:Anesthesia base and time units'
;

value $dcmfmt
   '0' = '0:No additional documentation'
   '1' = '1:Addl docum submitted non-DME EMC claim'
   '2' = '2:CMN/Rx/other doc. shows medic. necess'
   '3' = '3:Prior auth obtained and approved'
   '4' = '4:Prior auth requested but not approved'
   '5' = '5:CMN/Rx/other doc. shows not med necess'
   '6' = '6:CMN/Rx/other doc. approve after reject'
   '7' = '7:Recertification CMN/Rx/other docum.'
;

value $modif
   '  '  = 'Blank'
   '00'  = '00'
   '99'  = '99'
   other = 'Others'
;

value $id
   ' '  = 'Blank'
  other = 'Assigned'
;

run;

*************************************************************;

proc freq data=outdd.psbene&YYYY.x;
tables state_cd        /list missing;
tables ms_cd           /list missing;
tables birthyr         /list missing;
tables begyr*endyr     /list missing;
tables srvtype*typvcb  /list missing;
tables srvplac         /list missing;
tables speclty1*hcfpcl /list missing;
tables claimpyr        /list missing;
tables patzip          /list missing;
tables srvlzip         /list missing;
tables modf1*modf2     /list missing;
tables srvunit         /list missing;
tables _modtcsg        /list missing;
tables _aged           /list missing;
tables _med2nd         /list missing;
tables patid           /list missing;

format typvcb   $tpsvfmt.
       srvplac  $placfmt.
       hcfpcl   $specfmt.
       ms_cd    $mstafmt.
       state_cd $statfmt.
       patzip   $stname.
       srvlzip  $stname.
       modf1    $modif.
       modf2    $modif.
       srvunit  $mtsfmt.
            patid    $id.
;

title2 "&yyyy Medicare Physician/Supplier File: Reformatted Beneficiary Version";
run;

proc univariate data=outdd.psbene&YYYY.x;
var bchrge alloamt reimamt patliab payment lprdmt

    ndiag noline nunits;
run;

proc univariate data=outdd.psbene&YYYY.x;
var bchrge alloamt reimamt patliab payment lprdmt
    ndiag noline nunits;
where (_med2nd eq '0');
title3 'Excluding Records Where Medicare Is Not the Primary Payer';
run;


proc freq data=outdd.psbene&YYYY.x;
tables proc   /out=freq1 noprint;
tables modf1  /out=freq2 noprint;
tables modf2  /out=freq3 noprint;
tables provid /out=freq4 noprint;
tables diag1  /out=freq5 noprint;
tables diag2  /out=freq6 noprint;
tables diag3  /out=freq7 noprint;
tables diag4  /out=freq8 noprint;
```

```
title3 ' ';
run;

proc sort data=freq1;
by descending count;
run;

proc print n uniform data=freq1 (obs=50);
title3 'Top 50 Values of PROC';
run;

proc sort data=freq2;
by descending count;
run;

proc print n uniform data=freq2 (obs=50);
title3 'Top 50 Values of MODF1';
run;

proc sort data=freq3;
by descending count;
run;

proc print n uniform data=freq3 (obs=50);
title3 'Top 50 Values of MODF2';
run;

proc sort data=freq4;
by descending count;
run;

proc print n uniform data=freq4 (obs=50);
title3 'Top 50 Values of PROVID';
run;

proc sort data=freq5;
by descending count;
run;

proc print n uniform data=freq5 (obs=50);
title3 'Top 50 Values of DIAG1';
run;

proc sort data=freq6;
by descending count;
run;

proc print n uniform data=freq6 (obs=50);
title3 'Top 50 Values of DIAG2';
run;

proc sort data=freq7;
by descending count;
run;

proc print n uniform data=freq7 (obs=50);
title3 'Top 50 Values of DIAG3';
run;

proc sort data=freq8;
by descending count;
run;

proc print n uniform data=freq8 (obs=50);
title3 'Top 50 Values of DIAG4';
run;

proc print n uniform data=outdd.psbene&YYYY.x (obs=25);
title3 'Print of 25 Observations';
run;

proc contents data=outdd.psbene&YYYY.x;
title3 ' ';
run;

%jobend;

dm "log;    print file = I:\MCdb\mcr_fn\programs_2006\PHYbene_load.log replace";
dm "output; print file = I:\MCdb\mcr_fn\programs_2006\PHYbene_load.lst replace";
```

```
proc datasets kill;
%jobstart;
*==============================================================================*
Program     : mcr_1.sas
Project     : Maryland
Purpose     : Create intermediate Medicare dataset from preprocessed HCFA data
              1. Recodes:
                 spec = 999 to blank
                 svcplace = 50 or 60 to 10
              2. Create variables:
                 F_UNCV  (flag uncovered services)
                 f_county (flag whether patient residence county is known)
                 D_DAT_F (flag records with service year earlier than current analysis year)
                 F_YR00  (flag records with service in 2000)
                 F_FINMS (flag services with all financial vars = 0 or missing)
                 f_proc  (flag system used for procedure codes)
                 LOS     (length of service)
                 AGE     (calculate age at service)
                 AGE1299 (calculate age on 12/31 of analysis year)
                 f_agecat (flag age grouping)
                 numdxc  (recount # of non-zero non-blank diagnosis codes)
                 county_p, STATE_P, region_p, urban_p (patient geographic location variables)
                 F_PAYM  (flag payment type)
                 INSTATE ( provider geographic location variable)
                 payertyp (payer group type)
                 F_MODF  (flag for TC or SG in AMA modifiers, create by renaming _MODTCSG)
                 EDC1-EDC3 (expanded diagnosis clusters aggregated from ICD-9 diagnosis codes)
                           and associated flags F_PRMICD and F_ICDEDC
                 RVU (relative value units, rvu_stat (status flag) and F_PRCRVU (flag)
                 betosdet, betosagg (collapse procedure codes to detailed and aggregated BETOS codes)
                 F_SERV_A (flag services for analysis)
                 CLMPDATE - claim paid date
                 BEGINDT  - service begin date
                 ENDDT    - service end date
              3. Impute variables:
                 RVU (relative value units) and RUV_F (flag)
              4. Other:
                 Set BETOS, EDC and RVU values to missing for records not
                   in this analysis year
Input       : psben99x.sd2
Output      : mcr_1.sd2
Programmer  : Laurie Hamilton
Date Created: December 6, 2000

Notes       : This is a revision of Raymond Hu's 1999 program to remove code
              specific to 1999 edits and add code specific to 1999 edits
              Several recodes and flagging have also been added from the 1999
              programs used to build the practitioner report analysis files
Revised     : 1/16/01
              1. Add variables from last years analysis file (all no/yes have been flipped):
                 AGECAT
                 F_AGE (1/0 valid/invalid)
                 F_BET0S (1/0)
                 F_PATID (1/0 valid/invalid - note change from last year which was
                          1,2,3 invalid/4 valid)
                 F_PAYER (1/0)
                 F_RVUGE0 (>= 1 = 1, other = 0)
                 F_PAYMA (1= 0 or positive/0=negative or missing)
                 Rename F_PAYM to F_PAYMD
                 Rename f_proc to F_CPT

              2. Change lengths to get all as short as possible


              1/25/2001
              1. Modify AGECAT, change oldest category to 65 to 110
              2. Keep MS_CD (Medicare Status Code) and rename to MCRSTAT.
                 Note that per Jeff
                 McCartney this variable is inquiry based and could
                 overstate ESRD. Next year should see about getting it
                 from the MEDPAC denominator file.

              10/30/01
              Modifications for Year 2000 Data
              1. Rename f_yr00 to f_yr01, change dates as needed in code
              2. Rename age1299 to age1200, change dates as needed in code

              11/02/02
              Modifications for Year 2001 Data
              1. Rename f_yr__
              2. Rename age12__
              3. drop EDCs, f_betos
              4. do not flag demographic donor records (will do in analysis set if
                 one is needed
Modified: 02/01/2005
          laurie hamilton
          adjust f_yrxx flag
          adjust age12xx flag and calculations
          remove BETOS childhood immunizations recode
          adjust begyr value in d_dat_f coding

Modified: 01/03/2006
          Sireesha Thati
          input : ps_bene2004x.sas7bdat
          output: mcr_1.sas7bdat

Modified: 02/01/2007
```

```
          Sireesha Thati
          input : psbene2005x.sas7bdat
          output: mcr_1.sas7bdat
          NOTE: In Year 2004 we had a program called set_payment_study_medicare_04.sas, which
          was run after the phybene_load.sas. The i/p dataset for this pgm was psbene2004x.sas7bdat
          and the o/p dataset that was created is ps_bene2004x.sas7bdat.
          This program was basically mcr_2.sas, where we assign RVUs. In 2004 this was done
          because we needed medicare rvus for the payment study report.
          Hence in 2005 we will not need set_payment_study_medicare_04.sas. The program mcr_1.sas
          will be run after phybene_load.sas
*****************************************************************************

Revised : 01/11/2007
          Kasey Chen
          input : psbene2006x.sas7bdat
          output: mcr_1.sas7bdat
Note:     1. Comment out AGE and AGE1206.
          2. Calculated AGE1206_NR (it is not rounded) for 2006 data.
          2. Added libname and used redefined formats for 2006 data
             I:\MCdb\datacol\Formats\PERMANENT\BETOS_formats
                                                \RVU_formats.
**=============================================================================
*                       ORIGINAL PROGRAM HEADER
*===============================================================================
*
*  Program Name:  i:\mcdb\MCDB_Fn\Programs\Mcr_Dt.sas
*  Programmer:    Raymond Hu
*  Date Created:  1/12/99
*
*  Description:  Create MCDB based on the Medical Care Data Base Development Plan
*
*  Note:
*
*
*  DATE MODIFIED: 1/22/2000
*      1. Do not drop the ALLOWAMT>7000 and just flag them
*      2. Do not drop the CNTY_CD=999 and just flag them
*  DATE MODIFIED:  01/26/2000 by Raymond
*      1) Impute RVU for these with RVU=0 and rvu_stat notin P,B
*  DATE MODIFIED: 3/14/2000 by Raymond
*      1) create i_mcr=0 - service in PR_MCR.SD2
*      2) create i_mPAT=0 - patient info used in MCR_SERV.SD2
*
*==============================================================================;
%let inset  = psbene2006x;
%let outset = mcr_1;
options obs = max nosymbolgen center errors=2 nodate  pageno=1;

libname in  "I:\MCdb\mcr_fn\sasdata";
libname out "I:\MCdb\mcr_fn\sasdata";
*libname library "i:\mcdb\datacol\formats\permanent";

libname library ("I:\MCdb\datacol\Formats\PERMANENT\BETOS_formats",
                 "I:\MCdb\datacol\Formats\PERMANENT\RVU_formats");
%let progdrv = i;


title1 "Creating MCR Dataset for 2006 Data";
title2 " %sysfunc(putn(%sysfunc(time()), hhmm.)) %sysfunc(putn(%sysfunc(date()), weekdate.))";


*****************************************************************************

                           PART I - FORMATS

*****************************************************************************
;
options nosource2;

%include "&progdrv:\mcdb\datacol\formats\uszipfmt.sas";
%include "&progdrv:\mcdb\datacol\formats\mdstzipfmt.sas";
%include "&progdrv:\mcdb\datacol\formats\mdcyzipfmt.sas";
%include "&progdrv:\mcdb\datacol\formats\cptfmt.sas";
%include "&progdrv:\mcdb\datacol\formats\hcpcsfmt.sas";
%include "&progdrv:\mcdb\datacol\formats\revcode.sas";
%include "&progdrv:\mcdb\datacol\formats\icd9pr_06.sas";
%include "&progdrv:\mcdb\dataCol\formats\ctssafips.sas"; /* new - needed for county_p */

***%include "&progdrv:\mcdb\datacol\formats\format_betos_2005.sas";
***%include "&progdrv:\mcdb\datacol\formats\stnamefmt.sas";
***%include "&progdrv:\mcdb\datacol\formats\ssacntyfmt.sas";

options source2;

*****************************************************************************

                            PART II
                          DATA EDITS

*****************************************************************************
;
data temp  (drop =
                   cnty_cd state_cd
                   enddate
                   hcfpcl lprdmt noline typvcb _aged spclty_1
                                       rename = (f_paym = f_paymd /*f_proc = f_cpt*/));
```

```
 set in.&inset (rename = (_modtcsg = f_modf
                          alloamt = allow
                          bchrge  = bill
                          diag1   = dx1
                          diag2   = dx2
                          diag3   = dx3
                          diag4   = dx4
                          modf1   = mod1
                          modf2   = mod2
                          ndiag   = numdx
                          nunits  = units
                          patzip  = zip_p
                                                          srvlzip = zip_s
                          proc    = cpt
                                                          srvplac = svcplace
                                                          reimamt = reimb
                          srvtype = tos
                          srvunit = unitind
                          _med2nd = f_mcr2nd
                          specltyl = spec
                          ms_cd    = mcrstat ));


 ********************************************
 PART II OPERATION
 Recode spec
 ********************************************;

 if spec = '999' then spec='';

 ********************************************
 PART II OPERATION
 Recode svcplace = '50' or '60' (federally qualified
 health center or mass immunization center) to '10'
 (other unspecified facility)
 ********************************************;

 if svcplace in ('50', '60') then svcplace='10';

 ********************************************
 PART II OPERATION
 Remove . from diagnosis codes
 ********************************************;

 array test1 dx1-dx4;
 do i=1 to 4;
    if index(test1(i),'.')>0 then test1(i)=compress(test1(i),'.');
end;


 ********************************************
 PART II OPERATION
 Define uncovered services flag (F_UNCV)
 1=uncovered service
 ********************************************;

 length f_uncv $ 1;

 if allow in (0,.,.M) and
    reimb in (0,.,.M) then f_uncv='1';
 else
 f_uncv='0';


 ********************************************
 PART II OPERATION
 Flag for service year prior to current analysis year
 (D_DAT_F)
 1=Yes 0=No
 Note: For 1999, all begin years are 4 digits and = 1999
 ********************************************;

 length d_dat_f $ 1;

 d_dat_f = '0';
 if begyr lt '2006' then d_dat_f = '1';

 ********************************************
 PART II OPERATION
 Flag for service year > 2006
 (f_yr07)
 1=Yes 0=No
 Note: For 2000, all begin years are 4 digits and = 2000
 ********************************************;

 length f_yr07 $ 1;

 f_yr07 = '0';
 if begyr gt '2006' then f_yr07 = '1';

 ********************************************
 PART II OPERATION
 Flag to identify services with all financial
 variables = 0 or missing (F_FINMS)
 Coded 1/0
 ********************************************;
```

```
     length f_finms $ 1;

     f_finms = '0';
     if bill    in (0,.,.C,.M)  and
        allow   in (0,.,.C,.M) and
        reimb in (0,.,.C,.M) and
        patliab in (0,.,.C,.M) then f_finms = '1';



     ********************************************
     PART II OPERATION
     Flag procedure coding type (F_CPT)
     (1 is homegrown for MCDB and is not used here)
     (MCDB does not use 7)
     2 = HCPCS
     3 = CPT
     4 = ICD-9-CM
     5 = Revenue Code
     6 = missing/blank
     7 = 7,Invalid PROC Code
     ********************************************;
     length f_cpt $ 1;

     if cpt=''                                 then f_cpt='6';
     else
     if put(cpt,$HCPCS.) = '1'                 then f_cpt='2';
     else
     if put(cpt,$CPTfmt.) = '1'                then f_cpt='3';
     else
     if put(cpt,$ICD9PR.) = '1'                then f_cpt='4';
     else
     if put(cpt,$RevCode.) = '1'               then f_cpt='5';
     else                                           f_cpt='7';



     ********************************************
     PART II OPERATION
     Calculate length of service (LOS)

     Note: For 1999, all begin and end dates are complete
     and valid
     ********************************************;

     length los 3;

     begdate = mdy(begmo,begday,begyr);
     enddate = mdy(endmo,endday,endyr);

     *** same day service = 1 ***;
     if begdate = enddate then los = 1;
     *** if enddate > begin date calculate, the rest will be missing ***;
     else
     if enddate gt begdate then los = enddate - begdate;


     ********************************************
     PART II OPERATION
     Calculate age at service (AGE) and
     age as of 12/31 of analysis year (AGE12__)
     Missing day = 15, Missing month = 6 Both
     day and month missing = 7/1

     Kasey Chen: 02/06/2008
     Only AGE1206_NR calculated for 2006.
     ********************************************;
/*
     begmox =begmo;
     begdayx=begday;
     begyrx =begyr;
     endmox =endmo;
     enddayx=endday;
     endyrx =endyr;
     birthyrx=birthyr;
     birthmox=birthmo;
     birthddx=birthdd;


     *** input date variables ***;
     *-- set the year to 4-digit year if it is two-digit;
     *-- set day=15 if only day=missing;
     *-- set month=7 and day=1 if both month and day are missing;
     begyrx=begyr;
     endyrx=endyrx;


     if birthyrx notin (' ','00','0000') and
        birthmox notin (' ',' 0','00') and
        birthddx in (' ',' 0','00') then birthddx='15';
     else if birthyrx notin (' ','00','0000') and
             birthmox in (' ',' 0','00') and
             birthddx in (' ',' 0','00') then do;
             birthmox='07';
             birthddx='01';
          end;
```

```
    if begyrx notin (' ','00','0000') and
       begmox notin (' ',' 0','00') and
       begdayx in (' ',' 0','00') then begdayx='15';
    else if begyrx notin (' ','00','0000') and
            begmox  in (' ',' 0','00') and
            begdayx in (' ',' 0','00') then do;
            begmox='07';
            begdayx='01';
         end;

    if endyrx notin (' ','00','0000') and
       endmox notin (' ',' 0','00') and
       enddayx in (' ',' 0','00') then enddayx='15';
    else if endyrx notin (' ','00','0000') and
            endmox  in (' ',' 0','00') and
            enddayx in (' ',' 0','00') then do;
            endmox='07';
            enddayx='01';
         end;

    *--- set day=day-1 if it is not missing and the day is invalid;
    if begyrx ne ' ' and begmox ne ' ' and begdayx ne ' ' and input(begdayx,2.)>0 and
       mdy(input(begmox,2.),input(begdayx,2.),input(begyrx,4.))=.
       then begdayx=put(input(begdayx,2.)-1,z2.);
    if endyrx ne ' ' and endmox ne ' ' and enddayx ne ' ' and input(enddayx,2.)>0 and
       mdy(input(endmox,2.),input(enddayx,2.),input(endyrx,4.))=.
       then enddayx=put(input(enddayx,2.)-1,z2.);
    if birthyrx notin ('    ','0000') and birthmox notin (' ','00') and birthddx  notin (' ','00') and
       input(birthddx,2.)>0 then do;
       if mdy(input(birthmox,2.),input(birthddx,2.),input(birthyrx,4.))=.
          then birthddx=put(input(birthddx,2.)-1,z2.);
    end;

    *** create AGE ***;
   if birthyrx notin (' ','00','0000') and begyrx notin (' ','00','0000') and
       input(begyrx,4.) >= input(birthyrx,4.) then do;
       if input(begyrx,4.) - input(birthyrx,4.)=0 then age=0;
       else if put(mdy(input(birthmox,2.),input(birthddx,2.),input(birthyrx,4.)),julday3.)<=
          put(mdy(input(begmox,2.),input(begdayx,2.),input(begyrx,4.)),julday3.)
          then age=input(begyrx,4.)-input(birthyrx,4.);
       else age=input(begyrx,4.)-input(birthyrx,4.)-1;
    end;
    else if birthyrx notin (' ','00','0000') and endyrx notin (' ','00','0000') and
            input(endyrx,4.) >= input(birthyrx,4.) then do;
       if input(endyrx,4.) - input(birthyrx,4.)=0 then age=0;
       else if put(mdy(input(birthmox,2.),input(birthddx,2.),input(birthyrx,4.)),julday3.)<=
               put(mdy(input(endmox,2.),input(enddayx,2.),input(endyrx,4.)),julday3.)
          then age=input(endyrx,4.)-input(birthyrx,4.);
       else age=input(endyrx,4.)-input(birthyrx,4.)-1;
    end;*/

    ********************************************************************;
    *    age1205                                                       *;
    *      (based on item #15 of the Next Step in Developing the MCDB)  *;
    *      (based on item #12 of the Practitioner Report:SUMMARY...)    *;
    *    Kasey Chen: 02/06/2008 Only AGE1206_NR calculated for 2006.    *;
    ********************************************************************;
/*
    if birthyrx notin (' ','00','0000') and 1999 >= input(birthyrx,4.) then do;
       if 2006 - input(birthyrx,4.)=0 then age1206=0;
       else if put(mdy(input(birthmox,2.),input(birthddx,2.),input(birthyrx,4.)),julday3.)<=
               put(mdy(12,31,2006),julday3.)
          then age1206=2006-input(birthyrx,4.);
       else age1206=2006-input(birthyrx,4.)-1;
    end;
    else if birthyrx notin (' ','00','0000') and 1999 < input(birthyrx,4.) then do;
       if input(birthmox,2.)=12 and input(birthddx,2.)=31
          then age1206=2006-input(birthyrx,4.);
       else age1206=2006-input(birthyrx,4.)-1;
    end;

    if age1206<0 or age1206>110 then age1206=.;*/

    ********************************************************************;
    *    Modified by Kasey Chen, 02/02/2008                            *;
    *      Per client request for 2006 data, age and age1206 are not    *;
    *      applicable, age1206 is only variable used to present AGE     *;
    ********************************************************************;
    length age1206_nr 3;
    if birthyr notin (' ','00','0000') then do;
      if mdy(input(birthmo,2.),input(birthdd,2.),input(birthyr,4.))
         = . then age1206_nr = .;
      else age1206_nr =
        (mdy(12,31,2006)-
         mdy(input(birthmo,2.),input(birthdd,2.),input(birthyr,4.)))/365.25;
    end;

    if input(birthyr,4.) = . then age1206_nr = .;
    else if age1206_nr < 0 or age1206_nr > 110 then age1206_nr = .;

    ********************************************
    PART II OPERATION
    Create valid age category flag (f_agecat)
    ********************************************;

    if age1206_nr = .      then f_agecat='1';
    else
```

```
     if 0<=age1206_nr<=64   then f_agecat='2';
     else
     if 65<=age1206_nr<=110 then f_agecat='3';
     else
     if age1206_nr > 110    then f_agecat='4';


     *********************************************
     PART II OPERATION
     Recalculate # of Diagnosis Codes Reported (numdxc)
     Note: This is just a count, doesnt check validity of code
     *********************************************;

     length numdxc 3;

     array adiag dx1-dx4;

     numdxc=0;
     do i = 1 to 4;
       if adiag(i) notin ('00'x,'','00000','0','00','000','0000') and
          compress(adiag(i),'00'x) ne ''                          then numdxc+1;
     end;

     ************************************************************************
     PART III OPERATION
     Create new variable for patient county of residence
     county_p = county fips code - derived from SSA county code
     001 = ALLEGANY
     003 = ANNE ARUNDEL
     005 = BALTIMORE
     009 = CALVERT
     011 = CAROLINE
     013 = CARROLL
     015 = CECIL
     017 = CHARLES
     019 = DORCHESTER
     021 = FREDERICK
     023 = GARRETT
     025 = HARFORD
     027 = HOWARD
     029 = KENT
     031 = MONTGOMERY
     033 = PRINCE GEORGES
     035 = QUEEN ANNES
     037 = SAINT MARYS
     039 = SOMERSET
     041 = TALBOT
     043 = WASHINGTON
     045 = WICOMICO
     047 = WORCESTER
     510 = BALTIMORE CITY
     ************************************************************************;
     length county_p $ 3;

     county_p = put(cnty_cd,$ssafips.);

     **********************************************
     PART II OPERATION
     Flag whether patient   of residence is known
     (f_county)
     1 = yes, 0 = no
     **********************************************;

     length f_county $ 1;
     f_county='0';

     if county_p ne ' ' then f_county = '1';

/*****************
     **********************************************
     PART II OPERATION
     Create Extended Diagnosis Clusters (EDC1-EDC3) from
     ICD-9 Diagnosis Codes

     Explanation of the algorithm:
      1.Remove all duplicate diagnostic codes from the the vector
        store the result (unique DXs) in the UIDC vector.
      2. Scan through the UIDC vector and convert all IDCs to EDCs.
      3. There should not be any conversion for Lab/Radiology
         procedures that are in LABRAD format.
      4. Create EDC1,EDC2,EDC3 which the first 3 EDCs  to be
         converted. For example if ICD1 converted fine to EDC1 and
         ICD2 did not convert to EDC2--if ICD3 can be converted to
         EDC, then it should become EDC2 not EDC3.
      5. Drop EDC4-EDC4 because they are not needed and also drop
         Udx1-4 because they are just temporary variables
     **********************************************;

     length udx1-udx4 $ 5;
     length edc1-edc4 $ 5;

     array uicd[4] udx1-udx4; *for unique diagnostic codes;
     array icd[4]  dx1-dx4;
     array edc[4]  edc1-edc4;

     do i=1 to 4;
        uicd[i]=icd[i];
```

```
        if i>1 then do j=1 to i-1;
           if icd[j]=icd[i] then uicd[i]="";
        end;
     end;

     retain k;

     k=0;

     do i = 1 to 4;
       if not(put(uicd[i], ICD2EDC.) = ""  or  put(cpt,LABRAD.)="INV") then do;
         k=k+1;
         edc[k]= put(uicd[i], ICD2EDC.);
       end;
     end;

**************/

     ***********************************************
     PART II OPERATION
     Create Payment Category Flag (F_PAYM)
     1=negative payment
     2=0 payment
     3=positive payment
     4=missing payment
     ***********************************************;

     length f_paym $ 1;

     if payment=. then F_PAYM='4';
     else
     if payment<0 then F_PAYM='1';
     else
     if payment=0 then F_PAYM='2';
     else
     if payment>0 then F_PAYM='3';


     ***********************************************
     PART II OPERATION
     Create Variable to flag practitioner site of service
     (INSTATE)
     1=valid MD zipcode
     0=valid non-MD zipcode
     Missing = invalid zipcode
     ***********************************************;

     length instate $ 1;

     if put(zip_s, $mdstzip.) = '1' then instate='1';

     else
     if put(zip_s, $uszip.) = '1'  then instate='1';
     else
     instate='0';


     ***********************************************
     PART II OPERATION
     Create Payer Group Flag (payertyp)
     = Blank if age1206_nr is negative, missing or between 0 and 64
     = 4 (Medicare, non-HMO) if between 65 and 110
     ***********************************************;

     length payertyp $ 1;

     if 65 <= age1206_nr <=110 then payertyp = '4';
     else payertyp=' ';




     ***********************************************
     PART II OPERATION
     Assign BETOS Codes based on CPT-4/HCPCS Proc Codes

     Note: Do only for this years analysis records
     ***********************************************;

     length betosdet $ 3;
     length betosagg $ 1;

      *** Detailed BETOS Code ***;

     if f_cpt in ("2","3") then do;
         if put(cpt,EXP2BET.) in ("") then do;
            betosdet=put(cpt,PRC2BET.);
         end;
     end;

     *** aggregated BETOS Code ***;
     if      substr(betosdet,1,1)="M"          then betosagg="1";
     else if substr(betosdet,1,1)="P"          then betosagg="2";
     else if substr(betosdet,1,1)="I"          then betosagg="3";
     else if substr(betosdet,1,1)="T"          then betosagg="4";
     else if substr(betosdet,1,1)="D"          then betosagg="5";
     else if substr(betosdet,1,1)="O"          then betosagg="6";
```

```
else if substr(betosdet,1,1) in ("Y","Z") then betosagg="7";

**********************************************
RVUs
**********************************************;

*%include rvu.sas;

**********************************************
PART II OPERATION
Create flag for specialty categories (f_specl)

**********************************************;

length f_specl $ 1;

spclty_1=input(spec,3.);

if (0< spclty_1 le 63) or (70 le  spclty_1 le 76) or (80 le  spclty_1 le 85) or
   (90 le spclty_1 le 93) or (spclty_1=100) then f_specl='1';
else if spclty_1 =102 then f_specl='2';
else f_specl='3';


**********************************************
PART II OPERATION
Create Service and Claim Paid Dates
(Note: All Medicare Data Dates are Clean so

the imputation used for mcdb isn't needed, just
set the flag
**********************************************;

clmpdate = mdy(claimpmo,claimpdd,claimpyr);
begindt  = mdy(begmo,begday,begyr);
enddt    = mdy(endmo,endday,endyr);

length f_baddt $ 1;
f_baddt = '4';
label f_baddt = "Service Date Imputation Flag";

format clmpdate
       begindt
       enddt     mmddyy10.;

**********************************************
PART II OPERATION
Flag records which will be used in analysis
**********************************************;
*** Criteria are:

    Covered service
    Medicare is primary payer
    Service coded with valid CPT-4 or HCPCS procedure codes
    Procedure does not have 'TC' or SG' as modifier
    Procedure is not in BETOS category of durable medical equipment
    Service is in 2000
***;

if f_uncv eq '0'        and
   f_mcr2nd eq '0'      and

   f_cpt in ('2','3')  and
   f_modf eq '0'        and
   betosagg ne '5'      and
   f_yr07 = '0'         then f_serv_a = '1';
else
f_serv_a = '0';


**************************************************************************
PART II OPERATION
Create Flag for Valid Age Group
**************************************************************************;
length f_age $ 1;
if 65<=age1206_nr<=110 then f_age='1';
else f_age='0';

 **************************************************************************
PART II OPERATION
Create Flag for Valid Patient ID
**************************************************************************;
length f_patid $ 1;



 if patid=''            or
 compress(PATID,'0')='' or
 compress(patid,'9')='' then f_patid='0';
 else f_patid='1';


**************************************************************************
PART II OPERATION
Create Flag for Valid PAYERTYP
**************************************************************************;
length f_payer $ 1;
```

```
     label f_payer = 'Valid Payer Type (1/0)';
     if payertyp = '4' then f_payer = '1';
     else
     f_payer = '0';



     *********************************************
     PART II OPERATION
     Add Labels
     *********************************************;
     Label
                    f_age     =                 'Valid Age Flag(age1206_nr 65-110) (1/0)'
          f_patid   =         'Valid Patient ID (1/0)'
          /*age       =         'Patient Age at Service'*/
          age1206_nr   =      'Patient Age as of 12/31/2006, Not Rounded'
          allow     =         'Allowed Amount-AA ($)'
          bill      =         'Billed Charge-BC ($)'
          begindt   =         'Service From Date'
          begday    =         'Service From Day'
          begmo     =         'Service From Month'
          begyr     =         'Service From Year'
          betosdet  =         '2006 BETOS Category, Detailed'
          betosagg  =         '2006 BETOS Category, Aggregated'
          birthdd   =         'Patient Birth Day'
          birthmo   =         'Patient Birth Month'
          birthyr   =         'Patient Birth Year'
          clmpdate  =         'Claim Paid Date'
          claimpdd  =         'Claim Paid Day'
          claimpmo  =         'Claim Paid Month'
          claimpyr  =         'Claim Paid Year'
          county_p  =         'Patient County of Residence FIPS Code'
          d_dat_f   =         'Service Prior to 2000 (1/0)'
          dx1       =         'ICD-9-CM Diagnosis Code 1'
          dx2       =         'ICD-9-CM Diagnosis Code 2'
          dx3       =         'ICD-9-CM Diagnosis Code 3'
          dx4       =         'ICD-9-CM Diagnosis Code 4'
          f_agecat  =         'Age Category Code'
          f_county  =         'Patient Residence County Known (1/0)'
          enddt     =         'Service Thru Date'
          endday    =         'Service Thru Day'
          endmo     =         'Service Thru Month'
          endyr     =         'Service Thru Year'
          fedtaxid  =         'Practitioner Federal Tax ID'
          f_finms   =         'All 4 Financial Vars=0/Missing (1/0)'
          f_modf    =         'TC or SG in Modifiers I or II (1/0)'
          f_paym    =         'Payment Category (1=Neg,2=0,3=>0,4=Miss)'
          f_cpt    =         'Procedure Code Category (1-7)'
          f_specl   =         'Flag for Specialty Categories'
          f_uncv    =         'Uncovered (AA&RA=0/Miss) Service (1/0)'
          f_yr07    =         'Service Provided in 2006+ (1/0)'
          instate   =         'MD Service Location ZIP (1/0/blank)'
          f_serv_a  =         'Service Used in Analysis (1/0)'
          los       =         'Length of Service(1 for Same&Next Day)'
          mod1      =         'AMA Modifier I'
          mod2      =         'AMA Modifier II'
          numdxc    =         'Number of Diagnosis Codes, Calculated'
          units     =         'Number of Units for a Service'
          patliab   =         'Patient Liability-PL ($)'
          zip_p     =         'Patient Residence ZIP Code'
          payertyp  =         'Payer Type (4=MCRNHMO)'
          payment   =         'Payment=Sum(RA,PL) ($)'
          cpt       =         'CPT-4/HCPCS/Other Procedure Code'
          provid    =         'Servicing Practitioner Identifier'
          reimb     =         'Reimbursement Amount-RA ($)'
          sex       =         'Patient Sex'
          patid     =         'Patient ID, Encrypted'
          spec      =         'Practitioner Specialty'
          zip_s     =         'Service Location ZIP Code'
          svcplace  =         'Place of Service'
          tos       =         'Type of Service'
          unitind   =         'Service Unit Indicator'
          f_mcr2nd  =         'Medicare is Secondary Payer (1/0)'
          mcrstat   =         'Medicare Status Code';

run;

*****************************************************************************

                          PART II-B
                        SHORTEN LENGTHS WHENEVER POSSIBLE
*****************************************************************************;
data out.&outset (drop = i begdate
                  compress = binary);
     length age1206_nr numdx units 3;
*     length rvu_i 4;
     length begday begmo birthdd birthmo claimpdd claimpmo endday endmo   $ 2;
     length begyr endyr birthyr claimpyr $ 4;
     set temp;

     *** remove formats from all  vars, add back just for date variables ***;
     format _all_;

     format begindt enddt clmpdate mmddyy10.;
run;
```

```
proc contents data=out.&outset;
    title2 "QC Checks";
run;

proc freq data = out.&outset;
     tables f_cpt /list missing;
     tables county_p/list missing;
     tables betosdet/list missing;
     tables spec/list missing;
run;

proc print data = out.&outset (obs = 100);
     var birthmo birthyr begindt age1206_nr;
run;

%jobend;

***** save log and list files *****;

dm "log;    print file = I:\MCdb\mcr_fn\programs_2006\mcr_1.log replace";
dm "output; print file = I:\MCdb\mcr_fn\programs_2006\mcr_1.lst replace";
```

```
%jobstart;

*****************************************************
Program : mcr_2.sas
Input   : mcr_1.sas7bdat
Output  : mcr_2.sas7bdat
Modified: Sireesha Thati
          February 6, 2007
*********************************************************************
Modified: Kasey Chen
          01/12/2007
Note:     1. Changed service year to 2006.
          2. Copied GPCI components from ENC_2 for 2006.
          3. Used same medicare fee schedule value
             (CF2006) as CF2005.
          4. Disabled libname library
             "I:\mcdb\datacol\formats\permanent".
          5. Compared medicare carrier numbers with Enc2 of year 2006
             and removed 2 medicare carrier numbers from year 2005.

*********************************************************************;

     /* Original Program Header */
     ****************************************************************;
     * Program: P020Step6d.sas                                      *;
     * Programmer: Adrien Ndikumwami/Chris Hogan                    *;
     * Date: September 25, 2001                                     *;
     * Adapted to 2002 MCDB on September 10, 2003-Adrien            *;
     *--------------------------------------------------------------*;
     *  This program assign RVUs (Work,Practice, and Malpractice) to     *;
     *  valid procedures. Medicare Payment Rates are also computed.      *;
     *  The components are estimated for procedures with imputed RVUs    *;
     *  The following format is needed to run the program:               *;
     *  Format_RVU_2002.sas                                              *;
     ****************************************************************;

    OPTIONS mprint nosource2 errors = 2       obs = max;

*  libname library "I:\mcdb\datacol\formats\permanent";
         libname library "I:\MCdb\datacol\Formats\PERMANENT\RVU_formats";
    libname IN "I:\MCdb\mcr_fn\sasdata";
    %let sasDTin = mcr_1;
    %let sasDTout= mcr_2;

    * rvu and payment formats ;
/*
options nosource2;
 %INCLUDE "I:\mcdb\datacol\Formats\hcpcsfmt.sas";
 %INCLUDE "I:\mcdb\datacol\Formats\cptfmt.sas";
 %INCLUDE "I:\MCdb\datacol\Formats\icd9pr_05.sas";
 %INCLUDE "I:\mcdb\datacol\Formats\revcode.sas";
 %INCLUDE "I:\mcdb\datacol\Formats\Format_RVU_2005.SAS";
 %INCLUDE "I:\mcdb\datacol\Formats\Format_RVU_2005_COMP.SAS";
options source2;
*/
         ***************************************************************************;
     * NOTES FOR UPDATING THE CODE TO PROCESS 2005 CLAIMS DATA                  *;
     * Since 2002, MHCC wants not only the total RVU, but also the individual   *;
     * components of the RVU (work, malpractice, practice expense) and,         *;
     * what is far more difficult, the actual Medicare price for the service.   *;
     * That requires:                                                           *;
     *  1) Adding the GPCI (geographic pricing) data by service                 *;
     *     location (ZIP code mapped to Part B carrier locality),               *;
     *  2) Gap filling that as required,                                        *;
     *  3) Adding the actual lab fee schedule amounts, which are separate       *;
     *     for each state                                                       *;
     *  4) Gap filling lab fees.                                                *;
     *  5) Assigning the 2005 RVU components.                                   *;
     *  6) Identifying modifiers that would change the Medicare price, and      *;
     *     either changing the Medicare price there or flagging it as not       *;
     *     the correct price                                                    *;
     *  7) Updating all the service-specific things like the price for          *;
     *     mammograms                                                           *;
     *     If we go back to redo older files, we have to re-create              *;
     *     all the gpci and lab fee schedule information for the earlier years  *;
     *     The code is not written to price out all of the pricing-relevant     *;
     *     modifiers under Medicare.These will typically be marked with F_ODDSVC*;
     *     values.  The user must know enough to throw out the records with     *;
     *     F_ODDSVC,and not to use the RVU or Medicare price data marked on     *;
     *     those records.                                                       *;
     *                                                                          *;
     *  Other aspects of Medicare payment are also not modeled.  For example,   *;
     *  the status T codes are only paid if they are the only thing the doc does*;
     *  for the patient on that day.  I have not done any such checking of      *;
     *  relationships among all procedures billed by a doc on a date for a      *;
     *  patient.                                                                *;
     ***************************************************************************;

     ********************************************************************;
     *1. Crosswalk provider zip to carrier locality                     *;
     *   Use zip of location of service (service location zip)          *;
     *   If that does not work, use beneficiary's zip as gap fill       *;
     *   If still no valid carrier locality, wait and put in imputed values  *;
     *   for the numerical values that you need, using the all-MD averages   *;
     *   rather than pick one carrier-locality and use that .             *;
     ********************************************************************;
```

```
 DATA in.&sasDTout. (compress = binary
                drop = ratio_mp ratio_pe ratio_w
                          ascgrp carrier carrloc cf2006 cpt_4_rvu
                          facbet facility
                          gpci_mal gpci_pe gpci_work
                          labcode newmod
                          rvu_fac rvu_imp rvu_lab rvu_nfac
                          rvu_pf rvu_pn tempval
                          mcr_stat

 );
  SET IN.&sasDTin.;

     *** medicare carrier and locality, concatenated ***;
     length carrloc $ 7 ;
     carrloc = put(zip_s,$zipcarr.) ;
     if carrloc = '' then carrloc = put(zip_p,$zipcarr.) ;

     *** if still missing, do not impute yet ***;
     *** add GPCI components based on that ***;
     gpci_work = input(put(carrloc,$g06work.),6.2); *medicare work gpci, 2006;
     gpci_pe   = input(put(carrloc,$g06pe.  ),6.2); *medicare practice gpci, 2006;
     gpci_mal  = input(put(carrloc,$g06mal. ),6.2); *medicare malpractice gpci, 2006;

     ******************************************************************;
     *  Note:                                                         *;
     * There are only a small fraction of records for which location info*;
     * is missing for 2006, once the missing provider locations have been*;
     * gap-filled with the beneficiary locations                      *;
     * We can gap-fill with the average GPCI values for Maryland, after  *;
     * running through this once                                      *;
     * After running the code once, the dollar-weighted means were    *;
     * computed  and plug in as the gap-fill values                   *;
     ******************************************************************;

     /* Year 2005 --> */ *if gpci_work = . then gpci_work = 1.0261;
     /* Year 2005 --> */ *if gpci_pe   = . then gpci_pe   = 1.0929;
     /* Year 2005 --> */ *if gpci_mal  = . then gpci_mal  = 0.9213;

/***** THE GPCI COMPONENTS BELOW HAVE TO BE CHANGED FOR 2007  *****/

     if gpci_work = . then gpci_work = 1.0000;     /* <-- Year 2006 */
     if gpci_pe   = . then gpci_pe   = 1.1082;     /* <-- Year 2006 */
     if gpci_mal  = . then gpci_mal  = 0.9217;     /* <-- Year 2006 */

     ******************************************************************;
     * Calculate Lab Fee Schedule values here                          *;
     * The program identifies which procedure codes are for lab/radiology *;
     * and assigns  payment rate for each lab codes based on the carrier. *;
     ******************************************************************;

     length labcode 3 ;
     labcode = input(put(cpt,$L06CODE.),4.);  ***temp flag to identify lab/rad codes*;

     if labcode = 1 then do ;

        ******************************************************************;
        * This block is ordered by state to speed execution, with MD,DC,VA   *;
        * being the first three states on the list. Note that ND and SD have *;
        * the same carrier but different payment rate. The assignment uses   *;
        * carrier location instead of carrier. $L00821B is the format for ND *;
        * and $L00822B is the format for SD(see Fomat_RVU_2005.sas           *;
        *  TEMPVAL is the temporary variable to hold pmt value for lab code. *;
        ******************************************************************;

        length carrier $ 5 ;
        carrier = substr(carrloc,1,5) ;     ***medicare carrier number;

        if      carrier = '00901' then tempval = input(put(cpt,$100901b.),6.2) ;
        else if carrier = '00903' then tempval = input(put(cpt,$100903b.),6.2) ;
        else if carrier = '00904' then tempval = input(put(cpt,$100904b.),6.2) ;
        else if carrier = '00902' then tempval = input(put(cpt,$100902b.),6.2) ;
        else if carrier = '00865' then tempval = input(put(cpt,$100865b.),6.2) ;
        ELSE IF CARRIER = '00883' THEN TEMPVAL = INPUT(PUT(CPT,$L00883B.),6.2) ;
        ELSE IF CARRIER = '00884' THEN TEMPVAL = INPUT(PUT(CPT,$L00884B.),6.2) ;
        else if carrier = '00831' then tempval = input(put(cpt,$100831b.),6.2) ;
        else if carrier = '00510' then tempval = input(put(cpt,$100510b.),6.2) ;
        else if carrier = '00520' then tempval = input(put(cpt,$100520b.),6.2) ;
        else if carrier = '00832' then tempval = input(put(cpt,$100832b.),6.2) ;
        else if carrier = '31140' then tempval = input(put(cpt,$131140b.),6.2) ;
        else if carrier = '31146' then tempval = input(put(cpt,$131146b.),6.2) ;
        else if carrier = '00824' then tempval = input(put(cpt,$100824b.),6.2) ;
        else if carrier = '00591' then tempval = input(put(cpt,$100591b.),6.2) ;
        else if carrier = '00590' then tempval = input(put(cpt,$100590b.),6.2) ;
        else if carrier = '00511' then tempval = input(put(cpt,$100511b.),6.2) ;
        else if carrier = '00833' then tempval = input(put(cpt,$100833b.),6.2) ;
        else if carrier = '00826' then tempval = input(put(cpt,$100826b.),6.2) ;
        else if carrier = '05130' then tempval = input(put(cpt,$105130b.),6.2) ;
        else if carrier = '00952' then tempval = input(put(cpt,$100952b.),6.2) ;
        else if carrier = '00630' then tempval = input(put(cpt,$100630b.),6.2) ;
        else if carrier = '00650' then tempval = input(put(cpt,$100650b.),6.2) ;
        else if carrier = '00660' then tempval = input(put(cpt,$100660b.),6.2) ;
        else if carrier = '00528' then tempval = input(put(cpt,$100528b.),6.2) ;
        else if carrier = '31143' then tempval = input(put(cpt,$131143b.),6.2) ;
        else if carrier = '31142' then tempval = input(put(cpt,$131142b.),6.2) ;
        else if carrier = '00953' then tempval = input(put(cpt,$100953b.),6.2) ;
        else if carrier = '00954' then tempval = input(put(cpt,$100954b.),6.2) ;
```

```
        else if carrier = '00740' then tempval = input(put(cpt,$l00740b.),6.2) ;
        else if carrier = '00523' then tempval = input(put(cpt,$l00523b.),6.2) ;
        else if carrier = '00512' then tempval = input(put(cpt,$l00512b.),6.2) ;
        else if carrier = '00751' then tempval = input(put(cpt,$l00751b.),6.2) ;
        else if carrier = '05535' then tempval = input(put(cpt,$l05535b.),6.2) ;
        else if carrloc='0082001' then tempval = input(put(cpt,$l00821b.),6.2) ; *ND;
        else if carrloc='0082002' then tempval = input(put(cpt,$l00822b.),6.2) ; *SD;
        else if carrier = '00655' then tempval = input(put(cpt,$l00655b.),6.2) ;
        else if carrier = '31144' then tempval = input(put(cpt,$l31144b.),6.2) ;
        else if carrier = '00805' then tempval = input(put(cpt,$l00805b.),6.2) ;
        else if carrier = '00521' then tempval = input(put(cpt,$l00521b.),6.2) ;
        else if carrier = '00834' then tempval = input(put(cpt,$l00834b.),6.2) ;
        else if carrier = '00801' then tempval = input(put(cpt,$l00801b.),6.2) ;
        else if carrier = '00803' then tempval = input(put(cpt,$l00803b.),6.2) ;
        else if carrier = '14330' then tempval = input(put(cpt,$l14330b.),6.2) ;
        else if carrier = '00522' then tempval = input(put(cpt,$l00522b.),6.2) ;
        else if carrier = '00835' then tempval = input(put(cpt,$l00835b.),6.2) ;
        else if carrier = '00973' then tempval = input(put(cpt,$l00973b.),6.2) ;

        else if carrier = '00880' then tempval = input(put(cpt,$l00880b.),6.2) ;
        else if carrier = '05440' then tempval = input(put(cpt,$l05440b.),6.2) ;
        else if carrier = '00900' then tempval = input(put(cpt,$l00900b.),6.2) ;

        else if carrier = '31145' then tempval = input(put(cpt,$l31145b.),6.2) ;
        else if carrier = '00836' then tempval = input(put(cpt,$l00836b.),6.2) ;
        else if carrier = '00951' then tempval = input(put(cpt,$l00951b.),6.2) ;
        else if carrier = '00825' then tempval = input(put(cpt,$l00825b.),6.2) ;

****** Removed from year 2005 ***************;
*       else if carrier = '00870' then tempval = input(put(cpt,$l00870b.),6.2) ;
*       else if carrier = '00910' then tempval = input(put(cpt,$l00910b.),6.2) ;
****** Removed from year 2004 ***************;
        ***ELSE IF CARRIER = '16360' THEN TEMPVAL = INPUT(PUT(CPT,$L16360B.),6.2) ;
        ***ELSE IF CARRIER = '16510' THEN TEMPVAL = INPUT(PUT(CPT,$L16510B.),6.2) ;
     end ;

     ************************************************************************;
     * Compute the 2005 Medicare rates using the 2006 conversion factor.     *;
     * Keep track of what type of code this is for payment calculation       *;
     * purposes.  Rather than sprinkle the payment calculation through the   *;
     * program, just flag what needs to be done and calculate payments at end*;
     * For example, GPCI does not apply to lab values, etc.                  *;
     ************************************************************************;

*    cf2005=37.8975;     /* 2005 medicare fee schedule conversion factor - was 37.3374 in 2004, a 1.5% increase */

     cf2006=37.8975;     /* 2006 medicare fee schedule conversion factor - same as 2005 conversion factor */

     length pay_calc $1 ;   *** flag for what payment calculation to use*;
     pay_calc = '0' ;       *** value of zero means do not calculate payment*;

     **Remember that the end of the loop is long way down there*;
     if f_cpt in ('2','3') then do;

     ******************************************************
     * Work On Modifiers For 26 And Blank For Radiology    *
     * Impute Them By Site Of Service                      *
     ******************************************************;

     length id_26_05 id_tc_05 id_26_06 id_tc_06 $ 2 newmod $ 2 ;
     label   id_26_05 = 'CPT for which mod of 26 appears MFS 2005'
             id_tc_05 = 'CPT for which mod of TC appears MFS 2005'
             id_26_06 = 'CPT for which mod of 26 appears MFS 2006'
             id_tc_06 = 'CPT for which mod of TC appears MFS 2006' ;

     id_26_05 = put(cpt,$y05_26m.) ;
     id_tc_05 = put(cpt,$y05_tcm.) ;
     id_26_06 = put(cpt,$y06_26m.) ;
     id_tc_06 = put(cpt,$y06_tcm.) ;


     newmod = 'XX' ;

     *Identify facility-based claims(IP,OP,ASC,SNF);

     if svcplace in ('01',' 1','1 ',
                     '02','2 ',' 2',
                     '21','22','23','24',
                     '31','34',
                     '51','52'
                    )
      then facility = 1 ;
      else facility = 0 ;

     ***************************************************************;
     * ONLY PROCESS THE CLAIMS WHERE 26 OR TC MODIFIER MIGHT APPLY  *;
     * special exceptions noted by inspection of the data           *;
     * 93555, 93556 are out of line -- what is the solution to that *;
     * ASSIGN BASED ON EXISTING MODIFIER OR ON PLACE OF SERVICE     *;
     * MENTION OF INDIVIDUAL CPTS BASED ON INSPECTION OF 1999 DATA  *;
         ***************************************************************;

     if id_26_05 = '26' or id_26_06 = '26' or
        id_tc_05 = 'TC' or id_tc_06 = 'TC' then do ;

        if cpt in ('93555','93556') then newmod = '26' ;
        else
```

```
   if mod1 = '26' or mod2 = '26' then newmod = '26' ;
   else
   if mod1 = 'TC' or mod2 = 'TC' then newmod = 'TC' ;
   else
   if facility = 1 then newmod = '26' ;
   else
   newmod = '  ' ;
end ;

if newmod not in ('26','TC') then newmod = '  ' ;


***DETERMINE Y2006 MEDICARE ASC RATE based on the file ASCPUF06 from CMS(HCFA)*;
length ascgrp $ 1 ;
if svcplace = '24' then  ascgrp = put(cpt,$y06_asc.) ;

*************************************************************************;
* ADD 2006 TRANSITIONED RVUS, AND PUT ON LAB FEES AS WELL:                *;
* FOR 2006, THIS ARE MORE COMPLEX.  MUST KEEP THE COMPONENTS OF THE RVU    *;
* FOR AN EVENTUAL ANALYSIS OF SECTION 805, AND FOR CALCULATION OF PAYMENTS *;
* ONLY USE THE MEDICARE RVUS WHEN MEDICARE USES THEM FOR PAYMENT,          *;
* INCLUDING THE OBSOLETE CODES (STATUS D) THAT ARE HELD OVER INTO          *;
* THE CURRENT YEAR.                                                        *;
*************************************************************************;

*cpt || pricing mod for rvu merge**;
length cpt_4_rvu $ 7 ;
cpt_4_rvu = left(trim(cpt))|| left(trim(newmod));


length rvu_w rvu_pf rvu_pn rvu_mp 4 ;
label rvu_w  = 'Medicare phys fee schedule RVU, work component'
      rvu_pf = 'Medicare phys fee schedul RVU, pract exp facility'
      rvu_Pn = 'Medicare phys fee schedul RVU, pract exp non-facility'
      rvu_mp = 'Medicare phys fee schedul RVU, malpractice' ;


length mcr_stat $ 1 ;
mcr_stat = put(cpt_4_rvu,$rvu06st.) ;
If mcr_stat in ('A','D','H','R','T','N') then do ;

   rvu_w  = INPUT(PUT(CPT_4_rvu,$rvu06w.),6.2);
   rvu_pf = INPUT(PUT(CPT_4_rvu,$rvu06pf.),6.2);
   rvu_pn = INPUT(PUT(CPT_4_rvu,$rvu06pn.),6.2);
   rvu_mp = INPUT(PUT(CPT_4_rvu,$rvu06mp.),6.2);

end ;

***************************************************************;
* NOTE THAT THE COMPONENTS WILL ALL BE MISSING FOR CODES OF THE *;
* WRONG STATUS. DO NOT SET INDIVIDUAL COMPONENTS TO MISSING.    *;
* THERE ARE NUMEROUS TRUE ZEROS AMONG THESE VALUES              *;
* BUT IT IS OK TO SET THE TOTALS (RVU_FAC, RVU_NFAC) TO MISSING *;
***************************************************************;

rvu_fac  = sum(rvu_w,rvu_pf,rvu_mp) ;
rvu_nfac = sum(rvu_w,rvu_pn,rvu_mp) ;

if rvu_fac le 0 then rvu_fac = . ;
if rvu_nfac le 0 then rvu_nfac = . ;

**** For lab rvu, use the Maryland lab fee schedule ;
*approx rvu for lab based on medicare lfs*;
rvu_lab = input(put(cpt,$100901b.),6.2) ;
if rvu_lab ne . then rvu_lab = rvu_lab/cf2006 ;
if rvu_lab le 0 then rvu_lab = . ;

rvu_imp = input(put(cpt_4_rvu,$rvuimp.),6.2);    *y2006 imputed rvu***;
* for hogan test run ;
*rvu_imp = input(put(cpt_4_rvu,gaprvu.),6.2) ;
if rvu_imp le 0 then rvu_imp = . ;

*Starting from 2002, mammography procedures are not edited***;
*Code for the edits was deleted in 2002*********************;

* IMPLEMENT RULE FOR MEDICARE RVUS BY SITE ;
if facility = 1 then do ;

  if svcplace = '24' then do ;

      if ascgrp ne '' then do ;
         Facility=1;
         rvu = rvu_fac ;
         rvu_pe=rvu_pf;
         pay_calc = '2' ; * value for pmt based on facility rvu ;
       end;
       else do ;
          Facility=0;
          rvu = rvu_nfac ;
          rvu_pe=rvu_pn;
          pay_calc = '3'; * value for non-facility rvu ;
       end ;

  end; *end of svcplace='24' do loop;
  else do ;
      pay_calc = '2' ;
      Facility=1;
```

```
      rvu = rvu_fac ;
      rvu_pe=rvu_pf;
  end ;
end ;  *end of facility='1' do loop;
else do ;
  Facility=0;
  pay_calc = '3' ;
  rvu = rvu_nfac ;
  rvu_pe=rvu_pn;
end ;


* SPECIAL MEDICARE RULE FOR THERAPY CODES ;
* THIS IS MY BEST GUESS AT THERAPY BILLED UNDER MEDICARE FEE SCHEDULE;

if cpt gt '97001' and cpt le '97456'  then    do ;
    Facility=0;
    pay_calc = '3' ;
    rvu = rvu_nfac ;
    rvu_pe=rvu_pn;
end ;
if rvu ne . then f_imprvu='0';        ****no-- rvu not imputed****;
rvu_i=rvu;

**impute lab rvus using medicare lab fee data**;
if rvu_i = . and rvu_lab ne . then do ;
      rvu_i = rvu_lab ;
      pay_calc = '4' ; * calc medicare rate based no lab fee schedule ;
      rvu_pe=rvu_lab;  *All lab rvu is set to pract.expense***;
      rvu_w=0;
      rvu_mp=0;
     end ;
else  if rvu_i = . and rvu_imp ne . then do ;
      rvu_i = rvu_imp ;
      pay_calc = '5' ; * calc based on imputed total rvu ;
end ;

if rvu_i ne . and rvu=. then f_imprvu='1';  *yes--rvu imputed;
if rvu_i=.    and rvu=. then f_imprvu='2';  *no--rvu not imputed -- no donor;

*******************Impute component RVUs**********;

length facbet $5;

facbet=Put(Facility,1.)||left(trim(betosdet));


Ratio_pe=input(put(facbet,$rat_pe.),6.4);
Ratio_mp=input(put(facbet,$rat_mp.),6.4);
Ratio_w=input(put(facbet,$rat_w.),6.4);

***New code starts here****;
*************************************************************;
** Set components to missing for procs with wrong status code**;
** (D=Deleted,N=Non covered services,R=Restricted services   **;
** Note: Some have good values, do this only when RVU=.      **;
** See above the statement [if rvu_fac le 0 then rvu_fac = .]**;
*************************************************************;
if rvu=. and mcr_stat in ('D','N','R')then do;
    rvu_mp=.;
    rvu_pe=.;
    rvu_w=.;
end;
***New code ends here****;

If rvu_i ne . and (rvu_w in (.,0) and rvu_mp in (.,0) and rvu_pe in (.,0)) then do; *open do loop 1;
    rvu_pe=Ratio_pe*RVU_i;
    rvu_mp=Ratio_mp*RVU_I;
    rvu_w=Ratio_w*RVU_i;

        *******if components still not imputed***********;
    if (rvu_w=. and rvu_mp=. and rvu_pe=.) then do;      *Open do loop 2;
              if betosdet in ("T1A","T1B") then do;   *open do loop 3;
         rvu_pe=rvu_i;  *All lab rvu_i set to pract. expense***;
         rvu_w=0;
         rvu_mp=0;
               end;                                   *end do loop 3;
               else do;   *Open do loop 4;
              *****use facility total ratios from Compute_Impute_RVU_Components_2006.lst;***;
                 if facility=1 then do;
                   rvu_pe= 0.3050*RVU_I;
                   RVU_mp= 0.0607*RVU_I;
                   RVU_w = 0.6343*RVU_I;
                 end;
                 else do;
                   rvu_pe = 0.5617*RVU_I;
                   RVU_mp = 0.0324*RVU_I;
                   RVU_w  = 0.4059*RVU_I;
                 end;
             end;       *Close do loop 4;
    end;                                          *Close do loop 2;
 end;                                                        *Close do loop 1;
************************************************************ ;
* NOW CALCULATE MEDICARE PAYMENTS BASED ON TYPE OF CODE AND ;
* PAYMENT CALCULATION FLAG ;
* VALUES FOR PAY_CALC ARE:
* 0 = DO NOT CALCULATE PAYMENT ;
```

```
     * 1 = THIS IS SCREENING MAMMOGRAPHY (PRE-2002) ;
     * 2 = THIS IS TO BE CALCULATED USING FACILTIY RVUS ;
     * 3 = THIS IS TO BE CALCUALTED USING NONFACILTIY RVUS ;
     * 4 = THIS IS A LAB FEE SCHEDULE CODE  ;
     * 5 = THIS CODE HAS AN IMPUTED TOTAL RVU BUT NO RVU COMPONENTS ;
     * 6 = (SEE BELOW) - THIS WAS FLAGGED AS AN ODD SERVICE -- DO NOT USE MEDICARE PMT RATE ;
     * ********************************************************** ;

     label pay_mcr = 'Prelim Calc MCR Rate-DO NOT USE AT THIS POINT' ;

     if pay_calc = '1' then do ; * screening mammography, no gpci-obsolete starting 2002;
         pay_mcr = rvu_fac*cf2006 ;
     end ;
     else if pay_calc = '2' then do ; * facilty rvus to be used ;
         pay_mcr = sum(gpci_work*rvu_w,gpci_pe*rvu_pf,gpci_mal*rvu_mp)*cf2006 ;
     end ;
     else if pay_calc = '3' then do ; * nonfacility rvus to be used ;
        pay_mcr = sum(gpci_work*rvu_w,gpci_pe*rvu_pn,gpci_mal*rvu_mp)*cf2006 ;
     end ;
     else if pay_calc = '4' then do ; * lab fee schedule -- no gpci adjustment ;
        pay_mcr = rvu_lab*cf2006 ;
     end ;
     else if pay_calc = '5' then do ; * imputed rvu, gap fill gpci using average weights ;
       pay_mcr = rvu_i*sum(gpci_work*.470,gpci_pe*.489,gpci_mal*.041)*cf2006 ;
     end ;

     /*************
     *NOTE: This section will be implemented after odd services are identified;

     IF PAY_CALC > 0 AND F_ODDSVC NE '' THEN DO ;
          PAY_MCR = . ;
          PAY_CALC = '6' ; * ODD SERVICE -- DO NOT USE MEDICARE PAYMENT RATE;
     END ;
     ********/

     if pay_mcr le 0 then pay_mcr = . ;

     drop   id_26_05 id_tc_05;

  end;         ****end of the big loop[if f_cpt in ('2','3')]*****;

  else f_imprvu='0'; **non cpt/hcpcs codes are classified as 0-no-- rvu not imputed**;


 run;


proc contents;
  title4 "Contents of Data Set &sasDTout.";
run;


%jobend;


***** save log and list files *****;

dm "log;    print file = I:\MCdb\mcr_fn\programs_2006\mcr_2.log replace";
dm "output; print file = I:\MCdb\mcr_fn\programs_2006\mcr_2.lst replace";
```

```
proc datasets kill;
run;

%jobstart;

option obs = max;

*****************************************************************************************
Program    : mcr_3.sas
Project    : Maryland, 1999 Pricing Report, Chris Hogan
Purpose    : Identify debit/credit pairs based on duplication of all financials in pnum/patient/
             service begin/service end/procedure code/provid
Input      : mcr_2.sas7bdat
Output     : mcr_3.sas7bdat
Programmer : Chris Hogan
Date       :
Modified By: Year 2000 Laurie Hamilton/John May to correct some inconsistencies
             in coding

             Year 2001 Laurie Hamilton 12/27/2002
             1. Create F_INANAL flag identifying records passing global
                analysis screens
             2. Subset only records passing global screens and 0 le age1201 le 64 screen

             Year 2002 Po-Lun Chou, October 21, 2003
             Modify this program to run on entire MCDB 2002 data set

             MCDB program modified to run on Medicare

Modified By: Sireesha Thati 01/04/2006
                    Made the program compatible by changing the macro variable year to 04.

Modified By: Sireesha Thati 02/06/2007
                    Made the program compatible by changing the macro variable year to 05.

Modified By: Kasey Chen 01/14/2008
                    Made the program compatible by changing the macro variable year to 06.
*****************************************************************************************
;

options linesize = 123 pagesize = 86 center errors = 2 pageno = 1 obs = max;

%let inset = mcr_2;
%let outset = mcr_3;
%let year = 06;
%let progname = mcr_3;

libname in  "I:\mcdb\mcr_fn\sasdata";
libname out "I:\mcdb\mcr_fn\sasdata";

%let asize = 25;

footnote1 "Program: I:\MCDB\MCR_FN\Programs_2006\&progname..sas";
footnote2 "%sysfunc(putn(%sysfunc(time()), hhmm.)) %sysfunc(putn(%sysfunc(date()), weekdate.))";

title1 "Creating Year 2006 Medicare Practitioner Report Analysis Set";
title2 "Adding Debit Credit Pair Flags, Applying Global Analysis Screens==>MCR_3";


**********************************
Sort and subset those potentially
debit/credit pairs
**********************************
;
data temp
;
     set in.&inset ;

     length recnum 8;
     label recnum = 'Sequential Record # (From Sample)';
     recnum = _n_;

run;

**************************
Begin ID of Pairs
**************************
;

*** non-capitated records only, with non-missing by variables ***;
data debcred1;
     set temp        (keep = patid
                             begindt enddt
                             cpt f_cpt
                             recnum
                             allow bill reimb patliab
                             provid fedtaxid
                      );


        If provid  ne ' ' and
           patid   ne ' ' and
           cpt     ne ' ' and
           begindt ne .   and
           enddt   ne .   and
           sum(bill,allow,reimb,patliab) ne 0 and
           sum(bill,allow,reimb,patliab) ne . then output;
```

```
run;

*** sort by pairing variables ***;
proc sort data = debcred1;
     by patid begindt enddt cpt provid;
run;


*** subset those by groups with more than one record ***;
data pairs1;
     set debcred1;
     by patid begindt enddt cpt provid;
     if not (first.provid and last.provid) then output;
run;

*title3 'Groups With > 1 Service/By Groupings';
*proc print data = pairs1;
*     by patid begindt enddt cpt provid;
*     id patid begindt enddt cpt provid;
*
*     var bill allow reimb patliab billtype;
*run;


*** identify those with more than 25 records/group and drop ***;
proc summary nway data = pairs1;
     class patid begindt enddt cpt provid;

     output out = stats1 (drop = _type_);
run;

data stats2;
     set stats1;
     if _freq_ gt 25;
run;

proc print uniform noobs data = stats2;
title3 'Records with more than 25 in a Group';
run;

data pairs25;
     merge pairs1 (in=a)
           stats2 (in=b)
;
     by patid begindt enddt cpt provid;

     if a and not b;
run;

******************************
Identify pairs to delete
******************************
;
data delprv&year (keep = recnum);
     set pairs25;
     by patid begindt enddt cpt provid;

     *** number of records for a by grouping ***;
     retain numrecs;

     *** declare character array for ennumrecs id to match back to master file ***;
     length recid1-recid&asize $ 8;
     array recids   recid1-recid&asize;
     retain recid1-recid&asize;

     *** arrays for financial variables ***;
     array allows   allow1-allow&asize;
     array bills    bill1-bill&asize;
     array patlbs   patlb1-patlb&asize;
     array reimbs   reimb1-reimb&asize;

     retain allow1-allow&asize;
     retain bill1-bill&asize;
     retain patlb1-patlb&asize;
     retain reimb1-reimb&asize;

     *** array for drop flag ***;
     array drop     drop1-drop&asize;

     retain drop1-drop&asize;

     *** first record in a by group ***;
     *** initialize the arrays ***;

     if first.provid then do;

        do i = 1 to &asize;
           allows[i] = .;
           bills[i]  = .;
           patlbs[i] = .;
           reimbs[i] = .;
           recids[i] = ' ';
           drop[i]   = 0;
           numrecs = 0;
        end;
```

```
     end; *** first.provid ***;

        *** increment the counter for number of records within the group ***;
        numrecs = numrecs + 1;
        *** store the financial data, keep ennumrecs id ***;
        allows[numrecs] = allow;
        bills[numrecs]  = bill;
        patlbs[numrecs] = patliab;
        reimbs[numrecs] = reimb;
        recids[numrecs] = recnum;


     *** check for debit/credit pairs ***;
     if last.provid then do;


        do i = 1 to (numrecs - 1);
           do j = (i + 1) to numrecs;

              *** if each record has no match yet ***;
              if drop[i] = 0 and
                 drop[j] = 0 then do;

                 *** flag if all financials cancel ***;
                 if allows[i] = -allows[j] and
                    bills[i]  = -bills[j]  and
                    patlbs[i] = -patlbs[j] and
                    reimbs[i] = -reimbs[j] then do;

                        drop[i] = 1;
                        drop[j] = 1;

                    end;
              end;
           end;
        end;
     end;


     *** output recnum id for matches, this will be merged back to the main set
         (which is already in this order) for the actual drop ***;
     do i = 1 to numrecs;
        if drop[i] = 1 then do;
        recnum = recids[i];
        output;
        end;
     end;
run;

proc sort data=delprv&year;
     by recnum;
run;

proc sort data=temp;
     by recnum;
run;

data out.&outset (compress = binary drop = recnum);
     merge temp (in = r)
     delprv&year(in=debcred)
;
     by recnum;

     label f_dcpair   ='Member of Debit/Credit Pair? (1/0)';

     if r and debcred then do;

        f_dcpair='1';
        output;
    end;
    else
    if r then do;
       f_dcpair='0';
       output;
    end;
run;

*************************
Checks on Assignment of
Debit/Credit Pair Flags
*************************
;
title3 'Count # Debit/Credit Pairs';
proc freq data=out.&outset;
     table f_dcpair /list missing;

run;

title3 'Pairs Only - All Financials Should Sum to 0';
proc means data = out.&outset n sum;
     where f_dcpair = '1';
     var allow bill patliab reimb;

run;

title3 'Debit Credit Pairs Print of 1st 200 Obs';
proc print data = out.&outset (obs = 200);
```

```
     where f_dcpair = '1';
     var begindt enddt cpt provid bill allow reimb patliab;
run;

proc contents data = out.&outset;
  title1 "Contents";
run;

%jobend;


***** save log and list files *****;

dm "log;    print file = I:\MCdb\mcr_fn\programs_2006\mcr_3.log replace";
dm "output; print file = I:\MCdb\mcr_fn\programs_2006\mcr_3.lst replace";
```

```
proc datasets kill;
run;


%jobstart;

*==============================================================================*
Program     : mcr_4.sas
Project     : Maryland Practitioner Report
Purpose     : Flag Odd Services
Input       : mcr_3.sas7bdat
Output      : mcr_4.sas7bdat

Programmer  : Chris Hogan
Date Created: December 21, 2001
Revised     : Laurie Hamilton
Date        : February 2002

              Year 2001 Revisions Laurie Hamilton 12/28/02
              1. Finalize medicare payment (PAY_MCR) by setting
              to missing of service is flagged "ODD"
              2. Drop all services flagged "ODD"

              Year 2002 Revisions, Po-Lun Chou, October 21, 2003
              Modify this program to run on entire MCDB


              Revised to run on Medicare
              Note: The usual _4 (add specialties) was not run for Medicare since we
              presume their specialty coding meedt Medicare standards

Modified By: Sireesha Thati 01/04/2006
                    Made the program compatible for year 2004.

Modified By: Sireesha Thati 02/06/2007
                    Made the program compatible for year 2005.

Modified By: Kasey Chen 01/14/2008
                    Made the program compatible for year 2006.

Modified By: Po-Lun Chou
Date:        May 1, 2008
Note:        Modified 2 intermediate variables COND04 & COND05, which were used
                    to generate variable F_ODDSVC (Odd Service Flag).
*==============================================================================*;

options obs = max linesize = 123 pagesize = 90 center errors = 2 pageno = 1;

libname in  "I:\mcdb\mcr_fn\sasdata";
libname out "I:\mcdb\mcr_fn\sasdata";
libname library "I:\MCdb\datacol\Formats\PERMANENT\RVU_formats";

%let inset  = mcr_3;
%let outset = mcr_4;
%let year = 06;
%let progname = mcr_4;

title1 "Creating Year 2006 Medicare Practitioner Report Analysis Set";
title2 "Flagging Odd Services";

footnote1 "Program: I:\MCDB\MCR_FN\Programs_2006\&progname..sas";
footnote2 "%sysfunc(putn(%sysfunc(time()), hhmm.)) %sysfunc(putn(%sysfunc(date()), weekdate.))";


proc format;
     value $paycalc
     0 = '0: Do not calculate payment'
     1 = '1: Screening mammography (pre-2002)'
     2 = '2: Calculate using facility rvus'
     3 = '3: Calculate using nonfacility rvus'
     4 = '4: Lab fee schedule code'
     5 = '5: Imputed total rvu but no rvu components'
     6 = '6: Flagged as an odd service-do not use MCR rate'
;
     value miss
     . = 'Missing'
     low-0 = '<= 0'
     other = '> 0'
;
run;


data temp;
     set in.&inset  /* (drop = facility) */;

**************************************************
Create PLAN
**************************************************
;

     length plan $ 6 ;
     label plan = 'Plan Type';
     if payertyp = '4' then plan = 'MCR';
     else plan = ' ';

*****************************************************
Create CPTX
```

```
5 character cpt code
*****************************************************
;

     length cptx $ 5 ;
     label cptx = 'Five Char CPT Code' ;
     cptx = substr(cpt,1,5) ;

*****************************************************
Create Flags to Identify Odd Services
COND01-COND15
*****************************************************
;

     label
     cond01  = 'Member of Debit-Credit Pair'
     cond02  = 'Odd/Unusual CPTs E.G. Rx, Dental)'
     cond03  = 'Payment <= 1, Allowed Amount <= 0'
     cond04  = 'Identified Anesthesia Outside Of Range'
     cond05  = 'Identified Facility Bill'
     cond06  = 'Minor Service Assoc With Major Service'
     cond07  = 'Adjustment Record'
     cond08  = 'Payer not NONHMO,HMOFFS,HMOCAP'
     cond09  = 'Not First Dupe by Key, Descending Payment'
     cond10  = 'Actuaries Trim, Payment, Low'
     cond11  = 'Actuaries Trim, Payment, High'
     cond12  = 'Actuaries Trim, RVU_I, Low'
     cond13  = 'Actuaries Trim, RVU_I, High'
     cond14  = 'Covoth_I = 2, Possible Secondary Payer'
     cond15  = 'RVU_I <= 0'
     ;

     length cond01-cond15 3;
     array acond cond01-cond15;
     do i = 1 to 15;
        acond[i] = 0;
     end;

******************************************************
COND01
Member of Debit/Credit Pair
*****************************************************
;

*     if f_dcpair eq '1' then cond01 = 1;

******************************************************
COND02
Odd/Dental Orothodontic CPT Codes
*****************************************************
;

    if (cptx = '00000')                      or
       (cptx = 'RX   ')                      or
       (cptx ge 'D0001' and cptx le 'D7420') or
       (cptx ge 'D8001' and cptx le 'D8999') then cond02 = 1;

*****************************************************
COND03
Allowed Amount <= 0 for Non-Capitated Services
or
Payment <= 1 for Non-Capitated Services
or
Bill Charge Only
*****************************************************
;

    if allow le 0 or
       payment le 1 then cond03 = 1;
    else
    if bill ne 0 and

       (allow = 0 and reimb = 0 and patliab = 0 and payment = 0)
       then cond03 = 1; ;

****************************************************
COND04
Identified Anesthesia Outside of Anesthesia CPT Range

NOTE: This also recodes BETOS in certain instances
*****************************************************
;

%let aneslist = 'AN', '47', 'P1', 'P2', 'P3', 'P4', 'P5', '30',
                'QX', 'QZ', '23', 'QK', 'QS', 'QY', 'G8', 'G9',
                'AA', 'AB', 'AC', 'AD', 'AE', 'AF', 'AG'
;

     *** first find all anesthesia procedures and flag ***;

     length anesth $ 1;
     label anesth = 'Anesthesia Procedure';
     anesth = '0';

     if (cptx ge '00100' and cptx le '01999')     or
        cptx in ('99100','99116','99135','99140') or
        unitind = '2'                             or
```

```
        spec in ('004','049')                    or
        mod1 in (&aneslist)                      or
        mod2 in (&aneslist)
     then anesth = '1';

     *** services that are legitimate anesthesia codes are ok,
         others have to be marked as odd services ***;

     if anesth   = '1' then do;
        if betosdet ne 'P0 ' then do;
           cond04 = 1;
           betosdet = 'P0 ';
           betosagg = '2';
        end;
     end ;

*****************************************************
Create NEWMOD Variable Which Will
be Used to Adjust Payer-Submitted
Modifiers

Note: NEWMOD is reset to XX as the various procedures
are inspected for different modifier imputations.
So, the value of NEWMOD at the end of the run relates
only to last imputation.
*****************************************************
;
      length newmod $ 2;
      label newmod = 'Imputed Modifier/Last Imputation';
      newmod = 'XX';

*****************************************************
COND06
Minor Services Associated With Major Services
*****************************************************
;

 length split assist bilat $1;

     *** minor/split surgery ***;
     if  mod1 in ('52','53','54','55','56','62','57','62','65') or
         mod2 in ('52','53','54','55','56','62','57','62','65') then split = '1';
     *** assistance at surgery ***;
     if mod1 in ('80','81','82') or
        mod2 in ('80','81','82')
            then assist = '1';

     *** bilateral/multiple procedures ***;
     if mod1   = '51' or
        mod2   = '51' then newmod = '51' ;

     if mod2   = '50' or
        mod2   = '50' then newmod = '50' ;

     if newmod = '50' or
        newmod = '51' then bilat  = '1';

     if split  = '1'  or
        assist = '1'  or
        bilat  = '1'  then cond06 = 1;

*****************************************************
COND07
Adjustment Record
*****************************************************
;

     *if billtype not in ('1', '8') then cond07 = 1;

*****************************************************
Impute NEWMOD
for Modifier 26 and Blank for Radiology
(Imputed by Site of Service)
*****************************************************
;

     LENGTH id_26_05 id_tc_05 id_26_06 id_tc_06 $ 2 ;
     LABEL ID_26_05 = 'CPT FOR WHICH MOD OF 26 APPEARS MFS 2005'
           ID_TC_05 = 'CPT FOR WHICH MOD OF TC APPEARS MFS 2005'
           ID_26_06 = 'CPT FOR WHICH MOD OF 26 APPEARS MFS 2006'
           ID_TC_06 = 'CPT FOR WHICH MOD OF TC APPEARS MFS 2006';

     ID_26_05 = PUT(CPT,$Y05_26M.) ;
     ID_TC_05 = PUT(CPT,$Y05_TCM.) ;
     ID_26_06 = PUT(CPT,$Y06_26M.) ;
     ID_TC_06 = PUT(CPT,$Y06_TCM.) ;

      NEWMOD = 'XX' ;

     *** identify facility-based claims using the same algorithm used to impute
         RVUs ***;

     length facility $ 1;
     label facility = 'Service Location is a Facility (1/0)';

     if svcplace in ('01',' 1','1 ',
                     '02','2 ',' 2',
```

```
                    '21','22','23','24',
                    '31',
                    '34',
                    '51','52')
     then facility = '1';
     else facility = '0';


    *** impute NEWMOD ***;
     ************************************************************
     *only process the claims where 26 or tc modifier might apply
     *special exceptions noted by inspection of the data
     *93544, 93545 are out of line -- what is the solution to that
     *
     * assign based on existing modifier or, barring that, on place of svc
     * mention of individual cpts based on inspection of 1999 data
     * not sure if this runs correctly (cpt versus cptx)
     **************************************************************
     ;

     IF ID_26_05 = '26' OR ID_26_06 = '26' OR
        ID_TC_05 = 'TC' OR ID_TC_06 = 'TC' THEN DO ;
        if cpt in ('93555','93556') then newmod = '26' ;
        else if mod1 = '26' or mod2 = '26' then newmod = '26' ;
        ELSE IF  MOD1 = 'TC' OR MOD2 = 'TC' THEN NEWMOD = 'TC' ;
        else if FACILITY = 1 THEN NEWMOD = '26' ;
        ELSE NEWMOD = '  ' ;
     END ;
     IF NEWMOD NOT IN ('26','TC') THEN NEWMOD = '  ' ;

*****************************************************
COND05
Facility Fees
(May still require match to tax ids to clean this up)
*****************************************************
;

     if (spec in ('090','091','092','093') or
         mod1 = 'SG'                      or

         mod2 = 'SG')                     or
        (newmod = 'TC')                   then cond05 = 1;

*******************************************************
COND08
Ambiguous Payer Type
*******************************************************
;

     if plan eq '' then cond08 = 1;

****************************************************
COND14
Possible Secondary Payer Involved
****************************************************
;

     *if covoth_i = '2' then cond14 = 1;

****************************************************
COND15
RVU <= 0
****************************************************
;

     if rvu_i le 0 then cond15 = 1;

****************************************************
Add a Variable to Speed Sorting in the Next Step
****************************************************
;
     length sortvar $ 100;
     length chardate $ 7;
     length odd $ 2;

     *** this (prematurely figures) out the equivalent of f_oddsvc so it can be part of the sort ***;
     array conds(15) cond01-cond15;

     condcnt= sum(of cond01-cond15);

     if condcnt=0 then odd=' ';
     else
     if condcnt gt 1 then odd = '99';
     else
       do j=1 to 15;
          if conds{j}=1 then do;
             odd=put(j,z2.);
          end;
       end;

     *** make date character ***;

     chardate = put(enddt,julian7.);
     sortvar = trim(patid) || trim(chardate) || trim(cptx) || odd || trim(newmod);


****************************************************
```

```
Enforce Consistency In Betos Where Possible
Make Minor Fixes In The Betos Coding Of Some Services

Move 1999 betos to 2000 betos
The betos on the 1999 file did not properly collect the dollars
    for flu vaccine and for chiropractic
What I do here is only overwrite for two categories: O1B, O1G
*****************************************************
;

      length newbetdt $ 3 ;
      length newbetag $ 1 ;
      label newbetdt = 'Detailed BETOS, Reclassified'
            newbetag = 'Aggregated BETOS, Reclassified';

      newbetdt = betosdet ;
      newbetag = betosagg ;

*******************************************************************
BETOS codes for vaccines:
These are codes that I believe ought to be included in childhood
immunizations, based on a project I did for Pennsylvania Medicaid,
and codes that Adrien had on his list of vaccines but that were local
(homegrown) codes that did not get passed into the betos category
*******************************************************************
;

     if cpt in (
     '90645',
     '90646',
     '90647',
     '90648',
     '90716',
     '90737',
     '90748',
     'G0010',
     'Q0158',
     '90714',
     'Y8003',
     'Y8004',
     'Y8091',
     'Y8093',
     'Y8094',
     'Z0700',
     'Z0701',
     'Z0702',
     'Z0707',
     'Z0712',
     'Z0713',
     'Z0718',
     'Z0720',
     'Z0731',
     'Z0737',
     'Z8091' ) then do ;
        newbetag = '8' ;
        newbetdt = 'O1H' ;
     end ;


*************************************************************
The BETOS codes for some key automated multichannel tests changed
between 1999 and 2000. Keep these codes in the automated multichannel
test category.

Note for next years run:  a lot of the old automated multichannel test
codes appear to be still in use.  These are 80003 on up.  We might bring
those back into the database next year .
*************************************************************
;

     if ('80053' le cpt le '80092') then do ;
        newbetdt = 'T1B' ;
        newbetag = '4' ;
     end ;

*******************************************************************
Laparascopic Cholecystectomies

HCFA misplaced these when the codes changed in 2000
these are the paired 1999 and 2000 codes
*******************************************************************
;

     if cpt in ('56324','46570',
                '56340','47562',
                '56341','47563',
                '56342','47564') then do ;
        newbetdt = 'P8G' ;
        newbetag = '2' ;
     end ;

***********************************************
Combine missing/misc aggregate betos categories
***********************************************
;

     if newbetag in (' ','6','7','N') then newbetag = 'X' ;
```

```
     if newbetdt in (' ','N/A')       then newbetdt = 'XXX' ;

run ;


*=====================================================================================================*
                                   COND09
                          FLAG DUPLICATES BY SERVICE DATE/CPT
                          KEEP ONLY HIGHEST PAYMENT RECORD
*=====================================================================================================*
;
proc sort data = temp
          out = temp2
          tagsort;
     by sortvar descending payment ;
run ;

data temp3 (drop = sortvar);
     set temp2;
     by sortvar descending payment ;
     if first.sortvar ne 1 then cond09 = 1;
run;


*=====================================================================================================*

                                IMPOSE THE HCFA ACTUARIES' TRIM

                    -- Screen out bills with allowed less than one third or more than three
                       times the mean allowed charge for the entire database
                    -- restrict to only good records (those which will be used in the pricing
                       analysis
                    -- Note: This will not catch major outliers in the miscellaneous codes with
                             Medicare status = 'C'
*=====================================================================================================*
;

*** temp3x is good records only, used for the means ***;

data temp3x;
     set temp3;
     if plan = 'MCR' and
        sum (of cond01-cond15) = 0 then output;
run;

proc sort data = temp3x ;
     by cptx newmod ;
run ;

*** use only valid records to calculate means *** ;

proc means noprint
           mean
           data = temp3x;
     var payment ;
     output out = cptsum mean = payment_ ;
     by cptx newmod ;
run ;

*** note asymmetric actuaries trim ***;
*** note this step also assigns condition flags cond10-cond13;

proc sort data = temp3;
     by cptx newmod;
run;

data temp4 (drop = _type_ _freq_ cpt_4_rvu mcr_stat);
     merge temp3
           cptsum;
     by cptx newmod ;

          *** re-create variables cpt_4_rvu & mcr_stat ***;

     length cpt_4_rvu $ 7  mcr_stat $ 1 ;
     cpt_4_rvu = left(trim(cpt))|| left(trim(newmod));
     mcr_stat = put(cpt_4_rvu,$rvu06st.) ;

     *** do not use odd services ***;

     if sum (of cond01-cond15) = 0 then ok = '1';
     if mcr_stat ne 'C'             and
        ok = '1'                    and
        plan = 'MCR' then do ;

        if payment < .33*payment_ then cond10 = 1;
        else
        if payment >   6*payment_ then cond11 = 1;

        if rvu_i gt 0 and payment < .25*37.8975*rvu_i then cond12 = 1; * cf2006 = 37.8975 ;
        if rvu_i gt 0 and payment >   8*37.8975*rvu_i then cond13 = 1;

     end ;
run;


*****************************************************************************************************
                              ASSIGN ODD SERVICE FLAG
```

```
                              GET STATS
                              DROP RECORDS FAILING ODD SERVICES SCREEN
*****************************************************************************************************
;

data out.&outset (compress = binary
                  drop = condcnt
                         cptx ok
                         id_26_05 id_26_06 id_tc_05 id_tc_06
                         cond01-cond15
                         ok payment_ i j
                  );
     set temp4;
     length f_oddsvc $ 2;
     label f_oddsvc = 'Odd Service Flag';

     array conds(15) cond01-cond15;

     condcnt= sum(of cond01-cond15);

     if condcnt=0 then f_oddsvc=' ';
     else
     if condcnt gt 1 then f_oddsvc='99';
     else
       do j=1 to 15;
          if conds{j}=1 then do;
             f_oddsvc=put(j,z2.);
          end;
       end;

     * odd service -- do not use medicare payment rate;
     if pay_calc > 0 and f_oddsvc ne '' then do ;
          pay_mcr = . ;
          pay_calc = '6' ;
     end ;

     label pay_mcr = 'Theoretical Medicare Payment';

run;


proc contents data = out.&outset;
run;

proc freq data = out.&outset;
     tables newbetag/list missing;
     tables newbetdt/list missing;
     tables betosdet * newbetdt /list missing;
     tables f_oddsvc/list missing;
run;


%jobend;

***** save log and list files *****;

dm "log;    print file = I:\MCdb\mcr_fn\programs_2006\mcr_4.log replace";
dm "output; print file = I:\MCdb\mcr_fn\programs_2006\mcr_4.lst replace";
```

```
proc datasets kill;
run;

%jobstart;


*==========================================================================================*
Program     : mcr_5.sas
Project     : Maryland
Purpose     : Assign f_demog flag, create new county var to be used in analysis
              After distribution, create region_p variable
Input       : mcr4.sas7bdat
Output      : mcr5.sas7bdat
Programmer  : Laurie Hamilton
Date Created: February 8, 2002
Modified    : For Year 2001 Data
              December 10, 2002 lsh
              1. Sex is now part of patient id, the only demographic variable distributed
                 will be patient residence county
              2. Flag demographic donor record
              3. Distribute residence info
              4. Output is an interim dataset analysis rather than final

              For Year 2002 Data
              Po-Lun Chou, October 21, 2003
              Modify this program to run on entire Medicare 2002 data set

Modified By: Sireesha Thati 01/04/2006
                                      Made the program compatible by changing the macro variable year to 04.

Modified By: Sireesha Thati 02/07/2007
                                      Made the program compatible by changing the macro variable year to 05.

Modified By: Kasey Chen 01/15/2008
                                      Made the program compatible by changing the macro variable year to 06.
*==========================================================================================*
;

*****************************************************************************

                                HOUSEKEEPING

*****************************************************************************
;
options linesize = 123 pagesize = 90 center errors = 2 pageno = 1 obs = max;

%let inset  = mcr_4;
%let outset = mcr_5;
%let year = 06;
%let progname = mcr_5;

libname in  "i:\mcdb\mcr_fn\sasdata";
libname out "i:\mcdb\mcr_fn\sasdata";

footnote1 "Program: I:\MCDB\MCR_FN\Programs_2006\&progname..sas";
footnote2 "%sysfunc(putn(%sysfunc(time()), hhmm.)) %sysfunc(putn(%sysfunc(date()), weekdate.))";

options nosource2;
%include "I:\mcdb\datacol\formats\mdregfmt.sas";
options source2;

title1 "Creating Year 2006 Practitioner Report Analysis Set";
title2 "Flagging Demographic Donor Record and Adding New County Variable==>MCR_5";

**********************************
Count Number Needing Demographics
Spread
**********************************
;
title3 '# Needing Demographics Spread';
proc summary data = in.&inset nway missing;
     class patid county_p;
     output out = stats1;
run;

data check1;
     set stats1;
     by patid;
     if not (first.patid and last.patid) then output;
run;


****************************************
Identify demographic donor record
For each unique id, this is the record
for the first date of service, either
capitated or not
****************************************
;
proc sort data = in.&inset out = tempsort tagsort;
     by patid begindt;
run;

data temp2;
     set tempsort;
     by patid;
     length holdcnty $ 3;
```

```
     retain holdcnty;

     *** create variable indicating record used to spread demographics -
         it will be the first analysis record if one is available,
         otherwise just the first record ***;

     length f_demog  $ 1;
     label f_demog = 'Demographic Donor for Analysis File? (1/0)';
     f_demog = '0';

     *** first record gets the demog donor flag, put demog vars
         in hold ***;

     if first.patid then do;
        f_demog = '1';
        holdcnty = county_p;
     end;

     *** all records get the hold values ***;

     length cnty_p2 $ 3;
     cnty_p2 = holdcnty;

     *** initialize the hold vars on the last record for a patid ***;
     if last.patid then do;
        holdcnty = ' ';
     end;

run;


**********************
Create Final Set
**********************
;
title3 ' ';
data out.&outset (compress = binary);
     set temp2 (drop = holdcnty);
run;

proc contents data = out.&outset;
run;

%jobend;

***** save log and list files *****;

dm "log;    print file = I:\MCdb\mcr_fn\programs_2006\mcr_5.log replace";
dm "output; print file = I:\MCdb\mcr_fn\programs_2006\mcr_5.lst replace";
```

```
proc datasets kill;
run;

%jobstart;
*==============================================================================
*Program     : m:\mcdb\medicare\buildmcr\mcr_f.sas
*Project     : Maryland
*Purpose     : Create final deliverable dataset and accompanying codebook and
*              stats documentation
*Input       : mcr_5.sas7bdat
*Output      : mcr03_v1.sas7bdat
*              mcr03s_1.sas7bdat
*Programmer  : Laurie Hamilton
*Date Created: December 8, 2000
*
*Revised     : 1/29/2001 lsh

*              drop state_p (all = MD) and f_payma (not needed)
*              Year 2000 Modifications
*              lsh 12/14/2001
*              1. Move codebook frequencies to another program
*              2. Create 10% sample in this run
*Notes       : Modified 1/10/03 to keep TOS, this is after delivery to
*              MHCC
*
*Modified    : Sireesha Thati, 1/5/2006
*Input                        : mcr_5.sas7bdat
*Output                       : mcr04_v1.sas7bdat
*              mcr04s_1.sas7bdat
*                                       Made the program compatible for this year
*
*Modified    : Sireesha Thati, 2/16/2007
*Input                        : mcr_5.sas7bdat
*Output                       : mcr05_v1.sas7bdat
*              This year we do not create the 10% sample medicare file
*
*Modified    : Kasey Chen, 01/15/2008
*Input                        : mcr_5.sas7bdat
*Output                       : mcr06_v1.sas7bdat
*Note:         1. Changed service year to 2006.
*              2. Replace variable 'AGE1206' to AGE1206_NR'.
*              3. Comment out AGE for year 2006 data.
*==============================================================================
;

%let indrive = i;
%let outdrive = i;
%let progdrv = i;

option obs = max;

libname in  "&indrive:\mcdb\mcr_fn\sasdata";
libname out "&outdrive:\mcdb\mcr_fn\sasdata";

%let inset   = mcr_5;
%let outset  = mcr06_v1;

*%let sampint = 4;

options nosource2;
options center errors=2 obs = max nodate pageno=1 missing = ' ' linesize = 123 pagesize = 90;

footnote1 "program: i:\mcdb\mcr_fn\programs_2006\mcr_f.sas";
footnote2 "%sysfunc(putn(%sysfunc(time()), hhmm.)) %sysfunc(putn(%sysfunc(date()), weekdate.))";

title1 "2006 Medicare Dataset";
title2 " ";

*** subset required variables and create deliverable set ***;

data out.&outset (label = '2006 Maryland Medicare Non-HMO'
                  drop  = person j dig01-dig14 hex01-hex14 int01-int14 /* sampler */
                          facility
                  compress = binary );
     set in.&inset (keep = f_baddt
                           /*age*/
                           age1206_nr
                           allow
                           bill
                           begday
                           begmo
                           begyr
                           betosdet
                           betosagg
                           birthmo
                           birthyr
                           claimpdd
                           claimpmo
                           claimpyr
                           county_p
                           dx1
                           dx2
                           dx3
                           dx4
                           f_agecat
                           f_county
                           endday
```

```
                        endmo
                        endyr
                        fedtaxid
                        f_finms
                        f_imprvu
                        f_modf
                        f_paymd
                        f_cpt
                        f_uncv
                        f_yr07
                        instate
                        f_serv_a
                        los
                        mod1
                        mod2
                        numdxc
                        payment
                        units
                        patliab
                        zip_p
                        payertyp
                        cpt
                        provid
                        reimb
                        rvu
                        rvu_i
                        sex
                        patid
                        spec
                        zip_s
                        svcplace
                    /*  tos           */
                        unitind
                        f_mcr2nd
                        clmpdate
                        begindt
                        enddt
                        f_age

                        f_patid
                        mcrstat
                        rvu_pe
                        rvu_mp
                        rvu_w
                        f_oddsvc
                        f_demog
                        cnty_p2
                        plan
                        newbetag
                        newbetdt
                        newmod
                        facility
                        pay_calc
                        pay_mcr
                        )
;

     *********************************
     Identify Records for the Sample
     *********************************
;
     length person $ 14 ;
     person = patid;

     length dig01 - dig14 $ 1 ;
     length hex01 - hex14 $ 2 ;
     length int01 - int14 8 ;

     array digits (*) dig01 - dig14 ;
     array hexes  (*) hex01 - hex14 ;
     array ints   (*) int01 - int14 ;

     do j = 1 to 14 ;
         digits(j) = substr(person,j,1) ;

         hexes(j)  = put(digits(j),$hex2.) ;
         ints(j)   = input(hexes(j),hex2.) ;
     end ;

   /**
     label sampler = 'sampling integer 0 to 9' ;
     sampler = sum(of int01 - int14) ;
     sampler = mod(sampler,10) ;

     *********************************
     Set the flag for records in the
     sample
     F_INSAMP
     1 = In the 10% Sample
     0 = Not in the 10% Sample
     *********************************
;
     f_insamp = '0';
     label f_insamp = 'Service in the 10% Sample (1/0)';

     if sampler = &sampint then do;
        f_insamp = '1';
```

```
     end;

   **/
     *********************************
     Adjust Labels
     *********************************
;
     label betosagg = 'BETOS Category, Aggregated'
           betosdet = 'BETOS Category, Detailed';

     ***********************************************
     Create ENCNT_ID
     This is a record number which will be used to
     link back to an original record by position if
     needed
     ***********************************************
;
     length encnt_id $ 8;
     encnt_id=put(_n_,z8.);
     label encnt_id = 'Sequential Record Number';

     *********************************
     Create Facility Flag
     *********************************
;
     length f_fac $ 1;
     label f_fac = "Service Performed in a Facility? (1/0)";
     if facility = '1' then f_fac = '1';
     else
     if facility = '0' then f_fac = '0';
     else
     f_fac = facility;

     ***********************************************
     change category for NEWBETDT and NEWBETAG
     ***********************************************
;
     if newbetdt = "O1H" then newbetdt = "O1G";
     if newbetag = "8" then newbetag = "X";

     label rvu_mp   = 'Medicare Phys Fee Schedule RVU, Malpractice'
           rvu_pe   = 'Medicare Phys Fee Schedule RVU, Pract Expense'
           rvu_w    = 'Medicare Phys Fee Schedule RVU, Work Component'
           f_imprvu = 'RVU Imputation Flag (1/0,2)'
           f_yr07   = 'Service Provided in 2006+ (1/0)'
           rvu_i    = 'Assigned/Imputed RVU'
           rvu      = 'Assigned RVU'
          /* f_insamp = 'Service in 10% Sample? (1/0)'*/
           age1206_nr  = 'Patient Age as of 12/31/2006, not rounded'
           cnty_p2  = 'County to Use for Pract. Report Analysis'
           pay_calc = 'Method for Calculating Medicare Payment'
           pay_mcr  = 'Theoretical Medicare Payment';

     *********************************
     Output
     *********************************
;
     output out.&outset;
    * if f_insamp = '1' then output out.&sampset;

run;

proc contents data = out.&outset;
  title3 "MCR06_V1";
run;
/*
proc contents data = out.&sampset;
  title3 "MCR06S_1";
run;
*/
%jobend;

***** save log and list files *****;

dm "log;    print file = I:\MCdb\mcr_fn\programs_2006\mcr_f.log replace";
dm "output; print file = I:\MCdb\mcr_fn\programs_2006\mcr_f.lst replace";
```

```
proc datasets kill;
run;

%jobstart;

*==================================================================================
* Program      : I:\MCdb\mcr_fn\programs_2006\codebook_mcr_all_06.sas
* Project      : Maryland
* Purpose      : Create documentation statistics, complete file for DOCUMENTATION
*==================================================================================
* Input        : prv04_v2.sas7bdat
* Output       : Mcrcball06_doc.sas7bdat
*                codebook_mcr_all_06.txt for import to excel
* Programmer   : Laurie Hamilton
* Date Created : December 21, 2001
* Revised      : 12/06/2001 lsh
*                this function was originally in mcdb_f.sas
* Modified by  : Po-Lun Chou
* Date         : January 9, 2005
* Note         : Modify and run this program based on the 2003 MCDB data set to
*                create the 2003 codebook for overall data set
* Modified by  : Sireesha Thati
* Date          : January 26, 2006
* Note          : Modify and run this program based on the 2004 MCDB data set to
*                 create the 2004 codebook for Overall Medicare data set
*
* Modified by  : Sireesha Thati
* Date          : February 16, 2007
* Note          : Modify and run this program based on the 2005 medicare data set to
*                 create the 2005 codebook.
*
* Modified by  : Kasey Chen
* Date          : 01/15/2008
* Note          : Modify and run this program based on the 2006 medicare data set to
*                 create the 2006 codebook.
*
* Modified by  : Po-Lun Chou
* Date          : May 5, 2008
* Note          : Re-created variable F_AGECAT here because the codes which used in
*                 MCR_1.sas is incorrect.
*
* Modified by  : Sanee Maphungphong
* Date          : June 3, 2008
* Note          : Modify program too use ODS to output an Excel file with proper format.
*                 Will no longer creating the TXT fiel.
*==================================================================================;

options center obs = max errors = 2 nodate pageno = 1 missing = ' ';

libname in  "I:\MCdb\mcr_fn\sasdata";
libname out "I:\MCdb\mcr_fn\sasdata";

%let inset  = mcr06_v1;
%let inset2 = mcr06_v2;
%let outset = mcrcball06_doc;

title1 "2006 MCDB (Private Payer)Medicare Dataset";
title2 "Frequencies for Codebook";

footnote1 "program: I:\MCdb\mcr_fn\programs_2006\codebook_mcr_all_06.sas";
footnote2 "%sysfunc(putn(%sysfunc(time()), hhmm.)) %sysfunc(putn(%sysfunc(date()), weekdate.))";

options nosource2;
%include "I:\mcdb\mcr_fn\programs_2006\mcrfmt.sas";
%include "I:\mcdb\mcr_fn\programs_2006\dxfmt.sas";
options source2;


*********************************************************************************
NOTE:
We re-created variable F_AGECAT here because the code in MCR_1.sas is incorrect.
We need to be sure to fix the code in 2007 data before we use it.
*********************************************************************************;

data in.&inset2;
  set in.&inset (drop = f_agecat);

     ********************************************
     PART II OPERATION
     Create valid age category flag (f_agecat)
     ********************************************;
     length f_agecat $1;

     if age1206_nr = .      then f_agecat='1';
     else
     if 0<=age1206_nr<65    then f_agecat='2';
     else
     if 65<=age1206_nr<=110 then f_agecat='3';
     else
     if age1206_nr > 110    then f_agecat='4';

  label f_agecat = "Age Category Code";
run;


*********************************************************************************
               PART I - CONTENTS TO PICK UP VARIABLE LABELS AND TYPE
```

```
              GET LIST OF CHARACTER VARS
*****************************************************************************;

proc contents data = in.&inset2
     out = tempcont (keep = name label type nobs);
run;

data tempcont;
     set tempcont;
     name = upcase(name);
          if name in ('FILE_ID') then delete;
run;

title4 'tempcont';
proc print data = tempcont;
run;

data charfmts;
     input @ 1 name $ 9.
           @14 fmt  $ 9.;
cards;
begday       $dayfmt.
begindt      dtfmt.
begmo        $monfmt.
begyr        $missyr.
betosagg     $betosa.
betosdet     $betosd.
birthmo      $monfmt.
birthyr      $byr.
claimpdd     $dayfmt.
claimpmo     $monfmt.
claimpyr     $missyr.
clmpdate     dtfmt.
cnty_p2      $cyfmt.
county_p     $cyfmt.
cpt          $rptmiss.
dx1          $icd9dx.
dx2          $icd9dx.
dx3          $icd9dx.
dx4          $icd9dx.
encnt_id     $encnt.
endday       $dayfmt.
enddt        dtfmt.
endmo        $monfmt.
endyr        $missyr.
f_age        $f_ageny.
f_agecat     $agecat.
f_baddt      $bddt.
f_county     $dzipf.
f_cpt        $f_proc.
f_demog      $demog.
f_fac        $f_fac.
f_finms      $ffinms.

f_imprvu     $f_imprv.
f_mcr2nd     $medsec.
f_modf       $f_modf.
f_oddsvc     $oddsvc.
f_patid      $f_patid.
f_paymd      $f_paym.
f_serv_a     $fserva.
f_uncv       $funcv.
f_yr07       $fyr.
fedtaxid     $rptmiss.
instate      $instate.
mcrstat      $mcrstat.
mod1         $modf.
mod2         $modf.
newbetag     $nbetosa.
newbetdt     $nbetosd.
newmod       $modf.
patid        $rptmiss.
pay_calc     $paycalc.
payertyp     $paygp.
plan         $plan.
provid       $rptmiss.
sex          $sex.
spec         $speclty.
svcplace     $srvplac.
unitind      $srvunit.
zip_p        $rptmiss.
zip_s        $rptmiss.
;
run;

data charfmts;
     set charfmts;
     name = upcase(name);
run;


*****************************************************************************
              PART II - STATS FOR NUMERIC VARIABLES
*****************************************************************************;

proc means data = in.&inset2
           maxdec = 2;
```

```
     output out = numstat;
run;

title4 'numstat';
proc print data = numstat;
run;

proc transpose data = numstat
               out = numstat2;
run;

data numstat2;
     set numstat2;
     _name_ = upcase(_name_);
run;

*** calculate nmiss by getting nobs from contents ***;

data _null_;
     set tempcont (obs = 1 keep = nobs);
     call symput ('totrecs', nobs);
run;

data numstat3 (keep = name n nmiss min max mean);
     set numstat2 (rename = (_name_= name
                             col1  = n
                             col2  = min
                             col3  = max
                             col4  = mean));
     nmiss = &totrecs - n;
     if name in ('_TYPE_', '_FREQ_', 'BEGINDT', 'ENDDT', 'CLMPDATE') then delete;
run;

title4 'numstat3';
proc print data = numstat3;
run;


*******************************************************************************
              PART III - STATS FOR DISCRETE VARIABLES
*******************************************************************************;

*** write code to do freqs ***;

filename out1 "I:\mcdb\mcr_fn\programs_2006\freqlist.sas";

data _null_;
     set charfmts end = eof;
     file out1
          noprint
          lrecl = 100;

     if _n_ = 1 then do;
         put "proc freq data = in.&&inset2 noprint;";
     end;

     put "     tables " name "/missing out = " name ";";
     put "     format " name " " fmt ";";

     if eof then do;
        put "run;" /;
     end;
run;

*** write code to combine freq output ***;

filename out2 "I:\mcdb\mcr_fn\programs_2006\combine.sas";

data _null_;
     set charfmts end = eof;
     file out2
          noprint
          lrecl = 100;

     if _n_ = 1 then do;
        put "data charfreq (keep = name code explcode count);";
        put "     set " name ";";
     end;
     else do;
        put "data " name " (keep = name  code explcode count);";
        put "     set " name ";" /;
     end;

     put "     length name $9;";
     put "     length code $100;";
     put "     length explcode $100;" /;

     put "     name = """ name """;";
     put "     hold1 = put(" name "," fmt ");";
     put "     code = scan(hold1,1,""?"");";
     put "     explcode = scan(hold1,2,""?"");"/;
     put "run;" /;

     if _n_ ne 1 then do;
        put "proc append base = charfreq;";
        put "run;";
     end;
```

```
run;


*** run code ***;

%include "I:\mcdb\mcr_fn\programs_2006\freqlist.sas";
%include "I:\mcdb\mcr_fn\programs_2006\combine.sas";

*** delete files ***;

options noxwait;
x "erase I:\mcdb\mcr_fn\programs_2006\freqlist.sas";
x "erase I:\mcdb\mcr_fn\programs_2006\combine.sas";


*****************************************************************************
               PART IV - COMBINE STATS
*****************************************************************************;

data allstat;
     set numstat3
         charfreq;
run;

proc sort data = allstat;
     by name;
run;

proc sort data = tempcont;
     by name;
run;

data out.&outset;
     merge tempcont (in = incont)
           allstat  (in = instat);
     by name;

    length usetype $ 9;
     if name in ('BEGINDT', 'ENDDT', 'CLMPDATE') then usetype = 'Date';
     else
     if type = '1' then usetype = 'Numeric';
     else
     usetype = 'Character';

     if name in ('BEGYR', 'ENDYR', 'BIRTHYR', 'CLAIMPYR') then
        explcode = 'Year = ' || code;
     else
     if name in ('BEGMO', 'ENDMO', 'CLAIMPMO', 'BIRTHMO') then do;
        if code not in ('01 thru 12', ' ') then explcode = 'Month = ' || code;
     end;
     else
     if name in ('BEGDAY', 'ENDDAY', 'CLAIMPDD', 'BIRTHDD') then do;
        if code not in ('01 thur 31', ' ') then explcode = 'Day = ' || code;
     end;
     else
     if name = 'patid_bd' then do;
        explcode = 'Birthdates Per PNUM/PATID=' || code;
     end;

     if not (first.name) then do;
        name    = ' ';
        label   = ' ';
        usetype = ' ';
     end;

run;

proc contents data = out.&outset;
run;


*---------------------------*
* Create Excel Table here   *
*---------------------------*;

option nocenter nosource2;

%include 'c:\_Excel_Tagset\excltags.tpl';

* Modify the default style;

proc template;
  define style newstyle;
    parent = styles.default;
    style Systemtitle / borderwidth=1 font_size=1 font_weight=bold;
    style Header / font_size=1 font_weight=bold;
    style RowHeader / font_size=1 font_weight=bold;
    style systemFooter / font_size=1 font_weight=bold;
    style Data / borderwidth=1;

    * display data with commas and no decimal places *;
    * CAUTION: large numbers may result in errors    *;
    *          for negative values, values should be less than 100 billion *;
    *          for positive values, values should be less than 100 billion *;
    style mp_comma from data / tagattr='format:#,#,#,#0';
```

```
    * display data with commas and two decimal places *;
    * CAUTION: large numbers may result in errors    *;
    *          for negative values, values should be less than 100 million *;
    *          for positive values, values should be less than 1 billion *;
    style mp_comma2 from data / tagattr='format:#,#,#,#0.00';

    * display data with leading zeros…6 characters *;
    style mp_z6 from data / tagattr='format:000000';

    * display data with leading zeros…3 characters *;
    style mp_z3 from data / tagattr='format:000';
  end;
run;

%let _ODSDEST = tagsets.ExcelXP;


ods tagsets.ExcelXP path = "I:\MCdb\mcr_fn\document"
                    file = "codebook_mcr_all_06.xls"
                    style = newstyle
                    options (orientation="landscape");

ods tagsets.ExcelXP options (sheet_name = "2006 MCR Doc");

Proc print data = out.&outset noobs label split="*";
   var NAME                                                                                                  / style={just=L};
   var LABEL                  / style={just=L};
   var usetype                / style={just=L};
   var code                   / style={just=L};
   var explcode               / style={just=L};
   var COUNT                  / style={tagattr='format:#,#,#,#0'};
   var nmiss                  / style={tagattr='format:#,#,#,#0'};
   var min                    / style={tagattr='format:#,#,#,#0'};
   var mean                   / style={tagattr='format:#,#,#,#0'};
   var max                    / style={tagattr='format:#,#,#,#0'};

   label NAME                                = "*VARIABLE*NAME"
         LABEL      = "*VARIABLE*DESCRIPTION"
         usetype    = "*VARIABLE*TYPE"
         code       = "*Variable*Value"
         explcode   = "*Value*Description"
         COUNT      = "*Number of*Services"
         nmiss      = "Number of*Missing*Value"
         min        = "*Minimum*Value"
         mean       = "*Mean*Value"
         max        = "*Miximum*Value";
title1 "Maryland Medical Care Database";
title2 "Data Description Summary";
title3 "Universe: Outpatient Medical Services";
footnote1 "I:\MCdb\mcr_fn\programs_2006\codebook_mcr_all_06.sas";
footnote2 "%sysfunc(putn(%sysfunc(time()), hhmm.)) %sysfunc(putn(%sysfunc(date()), weekdate.))";
run;

ODS tagsets.excelxp CLOSE;


/********

data _null_;
     set out.mcrcball06_doc;
     file "I:\MCdb\mcr_fn\programs_2006\codebook_mcr_all_06.txt"
          noprint
          lrecl = 1000;

     put name     $8.
         '09'X
         label    $50.
         '09'X
         usetype  $9.
         '09'X
         code     $50.
         '09'X
         explcode $100.
         '09'X
         count    comma12.
         '09'X
         nmiss    comma12.
         '09'X
         min      comma12.
         '09'X
         mean     comma12.2
         '09'X
         max      comma12.
         '09'X;
run;

*********/

%jobend;

*** save log and output files ***;

dm "log;    print file = I:\MCdb\mcr_fn\programs_2006\codebook_mcr_all_06.log replace";
dm "output; print file = I:\MCdb\mcr_fn\programs_2006\codebook_mcr_all_06.lst replace";
```