2006 Maryland Medical Care Data Base Documentation
(RX MCDB)

Prepared for
The Maryland Health Care Commission
(MHCC)

April 2008

Table of Contents

page

1. Overview 1
2. Input Data from Private Payers 1
3. Database Development Modules 3
4. Summary of New Variables Creation 5
   4.1 National Drug Codes (NDCs) 6
   4.2 Drug Classification 7
   4.3. Delivery Systems and Coverage Type 8
   4.4 Unique Patient Identifier 8
   4.5 Debit/Credit Flag 8
   4.6 Linkage Between Pharmacy Data and Provider Directories Containing Specialty Information 8
   4.7 Linkage Between Pharmacy and Encounter Data 9
   4.8 FDA Drug Approval Date and Brand/Generic Indicator 9
   4.9 Mail Order 9
   4.10 Pharmacy Grouping and Location 10
   4.11 Drug Supply and Adjusted Prescriptions 10
   4.12 Analysis Flags 10
   4.13 Mail Order Analysis Flags 12
5. The Final 2006 Pharmacy Database 13

Appendix A – Security and Data Confidentiality
Appendix B – SAS Programs Used in 2006 Pharmacy MCDB Development
Appendix C – Payer-specific Quality Review Statements for 2006
Appendix D – Codebook

2006 Maryland Private Payer Pharmacy Database Documentation

1. Overview

The objective of the pharmacy database development effort is to produce a file that can be used by analysts and researchers to conduct quantitative studies on topics such as policy, drug utilization, and pricing. To arrive at this objective, a number of quality-assurance mechanisms are imbedded in the data processing software. During this process, Social & Scientific Systems (SSS) creates new analysis fields. We examine the components of the National Drug Codes (NDC), evaluate the field data quality, and create drug classification fields. We investigate the possibilities of linking the pharmacy data to the Provider Directories by DEA Number and practitioner identifiers (Practitioner ID and Federal Tax ID) and to the encounter data by patient identifiers (Patient ID, Date of Birth, and Sex). We also correct some fields that payers incorrectly coded, and we remove from the database any records that do not meet the criteria for inclusion. In addition, we generate descriptive statistics for the reported fields to evaluate the quality of the submitted data. The data quality reports are shared with payers to help them correct the data and provide the Maryland Healthcare Commission (MHCC) with more complete and accurate submissions in future years.

2. Input Data from Private Payers

During July and early August 2007, 24 payers submitted pharmacy data to MHCC. For the 2006 submittal, payers were required to use the COMAR 10.25.06 Data Submission Manual Formatted for 2006 Medical Care Data Base due June 30, 2007, the same manual used to submit Encounter data. A few payers received time extension waivers from MHCC. . Four payers – New York Life Insurance Company (P600), Principal Life Insurance Company (P690), ReliaStar Life Insurance Company (P730), Transamerica Life Company (P770), and P890-American Medical Security Life Insurance Company (formerly United Wisconsin) received exception waivers from MHCC and did not submit their Pharmacy data. However, MHCC requested that New York Life Insurance Company reports the total number of covered recipients, total number of claims processed and total amount paid by each TPA. Table 1 lists private payers submitting 2006 pharmacy data and shows the number of services and recipients for both 2005 and 2006.

Table 1. Private Payers Submitting Pharmacy Data for 2006

Payer ID	Organization Name	Number of RX Services 2006	Number of RX Services 2005	Number of RX Recipients 2006	Number of RX Recipients 2005
P020	Aetna Life & Health Insurance Co.	1,348,142	1,023,951	103,592	90,019
P030	Aetna U.S. Healthcare	1,667,009	1,716,067	135,542	144,471
P070	American Republic Insurance Co.	8,002	10,271	424	631
P130	CareFirst-BCBS of DC, Inc.	6,212,172	5,782,492	491,125	464,789
P131	CareFirst-BCBS of MD, Inc.	2,495,115	3,380,629	164,792	188,661
P160	CIGNA Healthcare Mid-Atlantic, Inc.	1,254,002	1,279,565	101,276	104,196
P280	Assurant/Fortis Benefits Insurance Co.	17,002	13,909	2,374	1,815
P320	Golden Rule Insurance Co.	61,915	53,051	8,094	6,013

Payer ID	Organization Name	Number of RX Services 2006	Number of RX Services 2005	Number of RX Recipients 2006	Number of RX Recipients 2005
P325	Graphic Arts Benefit Corporation	41,833	61,802	4,854	4,753
P330	Great-West Life & Annuity Insurance Co.	191,125	191,870	17,249	17,069
P350	Guardian Life Insurance Co. of America	192,256	215,286	17,117	19,912
P471	Unicare Life & Health Insurance Co.	128,056	133,943	11,674	12,878
P480	Kaiser Foundation Health Plan of Mid-Atlantic	2,741,080	2,635,067	202,377	198,898
P500	MAMSI Life and Health Insurance Co.	793,031	896,487	62,954	82,567
P510	Maryland Fidelity Insurance Co.	281	96,232	106	12,219
P520	MD-Individual Practice Association, Inc.	1,462,989	1,460,113	95,312	100,513
P530	MEGA Life & Health Insurance Co.	82,342	72,627	9,661	8,716
P620	Optimum Choice, Inc.	2,382,713	2,654,683	191,911	223,329
P680	Coventry Healthcare of Delaware, Inc.	782,930	425,226	51,940	36,734
P760	State Farm Mutual Automobile Insurance Co.	50,195	49,985	2,825	2,741
P820	United Healthcare Insurance Co.	1,037,812	936,634	87,875	78,099
P830	Trustmark Insurance Co.	957	1,524	70	123
P850	Union Labor Life Insurance Co.	56,068	60,949	3,494	3,372
P870	United Healthcare of the Mid-Atlantic, Inc.	70,094	89,884	7,008	8,898
Total		**23,077,121**	**23,242,247**	**1,773,646**	**1,811,416**

Each payer submitted data for 24 fields as required by the 2006 data submission manual. Seven of the 24 fields were added from the Provider and the Encounter files. The three most common specialty fields were added from the Provider file and the Coverage Type and Delivery System fields were added from the Encounter file. Starting with the 2006 data, two additional fields are added from the Encounter file: Enrollment Date and Disenrollment Date. Although payers were asked to submit encrypted Patient IDs, we could not verify that each payer complied with this guideline. As a result, SSS re-encrypted the Patient ID field submitted by payers to ensure confidentiality. Table 2 shows the list of fields for which data was submitted by individual payers for the 2006 Pharmacy Database.

Table 2. Fields Collected from Individual Payers for the 2006 Pharmacy Database

Field Name	Description
1. DEANUM	Practitioner DEA Number
2. DRUGQTY	Drug Quantity (No. of Units Dispensed)
3. DRUGSPL	Drug Supply (Est. Number of Supply Days)
4. FILLDD	Day Prescription Filled
5. FILLMO	Month Prescription Filled
6. FILLYR	Year Prescription Filled
7. NCPDP	Pharmacy Number (NCPDP)
8. NDC	National Drug Code (NDC-11 Chars)
9. PATID	Patient ID, Encrypted
10. RBILL	Billed Charge-BC ($)
11. RBIRTHMO	Patient Birth Month
12. RBIRTHYR	Patient Birth Year

Field Name	Description
13. RCOVTYPE	Coverage Type (Submitted in the Encounter file)
14. RDELIVTYP	Delivery System (Submitted in the Encounter file)
15. RDENRDATE	Patient Disenrollment Date
16. RENRDATE	Patient Enrollment Date
17. RPATLIAB	Patient Liability-PL ($)
18. RPATZIP	Patient Residence ZIP Code
19. RREIMB	Reimbursement Amount-RA ($)
20. RSEX	Patient Sex
21. RSPECL1	Practitioner/Supplier Specialty-1 (Submitted in the Provider file)
22. RSPECL2	Practitioner/Supplier Specialty-2 (Submitted in the Provider file)
23. RSPECL3	Practitioner/Supplier Specialty-3 (Submitted in the Provider file)
24. RZIP_S	Pharmacy ZIP Code

3. Database Development Modules

Pharmacy Database development begins by reading in and processing individual payer files (in ASCII format) with records at the prescription service level. This raw data then goes through eight database development modules. These modules are designed to assess the quality of the data and produce a cleaner and enhanced Pharmacy Database. Eight functions of the database development modules are described below. Database development security issues are discussed in Appendix A.

Module 1. Convert Pharmacy Data to SAS Format

- Verify prescription drug services and recipient counts.
- Generate contents, frequencies, and a sample print to verify that raw data were converted correctly.

Module 2. Evaluate the Quality of the NDC Field and the Drug Class Field Provided in the Pharmacy Database

- Convert Multum Access files and create a Multum NDC lookup data set.
- Adjust the structure of the NDC code provided in payers' submissions to 5-4-2 configuration, if needed.
- Examine the quality of the NDC field in the Pharmacy database by performing three different links: by 11-digit NDC code (labeler, product, and package codes), by the first nine digits from the NDC code (labeler and product codes), by the first five digits from the NDC code (labeler code).

Module 3. Adjust Selected Pharmacy Fields

- Convert zoned decimal to real numbers.
- Remove embedded +/- signs from financials.

- Add Payer ID.
- Adjust alignment and zero-padding of selected variables.

Module 4. Flag and Perform Year-Specific Cleanup

- Perform additional division by 100 on the three financial fields if needed.
- Create flags to identify records for non-Maryland residents.
- Create flags to identify records where Date Filled is missing or invalid or provided before the collection period.
- Create a flag for all services where the three financial variables are equal to zero or missing.
- Create flags to identify outlier records based on financial fields.

Module 5. Check Pharmacy-to-Encounter and Pharmacy-to-Provider Linkages

- Determine if the Pharmacy file links to the encounter file by Patient identifier.
- Determine if the Pharmacy file links to the Provider file by Practitioner Identifier.

Module 6. Generate Delivery Statistics

- Generate diagnostic statistics and data completeness report.

Module 7. Create Analysis fields

- Create new fields by deriving them from existing fields.
- Create new fields by linking the Pharmacy file to the Encounter file, the Provider file, and Multum Lexicon Database.
- Create analysis flags.

Module 8. Generating the Codebook

- Generate the codebook of the final Pharmacy Database.

During Pharmacy Database development, errors encountered were tracked and reported to MHCC and payers for examination and correction (See Appendix C, Payer-Specific Quality Review Statements for 2006). Database development is an iterative process that allows payers to become familiar with identified errors and resubmit data in cases of severe data problems. The data processing programs are included in Appendix B. The most frequently found problems in the 2006 Pharmacy data submissions were:

- Payment is greater than bill.
- Invalid or missing NDC code.
- Invalid or missing NCPDP.
- Patient residence not MD.
- Invalid or missing Pharmacy ZIP Code.
- Incorrect coding of drug compound.
- DEA Number information missing, therefore no link to the Provider files to capture provider specialty information.
- Problems with linking Pharmacy data to the Encounter data by Patient ID.
- File layout inconsistent with layout specified in the Submission Manual.

Database development security issues are discussed in Appendix A.

4. Summary of New Variables Creation

This year marked the beginning of the Prescription Drug Report. For the purpose of this report and for other studies using the database, 43 new fields were created in addition to the 24 fields submitted by payers. To facilitate the analysis of the NDC code, nine fields were added from the Multum Lexicon database. Twenty flags were also created for the purpose of identifying or screening records during the analysis.

New fields created for inclusion in the 2006 Pharmacy Database are listed in Table 3.

Table 3. New Fields Created for the 2006 Pharmacy Database

Field Name	Description
1. APPRDATE	FDA Approval Date - From Multum Lexicon
2. CHNIND	Chain Indicator
3. DRUG_ID	Multum Drug ID - From Multum Lexicon
4. FILLDATE	Created at MHCC's Request from Month, Day, and Year Fields
5. GBO	Generic/Brand Indicator - From Multum Lexicon
6. MMDC	Main Multum Drug Code - From Multum Lexicon
7. MULTDC1	Multum Drug Category 1 - From Multum Lexicon
8. MULTDC2	Multum Drug Category 2 - From Multum Lexicon
9. MULTDC3	Multum Drug Category 3 - From Multum Lexicon
10. MULTDC4	Multum Drug Category 4 - From Multum Lexicon
11. NDCLABL	National Drug Labeler-Product Code (See Section 2.1)
12. NDCPACK	National Drug Package Code (see Section 2.1)

Field Name	Description
13. PIDBDG	Unique Patient ID (PNUM/PATID/BRTHYR/BRTHMO/SEX)
14. PNUM	Unique Payer Number
15. RAGE	Patient Age at Service
16. RAGE1206	Patient Age at 12/31/2006
17. RF_10K	Flag for payment > $10K
18. RF_ANAL	Analysis Flag
19. RF_COV17	Medicare-Coverage Flag
20. RF_DBC	Debit/Credit Flag
21. RF_DEA	DEA Number (1 - Valid, 0 -Other, 2 - Blank)
22. RF_ELINK	Pharmacy/Encounter Link Flag
23. RF_FINMS	Flag Identifying Records with all Financial Fields = 0 or missing
24. RF_MD	Maryland Residency Flag
25. RF_MEDSUP	Medical Supply Indicator ((0 - No, 1 - Yes)
26. RF_MOANAL	Mail Order Analysis Flag (1 - Analysis, 0 - Exclude)
27. RF_MORDER	Mail order Flag (0 - No, 1 - Yes)
28. RF_NDC	NDC Flag
29. RF_NDRUG	New Drug Flag
30. RF_NDUP	Duplicate Set with at Least One Negative Bill
31. RF_PATID	Patient ID Flag
32. RF_PAYMD	Flag for Payment Category
33. RF_PLINK	Pharmacy/Provider Link Indicator
34. RF_SELF	Self Insurance Coverage Indicator
35. RF_YR07	Flag Identifying Records for Claims in 2007
36. RF_YR65	Age 65+ Flag
37. RMSCAT	Medical supply category - From Multum Lexicon
38. RN_ZIP_S	Adjusted Pharmacy ZIP Code (Using NACDS Data)
39. RPAYMENT	Calculated Payment (patient liability + reimbursement amount)
40. RPHARMCTY	Pharmacy County
41. RPHARMGRP	Pharmacy Group
42. RSCRIPT	Number of Adjusted or 30-day Script
43. R_NAGE1206	Patient Age as of 12/31/2006 (Not Rounded)

4.1 National Drug Codes (NDCs)

The NDC is one of the most important fields in the database because it identifies the prescription drug filled for each record. Three components make up the configuration of an NDC. First, the labeler code identifies the firm that manufactures the drug. For example, the code for Bristol Myers Squibb Co is 00087. This code is assigned by the Food and Drug Administration (FDA). Second, the product code identifies a drug brand name. For example, the product code of Monopril tablets at Bristol Myers Squib is 0158. The last component is a package code to differentiate between the various packages in which the drug is dispensed. The product and the package codes are assigned by each firm. Two new fields derived from the NDC field were created during Pharmacy Database development. The first field is the Labeler-Product, which consists of the first 9 characters of the NDC code. The second field is the package code,

which consists of the last 2 characters of the NDC code. The fields were created only for NDCs that could be identified using the Multum Lexicon Database.

4.2 Drug Classification

Prior to the 2002 Pharmacy Database, three fields (DC1, DC2, and DC3) were added to the database. These fields contained the first three (of seven possible) FDA drug classification codes associated with the NDC code on each prescription service. Starting with 2002 Pharmacy Database, four new fields (MULTDC1, MULTDC2, MULTDC3, and MULTDC4) were added to the database. These fields contain all four Multum drug category codes associated with the NDC code on each prescription service. The Multum NDC database includes a therapeutic chemical classification scheme that groups drugs according to the organ or the system on which they act and/or their therapeutic and chemical characteristics. There are 17 main categories and 277 sub-categories. Some main categories are also broken down into intermediary categories. There are 32 intermediary categories. For example, polyenes, azole antifungals, miscellaneous antifungals, and echinocandins are classified as antifungals, and antifungals are in turn classified as anti-infectives. Each NDC is classified under one or more drug categories. About 13 percent of all NDCs belong to more than one category. To illustrate, Sodium Bicarbonate (NDC 00002202902) belongs to three categories: urinary pH modifiers, minerals and electrolytes, and antacids. Table 4 lists the 17 major drug categories.

Table 4. List of Multum Major Drug Categories

Category ID	Major Drug Category Name
1	Anti-Infectives
19	Antihyperlipidemic Agents
20	Antineoplastics
28	Biologicals
40	Cardiovascular Agents
57	Central Nervous System Agents
81	Coagulation Modifiers
87	Gastrointestinal Agents
97	Hormones
105	Miscellaneous Agents
115	Nutritional Products
122	Respiratory Agents
133	Topical Agents
218	Alternative Medicines
242	Psychotherapeutic Agents
254	Immunologic Agents
331	Radiololologic Agents

4.3 Delivery Systems and Coverage Type

Payers are required to provide delivery systems and coverage type information in the Medical Encounter Database only. However, there is a need to identify the same information for prescription service recipients. For this purpose, two fields were copied from the Encounter to the Pharmacy Database: Delivery Systems and Coverage Type. After linking the Pharmacy and the Encounter files by payer, patient ID, date of birth, and sex, delivery system and coverage type values were copied from the Encounter file to the Pharmacy file. Since a patient could have more than one delivery system or coverage type, the one with more expenditure per patient was copied.

4.4 Unique Patient Identifier

A special patient identifier was created to link the Encounter file and the Pharmacy file. The special patient identifier was created by concatenating four fields submitted by payers and one field assigned by SSS: Payer Number, the encrypted Patient ID, Birth Year, Birth Month, and Sex. The new unique patient identifier is also used to sort the final Pharmacy file.

4.5 Debit/Credit Flag

A number of payers have submitted data with negative financial fields, suggesting that these may be credit adjustments. However, there may be some payers who make adjustments but store them in their system as positive numbers. Without reliable bill type information, it is difficult to know if they are real adjustments or not. Whenever negative financials are provided, it is easy to identify full credit adjustments because one bill (positive) is cancelled by another bill (negative). Some payers also have records that are duplicated. A record is duplicated if more than one prescription (NDC) was filled the same day (Fill Date) for the same patient (Patient ID-Date of Birth-Sex).

A debit/credit flag RF_DBC was created to identify debit/credit problems. The flag RF_DBC for all records without duplicate fill (no similar prescription is filled the same day for the same patient) is set to 0. About 97 percent of all the records in 2006 had no debit/credit problems. The remaining records go through three passes. In the first pass, if two bills for the same patient, the same NDC, and the same fill date have positive and negative payments canceling each other, the flag RF_DBC is set to 1. Using the remaining non-flagged records in the first pass, if two records are complete duplicates, RF_DBC is set to 2 and if the records are partial duplicates, RF_DBC is set 3. In the last pass, all records without a flag in the second pass have their flag RF_DBC set to 4.

4.6 Linkage Between Pharmacy Data and Provider Directories Containing Specialty Information

Starting with the 2001 data submission, payers are required to include the DEA Number instead of the Practitioner ID. In addition, payers were required to add the DEA Number to the provider file to allow linkage between pharmacy and provider data. This linkage is necessary to capture specialty information

available in the Provider file (RSPECL1, RSPECL2, and RSPECL3). About 91 percent of all the records in the Pharmacy Database were successfully linked to the Provider file. There is a 17 percent improvement compared to 2005. This is mainly due CareFirst BlueChoice's submission. 2006 is the first year that this payer populated the DEA Number field for all records.

4.7 Linkage between Pharmacy and Encounter Data

Payers are required to submit Encounter and Pharmacy files that can be linked to each other by Patient ID. In previous years, the linkage rate between the pharmacy and the encounter data was very low. Payers have been notified to ensure that patient IDs are encrypted the same way in both the Pharmacy and the Encounter files and that they do link. The issue of linkage between the Pharmacy and the Encounter data was re-emphasized this year. Any payer with a low linkage rate was contacted for further explanation or resubmission if necessary. In the process of linking the Encounter and the Pharmacy files, two fields were added from the Encounter to the Pharmacy file: Coverage Type and Delivery Systems. In general, most patients are associated with one Delivery System or Coverage Type. However, it is possible for a patient to change Coverage Type or Delivery System in the course of the year. For patients with multiple Coverage Types or Delivery Systems, MHCC decided that the most frequently occurring Delivery System or Coverage Type should be added from the Encounter to the Pharmacy file. In addition, the definition of a patient has been expanded to Patient ID-Birthdate-Sex. This is the key that was used in the linkage. A linkage flag (RF_ELINK) was created to indicate records of patients who are in both the Pharmacy and Encounter file. Overall, 95 percent of records in the Pharmacy file have a link in the Encounter file.

4.8 FDA Drug Approval Date and Brand/Generic Indicator

Starting in 2003, the Pharmacy Database was linked to the Multum Lexicon Database to capture drug approval date and brand/generic indicator. The drug approval date is the date when a specific drug was approved by FDA. The brand/generic indicator designates which drug is branded and which one is generic. A brand name drug is produced and sold under the original manufacturer's brand name. A generic drug has exactly the same ingredients as a brand name drug but is produced under a generic name. Generic drugs are produced after a brand name drug's patent has expired. A new flag (RF_NDRUG) was created to differentiate between new and old brand drugs. New branded drugs are drugs approved in the last 4 years (2003 - 2006) and old branded drugs are the ones approved by FDA more than 4 years ago (before 2003).

4.9 Mail Order

Since 2003, the Pharmacy Database has included a flag (RF_MORDER) indicating prescription services dispensed from mail-order pharmacies. Major payers were asked to provide a list of mail-order pharmacies and their National Council for Prescription Drug Program (NCPDP) number.

This compiled list was then used throughout the database to identify prescription services filled via mail order. Overall, about 5 percent of all prescription drug services were filled through this channel.

4.10 Pharmacy Grouping and Location

Pharmacies are grouped based on data provided by the National Association of Chain Drug Stores (NACDS). The major groupings used are chain, independent, super market/mass market, mail order, and other non-Maryland pharmacies. In addition, the same NACDS data were used to fill in the pharmacy ZIP Codes where the payer did not provide the field value. About 2 percent of all records with missing pharmacy ZIP Codes were filled using NACDS data.

4.11 Drug Supply and Adjusted Prescriptions

Drug supply is the estimated number of days of dispensed supply. Drug supply is very important for the prescription drug analysis because it is used for normalizing or adjusting the number of prescriptions. We normalize the number of prescriptions because the same drug is filled in different quantities and the normalized number of prescription serves as a standard unit of study. One normalized or adjusted prescription is equal to a 30-day drug supply. A 60-day drug supply is equivalent to two normalized or adjusted prescriptions. The number of normalized or adjusted prescriptions is stored in a field called RSCRIPT. This field represents the number of adjusted 30-day scripts or normalized scripts. Any drug supply of less or equal to 30 counts as one 1 script. Any drug supply between 31 and 60 days counts as two scripts and any supply over 60 days counts as 3 scripts.

4.12 Analysis Flags

An analysis flag (RF_ANAL) was created to identify records to be used in the analysis file. The flag was created to identify records used for the various prescription drug use and expenditures reports. The following 11 criteria were used to exclude records from the analysis (excluded records are identified by RF_ANAL= 0).

1. Exclude records not filled in current year (2006). These records are identified by RF_YR07=1.
2. Exclude records for Non-Maryland patients. These records are identified by RF_MD=0. If a patient is associated with at least one Maryland ZIP Code, the patient is considered a Maryland resident.
3. Exclude records with all financial variables missing or zero. These records are identified by RF_FINMS=1.
4. Exclude debit/credit record pairs - either two claims with same NDC code, same fill date, equal dollar amounts (one positive, one negative) or partial debit/credit pairs. These records are identified by RF_DBC=1.
5. Exclude records in duplicate sets where at least one of the bills is negative. These records are identified by RF_NDUP=1.
6. Exclude remaining records with negative payment. These records are identified by RF_PAYMD=1.
7. Exclude outlier records (claims with total payment > $10,000). These records are identified by RF_10K=1.

8. Exclude records with invalid NDC codes (blank, 0s, and 9s). These records are identified by RF_NDC=2,3,4.
9. Exclude records with invalid patient IDs (blank, 0s, and 9s). These records are identified by RF_PATID=1.
10. Exclude records for patients that are 65 years old and older. These records are identified by RF_YR65=1.
11. Exclude records of patients covered by Medicare plans (Medicare Supplemental and Medicare+Choice). These records are identified by RF_COV17=1.

Table 5 shows the number of recipients as well as the number of prescription services included in the analysis data set. Per-payer ratios of recipients and prescription services kept in the analysis file compared to the whole Pharmacy Database are also shown. Overall, 62 percent of all prescriptions (or 80 percent of all patients) were included in the analysis data set.

Table 5. Number of Services and Recipients Included in the 2006 Pharmacy Analysis Dataset

Payer ID	Organization Name	Number of RX Recipients*	% of Total RX Recipients	Number of RX Services	% of Total RX Services
P020	Aetna Life & Health Insurance Co.	77,991	75.29	765,365	56.77
P030	Aetna U.S. Healthcare	112,655	83.11	1,089,017	65.33
P070	American Republic Insurance Co.	395	93.16	5,835	72.92
P130	CareFirst BlueChoice, Inc..	410,837	83.65	4,177,886	67.25
P131	CareFirst MD	123,197	74.76	1,446,320	57.97
P160	CIGNA Healthcare Mid-Atlantic, Inc	82,433	81.39	838,972	66.90
P280	Fortis Insurance Co.	1,916	80.71	12,145	71.43
P320	Golden Rule Insurance Co.	6,467	79.90	43,387	70.08
P325	Graphic Arts Benefit Corporation	4,116	84.80	30,177	72.14
P330	Great-West Life & Annuity Ins. Co.	13,865	80.38	123,942	64.85
P350	Guardian Life Insurance Co.	14,670	85.70	137,922	71.74
P471	Unicare Life & Health Insurance Co.	10,222	87.56	93,170	72.76
P480	Kaiser Foundation Health Plan of Mid-Atlantic	158,054	78.10	1,442,383	52.62
P500	MAMSI Life Insurance Co.	56,647	89.98	589,766	74.37
P510	Maryland Fidelity Insurance Co.	99	93.40	219	77.94
P520	MD Individual Practice Association, Inc.	84,195	88.34	993,409	67.90
P530	MEGA Life & Health Insurance Co.	7,854	81.30	59,028	71.69
P620	Optimum Choice, Inc.	169,679	88.42	1,746,116	73.28
P680	Coventry Healthcare of Delaware, Inc	42,851	82.50	404,728	51.69
P760	State Farm Mutual Automobile Insurance Co.	2,130	75.40	22,540	44.90
P820	United Healthcare Insurance Co.	67,111	76.37	658,123	63.43
P830	Trustmark Insurance Co.	59	84.29	557	58.20
P850	Union Labor Life Insurance Co.	2,869	82.11	30,653	54.67
P870	United Healthcare of the Mid-Atlantic, Inc	5,889	84.03	51,238	73.12
Total		1,456,201	82.10	14,762,898	63.97

**Based on patient ID, birth date, and gender information (PATID, RBIRTHMO, RBIRTHYR, RSEX).*

4.13 Mail Order Analysis Flags

In 2006, a mail-order analysis flag (RF_MOANAL) was created to identify records that would be used to conduct a study similar to the 2004 Maintenance Drug Prescriptions Mail Order Purchase Study as mandated by Senate Bill 885. The mail-order analysis file is different from the regular analysis file because the latter includes the population over 64 years old as well as patients covered by Medicare plan. The following nine criteria were used to exclude records from the mail-order analysis file (excluded records are identified by RF_MOANAL= 0).

1. Exclude records not filled in current year (2006). These records are identified by RF_YR07=1.
2. Exclude records for Non-Maryland patients. These records are identified by RF_MD=0. If a patient is associated with at least one Maryland ZIP Code, the patient is considered a Maryland resident.
3. Exclude records with all financial variables missing or zero. These records are identified by RF_FINMS=1.
4. Exclude debit/credit record pairs - either two claims with same NDC code, same fill date, equal dollar amounts (one positive, one negative) or partial debit/credit pairs. These records are identified by RF_DBC=1.
5. Exclude records in duplicate sets where at least one of the bills is negative. These records are identified by RF_NDUP=1.
6. Exclude remaining records with negative payment. These records are identified by RF_PAYMD=1.
7. Exclude outlier records (claims with total payment > $10,000). These records are identified by RF_10K=1.
8. Exclude records with invalid NDC codes (blank, 0s, and 9s). These records are identified by RF_NDC=2,3,4.
9. Exclude records with invalid patient IDs (blank, 0s, and 9s). These records are identified by RF_PATID=1.

Table 6 shows the number of recipients as well as the number of prescription services included in the mail-order analysis data set. Per-payer ratios of recipients and prescription services kept in the mail-order analysis file compared to the whole Pharmacy Database are also shown. Overall, 76 percent of all prescriptions (or 88 percent of all patients) were included in the mail-order analysis data set.

Table 6. Number of Services and Recipients Included in the 2006 Pharmacy Mail Order Analysis Dataset

Payer ID	Organization Name	Number of RX Recipients *	% Total RX Recipients*	Number of RX Services	% of Total RX Services
P020	Aetna Life & Health Insurance Co.	86,276	83.28	991,003	73.51
P030	Aetna U.S. Healthcare	120,200	88.68	1,257,715	75.45
P070	American Republic Insurance Co.	408	96.23	6,313	78.89
P130	CareFirst BlueChoice, Inc..	427,711	87.09	4,625,382	74.46
P131	CareFirst MD	153,176	92.95	1,899,831	76.14
P160	CIGNA Healthcare Mid-Atlantic, Inc	86,084	85.00	931,723	74.30

Payer ID	Organization Name	Number of RX Recipients *	% Total RX Recipients*	Number of RX Services	% of Total RX Services
P280	Fortis Insurance Co.	1,926	81.13	12,303	72.36
P320	Golden Rule Insurance Co.	6,523	80.59	44,624	72.07
P325	Graphic Arts Benefit Corporation	4,272	88.01	32,186	76.94
P330	Great-West Life & Annuity Ins. Co.	14,191	82.27	132,534	69.34
P350	Guardian Life Insurance Co.	15,021	87.75	145,601	75.73
P471	Unicare Life & Health Insurance Co.	10,538	90.27	100,166	78.22
P480	Kaiser Foundation Health Plan of Mid-Atlantic	181,195	89.53	2,027,476	73.97
P500	MAMSI Life Insurance Co.	58,387	92.75	630,375	79.49
P510	Maryland Fidelity Insurance Co.	102	96.23	227	80.78
P520	MD Individual Practice Association, Inc.	88,740	93.10	1,125,694	76.94
P530	MEGA Life & Health Insurance Co.	7,897	81.74	59,586	72.36
P620	Optimum Choice, Inc.	175,408	91.40	1,882,287	79.00
P680	Coventry Healthcare of Delaware, Inc	44,460	85.60	436,526	55.76
P760	State Farm Mutual Automobile Insurance Co.	2,575	91.15	34,320	68.37
P820	United Healthcare Insurance Co.	70,944	80.73	773,338	74.53
P830	Trustmark Insurance Co.	66	94.29	775	80.98
P850	Union Labor Life Insurance Co.	3,370	96.45	44,152	78.75
P870	United Healthcare of the Mid-Atlantic, Inc	6,000	85.62	53,426	76.24
Total		**1,565,470**	**88.26**	**17,247,563**	**74.74**

**Based on patient ID, birth date, and gender information (PATID, RBIRTHMO, RBIRTHYR, and RSEX).*

5. The Final 2006 Pharmacy Database

The final 2006 Pharmacy Database (SAS dataset: MCDBRX06.SAS7BDAT) contains 23,077,121 service-level records for 1,773,646 recipients from 24 private payers. The file size is about 5GB, and it contains a total of 67 fields. Table 7 shows the number of records and the number of unique recipients as reflected in the final 2006 Pharmacy MCDB.

The codebook and SAS contents for MCDBRX06 are shown in Appendix C.

Please note that records used for analysis can be subset from the overall Pharmacy file using the analysis flag RF_ANAL. By the same token, records used in the mail-order analysis can be subset from the overall Pharmacy file using the mail-order analysis flag RF_MOANAL. For either case, value of 1 means a record was included in the analysis file for 2006. The analysis file is also delivered as separate file (ANALRX06).

Table 7. Number of Services and Recipients Included in the Final 2006 Pharmacy Database

Payer ID	Organization Name	Number of RX Recipients*	Number of RX Services
P020	Aetna Life & Health Insurance Co.	103,592	1,348,142
P030	Aetna U.S. Healthcare	135,542	1,667,009
P070	American Republic Insurance Co.	424	8,002
P130	CareFirst BlueChoice, Inc..	491,125	6,212,172
P131	CareFirst MD	164,792	2,495,115
P160	CIGNA Healthcare Mid-Atlantic, Inc	101,276	1,254,002
P280	Fortis Insurance Co.	2,374	17,002
P320	Golden Rule Insurance Co.	8,094	61,915
P325	Graphic Arts Benefit Corporation	4,854	41,833
P330	Great-West Life & Annuity Ins. Co.	17,249	191,125
P350	Guardian Life Insurance Co.	17,117	192,256
P471	Unicare Life & Health Insurance Co.	11,674	128,056
P480	Kaiser Foundation Health Plan of Mid-Atlantic	202,377	2,741,080
P500	MAMSI Life Insurance Co.	62,954	793,031
P510	Maryland Fidelity Insurance Co.	106	281
P520	MD Individual Practice Association, Inc.	95,312	1,462,989
P530	MEGA Life & Health Insurance Co.	9,661	82,342
P620	Optimum Choice, Inc.	191,911	2,382,713
P680	Coventry Healthcare of Delaware, Inc	51,940	782,930
P760	State Farm Mutual Automobile Insurance Co.	2,825	50,195
P820	United Healthcare Insurance Co.	87,875	1,037,812
P830	Trustmark Insurance Co.	70	957
P850	Union Labor Life Insurance Co.	3,494	56,068
P870	United Healthcare of the Mid-Atlantic, Inc	7,008	70,077
Total		1,773,646	23,077,104

**Based on patient ID, birth date, and gender information (PATID, RBIRTHMO, RBIRTHYR, and RSEX).*

Appendix A

Security and Data Confidentiality

This Appendix to 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. Our 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.

Once received in the MPDC, data are stored only on the original media or on one of the secured MPDC systems for processing. The original media are placed in the MPDC and are not removed until they have been prepared for return to MHCC. At all times, media are located in the MPDC until SSS is ready to

deliver the media to MHCC. At that point, a special courier pickup is arranged in a joint effort with MHCC. Both SSS and MHCC will check the completeness of the deliverable packages.
Disaster Recovery

At the time of the initial installation of the PC operating system and standard software, an image of each workstation is created using Altiris, SSS' desktop management software. The retained image serves as a mechanism for quickly reconfiguring the workstations in the event of failure. In addition, a Windows XP backup has been performed on each system to be retained offsite for recovery in the event of disaster. To protect confidentiality and adhere to the security requirements, both the Altiris and Windows XP backups will occur prior to the introduction of data. As substantial system changes occur, incremental backups of the "system state" will be performed to facilitate rapid system recovery.

The raw data will be retained in the format in which they were first received and stored in a strictly controlled locked room remote to the workstations and server. Additionally, other files necessary to rebuild the systems will be backed up to SDLT 320 cartridges and retained in the same remote storage facility.

In the event of a system failure, a replacement system will be acquired and quickly rebuilt with the retained system images. Upon rebuild, any data relevant to the failed system will be restored from tape. In the event of a more catastrophic failure, the replacement systems can quickly be rebuilt from the offsite backups at the remote office, if necessary.

Encryption of the Patient Identifier

The IDs on the Medical Services and Prescription Drug databases are encrypted by the payers prior to shipping to the MPDC. SSS will re-encrypt the IDs during the database development stage.

Physical Security of the Facilities

All Maryland data processing workstations and printers are secured in the MPDC with a cipher lock. Access to the room is limited to the key Maryland project personnel. The server resides in a separate cipher-locked, climate-controlled room. Both the server and the workstations are attached to a 100Mbs switch accessible only to the MPDC and segregated from SSS' enterprise.

Equipment Used in Data Collection

The workstations in the MPDC have all been installed with the Windows XP Professional Operating System. Each of these systems is configured with the NTFS file system to support assignment of directory- and file-level permissions. Additionally, each system is a member of a dedicated, standalone Windows XP Domain, specifically created and maintained for the MPDC.

The workstations dedicated to MCDB processing include a Pentium 4 3.0 GHz containing 2GB or RAM; two Pentium 4, 1.8Ghz; one Pentium 4, 1.6Ghz; and one Pentium 3, 500MHz workstations. The MPDC has also dedicated two Pentium 3, 500Mhz workstations, specifically to SHEA processing and a Pentium II workstation for in-processing functions. Attached to the in-processing workstation is an HP SureStore DAT8, an Exabyte EL820E, and a QualStar/Fujitsu 3490E tape drive, allowing for restoration of data from diverse sources to be used in the project.

SSS uses a Dell PowerEdge 600SC 1.8Ghz server running Windows XP Server for data storage, authentication services, and data backups. It has a formatted storage capacity of 756 GB. Attached to the PowerEdge server is a Quantum DLT 7000 Tape Drive supporting the DLT IV tape format. As noted,

Windows XP Active Directory is used for authentication and access control. Through the default group policy, password restrictions and lockout settings are enforced as specified in Maryland's "Computerized Record System Security Requirements" and "HIPAA Information Technology Security Standards."

An HP LaserJet 4200 DTN is installed in the MPDC for dedicated support of report processing and other hardcopy deliverables. A shredder is also located in the MPDC for immediate shredding of confidential materials.

Appendix B

SAS Programs Used In 2006 Pharmacy MCDB Development

```
*==============================================================================
*
*  Program Name:  i:\mcdb\datacol\step2_3pgm02\step3pgm02.sas
*
*  Description:   sasify health care provider data base
*                 include in each payer's pxxxsasprov.sas program.
*
*  Input files:   n/a
*
*  Output files:  n/a
*
*  DATE CREATED:  07/06/00 by N. Odaka
*  DATE MODIFIED: 07/10/00 by N. Odaka
*                 07/11/00 by N. Odaka  Add code to zero-fill left-adjusted values.
*                 08/25/00 by N. Odaka  After consultation with R. Hu, create
*                                       unique id for provider record. Check if
*                                       specialty code is valid and unique.
*                 08/30/00 by N. Odaka  Increase specialty variables SPECL from
*                                       15 to 84.
*==============================================================================
;
  ***************************************************************;
  *  Create unique provider record ID which will be used to keep ;
  *  the provider data in the same order as they were submitted. ;
  ***************************************************************;
  PRV_ID=put(_N_,z8.);

  *** check all the variables for special characters ***;
  if compress(PROVID,"&CHAR_LST")    ne '' then do;
     vstr(1)=trim(left(vstr(1)))||compress(PROVID,"&CHAR_LST");
     vcnt(1)+1;
  end;
  if compress(FEDTAXID,"&CHAR_LST")  ne '' then do;
     vstr(2)=trim(left(vstr(2)))||compress(FEDTAXID,"&CHAR_LST");
     vcnt(2)+1;
  end;
  if compress(LNAME,"&CHAR_LS3")  ne '' then do;
     vstr(3)=trim(left(vstr(3)))||compress(LNAME,"&CHAR_LS3.,");
     vcnt(3)+1;
  end;
  if compress(FNAME,"&CHAR_LS3")  ne '' then do;
     vstr(4)=trim(left(vstr(4)))||compress(FNAME,"&CHAR_LS3.,");
     vcnt(4)+1;
  end;
  if compress(SPECLTY1,"&CHAR_LS2")      ne '' then do;
     vstr(5)=trim(left(vstr(5)))||compress(SPECLTY1,"&CHAR_LS2");
     vcnt(5)+1;
  end;
  if compress(SPECLTY2,"&CHAR_LS2")  ne '' then do;
     vstr(6)=trim(left(vstr(6)))||compress(SPECLTY2,"&CHAR_LS2");
     vcnt(6)+1;
  end;
  if compress(SPECLTY3,"&CHAR_LS2")  ne '' then do;
     vstr(7)=trim(left(vstr(7)))||compress(SPECLTY3,"&CHAR_LS2");
     vcnt(7)+1;
  end;
  if compress(DEANUM,"&CHAR_LS2")  ne '' then do;
     vstr(7)=trim(left(vstr(7)))||compress(DEANUM,"&CHAR_LS2");
     vcnt(7)+1;
  end;
  if compress(MIDINIT,"&CHAR_LS2")  ne '' then do;
     vstr(8)=trim(left(vstr(8)))||compress(MIDINIT,"&CHAR_LS2");
     vcnt(8)+1;
  end;
  if compress(SUFFIX,"&CHAR_LS2")  ne '' then do;
     vstr(9)=trim(left(vstr(9)))||compress(SUFFIX,"&CHAR_LS2");
     vcnt(9)+1;
  end;
  if compress(CREDENT,"&CHAR_LS2")  ne '' then do;
     vstr(10)=trim(left(vstr(10)))||compress(CREDENT,"&CHAR_LS2");
     vcnt(10)+1;
```

```
  end;

  *** create report ***;
  if eof then do;
    *create report header;
    if sum(of vcnt1-vcnt10)>0 then do;
       file print;
       put;put;put;;
       put 'The following variable(s) have some characters not 0-9, a-z, A-Z or _';
       put;put;put;
    end;
    do i = 1 to 10;
       if vcnt(i)>0 then do;
          pvcnt=put(100*vcnt(i)/_N_,5.2)||'%';
          put i pvname. ' - ' vcnt(i) 5. ' records or ' pvcnt ' with special characters = '
vstr(i);
       end;
    end;
  end;

  LABEL
     PROVID  ='1. Practitioner/Supplier ID'
     FEDTAXID='2. Practitioner/Supplier Federal Tax ID'
     LNAME   ='3. Practitioner/Supplier Last Name'
     FNAME   ='4. Practitioner/Supplier First Name'
     SPECLTY1='5. Practitioner/Supplier Specialty-1'
     SPECLTY2='6. Practitioner/Supplier Specialty-2'
     SPECLTY3='7. Practitioner/Supplier Specialty-3'
     PRV_ID  ='Unique identifier for provider record'
     DEANUM  ='8. Practitioner DEA Number'
     MIDINIT ='9. Practitioner/Supplier Middle Initial'
     SUFFIX  ='10. Practitioner/Supplier Suffix'
     CREDENT ='11. Practitioner/Supplier Credential'
    ;
run;


*****************************************************************;
*    2) Keep unique records for each key variable combination      ;
*****************************************************************;
PROC SORT DATA=SASPROV out=&OUTFNAME nodupkey;
  BY PROVID FEDTAXID DEANUM SPECLTY1 SPECLTY2 SPECLTY3;
RUN;

%LET SPECLST='','.','0','000';
%LET CHAR_LS4=0123456789;

*****************************************************************;
*    3) Resort the data by PRV_ID within PROVID/FEDTAXID/DEANUM pair.  ;
*****************************************************************;
PROC SORT DATA=&OUTFNAME;
  BY PROVID FEDTAXID DEANUM PRV_ID;
RUN;

*************************************************************************;
*    4) Create unique PROVID/FEDTAXID pair with specl1-specl9.             ;
*       (Aggregate all the specialty code for each PROVID/FEDTAXID pair)   ;
*************************************************************************;
data out1.&OUTFNAME (drop=i j k not_ok speclty1 speclty2 speclty3 ZF1 ZF2 ZF3
                     SPECL10-SPECL84);

length fname $40 lname $40 provid $13;
  set &OUTFNAME;
  BY PROVID FEDTAXID DEANUM;
  length specl1-specl84 $3 scnt 3.;
  retain specl1-specl84 '' scnt 0;
  label
     scnt = '# of specialties for each PROVID'
     SPECL1='Practitioner/Supplier Specialty-1'
     SPECL2='Practitioner/Supplier Specialty-2'
     SPECL3='Practitioner/Supplier Specialty-3'
```

```
     SPECL4='Practitioner/Supplier Specialty-4'
     SPECL5='Practitioner/Supplier Specialty-5'
     SPECL6='Practitioner/Supplier Specialty-6'
     SPECL7='Practitioner/Supplier Specialty-7'
     SPECL8='Practitioner/Supplier Specialty-8'
     SPECL9='Practitioner/Supplier Specialty-9'
  ;

  array _in  (3)  $ speclty1 speclty2 speclty3;
  array _zf  (3)    zf1-zf3;
  array _out (84) $ specl1-specl84;

%macro chkspecB;
  do k=1 to 3;
  *******************************************;
  *  Convert to Z3.0 format.                ;
  *******************************************;
    if _in(k) notin (&SPECLST) and compress(_in(k),"&CHAR_LS4")='' then do;
      _zf(k) = input(_in(k),3.);
      _in(k) = put(_ZF(k),Z3.0);

  *******************************************;
  *  Check if valid specialty code.         ;
  *******************************************;
      if (0< _zf(k) le 63) or (70 le _zf(k) le 76) or (80 le _zf(k) le 85) or
         (90 le _zf(k) le 93) or (100 le _zf(k) le 103) then do;
        if scnt = 0 then do;
          _out(scnt+1) = _in(k);
          scnt = scnt + 1;
        end;

  *******************************************;
  *  Check if unique specialty code.        ;
  *******************************************;
        else if scnt > 0 then do;
          not_ok=0;
          do j=1 to scnt;
            if _in(k) = _out(j) then not_ok=1;
          end;
          if not_ok=0 then do;
            _out(scnt+1) = _in(k);
            scnt = scnt + 1;
          end;
        end;
      end;
    end;
  end;
%mend chkspecB;

  if first.deanum then do;
     scnt=0;
     do i=1 to dim(_out);
       _out(i)='';
     end;
  end;

  %chkspecB;

  if last.deanum then output;
run;

*******************************************;
*  5) Print out # of records imported      ;
*******************************************;
data _null_;

  *** # of unique pairs from imported PROV.TXT file ***;
  sid=open('work.sasprov');
  nlobs=attrn(sid,'NLOBS');
  file print;
  put;put;put;put;
```

```
  put "           " nlobs "Records imported for the P%substr(&PAYERNUM,6,3)PRV.txt.";

  *** # of unique pairs from imported PROV.TXT file ***;
  sid=open("out1.&OUTFNAME");
  nlobs=attrn(sid,'NLOBS');
  file print;
  put;put;put;put;
  put "           " nlobs "unique PROVID/FEDTAXID pairs imported from the
P%substr(&PAYERNUM,6,3)PRV.txt file have been saved in P%substr(&PAYERNUM,6,3)PRV.SD2.";
RUN;

*****************************************;
*  6) Contents, sample print, freq         ;
*****************************************;
PROC CONTENTS DATA=out1.&OUTFNAME;
RUN;

PROC PRINT DATA=out1.&OUTFNAME (OBS=50);
VAR PROVID FEDTAXID LNAME FNAME SPECL1 SPECL2 SPECL3
    SPECL4 SPECL5 SPECL6 SPECL7 SPECL8 SPECL9 SCNT DEANUM MIDINIT SUFFIX CREDENT;
TITLE5 "Print First 50 Obs From &Outfname";
Run;


PROC FREQ DATA=out1.&OUTFNAME;
  FORMAT  PROVID  $PROVID. DEANUM $DEANUM.
          LNAME FNAME MIDINIT SUFFIX CREDENT $GENERAL.
          FEDTAXID  $FEDTAX.
          SPECL1-specl9  $SPECLTY.
          ;
  TABLE provid deanum fedtaxid SPECL1 SPECL2 SPECL3
        specl4-specl9 SCNT /MISSING list;
  TITLE5 "Frequency Of Variables In The Provider File";
RUN;
```

```
%jobstart;

 ************************************************************************;
 *                                                                      *;
 *  Program: d:\mcdb\pharmacy\prog\Create_dea_tax_xref.sas              *;
 *                                                                      *;
 *  Programmer:    John May                                             *;
 *----------------------------------------------------------------------*;
 *  Description: Create DEA, Provid and Fedtaxid Cross Reference Files  *;
 *----------------------------------------------------------------------*;
 *  Date Created: 10/29/2002                                            *;
 *  Date Modified:  7/15/2004 - modified to output three reference files *;
 *  (DEA, PROVID and FEDTAX) with the three most common specialty codes. *;
 *  PROVID and FEDTAXID files will be specific by payer                 *;
 ************************************************************************;

%LET PROCYR=2006;

  LIBNAME indd 'D:\Mcdb\DataCol\RX_Step4_7\data' ACCESS=READONLY;
  LIBNAME outdd 'D:\Mcdb\DataCol\RX_Step4_7\data';

  title1  " %sysfunc(putn(%sysfunc(time()), hhmm.)) %sysfunc(putn(%sysfunc(date()), weekdate.))";
  title2 "MARYLAND HEALTH CARE COMMISSION";

  options nocenter sumsize=max mprint obs=max;

%macro hifrq(inf1=, outf1=, byvar1=, outf2=, byvar2=, byvar3=, outf3=);

*output one deanum record for each specialty code - frequency counts
 will be used in determining which codes are speclty_1, ,speclty_2, speclty_3
 First - check to see if only one record by key.  If so, we want to
 retain the specialty code order;
data one(keep=&byvar1 specl1-specl3 rf_multi totspec)
     more(keep=&byvar1 specl1-specl9);
  set &inf1; by &byvar1;
  if first.dea_ind and last.dea_ind then do;
    if specl4 ne ' ' then do;
         specl1='101'; specl2=' '; specl3=' '; rf_multi='1';
      if specl9 ne ' ' then totspec=9;
         else if specl8 ne ' ' then totspec=8;
      else if specl7 ne ' ' then totspec=7;
      else if specl6 ne ' ' then totspec=6;
      else if specl5 ne ' ' then totspec=5;
      else if specl4 ne ' ' then totspec=4;
    end;
       else do;
          rf_multi='0';
          if specl3 ne ' ' then totspec=3;
          else if specl2 ne ' ' then totspec=2;
          else totspec=1;
       end;
    output one;
  end;
  else output more;

proc print data=one(obs=10); title 'no dupes'; run;

data &outf1(drop=specl1-specl9);
 set more;
 spc1=specl1; output;
 if specl2 ne ' ' then do; spc1=specl2; output; end;
 if specl3 ne ' ' then do; spc1=specl3; output; end;
 if specl4 ne ' ' then do; spc1=specl4; output; end;
 if specl5 ne ' ' then do; spc1=specl5; output; end;
 if specl6 ne ' ' then do; spc1=specl6; output; end;
 if specl7 ne ' ' then do; spc1=specl7; output; end;
 if specl8 ne ' ' then do; spc1=specl8; output; end;
 if specl9 ne ' ' then do; spc1=specl9; output; end;
run;
```

```
proc print data=&outf1(obs=10);
var &byvar1 spc1;
title "Post transpose of &outf1";
run;

*sum up speclty counts;
proc summary data=&outf1 missing nway;
 class &byvar1 spc1;
 output out=&outf2(drop=_type_ rename=(_freq_=count));
 run;

proc print data=&outf2(obs=10);
 var &byvar1 spc1 count;
 title "sample print of summarized &outf1";
 run;

proc sort data=&outf2; by &byvar1 descending count spc1;
run;

*collapse file - putting out one record per
 deanum with 3 most frequent specialties.
 if two specialties have same freq - select
 by ascending specialty.
If more than 3 specialties associated with a DEANUM, set
 rf_multi to 1, speclty1 to '101', and speclty2/3 to blank ;

data &outf3(drop=spc1 count);
 set &outf2; by &byvar1;
 retain totspec specl1 specl2 specl3 rf_multi;
 if first.&byvar2 then do;
  totspec=0; specl1='   '; specl2='   '; specl3='   '; rf_multi='0';
 end;
 totspec+1;
 if totspec=1 then specl1=spc1;
 if totspec=2 then specl2=spc1;
 if totspec=3 then specl3=spc1;
 if totspec=4 then do;
  specl1='101'; specl2=' '; specl3=' '; rf_multi='1';
 end;
 if last.&byvar2 then output;
 run;

 *combine the final files;
data outdd.&outf3;
 set one &outf3;

proc print uniform noobs data=outdd.&outf3(obs=10);
 var &byvar1 specl1-specl3 totspec rf_multi;
 where totspec>1;
 title1 "sample print of outdd.&outf3";
run;

proc freq data=outdd.&outf3;
 table dea_ind
       rf_multi*totspec /list missing;
 format dea_ind $dea_ind. rf_multi $multi.;
 title1 "Freqs of DEA_IND and RF_MULTI from &outf3";
 run;

proc freq data=outdd.&outf3;
 where rf_multi='1';
  table rf_multi*specl1*specl2*specl3 /list missing;
title1 "Cross-freqs of RF_MULTI and SPECLTY Codes from &outf3";
run;

proc contents data=outdd.&outf3;
 title1 "Contents of &outf3";
 run;

%mend hifrq;
```

```
 proc format;
  value $dea_ind
   ' '='Missing'
   '1'='DEA is DEANUM'
   '2'='DEA is PROVID'
   '3'='DEA is FEDTAXID';

  value $multi
   ' '='Missing'
   '0'='Non-Multiple Specialties'
   '1'='Multiple Specialties';

 *flag invalid records;
  data all;
    set indd.prv_dea_master_&PROCYR;
    payer=upcase(payer);
    if specl1 ne ' ' then speclty_sw=1;
    else speclty_sw=0;

 *DEA MUST BE GREATER THAN ONE CHAR AND NOT BE ALL ZEROES OR ALL 9's;
    if (length(compress(deanum)) <2 or
       compress(deanum,'0') eq ' ' or
       compress(deanum,'9') eq ' ' or
       index(deanum,'N/A') NE 0 OR
       index(deanum,'NO DEA') NE 0 OR
           index(deanum,'UNKNOWN') NE 0 OR
       index(deanum,'NOT REQ') NE 0) then dea_sw=0;
    else dea_sw=1;

*PROVID MUST BE GREATER THAN ONE CHAR AND NOT BE ALL ZEROES OR ALL 9's;
    if length(compress(provid)) <2 or
      compress(provid,'0') eq ' ' or
          index(provid,'N/A') NE 0 OR
          index(provid,'UNKNOWN') NE 0 or
      compress(provid,'9') eq ' ' then prov_sw=0;
    else prov_sw=1;

 *FEDTAXID MUST BE GREATER THAN ONE CHAR AND NOT BE ALL ZEROES OR ALL 9's;
    if length(compress(fedtaxid)) <2 or
      compress(fedtaxid,'0') eq ' ' or
          index(fedtaxid,'N/A') NE 0 OR
          index(fedtaxid,'UNKNOWN') NE 0 or
      compress(fedtaxid,'9') eq ' ' then tax_sw=0;
    else tax_sw=1;

    *if _n_ gt 100000 then stop;
 run;

proc freq data=all;
 table dea_sw*speclty_sw
       prov_sw*speclty_sw
       tax_sw*speclty_sw /list missing;
title 'Initial cross freqs';
title2 'keep obs where speclty_sw>0 and dea_sw=1 or prov_sw=1 or tax_sw=1';
run;

*DEANUM PROCESSING
 begin processing to determine 3 most frequent specialties.
 transpose file, putting out one record per specialty to determine most common specialties;

*put out file with "valid" DEA numbers;
data DEA(keep=deanum fedtaxid payer fname lname specl1 specl2 specl3
         specl4 specl5 specl6 specl7 specl8 specl9 tax_sw provid);
 set all(where=(dea_sw=1 and speclty_sw>0));

proc sort data=dea; by deanum payer;
run;

data dea;
 set dea; by deanum payer;
 if (first.payer and last.payer) then payer_dupe='N';
```

```
 ELSE payer_dupe='Y';

proc freq data=dea;
 table payer*payer_dupe
       payer_dupe /list missing;
 title1' Frequency of duplicate DEANUM within a payer';
 title2 ' ';
 run;

proc print noobs uniform data=DEA(obs=20);
 where payer_dupe='Y';
 by deanum payer;
 id deanum payer;
 var fedtaxid fname lname specl1 specl2 specl3;
 title1 'Diagnostic - Print of records with Duplicate DEA witin Payer';
 run;

*try to get a non-blank name, fedtaxid and provid for each deanum;
data deaname(keep=deanum fname lname fedtaxid blank_name provid);
 length h_last $40 h_first $20 h_tax $11 h_provid $13 blank_name $1;
 retain h_last h_first h_tax h_provid blank_name;
 set DEA; by deanum;
 if first.deanum then do;
   h_last=' '; h_first=' '; h_tax=' '; h_provid=' '; blank_name='y';
 end;
 if tax_sw=1 and (lname ne ' ' or fname ne ' ') then do;
   h_first=fname; h_last=lname; h_tax=fedtaxid; h_provid=provid;
   blank_name='n';
 end;
 if last.deanum then do;
   fname=h_first; lname=h_last; fedtaxid=h_tax; provid=h_provid; output;
 end;

proc freq data=deaname;
 table blank_name /list missing;
 title1 'Percentage of non-blank Name/Fedtaxid';
 run;

*link names and fedtaxid back to deanum file;
data deanum(drop=blank_name);
 merge dea(in=a keep=deanum specl1-specl9)
       deaname(in=b); by deanum;
 if a;

*flag duplicate records with same dea/specl1-specl9 data;
proc sort data=deanum; by deanum specl1 specl2 specl3 specl4
                          specl5 specl6 specl7 specl8 specl9;
run;

*add dea_ind to indicate match by DEANUM;
data deanum;
 set deanum; by deanum specl1 specl2 specl3 specl4
                specl5 specl6 specl7 specl8 specl9;
 dea_ind='1';
 if (first.specl9 and last.specl9) then speclty_dupe='N';
 else speclty_dupe='Y';

proc freq data=deanum;
 table speclty_dupe /list missing;
 title1' Frequency of duplicate DEANUM and All Specialty Codes';
 title2 ' ';
 run;
*execute macro to read in all specialty codes and keep the three
 most common codes and set rf_multi indicator;
 %hifrq(inf1=deanum, outf1=dea_specltyty,
        byvar1=deanum provid fedtaxid fname lname dea_ind,
        outf2=dea2, byvar2=deanum,
        outf3=provider_dea_master);

*PROVID PROCESSING;
*Read in data - selecting observations that have valid provids
```

```
 and specialty codes. Special processing for P480 - add leading zeroes;
data prov(keep=payer prvid specl1-specl9);
 set all;
 length prvid $11;
 if prov_sw=1 and speclty_sw>0;
 prvid=trim(provid);

* Eliminate complete dupes;
 proc sort data=prov out=prv_nodupe(rename=(prvid=provid)) nodupkey;
   by payer prvid
    specl1 specl2 specl3 specl4 specl5 specl6 specl7 specl8 specl9;
run;

* add dea_ind to indicate match by provid;
data prv_nodupe;
 set prv_nodupe;
 dea_ind='2';

*execute macro to read in all specialty codes and keep the three
 most common codes and set rf_multi indicator;
 %hifrq(inf1=prv_nodupe, outf1=prv_speclty,
        byvar1=payer provid dea_ind,
        outf2=prv2, byvar2=provid,
        outf3=provider_provid_master);


*FEDTAXID PROCESSING;
*First, select fedtaxids that have valid taxids and specialty codes.
 Then eliminate dupes;
 proc sort data=all out=tax_nodupe nodupkey; by payer fedtaxid
    specl1 specl2 specl3 specl4 specl5 specl6 specl7 specl8 specl9;
 where tax_sw=1 and speclty_sw>0;
run;

* add dea_ind to indicate match by fedtaxid, change length of fedtaxid;
data tax_nodupe(keep=payer fedtaxid dea_ind specl1-specl9);
 set tax_nodupe(rename=(fedtaxid=taxid));
 length fedtaxid $11;
 fedtaxid=taxid;
 dea_ind='3';

*execute macro to read in all specialty codes and keep the three
 most common codes and set rf_multi indicator;
 %hifrq(inf1=tax_nodupe, outf1=tax_speclty,
        byvar1=payer fedtaxid dea_ind,
        outf2=tax2, byvar2=fedtaxid,
        outf3=provider_ftaxid_master);
%jobend;
```

```
*This reads all the prv directories &creates one file of selected vars;

%LET PROCYR=2006;

LIBNAME outdd 'D:\Mcdb\DataCol\RX_Step4_7\data';
OPTIONS NOXWAIT obs=max;

x 'DIR/s D:\MCDB\DATACOL\P???Prx.sas7bdat> d:\WORKAREA\TEMP_DIR.TXT';

data one;
  infile  "d:\WORKAREA\TEMP_DIR.TXT" missover;
  input  @31 payer $4.;
  If upcase(substr(payer,1,1))  = "P";
  If upcase(substr(payer,2,1))  = "X" or payer="" then delete;
run;

data _null_;
  set one end=eof;
  call symput('p'||left(_N_), payer);
  if eof then call symput('nobs',_N_);
run;

options mprint mlogic symbolgen;
%macro payrinfo;

%let num1=%eval(&nobs-1);
libname x ( %do i=1 %to &num1;
               "D:\MCDB\DATACOL\&&p&i\sasdata",
            %end;
              "D:\MCDB\DATACOL\&&p&i\sasdata");

%mend payrinfo;

%payrinfo;

%macro bldmast;

  %let i=1;
  data master(keep=deanum fedtaxid provid specl1-specl9 payer lname fname);
    length fedtaxid $11;
    set x.&&p&i..prx;
        payer="&&p&i";

  %do i=2 %to &nobs;
   data d&i(keep=deanum fedtaxid provid specl1-specl9 payer lname fname);
     length fedtaxid $11;
     set x.&&p&i..prx;
         payer="&&p&i";

    proc append base=master force data=d&i;
  %end;
  run;

%mend bldmast;

%bldmast;

data outdd.PRV_Dea_master_&PROCYR(compress=binary);
 set master;
run;

proc contents data=outdd.PRV_Dea_master_&PROCYR;
 title 'Proc Contents';
 run;
proc print data=outdd.PRV_Dea_master_&PROCYR(obs=10);
 where fedtaxid ne 'U';
 title 'Diagnostic Print';
run;
proc freq data=outdd.PRV_Dea_master_&PROCYR;
 table payer /list missing;
```

```
 title 'Payer Frequencies';
 run;

x 'DEL d:\WORKAREA\TEMP_DIR.TXT';
```

```
*******************************************************************;
*                                                                 ;
*  Program Name:  Run_Step4_7.SAS                                 ;
*                                                                 ;
*  Programmer:    Raymond Hu                                      ;
*                                                                 ;
*  Description:   Processing multiple jobs and save log, output   ;
*                 files into their corresponding subdirectory.    ;
*                                                                 ;
*  DATE CREATED:  5/17/2000                                       ;
*  DATE MODIFIED: 11/05/02 - added date macro variables           ;
*  DATE MODIFIED:  7/20/04 - modify the order in which some steps are ;
*                  executed.                                      ;
*                                                                 ;
*  Modified by:   Po-Lun Chou                                     ;
*  Date:          September 21, 2007                              ;
*  Purpose:       Modify and re-run this program for the payers who   ;
*                 re-submitted Provider or Pharmacy data after August ;
*                 15, 2007.                                       ;
*******************************************************************;


*****************************************************************;
*   1. Macro variables                                          ;
*****************************************************************;
options obs=max;

%LET CURYR2=06;
%LET CURYR4=2006;
%LET NXTYR2=07;
%LET NXTYR4=2007;
%let begdt="01jan&curyr2";
%let enddt="30jun&nxtyr2";
%let yrEndDt = "31DEC&CURYR4"d;

%let D_f1=P760;
%let D_f2=P---;
%let D_f3=P---;
%let D_f4=P---;
%let D_f5=P---;
%let D_f6=P---;
%let D_f7=P---;
%let D_f8=P---;
%let D_f9=P---;
%let D_f10=P---;
%let D_f11=P---;
%let D_f12=P---;
%let D_f13=P---;
%let D_f14=P---;
%let D_f15=P---;
%let D_f16=P---;
%let D_f17=P---;
%let D_f18=P---;
%let D_f19=P---;
%let D_f20=P---;
%let D_f21=P---;
%let D_f22=P---;
%let D_f23=P---;
%let D_f24=P---;
%let D_f25=P---;
%let D_f26=P---;
%let D_f27=P---;
%let D_f28=P---;
%let D_f29=P---;

/*
%let D_f1=P020;
%let D_f2=P030;
%let D_f3=P070;
```

```
%let D_f4=P130;
%let D_f5=P131;
%let D_f6=P160;
%let D_f7=P280;
%let D_f8=P320;
%let D_f9=P325;
%let D_f10=P330;
%let D_f11=P350;
%let D_f12=P471;
%let D_f13=P480;
%let D_f14=P500;
%let D_f15=P510;
%let D_f16=P520;
%let D_f17=P530;
%let D_f18=P620;
%let D_f19=P680;
%let D_f20=P760;
%let D_f21=P820;
%let D_f22=P830;
%let D_f23=P850;
%let D_f24=P870;
%let D_f25=P---;
%let D_f26=P---;
%let D_f27=P---;
%let D_f28=P---;
%let D_f29=P---;
*/

*****************************************************************;
*   2. Macro - count jobs                                       ;
*      to count total how many job submitted this time          ;
*      njob - output macro var                                  ;
*****************************************************************;
%macro job_cnt;
  %global njob;
  %let njob=0;
  %do i = 1 %to 25;
     %if "%substr(&&D_f&i,2,3)" ne "---" %then %do;
         %let njob=%eval(&njob+1);
     %end;
  %end;
%mend job_cnt;

%job_cnt;
%let TotJob=&njob;



*****************************************************************;
*   3. Macro - run step 4-7                                     ;
*****************************************************************;

%macro run_job (nJob=);

  %do i = 1 %to &nJob;
  %*** a) Create libnames*********;
    LIBNAME  INdata1     "d:\MCDB\DATACOL\&&D_f&i\SASDATA\";    ***Enc and provider files*;
    LIBNAME  RXIN     "d:\MCDB\DATACOL\&&D_f&i\SASDATA\";    ***RX file*;
    LIBNAME  ENCIN     "i:\MCDB\DATACOL\&&D_f&i\SASDATA";  ***Encounter patient summary files;
    LIBNAME  OUTdata1    "d:\MCDB\DATACOL\&&D_f&i\SASDATA\";
    LIBNAME  OUTdata2    "d:\MCDB\DATACOL\&&D_f&i\SASDATA\";
    LIBNAME LIBRARY "D:\Mcdb\DataCol\Formats\Permanent";


    %*** b) create macro variable needed by Excserv.sas ***;
    %let PayerNum=&&D_f&i;

    %*** c) display message in log window ***;
    %put <<---PayerNum=&&D_f&i>>;

    %*** d) lists the program code in the log ***;
```

```
options source2;



%********************** STEP 4 - NDC Categories Assignment ******************;
%let jobname=NDC_Categories_Assignment;

%put  ---Running &jobname..sas---;
%*** e) specify the log and output file names ***;
proc printto log="d:\MCDB\DATACOL\&&D_f&i.\Log_Out\%substr(&&D_f&i,1,4)&jobname..log"
           print="d:\MCDB\DATACOL\&&D_f&i.\Log_Out\%substr(&&D_f&i,1,4)&jobname..lst" new;

%*** f) submit each specified job ***;
%include "D:\Mcdb\DataCol\RX_Step4_7\PROG\&jobname..sas";

%*** g) release the log and output files and back to the default ***;
proc printto;
run;


%********************** STEP 5 - convert zd format ******************;
%let jobname=ZdConvRX;

%put  ---Running &jobname..sas---;
%*** e) specify the log and output file names ***;
proc printto log="d:\MCDB\DATACOL\&&D_f&i.\Log_Out\%substr(&&D_f&i,1,4)&jobname..log"
           print="d:\MCDB\DATACOL\&&D_f&i.\Log_Out\%substr(&&D_f&i,1,4)&jobname..lst" new;

%*** f) submit each specified job ***;
%include "D:\Mcdb\DataCol\RX_Step4_7\PROG\&jobname..sas";

%*** g) release the log and output files and back to the default ***;
proc printto;
run;


%********************** STEP 6 - set 5 screen variables *************;
%let jobname=ExcservRX;

%put  ---Running &jobname..sas---;
%*** e) specify the log and output file names ***;
proc printto log="d:\MCDB\DATACOL\&&D_f&i.\Log_Out\%substr(&&D_f&i,1,4)&jobname..log"
           print="d:\MCDB\DATACOL\&&D_f&i.\Log_Out\%substr(&&D_f&i,1,4)&jobname..lst" new;

%*** f) submit each specified job ***;
%include "D:\Mcdb\DataCol\RX_Step4_7\PROG\&jobname..sas";

%*** g) release the log and output files and back to the default ***;
proc printto;
run;

    %********************** STEP 7 - set 5 screen variables *************;
%let jobname=EnctlinkRX;

%put  ---Running &jobname..sas---;
%*** e) specify the log and output file names ***;
proc printto log="d:\MCDB\DATACOL\&&D_f&i.\Log_Out\%substr(&&D_f&i,1,4)&jobname..log"
           print="d:\MCDB\DATACOL\&&D_f&i.\Log_Out\%substr(&&D_f&i,1,4)&jobname..lst" new;

%*** f) submit each specified job ***;
%include "D:\Mcdb\DataCol\RX_Step4_7\PROG\&jobname..sas";

%*** g) release the log and output files and back to the default ***;
proc printto;
run;


%********************** STEP 8 - Debit/Credit Analysis ******************;
%let jobname=Debit_Credit_Analysis;

%put  ---Running &jobname..sas---;
```

```
    %*** e) specify the log and output file names ***;
    proc printto log="d:\MCDB\DATACOL\&&D_f&i.\Log_Out\%substr(&&D_f&i,1,4)&jobname..log"
               print="d:\MCDB\DATACOL\&&D_f&i.\Log_Out\%substr(&&D_f&i,1,4)&jobname..lst" new;

    %*** f) submit each specified job ***;
    %include "D:\Mcdb\DataCol\RX_Step4_7\PROG\&jobname..sas";

    %*** g) release the log and output files and back to the default ***;
    proc printto;
    run;



    %********************** STEP 8A - statistics ************************;
    %let jobname=statChkRX;

    %put  ---Running &jobname..sas---;
    %*** e) specify the log and output file names ***;
    proc printto log="d:\MCDB\DATACOL\&&D_f&i.\Log_Out\%substr(&&D_f&i,1,4)&jobname..log"
               print="d:\MCDB\DATACOL\&&D_f&i.\Log_Out\%substr(&&D_f&i,1,4)&jobname..lst" new;

    %*** f) submit each specified job ***;
    %include "D:\Mcdb\DataCol\RX_Step4_7\PROG\&jobname..sas";

    %*** g) release the log and output files and back to the default ***;
    proc printto;
    run;


  %end;

%mend run_job;

options noxwait xsync;
%macro chk_job (nJob=);
  x "del d:\workarea\Log_sum.txt";

  %do i = 1 %to &nJob;
    %*** Turn ON page break, SAS date, page number ***;
    options formdlim='' date number pageno=&i;

    %put <<Check PayerNum=&&D_f&i>>;

    %********************** STEP 4 - NDC Categories assignment *******************;
    %let jobname=NDC_categories_assignment;
    %*** a) define libname ***;
    %let logname=d:\MCDB\DATACOL\&&D_f&i.\Log_out\%substr(&&D_f&i,1,4)&jobname..log;
    %*** b) display message in log window ***;
    %put ---Check &jobname..sas---;

    %*** c) submit each specified job ***;
    %let header=yes;
    %include "i:\MacroSAS.Lib\Log_chk.sas";

    %*** Turn OFF page break, SAS date, page number ***;
    options formdlim=' ' nodate nonumber;



    %********************** STEP 5 - convert zd format *******************;
        %let jobname=ZdConvRX;
            %*** a) define libname ***;
    %let logname=d:\MCDB\DATACOL\&&D_f&i.\Log_out\%substr(&&D_f&i,1,4)&jobname..log;
    %*** b) display message in log window ***;
    %put ---Check &jobname..sas---;

    %*** c) submit each specified job ***;
    %let header=yes;
    %include "i:\MacroSAS.Lib\Log_chk.sas";

    %*** Turn OFF page break, SAS date, page number ***;
```

```
    options formdlim=' ' nodate nonumber;

    %********************** STEP 6 - set 5 screen variables *************;
    %let jobname=ExcservRX;
    %*** a) define libname ***;
    %let logname=d:\MCDB\DATACOL\&&D_f&i.\Log_out\%substr(&&D_f&i,1,4)&jobname..log;
    %*** b) display message in log window ***;
    %put ---Check &jobname..sas---;

    %*** c) submit each specified job ***;
    %let header=no;
    %include "i:\MacroSAS.Lib\Log_chk.sas";;

        %********************** STEP 7 - set 5 screen variables *************;
    %let jobname=EnctlinkRX;
    %*** a) define libname ***;
    %let logname=d:\MCDB\DATACOL\&&D_f&i.\Log_out\%substr(&&D_f&i,1,4)&jobname..log;
    %*** b) display message in log window ***;
    %put ---Check &jobname..sas---;

    %*** c) submit each specified job ***;
    %let header=no;
    %include "i:\MacroSAS.Lib\Log_chk.sas";


    %********************** STEP 8 - Debit/Credit Analysis *************;
    %let jobname=Debit_credit_analysis;
    %*** a) define libname ***;
    %let logname=d:\MCDB\DATACOL\&&D_f&i.\Log_out\%substr(&&D_f&i,1,4)&jobname..log;
    %*** b) display message in log window ***;
    %put ---Check &jobname..sas---;

    %*** c) submit each specified job ***;
    %let header=no;
    %include "i:\MacroSAS.Lib\Log_chk.sas";


    %********************** STEP 8A - statistics ************************;
    %let jobname=statChkRX;
    %*** a) define libname ***;
    %let logname=d:\MCDB\DATACOL\&&D_f&i.\Log_out\%substr(&&D_f&i,1,4)&jobname..log;
    %*** b) display message in log window ***;
    %put ---Check &jobname..sas---;

    %*** c) submit each specified job ***;
    %let header=no;
    %include "i:\MacroSAS.Lib\Log_chk.sas";
  %end;

%mend chk_job;

*****************************************************************;
*   3. Submit all the specified jobs  for step 4 - 8            ;
*****************************************************************;
options formdlim='';
%run_job(nJob=&TotJob);

*****************************************************************;
*   4. Check all the logs of these submitted jobs               ;
*****************************************************************;
title;
options nosource2;
%chk_job (nJob=&TotJob);
options source2;
```

```
*%JOBSTART;
/*
*==============================================================================
*
*  Program Name:  I:\MCdb\DataCol\RX_STEP4_7\prog\ZdConvRX.SAS
*
*  Programmer:    Raymond Hu/Adrien Ndikumwami/John May
*
*  Description:   CONVERT NUMERIC VARS INTO NUMERIC FORMAT AND DELETE
*                 DEFINITE INVALID MEDICAL SERVICE RECORDS FROM THE ENCOUNTER FILE
*
*  Instructions:  TO USE THE PROGRAM:
*                 1. Copy the program to I:\MCdb\DataCol\Pxxx\Programs\
*                 2. Change:
*                    PAYERNUM=Pxxx -> number assigned to payer
*
*  Input files:   P890RX0  - 2001 Pharmacy DATABASE
*  Output files:  P890RX1  - 2001 Pharmacy DATABASE with numeric variable
*                                 in numeric format
*
*  DATE MODIFIED: 10/31/02 - COMBINE 4 DATASTEPS CREATING RX1
*                          - added pnum check to strip off $ from specific payers
*
*  DATE MODIFIED: 07/25/03 - Updated to get rid of $, . in P680
*                 07/20/04 - kept for 2003 data.
*                 07/06/06 - TOOK OUT $, . IN P680
*==============================================================================
*/

/*
OPTIONS NOCENTER ;

******************************
* Macro variable:
*   PAYERNUM - payer who sent the data
**;
*/

/*
%LET PAYERNUM=P680;

LIBNAME  INdata1     "I:\MCDB\DATACOL\&PAYERNUM\SASDATA\" ACCESS=READONLY;    ***Enc and provider
files*;
LIBNAME  RXIN     "d:\MCDB\DATACOL\&PAYERNUM\SASDATA\" ACCESS=READONLY;    ***RX file*;
LIBNAME  OUTdata2    "d:\MCDB\DATACOL\&PAYERNUM\SASDATA\";
LIBNAME  LIBRARY    "I:\MCDB\DATACOL\FORMATS\PERMANENT";
*/

TITLE1 "MARYLAND HEALTH CARE COMMISSION";
TITLE2 "STEP 4: ZD NUMERIC VARIABLES CONVERSION FROM THE Pharmacy FILE &PAYERNUM";
footnote1 "%sysfunc(putn(%sysfunc(time()), hhmm.)) %sysfunc(putn(%sysfunc(date()), weekdate.))";
footnote2 "program: D:\Mcdb\DataCol\RX_Step4_7\prog\ZdConvRX.sas";


*********************************************************************;
*   SAS MACROs                                                      ;
*********************************************************************;
%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=);

 pnum=upcase(substr("&PAYERNUM",1,4));
   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 ZDCONV1: 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;

data outdata2.P%substr(&PAYERNUM,2,3)rx1(COMPRESS=BINARY drop=pnum);
  set RXIN.P%substr(&PAYERNUM,2,3)rx0;
  length rbchrge rreimamt rpatliab 8 drugcmp 3;

*********************************************************************;
*   1. Convert financial vars into numeric vars                     ;
*      a)convert if it is in ZDw. format                            ;
*      b)convert '00000-123' to -123                                ;
*      c)convert '+000000123' to 123 or '000000+123' to 123         ;
*      BCHRGE_  --> BCHRGE                                          ;
*      REIMAMT_ --> REIMAMT                                         ;
*      PATLIAB_ --> PATLIAB                                         ;
```

```
*********************************************************************;

  *** a)convert if it is in ZDw. format                              ;
  *** b)convert '00000-123' to -123                                  ;
  *** c)convert '+000000123' to 123 or '000000+123' to 123           ;

  %zdconv1(invar=rBCHRGE_,outvar=rBCHRGE);
  %zdconv1(invar=rREIAMT_,outvar=rREIMAMT);
  %zdconv1(invar=rPATLIB_,outvar=rPATLIAB);

  *** drugcmp - if A or B -->1,2 ***;
  if drugcmp_ in ('A','B') then do;
     %zdconv1(invar=drugcmp_,outvar=drugcmp);
  end;
  else DRUGCMP=input(DRUGCMP_,2.);

  label
     rBCHRGE  = 'Billed Charge'
     rREIMAMT = 'Reimbursement Amount'
     rPATLIAB = 'Patient Liability'
     drugcmp = 'Drug Compound'
  ;

*********************************************************************;
*   2. Convert Count vars, DRUGQTY_,DRUGSPL_, into numerical vars     ;
*********************************************************************;
  length DRUGQTY 4 DRUGSPL 3;

  %zdconv1(invar=DRUGQTY_,outvar=DRUGQTY);
  %zdconv1(invar=DRUGSPL_,outvar=DRUGSPL);

  label
    DRUGQTY = 'Number of units dispensed'
    DRUGSPL = 'Estimated # of dispensed supply days'
  ;

*********************************************************************;
*   3. Convert the other character vars (with digit only) by using ZDw.  ;
*      and put back as character vars.                                 ;
*   4. Output SAS file, ENCNT1.SD2                                     ;
*      BIRTHYR,BIRTHMO,BIRTHDD - if ZD data                            ;
*      FILLYR,FILLMO,FILLDAY  - if ZD data                             ;
*      DRUGCMP - if A or B --> 1 or 2                                  ;
*      rPATZIP - if ZD data                                             ;
*      PHAZIP - if ZD data                                             ;
*      rSEX - if A or B --> 1 or 2                                      ;
* ???     NCPDP -                                                      ;
* ???     NDC -                                                        ;
*********************************************************************;

  *** BIRTHYR,BIRTHMO,BIRTHDD - if ZD data ***;
  %zdconv2(invar=rBIRTHYR);
  %zdconv2(invar=rBIRTHMO);
  %zdconv2(invar=rBIRTHDD);

  *** FILLYR,FILLMO,FILLDD - if ZD data ***;
  %zdconv2(invar=FILLYR);
  %zdconv2(invar=FILLMO);
  %zdconv2(invar=FILLDD);


  *** rSEX - if A or B -->1,2 ***;
  if rsex in ('A','B') then do;
     %zdconv2(invar=rSEX);
  end;

  *** rPATZIP - if ZD data              ***;
  %zdconv2(invar=rPATZIP);

  *** PHAZIP - if ZD data              ***;
  %zdconv2(invar=PHAZIP);
```

```
  *** NCPDP - ???                      ***;

  *** NDC - ???                      ***;

run;

*********************************************************************;
*   13. Contents and sample print                                   ;
*********************************************************************;
proc contents data=outdata2.P%substr(&PAYERNUM,2,3)rx1;
  title3 "P%substr(&PAYERNUM,2,3)rx1.SD2 - pharmacy file after ZD conversion";
run;

proc means data=outdata2.P%substr(&PAYERNUM,2,3)rx1 maxdec=2;
   label
     rBCHRGE  = ''
     rREIMAMT = ''
     rPATLIAB = ''
     drugcmp = ''
     DRUGQTY = ''
     DRUGSPL = ''
   ;
run;


proc print data=outdata2.P%substr(&PAYERNUM,2,3)rx1 (obs=25);
run;

proc freq data=outdata2.P%substr(&PAYERNUM,2,3)rx1;
  tables DRUGCMP*DRUGCMP_ DRUGQTY*DRUGQTY_  DRUGSPL*DRUGSPL_/ list missing;
  title3 "Check created numeric SAS data: DRUGQTY, DRUGSPL, DRUGCMP";
run;


*%JOBEND;

/*
*** save log and output files;
dm'log;   print file=D:\MCdb\DataCol\&PAYERNUM.\log_out\P%substr(&PAYERNUM,2,3)ZdConvRX.log
replace';
dm'output;print file=D:\MCdb\DataCol\&PAYERNUM.\log_out\P%substr(&PAYERNUM,2,3)ZdConvRX.lst
replace';
*/
```

```
*%JOBSTART;
/*
*==============================================================================
*
*  Program Name: EXCServRX.SAS
*
*  Programmer:    Raymond Hu/Adrien Ndikumwami/John May
*
*  Description:   DELETE DEFINITE INVALID MEDICAL SERVICE RECORDS FROM THE Pharmacy FILE
*
*
*  Input files:   PXXXrx2   - Pharmacy file with Provider data
*  Output files:  PXXXrx2   - Pharmacy DATABASE
*                             with deletion flags, spec codes and flag-flag
*
*  DATE MODIFIED: 10/16/2000
*                 1) set -999 to .U for financials according to Sophie
*
*
*  DATE MODIFIED: 10/16/2000 by Adrien
*                 Add length statement to enforce a uniform var length
*                 in order to combine all payers.
*                 Remove Fillyrx from drop statement-we need this field for
*                 non-y2k Fillyr.
*
*  DATE MODIFIED: 11/01/2002 by John
*                 replace literal dates with macro variables
*                 the link to provider was done in previous step, so don't
*                 need to merge with ELR file.  Read in RX1B and output rx2
*                 with corrected fields and additional flags set.
*
*  DATE MODIFIED: 07/25/2003 by John
*                 change state/zip verification from stnamefmt to mdstzipfmt
*                 added code to create outlier format and create RF_FINODD
*                 and RF_BCOUT flags.
*  DATE MODIFIED: 08/20/2003 by Adrien
*                 -999 are legitimate values and should not be set to .U
*
*  DATE MODIFIED: 07/20/04 by John
*                 compute rage, rage1203 using logic identical to Encounter.
*                 substitute space for U for deanum/phazip when U=Unknown.
*                 Since we are not including output from ENCTLINKRX, the input
*                 file to this comes from ZDCONVRX and is RX1, not RX1B.
*                 Per Doug Roth(email 8/2/04) Payer P850 birth dates 2004-2019
*                 should be set to 1904-1919.
*  DATE MODIFIED: 09/23/04 by John
*                 Added code for RF_PAOUT flag.
*  DATE MODIFIED: 7/25/05 by John
*                 Simplified code for FILLDATE using code from Step8_RX_Allpayers_1.
*                 Added logic (From Encounter code) to scramble Patid.
*                 Moved RF_NDC from Step8_RX_Allpayers_1 to Exservrx
*                 Moved RF_NAGE, RF_NDRUG, RF_MORDER. RF_MD and code to add/delete zeroes
*                 from NCPDP from Create_RX_Anaysis
*                 Added NCPDP_ST, RF_DEA variables
*                 Added MDC1 - MDC4 variables
*                 Moved up PIDBDG logic
*                 Add DEABUS variable
*  DATE MODIFIED: 6/30/06 - RENAMED DC1-DC4 TO MULTDC1-MULTDC4 (moved from Step8_RX_Allpayers_1
*                 added RSCRIPT.  DELETED MDC1-MDC4.
*  DATE MODIFIED: 7/30/07 - ADDED CODE TO COMPRESS OUT INVALID CHARACTERS (0,9,U) FROM
*  DEA, NDC, NCPDP, PATID, RPATZIP AND PHAZIP.
*==============================================================================
*/

OPTIONS NOCENTER;

******************************
* Macro variable:
*   PAYERNUM - payer who sent the data
**;
```

```
**will be assigned at calling macro**;
/*
%LET PAYERNUM=P500;

%LET CURYR2=05;
%LET CURYR4=2005;
%LET NXTYR2=06;
%LET NXTYR4=2006;
%let begdt="01jan&curyr2";
%let enddt="30jun&nxtyr2";
%let yrEndDt = "31DEC&curyr4"d;   *for computing age12yy;

LIBNAME  INdata1     "d:\MCDB\DATACOL\&PAYERNUM\SASDATA" ACCESS=READONLY;    ***Enc and provider
files*;
LIBNAME  RXIN     "d:\MCDB\DATACOL\&PAYERNUM\SASDATA" ACCESS=READONLY;    ***RX file*;
LIBNAME  OUTdata1    "d:\MCDB\DATACOL\&PAYERNUM\SASDATA";
LIBNAME  OUTdata2    "d:\MCDB\DATACOL\&PAYERNUM\SASDATA";
LIBNAME  LIBRARY     "d:\MCDB\DATACOL\FORMATS\PERMANENT";
*/

TITLE1 "MARYLAND HEALTH CARE COMMISSION";
TITLE2 "STEP 6: FLAG INVALID RX SERVICE RECORDS FROM THE PHARMACY FILE &PAYERNUM";
footnote1 "%sysfunc(putn(%sysfunc(time()), hhmm.)) %sysfunc(putn(%sysfunc(date()), weekdate.))";
footnote2 "program: D:\Mcdb\DataCol\RX_Step4_7\prog\ExcservRX.sas";

data P%substr(&PAYERNUM,2,3)rx2(drop = birthdate holdpid old1-old14 new1-new14 i
                                   dobyr CATEGORY_ID COMPRESS_CHK
                                RENAME=(DC1=MULTDC1 DC2=MULTDC2 DC3=MULTDC3 DC4=MULTDC4));

  LENGTH DRUGCMP_ $1  DRUGQTY_ $11 DRUGSPL_ $6  FILLDD   $2
       FILLMO   $2  FILLYR   $4  NCPDP    $7  NDC      $12 PATID    $14
       PHAZIP   $5  RBCHRGE_ $10 RBIRTHDD $2  RBIRTHMO $2  RBIRTHYR $4
       RPATLIB_        $10 RPATZIP  $5  RREIAMT_ $10 RSEX    $1;

  length rd_zip_f rd_dat_f RF_YR&nxtyr2 rf_finms  $1
          rF_PAYM $1;
  length R_NAGE12&curyr2 3;
  length fillyrx $4;

  set RXIN.P%substr(&PAYERNUM,2,3)rx1(rename=(patid=patid_raw));

 length pnum $4;
  pnum=upcase(substr("&PAYERNUM",1,4));

*****************************************************************
Add new variable rscript -- based on the definition from claudia
*****************************************************************;
  length rscript 3;
  if drugspl<= 30 then rscript = 1;
    else if 30 < drugspl <= 60 then rscript = 2;
    else if drugspl > 60 then rscript = 3;

*****************************************************************
*  New for 2006 data - COMPRESS OUT INVALID CHARACTERS, IF NOTHING
*  IS LEFT, THEN MOVE SPACES INTO THE VARIABLE
*****************************************************************;
LENGTH COMPRESS_CHK $11;
%MACRO COMPCHK(VAR);
COMPRESS_CHK=COMPRESS(&VAR,'0''9''U');
IF COMPRESS_CHK=' ' THEN &VAR=' ';
%MEND COMPCHK;
%COMPCHK(DEANUM);
%COMPCHK(NDC);
%COMPCHK(NDC2);
%COMPCHK(NDC9);
%COMPCHK(NCPDP);
%COMPCHK(PATID_RAW);
%COMPCHK(RPATZIP);
%COMPCHK(PHAZIP);

*Create new flag, RF_DEA, based on DEA Number;
```

```
  length RF_DEA $1;
  RF_DEA = Put(DEANUM, $fdeanum.);

*Create NDC flag: 0=Match Multum NDC,1=No Match, 2=Blank, 3=0s,4=9s ;
   if NDC9="" then do;
      if compress(ndc)="" then RF_NDC="2";
      else if compress(ndc,"0")="" then RF_NDC="3";
      else if compress(ndc,"9")="" then RF_NDC="4";
          else RF_NDC="1";
   end;
   else RF_NDC="0";

/*If category_id is . then change it to blank - 09/18/2006 */
  length RMSCAT $5;
  RMSCAT = put(CATEGORY_ID, 5.);
        if CATEGORY_ID = . then RMSCAT = ' ';

/**Added because we found rf_medsup = blank(shd be 0/1) when we ran RX_step8_Allpayers_1.sas--
09/18/2006**/
        if RMSCAT = ' ' and rf_medsup = ' ' then rf_medsup = '9';

/*** remove / add the extra leading zero for some payers ***/

/** per Ben's e-mail dated 02-14-2005, he agreed with Adrien's suggestion to **/
/** remove the extra leading zero for 4 payers, P325, P510, P680, & P330;    **/
/** and per Adrien's request, also remove the extra leading zero for P830    **/
/** and add the extra leading zero for P160                                  **/
/* these are still needed in 2005 */
  if pnum in ('P330','P510','P830') then do;
    if length(ncpdp) = 7 and substr(ncpdp, 1, 1) = '0' then do;
      ncpdp = substr(ncpdp, 2, 6);
    end;
  end;
  else if pnum = 'P160' then do;
    if substr(ncpdp, 1, 1) = '' then do;
      if substr(ncpdp, 2, 1) ne '' then substr(ncpdp, 1, 1) = '0';
    end;
  end;

 *Create new variable - NCPDP_ST;
LENGTH NCPDP_ST $2;
 NCPDP_ST=SUBSTR(NCPDP,1,2);

***** Redefine flag RF_NDRUG, based on APPRDATE and GBO because New-Branded should now
    be defined as approved by FDA between 2002 and 2005, and Branded-Not new are approved
    prior to 2002 AN-7/10/2006****;

  length RF_NDRUG $1;
  if gbo = 'N' and '01JAN2002'd <= apprdate <= "31DEC&CURYR4"d then rf_ndrug = '1';
  else if gbo = 'N' and not ('01JAN2002'd <= apprdate <= "31DEC&CURYR4"d) then rf_ndrug = '2';
  else if gbo = 'G' then rf_ndrug = '3';
  else rf_ndrug = '4';

* add new flag, RF_MORDER, based on NCPDP;
  length RF_MORDER $1;
  rf_morder = put(ncpdp, $moph.);

  *Still valid for 2006 data;
  if pnum in ('P480') then rREIMAMT=(rBCHRGE - rPATLIAB); ** Per Ben's e-mail on 9/05/03 **;

  /*  NO LONGER NECESSARY - USING COMPRESS TO IDENTIFY U,0,9 AS INVALID
  *substitute spaces for U for DEANUM and PHAZIP;
   If Deanum='U' then Deanum=' ';
   if Phazip='U' then Phazip=' ';
  */

  *** a) According to Pal call on 8/8/2000 Bruce Burget at 301-625-4300 ***;
  ***    -99900000 means not available                                    ***;

  /*   COMMENT OUT UNLESS NECESSARY - NOT NECESSARY FOR 2006 DATA
```

```
  if pnum in ('P480') then do;
     if rBCHRGE  = -99900000 then rBCHRGE  = .;
     rREIMAMT=(rBCHRGE - rPATLIAB);              ***Per Ben's e-mail on 9/05/03 **;
     if patid_raw="" then delete;  *There are 812 records in 2005 to delete. May not be needed in
2006**;

      if rREIMAMT = -99900000 then rREIMAMT = .;
     if rPATLIAB = -99900000 then rPATLIAB = .;
   end;
  */

  ***Edit all the financials for P325 with negative value--multiply by 10. The - values took 1
digit***;
/*  THE FOLLOWING EDIT IS NOT EFFECTIVE FOR 2006 DATA
   if pnum in ('P325') then do;
       if .< rBCHRGE <0   then rBCHRGE  = rBCHRGE*10;
       if .< rREIMAMT<0  then rREIMAMT = rREIMAMT*10;
       if .< rPATLIAB<0  then rPATLIAB = rPATLIAB*10;
   end;
*/

  *** c) Set sex = M/F to 1/2 ***;
  if RSEX='M' then RSEX='1';
  else if RSEX='F' then RSEX='2';

  *** c) Set sex =3 when sex=0 for P480***;
  if PNUM='P480' and RSEX='0' then RSEX='3';


  *******************************************************;
  *                                                     ;
  *   **** Divided by 100 ***                           ;
  *        since payers submitted financial variable    ;
  *        have two implied decimal places              ;
  *                                                     ;
  *******************************************************;
*STILL REQUIRED FOR 2006;
  if pnum notin ('P160','P471','P530') then do;
     if rBCHRGE>.Z   then rBCHRGE  = rBCHRGE/100;
     if rREIMAMT>.Z  then rREIMAMT = rREIMAMT/100;
     if rPATLIAB>.Z  then rPATLIAB = rPATLIAB/100;
  end;


*******************************************************************************;
*   Per Doug Roth email dated 8/2/04 the following correction needs to be done.    ;
*   Due to 2 digit year Y2k mistake - P850 birthdates 2004-2019 should be 1904-1919;
*******************************************************************************;
/*  NOT NECESSARY FOR 2006 DATA
 if pnum='P850' then do;
   if substr(rbirthyr,1,2)='20' and
      ('04'<=substr(rbirthyr,3,2)<='19') then rbirthyr='19'||substr(rbirthyr,3,2);
 end;
*/

*******************************************************************************;
*   Per Union Labor's Carole Shoulders, the last 2 digit of a patient Id are not the *;
*   same on RX and Encounter file. Remove the last 2 digit of patid for both files   *;
*******************************************************************************;
/***Not needed as of 2005 MCDB
if pnum='P850' then do;
  patid_raw=substr(patid_raw,1,9);
 end;
***/
*******************************************************************************;
*   Per Guardian's Gail Puls (e-mail on 9/24/2004), if the 6th byte is M, then the *;
*   7th and the 8th byte should be set to missing.                                 *;
*******************************************************************************;
/* 2005 - keeping this in increases the linkage slightly.
   FOR 2006 DATA THERE ARE STILL RECORDS WHERE M IS IN POS 6=M */
if pnum='P350' then do;
```

```
if  substr(patid_raw,6,1)= "M" then substr(patid_raw,7,2)="  ";
end;

*********************************************
     Encrypt Patient ID and create PIDBDG
*********************************************;
     length holdpid $ 14;
     holdpid = left(patid_raw);

     *** encrypt ***;
     length old1-old14 $ 1;
     length new1-new14 $ 1;

     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;

 length pidbdg $25;
 if patid ne ' ' then
        pidbdg = pnum    ||
                 patid   ||
                 rbirthyr ||
                 rbirthmo ||
                 rsex;

*****************************************************************************;
*                                                                            ;
*   Set deletion flags  1=deletion and 0=keep                                ;
*     1. Set deletion flag, rD_ZIP_f,                                        ;
*     2. Set deletion flag, rD_DAT_f, for Service provided before 1999       ;
*     3. Create rage and rage1200                                            ;
*     ??? f_cap f_uncv                                                       ;
*   Set flag flags  1=flaged and 0=not flaged                                ;
*    11. Set flag flag, F_YR01, for Service provided after 1998              ;
*    12. Set flag flag, f_finms, for not covered non-capitated service       ;
*                                                                            ;
*****************************************************************************;

*Create new variable - DEABUS;
LENGTH DEABUS $1;
DEABUS=PUT(DEANUM,$FDEABUS.);

label
    DEABUS = 'DEA Business Activity Code'
    rage     = 'Patient service Age'
    rage12&curyr2 = "Patient Age at Dec &curyr4"
    rd_zip_f = '1=Others,0=MD'
    rd_dat_f = "1=before &begdt or after &enddt or missing"
    rf_yr&nxtyr2 = "1: >=&nxtyr4 0: <&nxtyr4"
    rf_finms = 'All finan. var. are zero or missing'
    rF_PAYM   = 'Payment:1=negative, 2=0, 3=+, 4=missing'
    rpayment  = 'sum(REIMAMT,PATLIAB)'
    PNUM     = 'Payer unique ID (SSS)'
    fillyrx  =' Y2K Fillyr'
    RF_NDC  ="NDC Flag 0=Match Multum,1=No Match, 2=Blank, 3=0s,4=9s"
    PATID_RAW = 'ORIGINAL PATID'
    rscript   = "Number of Adjusted or 30-day Script";

  ************************************
  Create Filldate - moved up simplified code from Step8_RX_AllPayers_1.SAS.
  ************************************
  ;
```

```
  FILLDATE=MDY(
           INPUT(FILLMO,2.),
           INPUT(FILLDD,2.),
           INPUT(FILLYR,4.)
            );

  FILLYRX=FILLYR;

  ***NEW CODE FOR 2003 - CALCULATE rage AND rage12 SAME WAY AS ENCOUNTER;
***************************
Age at Service - creating birthdate using same logic as above.
***************************
;
    *** have birth year ***;
     if '1700'<rbirthyr<'9999' then do;
        *** have month, use the 15th to calculate date ***;
        if ('01'<=rbirthmo<='12') then birthdate = mdy(rbirthmo,15,rbirthyr);
        *** don't have month, use July 15 to calculate date ***;
        else birthdate = mdy(7,15,rbirthyr);
     end;
     *** no birth year information, birthdate set to missing **;
     else
     birthdate = .;

     if birthdate=. then rage=.;
     else rage=round((filldate-birthdate)/365.25,1);

     if rage < 0 or rage >110 then rage = .M;



 ************************************************
     Calculate Age as of December of Current Year
 ************************************************;

       ***Have birth year***;
       if '1700'<rbirthyr<'9999' then do;
         ***Have month, use the 15th to calculate date***;
        if ('01'<=rbirthmo<='12') then birthdate=mdy(rbirthmo,15,rbirthyr);
        ***Don't have month, use July 15 to calculate date***;
        else birthdate=mdy(7,15,rbirthyr);
        end;
       ***No birth year information, birthdate set to missing***;
       else birthdate=.;

       if birthdate=. then rage12&curYr2=.;
       else rage12&curYr2=round((&yrEndDt-birthdate)/365.25,1);

       if rbirthyr= . then rage12&curYr2=.;
       else
       if rage12&curYr2 < 0 or rage12&curYr2 > 110 then rage12&curYr2= .M;

**********************************************
  Calculate RNAGE - as of 12/31 Current year (not Rounded)
**********************************************;
  dobyr=input(rbirthyr,5.);

  if birthdate=. then r_nage12&curYr2 =.;
  else r_nage12&curYr2 =(&yrEndDt-birthdate)/365.25;

 ************************************************************************;
 * Patients born after &yrEndDt will be set to .M                                                    *;
                                        *;
 * However rounded rage12&curYr2 (Age as of December of Current Year) will *;
 * be set to zero if patients were born in the four months after &yrEndDt  *;
 ************************************************************************;
  if dobyr= . then r_nage12&curYr2 =.;
  else if r_nage12&curYr2 < 0 or r_nage12&curYr2 > 110 then r_nage12&curYr2 = .M;
```

```
  ******************************************************************;
  *   t) PAYMENT                                                   *;
  *      (based on item #17 of the Next Step in Developing the MCDB)  *;
  ******************************************************************;
  rpayment=sum(rreimamt,rpatliab);

  ******************************************************************;
  *   t) rF_PAYM                                                    *;
  *      1=negative payment                                        *;
  *      2=0 payment                                               *;
  *      3=positive payment                                        *;
  *      4=missing payment                                         *;
  *      (based on item #17 of the Next Step in Developing the MCDB)  *;
  ******************************************************************;
  if rpayment=. then rf_paym='4';
  else if rpayment<0 then rf_paym='1';
  else if rpayment=0 then rf_paym='2';
  else if rpayment>0 then rf_paym='3';

    rf_finms = '0';

  ********************************************************************************;
  *   1. Set deletion flag, D_ZIP_f,                                               ;
  *      drop if zipcode not Maryland except P690 (PATZIP)                         ;
  ********************************************************************************;

   if upcase(put(rPATZIP,$mdstzip.))='1' then rd_zip_f='0';
   else rd_zip_f='1';

   ***********************************************************************;
  *   2. Set deletion flag, D_DAT_f, for Service provided before 2000       ;
  *    a) fillyr are missing                                                ;
  *    b) if filldt < 1/1/curyr or > 06/30/nxtyr, exclude (RD_DAT_F).       ;
  *    c) if filldt is 1/1/01 or later, flag (RF_YR01).                     ;
  ***********************************************************************;

  rd_dat_f='0';
  rf_yr&nxtyr2 = '0';

 *** a) fillyr is missing ***;
  if input(fillyrx,4.)=. then do;
    rd_dat_f='1'; rf_yr&nxtyr2='1';
  end;
  ELSE DO;

  *** b) if fill date is before &begdt or after &enddt then exclude. ***;
  if filldate < &begdt.d or filldate > &enddt.d then rd_dat_f='1';

  *** c) if fill date is equal or later processing year +1, keep and flag these. ***;
  else if fillyrx >= &nxtyr4 then rf_yr&nxtyr2='1';
  END;



  ***********************************************************************;
  *   4. Set flag, f_finms, if the four financial vars are equal           ;
  *         to zero or missing AND f_finzr is all 3 vars =0                ;
;
  *                                                                        ;
  ***********************************************************************;

  if rbchrge in (0,.,.C,.M)  and
     rreimamt in (0,.,.C,.M) and
     rpatliab in (0,.,.C,.M) then  rf_finms='1';

  if rbchrge =0  and rreimamt=0 and rpatliab=0 then rf_finzr='1';
  else rf_finzr='0';



  ***********************************************************************;
```

```
  *   5. Set new flags, rf_10K, rf_yr65, and rf_patid                      ;
  ***********************************************************************;

  length rf_10K rf_yr65 rf_patid $1;

  if rpayment > 10000 then rf_10K = '1';
  else rf_10K = '0';

  *if rage12&CURYR2 in (., .M) then rf_yr65 = '2';
  *else if rage12&CURYR2 >= 65 then rf_yr65 = '1';
  *else if rage12&CURYR2 < 65 then rf_yr65 = '0';

  ** The Calculation of rf_yr65 is done with non-rounded age now instead of rage12&CURYR2 **;
  if r_nage12&CURYR2 in (., .M) then rf_yr65 = '2';
  else if r_nage12&CURYR2 >= 65 then rf_yr65 = '1';
  else if r_nage12&CURYR2 < 65 then rf_yr65 = '0';

  if compress(patid, "0")=" " then rf_patid = '1';
  else if compress(patid, "9")=" " then rf_patid = '2';
  else if patid = " " then rf_patid = '3';
  else rf_patid = '0';

run;

*CREATE THE RF_MD FLAG.  THIS INDICATES WHETHER OR NOT ANY PATIENT
 RECORD HAD A MARYLAND ZIPCODE;

*SUMMARIZE DATA BY PIDBDG AND RD_ZIP_F;
PROC SUMMARY DATA=P%substr(&PAYERNUM,2,3)rx2 MISSING NWAY;
  CLASS PIDBDG RD_ZIP_F;
  OUTPUT OUT=TEMPSUM(DROP=_TYPE_ _FREQ_);
RUN;

*KEEP ALL PATIENT IDS WITH AT LEAST ONE MD ZIPCODE RECORD;
DATA TEMP_MD(KEEP=PIDBDG);
  SET TEMPSUM;
  IF RD_ZIP_F='0';
RUN;

*CREATE INDEX OF MARYLAND PATIENT IDs;
PROC DATASETS LIBRARY=WORK;
  MODIFY TEMP_MD;
  INDEX CREATE PIDBDG /UNIQUE;
RUN;

DATA P%substr(&PAYERNUM,2,3)rx2;
 SET P%substr(&PAYERNUM,2,3)rx2;
 LENGTH RF_MD $1;
 *READ IN INDEXED FILE;
  SET  TEMP_MD KEY=PIDBDG /UNIQUE;

*RESET _ERROR_ IF NOMATCH;
  IF _IORC_ EQ 1230015 THEN _ERROR_=0;

*IF PIDBDG IS FOUND ON INDEXED FILE, THEN THERE IS AT LEAST ONE MD RECORD FOR THE PATIENT;
 IF _IORC_ NE 1230015 THEN RF_MD='1';
 ELSE RF_MD='0';

LABEL RF_MD    = "1-MD, 0-non MD";


*calculate bcpqty and papqty- eliminate division by zero and where
 drugqty is a non-whole number less than 1, i.e., .01;

data bc_pa_pqty(keep=pnum ndc bcpqty papqty drugqty rbchrge rpayment);
 set P%substr(&PAYERNUM,2,3)rx2(keep=pnum drugqty rbchrge ndc rpayment);
  if drugqty in(. 0) or
     0 <drugqty <1 then drugqty=1;
  bcpqty=abs(rbchrge/drugqty);
  papqty=abs(rpayment/drugqty);
run;
```

```
proc print uniform noobs data=bc_pa_pqty(obs=10);
 var pnum ndc drugqty rbchrge bcpqty rpayment papqty;
title5 'sample print of bcpqty and papqty calculation';
run;

*compute mean and standard deviation of billcharge/payment by pnum and NDC;
 proc means data=bc_pa_pqty missing nway noprint;
  class pnum ndc;
  var bcpqty papqty;
  output out=summ mean=meanbc meanpa std=sdbc sdpa;
 run;

proc print uniform noobs data=summ(obs=10);
title5 'sample print of output from means';
run;

*create lulimit/palimit var;
data lulimit;
 set summ;
 if sdbc=. then sdbc=0;
 lulimit=meanbc + (2.5*sdbc);
 if sdpa=. then sdpa=0;
 palimit=meanpa + (2.5*sdpa);

proc print data=lulimit(obs=10);
 title5 'sample print of lulimit file';
 run;

PROC MEANS DATA=LULIMIT N NMISS MEAN MIN MAX SUM;
TITLE5 'MEANS OF LULIMIT FILE';
RUN;

*create lulimit format;
data lulimit2(keep=start label fmtname);
 length label $12 start $16;
 set lulimit end=eof;
 start=pnum||ndc;
 label=put(lulimit,12.5);
 fmtname='$lulim';
 output;
 if eof then do;
  fmtname='$lulim';
  start='other';
  label=' ';
  output;
 end;

proc print uniform noobs data=lulimit2(obs=10);
title5 'sample print of Bill Charge Limit format data';
run;

proc sort data=lulimit2 nodupkey; by start; run;

 proc format cntlin=lulimit2; run;

*create palimit format;
data palimit2(keep=start label fmtname);
 length label $12 start $16;
 set lulimit end=eof;
 start=pnum||ndc;
 label=put(palimit,12.5);
 fmtname='$palim';
 output;
 if eof then do;
  fmtname='$palim';
  start='other';
  label=' ';
  output;
 end;
```

```
proc print uniform noobs data=palimit2(obs=10);
title5 'sample print of Payment Limit format data';
run;

proc sort data=palimit2 nodupkey; by start; run;

proc format cntlin=palimit2; run;

data outdata1.P%substr(&PAYERNUM,2,3)rx2(drop=RF_key drugqty_RF
                        papqty bcpqty pexpay compress=binary);
set P%substr(&PAYERNUM,2,3)rx2;
length Rf_finodd Rf_bcout pex_ind $1 RF_key $16 lu_limit pa_limit 8;

*code for RF_FINODD flag;
if round(rpayment,.1)=round(rbchrge,.1) then Rf_finodd='0';
if round(rpayment,.1)>round(rbchrge,.1) then Rf_finodd='1';
if round(rpayment,.1)<round(rbchrge,.1) then Rf_finodd='2';


*code for pex_pay;
 if rf_finodd='1' then do;
  if rpayment=0 then pex_ind='0';
  else do;
    pexpay=round((rpayment-rbchrge)/rpayment,.01);
    if pexpay<=.10 then pex_ind='1';
    else if pexpay<=.25 then pex_ind='2';
    else if pexpay<=.50 then pex_ind='3';
    else pex_ind='4';
  end;
 end;

*code for RF_BCOUT/RF_PAOUT flag;
  RF_key=pnum||ndc;
  lu_limit=input(put(RF_key,$lulim.),12.5);
  pa_limit=input(put(RF_key,$palim.),12.5);
  if drugqty in(. 0) or
     0 <drugqty <1 then drugqty_RF=1;
  else drugqty_RF=drugqty;
  bcpqty=round((rbchrge/drugqty_RF),0.00001);
  if -lu_limit <= bcpqty <= lu_limit then Rf_bcout='0';
  else Rf_bcout='1';

  papqty=round((rpayment/drugqty_RF),0.00001);
  if -pa_limit <= papqty <= pa_limit then Rf_paout='0';
  else Rf_paout='1';

*** Add encounter ID for specialty purpose ***;
  recnum = _n_;


  label
    multdc1 = 'Multum Drug Category 1'
    multdc2 = 'Multum Drug Category 2'
    multdc3 = 'Multum Drug Category 3'
    multdc4 = 'Multum Drug Category 4'
    lu_limit = 'Lower or upper charge limit'
    pa_limit = 'Lower or upper Payment limit'
    rf_bcout = 'Bill Charge - 0=Not outlier, 1= Outlier'
    rf_paout = 'Payment - 0=Not outlier, 1= Outlier'
    rf_finodd = '0=pymt=bill, 1=pymt>bill, 2=pymt<bill'
    pex_ind = 'Percentage of overpayment'
    recnum = 'Record Sequential Numbers'
    R_NAGE12&CURYR2 = "Patient Age as of 12/31/&CURYR4 (Not Rounded)"
    RF_NDRUG   = "New Drug Flag"
    RF_MORDER  = "Mail Order Flag(0=No, 1=Yes)"
    RF_DEA = "DEANUM(1=Valid, 0=Other, 2=Blank)"
    PATID = "PATIENT ID (SCRAMBLED)"
    NCPDP_ST = 'STATE CODE - 1ST TWO POS OF NCPDP'
   ;
RUN;
```

```
proc freq data=outdata1.P%substr(&PAYERNUM,2,3)rx2;
  tables rd_zip_f rd_dat_f / list missing;
  format rd_zip_f $d_zip_f. rd_dat_f $d_dat_f.;
  title5 "Check all the deletion flags: 1=deletion,0=keep";
run;

proc freq data=outdata1.P%substr(&PAYERNUM,2,3)rx2;
  tables rf_YR&nxtyr2 rf_finms /list missing;
  tables rf_bcout /list missing;
  tables rf_paout /list missing;
  tables rf_bcout*rf_paout /list missing;
  tables rf_finodd /list missing;
  format rf_yr&nxtyr2 $f_&nxtyr2.yr. rf_finms $f_finms.;
  format rf_bcout $f_bcout. rf_paout $f_paout. rf_finodd $f_finodd.;
  title5 "Check all the questionable flags: 1=flag questionable";
run;

proc freq data=outdata1.P%substr(&PAYERNUM,2,3)rx2;
 where rf_finodd='1';
  tables pex_ind /list missing;
  format pex_ind $pex_fmt.;
  title5 "Check Percentage of Excessive Payment";
run;

proc print data=outdata1.P%substr(&PAYERNUM,2,3)rx2 (obs=25);
 where rf_bcout='1';
  title5 "rx2 - checking where rf_bcout=1";
run;

proc print data=outdata1.P%substr(&PAYERNUM,2,3)rx2 (obs=25);
 where rf_paout='1';
  title5 "rx2 - checking where rf_paout=1";
run;

proc print data=outdata1.P%substr(&PAYERNUM,2,3)rx2 (obs=25);
 where rf_finodd='1';
  title5 "rx2 - checking where rf_finodd=1";
run;

************************************************************************;
*  10. Cross freq of deletion flag,                                    ;
*      D_ZIP_f* D_DAT_f*d_dnt_f*D_INP_f                                ;
************************************************************************;
proc freq data=outdata1.P%substr(&PAYERNUM,2,3)rx2;
  tables rd_zip_f*rd_dat_f / list missing;
  *format rd_zip_f $d_zip_f. d_dat_f $d_dat_f.;
  title5 "Check all the deletion flags: 1=deletion,0=keep";
run;

proc freq data=outdata1.P%substr(&PAYERNUM,2,3)rx2;
  tables rf_yr&nxtyr2*rf_finms /list missing;
  *format f_yr&nxtyr2 $f_&nxtyr2.yr. f_finms $f_finms. f_rule $f_rule. f_proc $fproc.;
  title5 "Check all the questionable flags: 1=flag questionable";
run;


************************************************************************;
*   13. Contents and sample print                                      ;
************************************************************************;
proc contents data=outdata1.P%substr(&PAYERNUM,2,3)rx2;
  title5 "outdata1.P%substr(&PAYERNUM,2,3)rx2 - Pharmacy file after setting exclusion flags";
run;
proc means data=outdata1.P%substr(&PAYERNUM,2,3)rx2 maxdec=2;
   label
     rBCHRGE  = ''
     rREIMAMT = ''
     rPATLIAB = ''
     drugcmp = ''
     DRUGQTY = ''
     DRUGSPL = ''
     rage = ''
```

```
     rage12&CURYR2 = ''
     rPayment = ''
   ;
run;

proc print data=outdata1.P%substr(&PAYERNUM,2,3)rx2 (obs=15);
 where rage12&curyr2=.M;
  title5 "rx2 - checking where rage12&curyr2=.M";
run;

*%JOBEND;

/*
*** save log and output files;
dm'log;   print file=D:\MCdb\DataCol\&PAYERNUM\log_out\P%substr(&PAYERNUM,2,3)ExcServRX.log
replace';
dm'output;print file=D:\MCdb\DataCol\&PAYERNUM\log_out\P%substr(&PAYERNUM,2,3)ExcServRX.lst
replace';
*/
```

```
proc datasets kill;
run;

*%jobstart;
*******************************************************************************
*  Program Name:  Debit_Credit_Analysis.sas
******************************************************************************
*  Description:   Identifies and flags the debit/credit payment records that cancel
*                 each other out and duplicates.
******************************************************************************
*  Input files:   d:\mcdb\datacol\&payernum\data\&payernumrx2
*
*  Output files:  d:\MCdb\DataCol\RX_Step4_7Pgm01\Data\&payernumrx3.sas7bdat
**  PROGRAMMER:  J May/A. Ndikumwami
*  DATE CREATED: 9/9/2003
*   Modified: 10/20/2005 to use bill charge for + and - cancellation.
*   Modified:  7/7/06 JMAY - changed setting of RF_NDUP from space to 0
*******************************************************************************;
/*
%let payernum= P070;
libname RXIN "d:\MCdb\DataCol\&payernum\SASdata";
libname outdata1 "d:\MCdb\DataCol\&PayerNUM\SASData";
LIBNAME LIBRARY "D:\Mcdb\DataCol\Formats\Permanent";

*LIBNAME  LIBRARY    "I:\MCDB\DATACOL\FORMATS\";
*/

options obs=max;

footnote1 "%sysfunc(putn(%sysfunc(time()), hhmm.)) %sysfunc(putn(%sysfunc(date()), weekdate.))";
footnote2 "program: D:\Mcdb\DataCol\RX_Step4_7\prog\Debit_Credit_Analysis.sas";

*READ IN RX3 AND CREATE VARIABLES TO BE USED IN DUPLICATE PROCESSING;

data rx4;
 set RXIN.&payernum.rx3;
 rcd_id=_n_;
 length pnum $4;
 pnum="&payernum";
 *round rpayment to whole number - if d/c pair within $1 then it is a dupe;
 tot_bill=round(abs(rbchrge),1);
 if .z<rbchrge<0 then neg_fin='y';
 else neg_fin='n';

 *Take care of P130 where bill is not - for the credit;

 if rbchrge>0 and rpayment<0 then do;
   neg_fin='n';
  tot_bill=round(abs(rpayment),1);
 end;

run;

 proc sort data=rx4 out=rx4_sort;
    by pnum patid rbirthyr rbirthmo rbirthdd rsex ndc fillyr fillmo filldd tot_bill;
 run;

*SPLIT FILE INTO POSITIVE PAIRS AND NEGATIVE PAIRS AND RECORDS WITH NO DUPLICATES;

 data neg_pairs(rename=(rcd_id=neg_rcd_id rpayment=neg_rpayment rbchrge = neg_rbchrge))
      pos_pairs(rename=(rcd_id=pos_rcd_id))
      all_pairs;
  set rx4_sort;
    by pnum patid rbirthyr rbirthmo rbirthdd   rsex ndc fillyr fillmo filldd;
  if not(first.filldd and last.filldd) then do;
    output all_pairs;
    if neg_fin='y' then output neg_pairs;
    else output pos_pairs;
  end;
 run;
```

```
*COULD HAVE MULTIPLE CANCEL RECORDS - SO ADD ANOTHER FIELD TO THE SORT FIELDS
 FOR USE IN THE MERGE;

data pos_pairs;
  set pos_pairs;
 by pnum patid rbirthyr rbirthmo rbirthdd rsex ndc fillyr fillmo filldd tot_bill;
 retain key_cnt;
 if first.tot_bill then key_cnt=0;
 key_cnt+1;
run;
proc sort data=pos_pairs;
   by pnum patid rbirthyr rbirthmo rbirthdd rsex ndc fillyr fillmo filldd tot_bill key_cnt;
 run;

data neg_pairs;
  set neg_pairs;
 by pnum patid rbirthyr rbirthmo rbirthdd rsex ndc fillyr fillmo filldd tot_bill;
 retain key_cnt;
 if first.tot_bill then key_cnt=0;
 key_cnt+1;
run;
proc sort data=neg_pairs;
   by pnum patid rbirthyr rbirthmo rbirthdd rsex ndc fillyr fillmo filldd tot_bill key_cnt;
 run;

*GETTING REC_IDS OF CANCELLED RECORDS THAT WILL BE DELETED.
 MUST VERIFY THAT THE PAYMENTS ARE EQUAL OPPOSITES;

data del_can_pairs (keep=rcd_id rf_dbc rf_ndup);
  merge pos_pairs(in=a) neg_pairs(in=b);
  by pnum patid rbirthyr rbirthmo rbirthdd rsex ndc fillyr fillmo filldd tot_bill key_cnt;
  length rf_ndup $1; *Used later to flag duplicates with negative members-Needs it for PROC
APPEND;
   rf_ndup='0';
  if a and b then do;
     posneg_bc = sum(rbchrge, neg_rbchrge);
     posneg_pay = sum(rpayment,neg_rpayment);
     if (-1 < posneg_bc < 1) or (-1 < posneg_pay < 1) then rf_dbc = '1';
         rcd_id=neg_rcd_id;
     output; *negative member of the pair;
     rcd_id=pos_rcd_id;
     output; *positive member of the pair;
  end;
run;


proc sort data=del_can_pairs;
 by rcd_id;
run;

proc sort data=all_pairs;
 by rcd_id;
run;

*DELETE RECORDS FROM PAIRED RECORDS THAT CANCEL EACH OTHER OUT;
DATA CHK_POS_DUPES(DROP=NEG_FIN);
 merge del_can_pairs(in=a) all_pairs(in=b);
 by rcd_id;
 if b and not a then output chk_pos_dupes;

*RE-SORT THE DATA TO SEE IF RECORDS HAVE SAME FINANCIAL CHARGES;

proc sort data=chk_pos_dupes;
 by pnum patid rbirthyr rbirthmo rbirthdd rsex ndc fillyr fillmo filldd  rbchrge;
run;

*NOW IDENTIFY REMAINING DUPLICATE RECORDS - IF DUPLICATES EXIST, ONLY KEEP FIRST RCD;

data dupe_chk(keep=rcd_id rf_dbc rf_ndup);
  set chk_pos_dupes(drop=rf_ndup); *dropping rf_ndup is important for the RETAIN statement;
 by pnum patid rbirthyr rbirthmo rbirthdd rsex ndc fillyr fillmo filldd rbchrge;
```

```
 retain rf_ndup;
  if first.filldd and not(last.filldd) then do;
     if rbchrge<0 then rf_ndup='1';  *If negative charge exists, it is the first record of the
key;
     else rf_ndup='0';
 end;
if first.filldd and last.filldd then do;
   rf_dbc='4'; *Adjusted;
   *rf_ndup=''; *this flag is not applicable for adjusted data*;
 end;
 else if first.rbchrge and last.rbchrge then do;
   rf_dbc='3'; *partial dupes;
  end;
  else if not(first.rbchrge and last.rbchrge) then do;
                rf_dbc='2';
  end;
run;

*CONCATENATE THE RECORDS THAT HAD CANCELED PAIRS WITH THE DE-DUPED RECORDS TO CREATE
 THE DBC FORMAT;

proc append base=del_can_pairs data=dupe_chk;
run;

/***RF_dbc****/

data dbc(keep=start label fmtname);
 set del_can_pairs end=eof;
 start=put(rcd_id,8.);
 label=rf_dbc;
 fmtname='$rf_dbc';
 output;
 if eof then do;
  fmtname='$rf_dbc';
  start='other';
  label='0';
  output;
 end;
run;
proc sort data=dbc nodupkey;
 by start;
run;

proc format cntlin=dbc;
run;


/***RF_dbc****/

data dup(keep=start label fmtname);
 set del_can_pairs end=eof;
 start=put(rcd_id,8.);
 label=rf_ndup;
 fmtname='$rf_ndup';
 output;
 if eof then do;
  fmtname='$rf_ndup';
  start='other';
  label='0';
  output;
 end;
run;
proc sort data=dup nodupkey;
 by start;
run;

proc format cntlin=dup;
run;


data outdata1.&payernum.rx4(drop=rcd_ida tot_bill compress=binary);
```

```
 length rf_dbc $1;
 set rx4;
 rcd_ida=put(rcd_id,8.);
 rf_dbc=put(rcd_ida,$rf_dbc.);
 rf_ndup=put(rcd_ida,$rf_ndup.);

run;
proc freq data=outdata1.&payernum.rx4;
 table rf_dbc /list missing;
 format rf_dbc $dbcfmt.;
 title "&payernum - Debit Credit ";
run;


proc freq data=outdata1.&payernum.rx4;
 table rf_dbc*rf_ndup /list missing;
format rf_dbc $dbcfmt.;
 title "&payernum - Check duplicates";
run;
proc print uniform data=outdata1.&payernum.rx4(obs=50);
 var pnum patid rbirthyr rbirthmo rbirthdd rsex ndc fillyr fillmo filldd rbchrge rf_dbc rf_ndup;
 title "Sample obs from &payernum.rx4";
run;

proc contents data=outdata1.&payernum.rx4;
 title "contents of &payernum.rx4";
run;

*%jobend;

/*
dm "log;    print file = D:\Mcdb\DataCol\RX_Step4_7\log_out\Debit_Credit_Analysis_&payernum..log
replace";
dm "output; print file = D:\Mcdb\DataCol\RX_Step4_7\log_out\Debit_Credit_Analysis_&payernum..lst
replace";
*/
```

```
*%jobstart;

/*
*==============================================================================
*
*  Program Name:  D:\MCDB\DataCol\RX_STEP4_7\NDC_Categories_Assignment.SAS
*
*  Description:   Change NDC field to 12-characters and Create NDC9 and its
*                 corresponding NDC2 whenever the payer's labeler-product code matches
*                 The FDA's labeler product codes.
*
*  Input files:   I:\MCDB\DATACOL\&PAYERNUM\DATA\&PNUM.RX
*                 SASSIFIED RX AND SAVED IN THE PAYER-SPECIFIC DIRECTORY
*
*  Format file:   I:\MCDB\DATACOL\NDC\multum\data\multum_ndc_2003
*                 July 06, 2003 Denormalized Multum NDC file

*  Output files:  I:\MCDB\DATACOL\&PAYERNUM\DATA\&PAYERNUM.RX0 -
*
*
*  08/20/01 Added drug classes to the file.
*  06/30/06 Changed name of input file, added rf_MedSup, Category_ID
*
*==============================================================================
*/

/*
%LET PayerNUM=P070;
%let curyr4=2006;
*/

LIBNAME  F  'D:\Mcdb\DataCol\Formats\ndc\Multum\data';
LIBNAME  RX "d:\MCDB\DATACOL\&PAYERNUM\sasdata" access=readonly;
LIBNAME  RX0 "d:\MCDB\DATACOL\&PAYERNUM\sasdata";

footnote1 "%sysfunc(putn(%sysfunc(time()), hhmm.)) %sysfunc(putn(%sysfunc(date()), weekdate.))";
footnote2 "program: D:\Mcdb\DataCol\RX_Step4_7\prog\NDC_Categories_Assignment.sas";


*************************************************************;
* 1. Create NDC9 and NDC2                                   *;
*************************************************************;

data _ndc(rename=(ndc_code=ndc));
  set f.multum_hcfa_ms_ndc_&curyr4 (keep=ndc_code dc1-dc4 rf_MedSup Category_ID gbo apprdate
               main_multum_drug_code drug_id);
  length ndc9 $9 ndc2 $2  MMDC $5;
  NDC9=substr(ndc_code,1,9);
  NDC2=substr(ndc_code,10,2);
  MMDC=trim(put(main_multum_drug_code,5.));
 if compress(MMDC)='.' then MMDC='';

  label NDC9="09. Labeler-Product Code"
        NDC2="09. Package Code"
        MMDC="Main Multum Drug Code"
        DRUG_ID="Multum Drug ID";
drop main_multum_drug_code;
run;


proc sql;
create table rx0.&payernum.rx0 as
select a.*, b.*
from rx.&payernum.rx a
left join _ndc b
on a.ndc=b.ndc;
quit;

run;
```

```
proc freq data=rx0.&payernum.rx0;
  where ndc9="";
  tables ndc9/missing;
  title "Count of invalid NDC codes  after Processing Step 3 for Payer &PAYERNUM";
run;

proc freq data=rx0.&payernum.rx0;
  where DC1=' ';
  tables DC1/missing;
  title "Count of Missing Drug Category 1 after Processing Step 3 for Payer &PAYERNUM";
run;


proc freq data=rx0.&payernum.rx0;
  tables apprdate/missing;
  format apprdate year4.;
  title "Freq by approval date (year)";
run;

proc contents data=rx0.&payernum.rx0;
  title "Contents of RX file after Processing Step 3 for Payer &PAYERNUM";
run;


proc print data=rx0.&payernum.rx0 (obs=50);
  title "Dump of 50 records of RX file after Processing Step 3 for Payer &PAYERNUM";
run;


*%jobend;

/*
*** save log and output files;
dm'log;   print file=d:\MCDB\DATACOL\&PAYERNUM\log_out\&PAYERNUM.NDC_Categories_Assignment.log
replace';
dm'output;print file=d:\MCDB\DATACOL\&PAYERNUM\log_out\&PAYERNUM.NDC_Categories_Assignment.lst
replace';
*/
```

```
*%JOBSTART;
/*
*==============================================================================
*
*  Program Name:  ENCTLINKrx.SAS
*
*  Description:   Step 5: Examine the links: Encounter file to the RX file
*                                            RX file to the Provider file.
*
*                 Due to the change in the linkage between Provider and RX data,
*                 this is a re-write of original ENCTLINKRX.sas.  It keeps
*                 the first part of the original which checks the link between
*                 Encounter and RX data, but then matches the RX DEA to a cross
*                 reference file of Provider DEA/FEDTAXID'S.  The specialty codes,
*                 rf_multi indicator and what type of linkage (Provider DEA or
*                 Provider Fedtaxid) are linked back to the RX file.
*
*                 2002 - Modified to execute three formats (DEANUM, PROVID, FEDTAXID) instead
*                        of one to get the Provider information.
*                 2002-  Modified to extract delivery system and coverage type info from
*                        the encounter file.
*                        This program uses encounter data summarized at the patient level.
*                        The program Create_Encounter_Patient_Link_Driver.sas needs to be run
*                        before Step 4-7. AN 8/17/2003.
*                 2006   took out the need for the Create_Encounter_patient_Link_Driver.  Now,
*                        this will be done on a payer by payer basis.
*
*  Instructions:  USE THE PROGRAM WITH RUN_STEP4_7.SAS.
*
*  Input files:   PxxxRX1   -2002 Pharmacy after NDC code Examination & ZD conv
*                 PxxxENC   -2002 ENCOUNTER DATABASE AFTER STEP 2
*                 DEA_XREF  -2002 FORMAT OF PROVIDER XREF CONTAINING DEA & SPECLTIES
*                 PxxxREC   -2002 Patient Level Ecounter summary.
*  Output files:  PxxxRX1B   -2001 RX/PROVIDER LINK DATABASE
*
*  DATE MODIFIED: 10/31/2002
*                 7/18/2003
*                 8/16/2003
*                 9/5/2003 - call to $patsex format to correct P160 sex field
*                 9/16/2003 - call to $patzip format to correct P660 zipcode
*  Modified 7/7/06 JMAY - moved RPHARMCTY RF_SELF RN_ZIP_S CHNIND, RPHARMGRP FROM
*    MCDB_RX to Enctlinkrx
*  Modified 7/17/06 AN - added enrollment and disenrollment date.
*==============================================================================
*/

OPTIONS NOCENTER;

/*
LIBNAME LIBRARY "D:\Mcdb\DataCol\Formats\Permanent";      ***Format Lib*;

%let PAYERNUM=P500;

LIBNAME  INdata1     "d:\MCDB\DATACOL\&PAYERNUM\sasDATA" access=readonly;    ***Enc and provider
files*;
LIBNAME  RXIN     "d:\MCDB\DATACOL\&PAYERNUM\sasDATA" access=readonly;    ***RX file*;
LIBNAME  ENCIN     "I:\MCDB\DATACOL\&PAYERNUM\SASDATA" access=readonly;  ***Encounter patient
summary files;
LIBNAME  OUTdata1    "d:\MCDB\DATACOL\&PAYERNUM\sasDATA";
LIBNAME  OUTdata2    "d:\MCDB\DATACOL\&PAYERNUM\sasDATA";
*/
option nosource2;
%include "D:\Mcdb\DataCol\Formats\Format_Chain_indicator.sas";
%include "I:\MCdb\datacol\Formats\mdcyzipfmt.sas";
%include "D:\MCdb\datacol\Formats\Format_MD_Pharmacy_ZIP.sas";
option source2;

************************************************;
* Macro variable assigned in Run_Step4_7.sas:   ;
*   PAYERNUM - PxxxSxxx payer who sent the data ;
```

```
************************************************;
TITLE1 "MARYLAND HEALTH CARE COMMISSION";
TITLE2 "STEP 5: VERIFY LINK BETWEEN RX AND ENCOUNTER /PROVIDER FILE - &PAYERNUM";
footnote1 "%sysfunc(putn(%sysfunc(time()), hhmm.)) %sysfunc(putn(%sysfunc(date()), weekdate.))";
footnote2 "program: D:\Mcdb\DataCol\RX_Step4_7\prog\EnctlinkRX.sas";

proc format;
  value $plink
  ' '='No Link'
  '1'='Link By DEANUM'
  '2'='Link By Provid'
  '3'='Link By Fedtaxid';
  value $elink
  ' '='No Link'
  '1'='Link By Patid-DOB-SEX';

  value $multi
   ' '='No Link'
   '0'='Non-Multiple Specialties'
   '1'='Multiple Specialties';
run;

************************************************************************;
*                                                                      ;
*  Check the link between Encounter and pharmacy data by PATID-DOB-SEX   ;
*                                                                      ;
************************************************************************;

*************************************************************************
 Summarize Encounter data by pidbdg/delivtyp, pidbdg/covtype
 *************************************************************************;
proc means data=encin.&PAYERNUM._enc1 missing noprint;
class pidbdg delivtyp covtype enrolldt disenrolldt;
types pidbdg
      pidbdg*delivtyp
      pidbdg*covtype
      pidbdg*enrolldt*disenrolldt;
output out=sumenc(rename=(_freq_=count)) N=;
run;

 data totserv(keep=pidbdg count)
     delv(keep=pidbdg delivtyp count)
     covr(keep=pidbdg covtype count)
         enroll1(keep=pidbdg enrolldt disenrolldt count);
 set sumenc;
 if _type_=20 then output covr;
 else if _type_=24 then output delv;
 else if _type_=19 then output enroll1;
 else if _type_=16 then output totserv;

 *** Whenever a patient has 2 or more enrollment date, choose the earliest. Choose the most
    recent disenrollment date. We need a temporary holder patients who have not disenrolled.
    Otherwise a missing will not be captured as the most recent disenrollment date   ***;

 data enroll2;
  set enroll1;
  if disenrolldt=. then disenrolldt='30jun2006'd;
 run;


 proc summary nway data=enroll2;
 class pidbdg;
 var enrolldt disenrolldt;
 output out=enroll (drop=_type_ _freq_)
 min=min_enrolldt min_disenrolldt max=max_enrolldt max_disenrolldt;
 run;

***Reset back all dates >= 4/30/2006 (last day of reporting) to missing***;

data enroll_fin;
  set enroll(keep=pidbdg min_enrolldt max_disenrolldt);
```

```
  by pidbdg;
  if max_disenrolldt>='30apr2006'd then max_disenrolldt=.;
  rename min_enrolldt=enrolldt   max_disenrolldt=disenrolldt;
run;

proc sort data=delv; by pidbdg descending count;
run;

*retain delivtyp with highest count;
data delv_fin(drop=count);
  set delv;
  by pidbdg;
  if first.pidbdg;
run;

proc sort data=covr; by pidbdg descending count;
run;

*retain covtype with highest count;
data covr_fin(drop=count);
  set covr;
  by pidbdg;
  if first.pidbdg;
run;

*merge four files together
 will have the most common delivtyp, covtype and encounter service count by
 unique recipients;
data &PAYERNUM.rec(rename=(count=encserv));
 merge totserv delv_fin covr_fin enroll_fin;
 by pidbdg;
run;

proc print data=&PAYERNUM.rec(obs=25);
title3 "Sample print of &PAYERNUM Patient IDs";
run;

proc freq data=&PAYERNUM.rec;
 table delivtyp covtype /list missing;
title3 "Freq of &PAYERNUM Delivery System and Coverage Type";
run;

proc means n nmiss min max sum data=&PAYERNUM.rec;
title3 "Count of &PAYERNUM total services";
run;

 ************************************************************************;
*                                                                        ;
*  Check the link between Encounter and pharmacy data by PATID-DOB-SEX   ;
*                                                                        ;
************************************************************************;

proc sort data=RXIN.&PAYERNUM.RX2 out=PHARM;
  by PIDBDG ;
run;

****************************************************;
*  Check the link between Encounter data           *;
*  and pharmacy data by PATID-DOB-SEX              *;
****************************************************;

data OUTdata2.P%substr(&PAYERNUM,2,3)RX3;
  merge &PAYERNUM.REC(in=a)
        PHARM (in=b)
  ;
  LENGTH RF_ELINK $1;
  by PIDBDG;
  if  b;
  if  a then RF_ELINK="1";

* add new flag, RF_COV17, based on covtype;
```

```
  length RF_COV17 $1;
  if covtype in ('1', '7') then RF_COV17 = '1';
  else RF_COV17 = '0';

length RPHARMCTY $3 RF_SELF $1 RN_ZIP_S $5;
  ******************************************************
Add new flag rf_self -- 1 - self insured coverage type
                        0 - fully insured
                        2 - other
******************************************************;
  if covtype in ('3','8') then RF_SELF = '1';
    else if covtype in ('1','2','4','5','6','7') then RF_SELF = '0';
    else RF_SELF = '2';

*********************************************************************************************
Add new variables rn_zip_s and Rpharmcty using NACDS data (sent by Linda Bartnyska on 11/16/05)
*********************************************************************************************;
  ncpdp6=substr(NCPDP,1,6);
  RN_ZIP_S=put(ncpdp6,$phzip.);
  if RN_ZIP_S="00000"  then RN_ZIP_S=PHAZIP;
  RPHARMCTY=put(RN_ZIP_S,$mdcyzip.);

  Label RF_COV17   = "Medicare Coverage Flag"
        COVTYPE  = "Coverage Type (From Encounter File)"
        DELIVTYP = "Delivery System(From Encounter File)"
        RPHARMCTY = "Pharmacy County"
               RF_SELF   = "Self Insurance Coverage Indicator"
               RN_ZIP_S  = "Adjusted Pharmacy Zip Code (Using NACDS data)";

*************************************************************
Add Chain indicator based on file sent by Linda on 11/16/2005.
*************************************************************;
  length CHNIND $1;
  CHNIND=put(NCPDP,$chnind.);

***Add additional MD Rx using pharmacy zip and the 1st 2 digit of NCPDP****;
  if CHNIND="O" and(put(PHAZIP,$mdstzip.)='1' OR
                    substr(NCPDP,1,2)="21") then CHNIND="X";
  if RF_MORDER='1' then CHNIND="M";

  Label CHNIND="Chain Indicator";

*************************************************************************
Add Pharmacy group based e-mail from Linda, Ben, and Claudia on 11/16/2005
*************************************************************************;
  length RPHARMGRP $1;
  if chnind in ("C","X")     then RPHARMGRP="1";        *Chain;
  if chnind in ("A","I","U") then RPHARMGRP="2";        *Independent;
  if chnind in ("G","D")     then RPHARMGRP="3";        *Supermarket/Mass Market;
  if chnind in ("M")         then RPHARMGRP="4";        *Mail order;
  if chnind in ("O")         then RPHARMGRP="9";        *Other-non MD;

  Label RPHARMGRP="Pharmacy Group";

 ****************************************************;
*  Check the link between Provider data              *;
*  and pharmacy data by DEANUM, PROVID, and FEDTAXID *;
*  The formats used were created by the program      *;
*  Create_xref_format.sas                            *;
****************************************************;

length rf_plink rf_elink rf_multi $1 speclty1-speclty3 $3 payer $4
        dea_payer $15;
 payer=upcase(substr("&PAYERNUM",1,4));
 dea_payer=payer||deanum;

 label
   rf_elink = 'Encounter Linkage Indicator'
   rf_plink = 'Provider Linkage Indicator'
   rf_multi = 'Multiple Specialties Indicator'
   speclty1 = 'Practitioner/Supplier Specialty-1'
```

```
   speclty2 = 'Practitioner/Supplier Specialty-2'
   speclty3 = 'Practitioner/Supplier Specialty-3';

 xref_link=put(deanum,$deaxref.); /* first, link by DEANUM */
 if xref_link=' ' then
  xref_link=put(dea_payer,$prvxref.); /* then, link by PROVID */
 if xref_link=' ' then
  xref_link=put(dea_payer,$taxxref.); /* then, link by FEDTAXID */

 if xref_link=' ' then do;
   rf_plink=' ';
   rf_multi=' ';
   speclty1='   ';
   speclty2='   ';
   speclty3='   ';
 end;
 else do;
   rf_plink=substr(xref_link,1,1);
   rf_multi=substr(xref_link,2,1);
   speclty1=substr(xref_link,3,3);
   speclty2=substr(xref_link,6,3);
   speclty3=substr(xref_link,9,3);
 end;

run;

****Statistics;

proc sql;
create table elinkd as
select count(distinct pidbdg) as Recip,  rf_elink
from OUTdata2.P%substr(&PAYERNUM,2,3)RX3
group by rf_elink;
quit;

proc sql;
create table elinks as
select count(distinct pidbdg) as TRecip
from  OUTdata2.P%substr(&PAYERNUM,2,3)RX3;
quit;

data elink;
set elinkd;
if _n_=1 then
set elinks;

PercLink=Recip/Trecip;
run;

proc print data=elink label;
var rf_elink recip PercLink;
sum recip PercLink;
label recip= "RX Unique Recipients" PercLink="% Link" ;
format rf_elink $elink. recip comma8.  percLink percent6.2;
title5 "Number of Recipients (PATID-DOB_SEX ) by Encounter Link Indicator";
run;

proc freq data=OUTdata2.P%substr(&PAYERNUM,2,3)RX3;
  table rf_elink rf_plink
        rf_multi /list missing;
  format rf_elink $elink. rf_plink $plink. rf_multi $multi.;
 title5 "Frequencies of Selected Variables from RX3 - Prescription services";
run;

proc print data=OUTdata2.P%substr(&PAYERNUM,2,3)RX3(obs=15);
 title5 "Sample Print of RX3";
run;

proc contents data=OUTdata2.P%substr(&PAYERNUM,2,3)RX3;
 title5 "Contents of RX3";
run;
```

```
 /*
*** save to the place you specified ***;
dm 'log; print file = d:\MCdb\DataCol\&PAYERNUM\log_out\P%substr(&PAYERNUM,2,3)EnctlinkRX.log
replace';
dm 'output; print file = d:\MCdb\DataCol\&PAYERNUM\log_out\P%substr(&PAYERNUM,2,3)EnctlinkRX.lst
replace';
*/

*%JOBEND;
```

```
*%JOBSTART;
/*
*==============================================================================
*
*  Program Name:  i:\MCdb\DataCoP4l\RX_STEP4_7PGM00\StatChkRX.SAS
*
*  Date created:  5/15/2000
*
*  Description:   New Step 7: EVALUATE THE QUALITY OF EXCLUSION VARIABLES
*                 USING THE pharmacy DATABASE.
*
*  Instructions:  TO USE THE PROGRAM:
*                 1. Copy the program to the payer specific directory,
I:\MCdb\DataCol\PxxxSxxx\Programs\
*                    (for PxxxSxxx see the list of payer posted on the wall).
*                 2. Change:
*                    PAYERNUM=PxxxSxxx -> number assigned to payer
*
*  Input files:   PAYERNUM rx4  - Current Pharmacy DATABASE
*
*  DATE MODIFIED: 11/01/02
*                 added macro variables to process year to year data
*                 updated format for f_link
*
*  DATE MODIFIED: 07/28/03
*                 changed call to stnamefmt to mdstzipfmt, all inputs/outputs are to I drive.
*
*  Needed:      a) zip code,diag code,proc code (home grown),
*               b) diag code, srvtype,f_cap, f_uncv
*                       instate, payergp, region
*  DATE MODIFIED: 08/16/03
*                 added stats for delivery systems, coverage type, rf_finzr, linkage and
*                 outlier stats
*  DATE MODIFIED: 08/02/04
*                 taking out the Displays from old Step 5 Enctlinkrx - done separately
*                 added displays for debit/credit analysis
*  DATE MODIFIED: 7/26/05 - J May
*                 Moved Freq of RF_NDC from Step 8
*                 Moved Freq if RF_NDRUG, RF_MORDER, R_NAGE12&CURYR2 from Create_RX_Analysis
*                 Added Freq of New VariablesNCPDP_ST, RF_DEA, DEABUS
*==============================================================================
*/

OPTIONS NOCENTER;


*******************************
* Macro variable:
*   PAYERNUM - payer who sent the data
**;

/*
%LET PAYERNUM=P070;
%LET CURYR2=05;
%LET CURYR4=2005;
%LET NXTYR2=06;

LIBNAME  INdata1     "d:\MCDB\DATACOL\&PAYERNUM\SASDATA";    ***Enc and provider files*;
LIBNAME  RXIN     "d:\MCDB\DATACOL\&PAYERNUM\SASDATA";    ***RX file*;
LIBNAME  ENCIN     "d:\MCDB\DATACOL\&PAYERNUM\SASDATA";  ***Encounter patient summary files;
LIBNAME  OUTdata1    "d:\MCDB\DATACOL\&PAYERNUM\SASDATA";
LIBNAME  OUTdata2    "d:\MCDB\DATACOL\&PAYERNUM\SASDATA\";
LIBNAME LIBRARY "D:\Mcdb\DataCol\Formats\Permanent";
*/

TITLE1 " MARYLAND HEALTH CARE COMMISSION";
TITLE2 " 2005 RX DECRIPTIVE STATISTICS USING pharmacy DATABASE (&PAYERNUM)";

footnote1 "%sysfunc(putn(%sysfunc(time()), hhmm.)) %sysfunc(putn(%sysfunc(date()), weekdate.))";
footnote2 "program: D:\Mcdb\DataCol\RX_Step4_7\prog\StatChkRX.sas";
```

```
************************************;
*   Create blank variables to left *;
*   adjustify title statements     *;
************************************;
data _null_;
  length lBlank $40 rBlank $120;
  lBlank=repeat(' ',39);
  call symput('lBlank',lBlank);
  rBlank=repeat(' ',119);
  call symput('rBlank',rBlank);
run;


*************************************************************************;
*    1. Create flags                                                    ;
*       Rf_pzip - 0=valid 1=invalid                                     ;
*       Rf_phZip - 0=valid 1=invalid                                    ;
*************************************************************************;
data ENCNT;
  set indata1.P%substr(&PAYERNUM,2,3)RX4 end=eof;
  length Rf_pzip Rf_phZip $1;

  if trim(left(Rpatzip))='' then Rf_pzip='2';
  else if upcase(put(rPATZIP,$mdstzip.))='1' then rf_pzip='0';
   else rf_pzip='1';

  if trim(left(phazip))='' then Rf_phZip='2';
  else if upcase(put(phaZip,$uszip.))="1" then Rf_phZip="0"; *Pharmacies can be anywhere in the
country;
  else  Rf_phZip="1";

*CONVERT "OTHER" NCPDP STATE CODES TO 99 TO FACILITATE FREQ ORDER;
  length NCPDPST $2;
  IF NCPDP_ST NOT IN('  ','08','09','21','39','49','51') THEN NCPDPST='99';
  ELSE NCPDPST=NCPDP_ST;

*GET YEAR FROM APPRDATE FOR FREQS;
  APPYR=YEAR(APPRDATE);

run;

*************************************************************************;
*   2. Check the four financial vars -                                  ;
*      rBCHRGE,rREIMAMT,rPATLIAB                                        ;
*************************************************************************;
proc means data=indata1.P%substr(&PAYERNUM,2,3)rx4 maxdec=2 n mean std min max p25 p50 p75 p95;
  label rBCHRGE='' rREIMAMT='' rPATLIAB='';
  var rBCHRGE rREIMAMT rPATLIAB;
  title3 "1. Check basic statistics of the three financial vars";
  title5 " (Rule of thumb: BCHRGE =  REIMAMT + PATLIAB)";
  title6 "&lBlank For example of 2001 P620,";
  title7 "&lBlank Mean of BCHRGE  = $53.92";
  title8 "&lBlank Mean of REIMAMT = $40.10";
  title9 "&lBlank Mean of PATLIAB = $13.82";
run;

proc freq data=indata1.P%substr(&PAYERNUM,2,3)rx4;
  tables (rBCHRGE rREIMAMT rPATLIAB) /list missing;
  tables rBCHRGE*rREIMAMT*rPATLIAB /list missing;
  format rBCHRGE rREIMAMT rPATLIAB CHGfmt.;
  title3 "2. Check the ranges of three financial vars";
run;

proc freq data=encnt;
  tables rf_bcout /list missing;
  tables rf_paout /list missing;
  tables rf_bcout*rf_paout /list missing;
  tables rf_10K /list missing;
  format rf_bcout $f_bcout. rf_paout $f_paout. rf_10K $rf_10K.;
```

```
 title3 "3. Check for Financial Variables Outliers";
run;



proc freq data=encnt;
  tables rf_finodd /list missing;
  format rf_finodd $f_finodd.;
 title3 "4. Check for Bill Overpayment";
run;

proc freq data=encnt;
 where rf_finodd='1';
  tables pex_ind /list missing;
  format pex_ind $pex_fmt.;
   title5 "***Percentage of Bill Overpayment (Rf_Finodd=1)***";
run;

proc freq data=encnt;
  tables rf_finms rf_finzr /list missing;
  tables Rf_paym /list missing;
  tables rBCHRGE rREIMAMT rPATLIAB /list missing;
  format Rf_paym $f_paym. ;
  format rf_finms $f_finms. rf_finzr $f_finzr.;
  format rBCHRGE rREIMAMT rPATLIAB fmtNine.;
 title3 "5. Check for Other Financial Variables Anomalies";
run;



proc means data=indata1.P%substr(&PAYERNUM,2,3)rx4 maxdec=2 n mean std min max p25 p50 p75 p95;
   label
     drugcmp = ''
     DRUGQTY = ''
     DRUGSPL = ''
     rAGE = ''
     rAGE12&curyr2 = ''
        R_NAGE12&curyr2 = ''
     rPayment = ''
   ;
   var drugcmp DRUGQTY DRUGSPL rAGE rAGE12&curyr2 R_nage12&CURYR2 rpayment;
 title3 "6. Check Mean of Other Numerical Variables";
run;

PROC FREQ DATA=ENCNT;
 TABLE RF_NDC RF_NDRUG MULTDC1 GBO /LIST MISSING;
  FORMAT GBO $GBO. rf_ndrug $ndrug. RF_NDC $F_NDC. MULTDC1 $MD3CAT.;
TITLE3 '7. Check Drug Code Variables';
RUN;

proc freq data=encnt;
  tables NCPDPST RF_MORDER /LIST MISSING;
format RF_MORDER $F_MOPH. NCPDPST $SSAST.;
TITLE3 '8. Check National Council of Prescription Drug Programs ( NCPDP) Number';
RUN;

proc freq data = ENCNT;
  tables NCPDP / list missing;
  where RF_MORDER = '1';
  TITLE4 '***Check for Mail Order (NCPDP)***';
run;

PROC FREQ DATA=ENCNT;
 TABLE R_NAGE12&CURYR2 RAGE RAGE12&CURYR2 RSEX RF_YR65 /list missing;
 FORMAT Rage Rage12&curyr2 agecat. R_NAGE12&CURYR2 Nagecat. rf_yr65 $rf_65yr. rsex $sex.;
TITLE3 '9. Check Age and Gender';
RUN;

PROC FREQ DATA=ENCNT;
 TABLES RF_DEA DEABUS/list missing;
 FORMAT RF_DEA $DEAFMT. DEABUS $BUSNAM.;
```

```
TITLE3 '10. Check DEA Number';
RUN;

PROC FREQ DATA=ENCNT;
  tables Rd_dat_f / list missing;
  format Rd_dat_f $d_dat_f.;
TITLE3 '11. Check Data Submission Range';
RUN;

PROC FREQ DATA=ENCNT;
  tables Rbirthyr Rbirthmo fillyrx*fillmo fillyrx fillmo filldate Rf_yr&NXTYR2 APPYR enrolldt
disenrolldt /LIST MISSING;
FORMAT APPYR APPYR. RBIRTHYR $YEAR. enrolldt disenrolldt filldate YEAR4.;
TITLE3 '12. Check Date Variables';
RUN;

PROC FREQ DATA=ENCNT;
 TABLES Rf_pzip Rf_phZip RF_MD /list missing;
 FORMAT Rf_pzip $d_zip_f. Rf_phZip $f_valid. ;
TITLE3 '13. Check Zip Code Variables';
RUN;

PROC FREQ DATA=ENCNT;
 tables rf_patid  /list missing;
  format rf_patid $f_patid. ;
TITLE3 '14. Check Patient ID';
RUN;


proc freq data=encnt;
 table rf_dbc /list missing;
 format rf_dbc $dbcfmt.;
 title3 "15. &payernum - Debit Credit Analysis";
 run;

proc freq data=encnt;
 table RMSCAT rf_medsup * RMSCAT /list missing;
 format RMSCAT $MS_CATEGORY. rf_medsup $medsup.;
 title3 "16. &payernum - Medical Supply Category";
 run;

 proc sql;
create table elinkd as
select count(distinct pidbdg) as Recip,  rf_elink
from indata1.P%substr(&PAYERNUM,2,3)RX4
group by rf_elink;
quit;

proc sql;
create table elinks as
select count(distinct pidbdg) as TRecip
from  indata1.P%substr(&PAYERNUM,2,3)RX4;
quit;

data elink;
set elinkd;
if _n_=1 then
set elinks;

PercLink=Recip/Trecip;
run;

proc print data=elink label;
var rf_elink recip PercLink;
sum recip PercLink;
label recip= "RX Unique Recipients" PercLink="% Link" ;
format rf_elink $elink. recip comma8.  percLink percent6.2;
title3 "17. Number of Recipients (PATID-DOB_SEX ) by Encounter Link Indicator";
run;

 proc freq data=indata1.P%substr(&PAYERNUM,2,3)RX4;
```

```
  table rf_elink covtype delivtyp /list missing;
  format rf_elink $elink.;
 title3 "18. Check RX-Encounter Link - Services";
run;

proc freq data=indata1.P%substr(&PAYERNUM,2,3)RX4;
  table rf_plink /list missing;
  format rf_plink $plink.;
 title3 "19. Check RX-Provider Link";
run;


****************************************************;
*    Sample print                                  ;
****************************************************;
proc print data=indata1.P%substr(&PAYERNUM,2,3)rx4 (obs=10);
  title3 "20. Sample print";
run;


proc contents data=indata1.P%substr(&PAYERNUM,2,3)rx4 ;
  title3 "21. Contents of RX4";
run;

****************************************************;
*     print special case                           ;
****************************************************;
proc print data=indata1.P%substr(&PAYERNUM,2,3)rx4;
  where (pnum='P710' and rreimamt in (9999.99,-9999.99) ) or
        (pnum='P471' and rreimamt in (9999.99,-9999.99) ) or
        (pnum='P310' and rpayment in (9999.99,-9999.99) ) or
        (pnum='P480' and rpatliab >=500000 )
        ;
  title3 "22. Sample print for special case";
run;

*%JOBEND;

/*
*** save log and list files;
dm'log;   print file=i:\MCdb\DataCol\&PAYERNUM.\log_out\P%substr(&PAYERNUM,2,3)StatChk.log
replace';
dm'output;print file=i:\MCdb\DataCol\&PAYERNUM.\log_out\P%substr(&PAYERNUM,2,3)StatChk.lst
replace';
*/
```

```
%jobstart;
*******************************************************************;
*                                                                 ;
*  Program Name:  D:\MCdb\DataCol\RX_Step8\Step8_RX_AllPayers_0.SAS   ;
*                                                                 ;
*  Programmer:    Adrien Ndikumwami                                ;
*                                                                 ;
*  Description:   Scan through the directories for RX files       ;
*                 and concatenate them into one file that will be ;
*                 for initial statistical analysis.               ;
*                                                                 ;
*  DATE CREATED: 12/16/2002 by Adrien Ndikumwami                  ;
*                                                                 ;
*                                                                 ;
*******************************************************************;
options nodate ls=148 ps=60 pageno=1;

Libname db "D:\MCDB\datacol\RX_step8\Data";
LIBNAME LIBRARY 'D:\MCDB\PHARMACY\FORMATS';

****************************************;
*   Create macro for MHCC title         *;
****************************************;

Title1 "MARYLAND HEALTH CARE COMMISION - PHARMACY STEP 8 -- Run on &sysday, &sysdate at
&systime";


*******************************************************************;
*  Steps:                                                         *;
*  1. Find all the directories used by  MCDB data and create a    *;
*     a libname including all those directories.                  *;
*  2. Find all RX files present in those directories              *;
*  3. Concatenate all the files                                   *;
*                                                                 *;
*******************************************************************;

*******************************************************************;
*  1. Find all the directories used by  MCDB data and create a    *;
*    a libname including all those directories                    *;
*******************************************************************;

OPTIONS NOXWAIT;

x 'DIR/A:d D:\MCDB\DATACOL\P??? > d:\WORKAREA\TEMP_DIR.TXT';


data one;
  infile  "D:\WORKAREA\TEMP_DIR.TXT" firstobs=6;
  input  @40 payer $4.;
  If upcase(substr(payer,4,1)) = "X" or payer="" then delete;
run;

data _null_;
  set one end=eof;
  call symput('p'||left(_N_), payer);
  if eof then call symput('nobs',_N_);
run;


************Start macro to assign libname*****************;
**********************************************************;


%macro payrinfo;

  libname rx ( %do i=1 %to &nobs;
                              "D:\MCDB\DATACOL\&&p&i\sasdata",
               %end;
                              "D:\MCDB\DATACOL\&&p&nobs\sasdata");
```

```
%mend payrinfo;

%payrinfo;



*********************************************************************;
*  2. Find all RX files present in those directories                 *;
*********************************************************************;

PROC CONTENTS NOPRINT DATA=rx._ALL_
                       OUT=two(KEEP=MEMNAME nobs);
run;

proc sort nodupkey data=two out=three;    *keep 1 record per payer;
   by memname ;
   where substr(memname,5,3)="RX4";
run;

proc print data=three;
  sum nobs;
  title2 "Number of observations for each payer before concatenation";
  label memname="Payer file" nobs="No. of Records";

run;

data _null_;
  set three  end=eof;
  call symput('pfn'||left(_N_), memname);  * Macro var for payers' filenames;
  if eof then call symput('npayer',_N_);   * Macro var for total number of payers;
run;


*********************************************************************;
*  3. Concatenate all the files                                      *;
*********************************************************************;


******Start macro to append all the files using payer filename********;
***************automatically generated by call symput*****************;
*********************************************************************;


%macro appayer;    *to append all payers files;

  **First file in sequence serves as base for appending;

  %do j=1 %to 1;
  %put ######### Getting   &&pfn&j ###########;
     data db.mcdbrx06_0;
     set rx.&pfn1  ;
   run;
   %end;

  *Append file 2 to last to the base;

  %do j=2 %to &npayer;
  %put ######### Appending   &&pfn&j ###########;

      proc append base= db.MCDBRX06_0  data= rx.&&pfn&j;
  run;
  %end;

%mend appayer;

%appayer;

PROC PRINT UNIFORM NOOBS N DATA=DB.MCDBRX06_0(OBS=15);
TITLE1 'SAMPLE PRINT';
RUN;
```

```
PROC FREQ DATA=DB.MCDBRX06_0;
 TABLE PNUM PNUM*RF_PLINK PNUM*RF_ELINK /LIST MISSING;
 TITLE1 'LINKAGE BY PAYER';
 RUN;

*Create a 10% sample for test purpose****;

data db.mcdbrx06_t;
 set db.MCDBRX06_0;
 r=ranuni(12345);
 if r<0.1;
run;


***Delete temporary file;

x 'DEL D:\WORKAREA\TEMP_DIR.TXT';

%jobend;

*** save log and output files;

dm'log;   print file=D:\MCdb\DataCol\RX_Step8\log_out\STEP8_RX_AllPayers_0.log replace';
dm'output;print file=D:\MCdb\DataCol\RX_Step8\log_out\STEP8_RX_AllPayers_0.lst replace';
```

```
%jobstart;

************************************************************************;
*                                                                      ;
*  Program Name:  D:\MCdb\DataCol\RX_Step8\Step8_RX_Stats.SAS          ;
*                                                                      ;
*  Programmer:    Adrien Ndikumwami                                    ;
*                                                                      ;
*  Description:   Create Summary Statistics from MCDBRX05_0, file      ;
*                 resulting from the concatenation of all payers       ;
*                                                                      ;
*  DATE CREATED: 10/17/2000 by Adrien Ndikumwami                       ;
*  DATE MODIFIED: 8/22/2003 by John May                                ;
*                 Added CURYR macros, Coverage type, Delivery Systems  ;
*                 freqs and converted to Multum NDC data               ;
*                                                                      ;
*               : 8/29/2005 by Sireesha Thati                          ;
*                               Added tables for Financial Variables - Outliers Payment ;
*                 Mail Order, Business Activity, Specialty Codes,      ;
*                 Check 999's and Values - Billed Charge, Reimbursement ;
*                                          Amount, Patient Liability   ;
*                 Date Variables - Filldd , Fillmo , Fillyr, Rbirthdd, ;
*                               Rbirthmo Rbirthyr and Drug Approval Date. ;
*                 NDC Level Analysis - Brand / Generic Indicator       ;
*                                      New Vs Old Drugs                ;
*                                                                      ;
*  Modified by:   Po-Lun Chou                                          ;
*  Date:          August 30, 2007                                      ;
*  Purpose:       Modified this program for year 2006 data.            ;
************************************************************************;

options nodate pageno=1 obs=max;

*set CURYR2/CURYR4 to Current Processing year;
%LET CURYR2=06;
%LET CURYR4=2006;

libname ndc     "D:\Mcdb\DataCol\Formats\ndc\Multum\data";
libname db      "D:\Mcdb\DataCol\RX_Step8\data";
libname library "D:\Mcdb\DataCol\Formats\Permanent";
data _null_;
 wdate= put(today(),worddate.);
 call symput ('tdate', wdate);
run;

Title1 "MARYLAND MEDICAL CARE DATA BASE";
Title2 "&curyr4 PHARMACY DATA DIAGNOSTIC STATISTICS (STEP 8)";
Footnote j=r "Run Date: &tdate";

************************************************;
*   0. Formats                                 *;
************************************************;

************** Include formats;
options nosource2;
%include "D:\Mcdb\DataCol\RX_Step8\programs\Template_ODS_Step8.sas";
%include "I:\mcdb\DataCol\formats\uszipfmt.sas";
%include "D:\mcdb\DataCol\formats\DrugCategories_Step8_2006.sas";
options source2;


********** Create format from datasets;

*************************************************************;
* 1. Create label format for NDC codes                      *;
*************************************************************;

data _ndc;
* set brand_description length to length of longest entry;
  length brand_description $52;
```

```
  set ndc.multum_hcfa_ms_ndc_2006 (keep=ndc_code brand_description);
  ndc9=substr(ndc_code,1,9);

run;

proc sort data=_ndc out=ndc1(keep=ndc9 brand_description);
  by ndc9 descending brand_description;
run;

data fmtndc1(drop=ndc9 brand_description);
  length start $9. label $62.;
  set ndc1;
  by ndc9 descending brand_description;
        if first.ndc9;
  start=ndc9;
  label=ndc9||" "||brand_description;
  fmtname="lbndc";
  type='c';
  output;
run;

proc format cntlin=fmtndc1 ;
run;


********** Create format;

proc format;
  value $ct
        '1'   = '1 Mcare*Suppl.'
        '2'   = '2 Indiv.*Plan'
        '3'   = '3 Emp-Sp*Self'
        '4'   = '4 Emp-Sp*Insured'
        '5'   = '5 Public*Empl.'
        '6'   = '6 CSHBP'
        '7'   = '7 Mcare*+Choice'
        '8'   = '8 Taft*Hartley'
        other = '9 Unknown';

  value $ctagg
        '2'   = '2'
        '3'   = '2'
        '4'   = '2'
        '6'   = '1'
        other = '3';

  value $ctaggd
        '1' = 'Coverage Type 6'
        '2' = 'Coverage Type 2,3,4'
        '3' = 'Other Coverage';

  value $ds
        '1'   = '1 HMO'
        '2'   = '2 PPO/POS'
        '3'   = '3 PPO+*Man.Care'
        '4'   = '4 Indem.*Care'
        '5'   = '5 HMO-POS*Rider'
        other = '9 Unknown';

  value $dbc_f
        '0' = '[0]*NON-DEBIT/*CREDIT'
        '1' = '[1]*DEB/CRED*SET'
        '2' = '[2]*COMPLETE*DUPL. SET'
        '3' = '[3]*PARTIAL*DUPL. SET'
        '4' = '[4]*NON-D/C*AFTER ADJ.';

  value $mth
        01-12 = 'VALID'
        .     = 'MISSING'
        other = 'INVALID';
```

```
value $day
      01-31 = 'VALID'
      .     = 'MISSING'
      other = 'INVALID';

value $yr
      .         = 'MISSING'
      1890-1942 = '1890-1942'
      1943-1989 = '1943-1989'
      1990-2007 = '1990-2007'
      other     = 'OTHER';

value $f_yr
      2006  = '2006'
      2007  = '2007'
      .     = 'MISSING'
      other = 'OTHER';

value $sex
      '1' = 'MALE'
      '2' = 'FEMALE'
      '3' = 'NOT CODED'
      ' ' = 'BLANK';

value $f_valid
      '0' = 'VALID'
      '1' = 'INVALID'
      '2' = 'BLANK';

value agecat
      low--1   = '<0'
      0-17     = '0-17'
      18-64    = '18-64'
      65-110   = '65-110'
      111-high = '>110'
      .        = 'MISSING'
      .C       = 'MISSING'
      .U       = 'MISSING'
      .M       = 'MISSING';

value $f_paym
      '1' = 'NEGATIVE*PAYMENT'
      '2' = 'ZERO*PAYMENT'
      '3' = 'POSITIVE*PAYMENT'
      '4' = 'MISSING*PAYMENT';

value $f_finms
      '1' = 'ALL FIN*VAR EQ 0* OR MISS.'
      '0' = 'ALL FIN*VAR NE 0* OR MISS.';

VALUE fmtNine
        .C = '.C'
         . = '.'
        .M = '.M'
        .U = '.U'
      -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'
      -9.99 = '-9.99'
     -99.99 = '-99.99'
    -999.99 = '-999.99'
   -9999.99 = '-9999.99'
  -99999.99 = '-99999.99'
 -999999.99 = '-999999.99'
       9.99 = '9.99'
      99.99 = '99.99'
     999.99 = '999.99'
    9999.99 = '9999.99'
   99999.99 = '99999.99'
  999999.99 = '999999.99'
 9999999.99 = '9999999.99'
      other = 'NON-999s';

 VALUE CHARGEL
       .M        = 'Missing'
       .U        = 'Missing'
       .         = 'Missing'
       .C        = 'Missing'
       LOW-<0    = '[1]*< 0'
       0         = '[2]*0'
       0<-25     = '[3]*0<-25'
       25<-100   = '[4]*25<-100'
       100<-500  = '[5]*100<-500'
       500<-HIGH = '[6]*500+'
       OTHER     = 'OTHER';

 VALUE CHARGES
       .M        = 'MISSING'
       .U        = 'MISSING'
       .         = 'MISSING'
       .C        = 'MISSING'
       LOW-<0    = '< 0'
       0         = '0'
       0<-25     = '0<-25'
       25<-100   = '25<-100'
       100<-500  = '100<-500'
       500<-HIGH = '500+'
       OTHER     = 'OTHER';

value drgcmp
       1 = 'NON-COMPND'
       2 = 'COMPOUND'
      .M = 'MISSING'
      .U = 'MISSING'
      .C = 'MISSING'
      .  = 'MISSING';

 value spec_f
       0 = 'NO SPECLTY'
       1 = '1-SPECLTY'
       2 = '2-SPECLTY'
       3 = '3-SPECLTY';

 value $keep_f
       '0' = 'EXCLUDED'
       '1' = 'KEPT';

 value $f_miss
       '0' = 'MISSING'
       '1' = 'NON-MISS.';

 value $d_zip_f
       '1' = 'NON-MD/*INVALID'
```

```
        '0' = 'MD RES.'
        '2' = 'BLANK';

  value $d_dat_f
        '1' = "<&curyr4./*INVALID"
        '0' = "&curyr4 +";

  value $bcout_f
        '0'='NORMAL'
        '1'='OUTLIER';

  value $fodd_f
        '0' = 'PAYMENT*=BILL'
        '1' = 'PAYMENT*>BILL'
        '2' = 'PAYMENT*<BILL';

  VALUE $PLINK
        ' ' = 'NO LINK'
        '1' = 'LINK BY*DEANUM'
        '2' = 'LINK BY*PROVID'
        '3' = 'LINK BY*FEDTAXID';

  VALUE $ELINK
        ' ' = 'NO LINK'
        '1' = 'LINK BY *ID-DOB-SEX';

  Value $m_order
        ''  = 'MISSING'
        '1' = 'MAIL ORDER'
        '0' = 'RETAIL';

  Value $B_day
        00    = '00'
        01-31 = '01-31'
        .     = 'MISSING'
        other = 'OTHER';

  Value $pout_f
        '1' = 'PAYMENT*>10K'
        '0' = 'PAYMENT*<=10K';

  Value $a_yr
        .         = 'MISSING'
        1900-2003 = '2003<'
        2004      = '2004'
        2005      = '2005'
        2006      = '2006'
        2007      = '2007'
        other     = 'OTHER';

  Value $enr_yr
        .         = 'MISSING'
        0001-2005 = '<=2005'
        2006      = '2006'
        2007      = '2007'
        Other     = '2008+';

  value $ssast
        '08'  = '1-DE'
        '09'  = '2-DC'
        '21'  = '3-MD'
        '39'  = '4-PA'
        '49'  = '5-VA'
        '51'  = '7-WV'
        OTHER = '8-OTHER';
run;


*************************************************************************;
*    Prepare all variables used in the summary reports                  *;
*************************************************************************;
```

```
data one(rename=(RBCHRGE=RBILL NDC9=NDCLABL RREIMAMT=RREIMB PHAZIP=RZIP_S)
         drop= DEANUM NCPDP NCPDP_ST apprdate speclty1 speclty2 speclty3);
     length pnum $8 appr_yr $8;
     set db.mcdbrx&curyr2._0(drop=drugcmp_   drugqty_   drugspl_  rpatlib_    rreiamt_
                               rbchrge_   lu_limit   pex_ind   CHNIND
                               MMDC       MULTDC2    MULTDC3   MULTDC4     RF_COV17
                               RF_SELF    RF_YR07    RN_ZIP_S  RPATZIP     RPHARMCTY
                               RPHARMGRP  R_NAGE1206 PATID     RF_DEA      RF_NDC
                               drug_id    encserv    enrolldt  disenrolldt ncpdp6
                               ndc2       pa_limit   patid_raw payer       pex_ind
                               rcd_id     recnum     rf_MedSup rf_multi    rf_ndup
                               rf_yr65    rscript    xref_link Rf_paout   dea_payer
                               neg_fin    rage);

  length  Rf_phZip $1 RECORDS 3 f_keep $1 f_ncpdp $1 f_dea $1 ;
  aggct=put(covtype,$ctagg.);
  if NDC9='' then f_NDC='1'; else f_NDC='0';
  if ndc='' then f_ndc='2';
  if deanum='' then f_dea='0'; else f_dea='1';
  if ncpdp='' then f_ncpdp='0'; else f_ncpdp='1';
  if rd_dat_f='0' and rd_zip_f='0' then f_keep="1" ; else f_keep="0";
  RECORDS=1;
  f_spec=(speclty1 ne '')+(speclty2 ne '') +(speclty3 ne '');

  if trim(left(phazip))="" then Rf_phZip="2";
  else if put(phazip,$uszip.)="1" then Rf_phZip="0";
  else Rf_phZip="1";
  if covtype='' then covtype='9';
  if delivtyp='' then delivtyp='9';

  mdcl=put(compress(multdc1),$mdcat.);
* fillyr=fillyrx;
  appr_yr=put(year(apprdate),z4.);
* CONVERT "OTHER" NCPDP STATE CODES TO 99 TO FACILITATE FREQ ORDER;
  length NCPDPST $2;
  IF NCPDP_ST NOT IN('  ','08','09','21','39','49','51') THEN NCPDPST='99';
  ELSE NCPDPST=NCPDP_ST;
run;

**********************************************;
* 2. Summary Reports                         *;
* Note for Printer Setup: Font:8 Courier New  *;
* LANDSCAPE, LM=0.4" RM=0.4" TM=0.55" BM=0.3" *;
**********************************************;

options orientation = landscape;

*********************************************************;
*     Table 1: Number of Services by Payer               *;
*********************************************************;

ODS PDF STYLE=CUSTSTEP8 BODY="D:\Mcdb\DataCol\RX_STEP8\Step8_2006.PDF";
options missing="-";

proc report data=one nowd box headline split='*' ;
         column pnum f_keep,records  pkept Total;
     define pnum / group  format=$pname. 'PAYER' style={just=l cellwidth=40%};
     define f_keep  / group across format=$keep_f. '' style={just=r outputwidth=10%};
     define total / computed format=comma10. 'TOTAL' ;
     define records/  format=comma10. ;
     define pkept / computed format=percent8.2 'PERCENT*KEPT'  ;
     compute total ;
          total=sum(_c2_,_c3_);
     endcomp;
     compute pkept;
          pkept=_c3_/(sum(_c2_,_c3_));
     endcomp;
     compute after;
       pnum="Total:";
     endcomp;
```

```
     rbreak after/dol dul summarize;
     Title3 "Table 1: Number of Services by Payer-All Records";
     Footnote1 "Rule for Record Exclusion: (1) Year Filled < &curyr4 or missing OR (2) non-MD
zipcode";
run;

*******************************************************************;
* Table 2: Number of Services by  Payer-Variables used for exclusion *;
*******************************************************************;
proc report data=one nowd box  headline split='*';
     column pnum rd_dat_f,RECORDS rd_zip_f,RECORDS   Total;
     define pnum / group width=55 format=$pname. 'PAYER' style={just=l cellwidth=40%};
     define rd_dat_f/ group across format=$d_dat_f.  style={just=r outputwidth=10%} '';
     define rd_zip_f/ group across format=$d_zip_f.   style={just=r outputwidth=10%} '';
     define RECORDS/  format=comma10. ;
     define total / computed format=comma10. 'TOTAL' ;
     compute total ;
          total=sum(_c3_,_c4_);
     endcomp;
     compute after;
       pnum="Total:";
     endcomp;
     rbreak after/dol dul summarize;
     Title3 "Table 2: Number of Services by Payer-Variables Used for Record Exclusions";

run;


*************************************************************************;
* Table 3: Financial Variables Analysis--Kept Records Only: Billed Charge *;
*************************************************************************;

proc report data=one nowd box  headline split='*';

     where rd_dat_f='0' and rd_zip_f='0';
     column pnum RECORDS=fin_n RBILL=fin_mean RBILL=fin_sum
                         RBILL=fin_min RBILL=fin_max RBILL=fin_std ;
     define pnum / group width=50 format=$pname. 'Payer' style={just=l cellwidth=40%};
     define fin_n / analysis n format=Comma10. '# Records' style={just=r cellwidth=9%};
     define fin_mean/analysis mean format=comma8.2 'Mean*Billed*Charge' style={just=r
cellwidth=9%};
     define fin_sum/ analysis sum format=comma15. 'Total*Billed*Charge' style={just=r
cellwidth=10%};
     define fin_min/ analysis min format=comma12.2 'Minimum*Billed*Charge' style={just=r
cellwidth=9%};
     define fin_max/ analysis max format=comma11.2 'Maximum*Billed*Charge' style={just=r
cellwidth=9%};
     define fin_std/ analysis std format=comma10.2 'STD*Billed*Charge' style={just=r
cellwidth=9%};
     compute after;
        pnum="Total:";
     endcomp;
     rbreak after/dol dul summarize;
     Title3 "Table 3: Financial Variables: Billed Charge";
      Footnote2 "This table is generated after applying the exclusion rule";
run;


*******************************************************************************************;
* Table 4: Financial Variables Analysis--Kept Records Only: Patient Liability           *;
*******************************************************************************************;

proc report data=one nowd box  headline split='*';
     where rd_dat_f='0' and rd_zip_f='0';
     column pnum records=fin_n rpatliab=fin_mean rpatliab=fin_sum
                         rpatliab=fin_min rpatliab=fin_max rpatliab=fin_std ;
     define pnum / group width=50 format=$pname. 'Payer' style={just=l cellwidth=40%};
     define fin_n / analysis n format=Comma10. '# Records' style={just=r cellwidth=9%};
     define fin_mean/analysis mean format=comma8.2 'Mean*Patient*Liability' style={just=r
cellwidth=9%};
```

```
     define fin_sum/ analysis sum format=comma15. 'Total*Patient*Liability' style={just=r
cellwidth=10%};
     define fin_min/ analysis min format=comma12.2 'Minimum*Patient*Liability' style={just=r
cellwidth=9%};
     define fin_max/ analysis max format=comma11.2 'Maximum*Patient*Liability' style={just=r
cellwidth=9%};
     define fin_std/ analysis std format=comma10.2 'STD*Patient*Liability' style={just=r
cellwidth=9%};
     compute after;
        pnum="Total:";
     endcomp;
     rbreak after/dol dul summarize;
     Title3 "Table 4: Financial Variables: Patient Liability";
run;


*******************************************************************************************;
* Table 5: Financial Variables Analysis--Kept Records Only: Reimbursement Amount           *;
*******************************************************************************************;

proc report data=one nowd box  headline split='*';
     where rd_dat_f='0' and rd_zip_f='0';
     column pnum records=fin_n RREIMB=fin_mean RREIMB=fin_sum
                         RREIMB=fin_min RREIMB=fin_max RREIMB=fin_std ;
     define pnum / group width=50 format=$pname. 'Payer' style={just=l cellwidth=40%};
     define fin_n / analysis n format=Comma10. '# Records' style={just=r cellwidth=9%};
     define fin_mean/analysis mean format=comma7.2 'Mean*Reimb.*Amount' style={just=r
cellwidth=9%};
     define fin_sum/ analysis sum format=comma15. 'Total*Reimb.*Amount' style={just=r
cellwidth=10%};
     define fin_min/ analysis min format=comma10.2 'Minimum*Reimb.*Amount' style={just=r
cellwidth=9%};
     define fin_max/ analysis max format=comma10.2 'Maximum*Reimb.*Amount' style={just=r
cellwidth=9%};
     define fin_std/ analysis std format=comma10.2 'STD*Reimb.*Amount' style={just=r
cellwidth=9%};

     compute after;
        pnum="Total:";
     endcomp;
     rbreak after/dol dul summarize;
     Title3 "Table 5: Financial Variables: Reimbursement Amount";
run;


*************************************************************************;
* Table 6: Financial Variables Analysis--Kept Records Only: Mean Summary *;
*************************************************************************;

proc report data=one nowd box  headline split='*';
     where rd_dat_f='0' and rd_zip_f='0';
     column pnum records=fin_n RBILL RREIMB rpatliab rpayment;
     define fin_n / analysis n format=Comma10. '# Records' style={just=r cellwidth=9%};
     define pnum / group width=50 format=$pname. 'Payer' style={just=l cellwidth=40%};
     define RBILL/  analysis mean format=comma8.2 'Mean*Billed*Charge' style={just=r
cellwidth=9%};
     define RREIMB/ analysis mean format=comma8.2 'Mean*Reimb.*Amount' style={just=r
cellwidth=9%};
     define rpatliab/ analysis mean format=comma8.2 'Mean*Patient*Liability' style={just=r
cellwidth=9%};
     define rpayment/ analysis mean format=comma8.2 'Mean*Payment' style={just=r cellwidth=9%};
     compute after;
        pnum="Total:";
     endcomp;
     rbreak after/dol dul summarize;
     Title3 "Table 6: Financial Variables: Mean Summary for all Fin. Variables";
run;


*************************************************************************;
* Table 7A: Financial Variables Analysis--Billed Charge Outlier          *;
```

```
************************************************************************;

proc report data=one nowd box  headline split='*';
     where rd_dat_f='0' and rd_zip_f='0';
     column pnum rf_bcout,records pOut Total;
        define pnum / group width=50 format=$pname. 'PAYER* ' style={just=l cellwidth=40%};
     define rf_bcout/group across format=$bcout_f. '' style={just=r outputwidth=9%} ;
     define total /computed format=Comma12. 'TOTAL* ' ;
     define records/format=Comma12. '' ;
        define pOut / computed format=percent8.2 'PERCENT*OUTLIER' ;
     compute total ;
          total=sum(_c2_,_c3_,_c4_,_c5_);
     endcomp;
    compute after;
      pnum="Total:";
     endcomp;
        compute pOut;
         pOut=_c3_/(sum(_c2_,_c3_));
     endcomp;
     rbreak after/dol dul summarize;
     Title3 "Table 7A: Financial Variables: Outliers";
        footnote3 "Outlier is defined as billed charge > absolute value of Mean + 2 std
deviation";
run;


************************************************************************;
* Table 7B: Financial Variables Analysis-- Outliers - Payment           *;
************************************************************************;

proc report data=one nowd box  headline split='*';
     where rd_dat_f='0' and rd_zip_f='0';
     column pnum rf_10k,records pOut Total;
        define pnum / group width=50 format=$pname. 'PAYER* ' style={just=l cellwidth=37%};
     define rf_10k/group across format=$pout_f. ''  style={just=r outputwidth=10%};
     define total /computed format=Comma10. 'TOTAL* ' ;
     define records/format=Comma10. '' ;
        define pOut / computed format=percent8.2 'PERCENT*>10k' ;
     compute total ;
          total=sum(_c2_,_c3_);
     endcomp;
    compute after;
      pnum="Total:";
     endcomp;
        compute pOut;
         pOut=_c3_/(sum(_c2_,_c3_));
     endcomp;
     rbreak after/dol dul summarize;
     Title3 "Table 7B: Financial Variables: Payment Outliers";
        footnote3 "Outlier is defined as billed charge > absolute value of Mean + 2 std
deviation";
run;


************************************************************************;
* Table 8: Financial Variables Analysis--Billed Charge Vs Payment        *;
************************************************************************;

proc report data=one nowd box  headline split='*';
     where rd_dat_f='0' and rd_zip_f='0';
     column pnum rf_finodd,records pOdd Total;
     define pnum / group width=50 format=$pname. 'PAYER* ' style={just=l cellwidth=40%};
     define rf_finodd/group across format=$fodd_f. width=16 '' style={just=r outputwidth=9%};
     define total /computed format=Comma10. 'TOTAL* '  ;
     define records/format=Comma10. '' ;
     define pOdd / computed format=percent8.2 '%PAYMENT*>BILL' ;
     compute total ;
     total=sum(_c2_,_c3_,_c4_,_c5_);
     endcomp;
     compute after;
     pnum="Total:";
```

```
     endcomp;
     compute pOdd;
     pOdd=_c4_/(sum(_c2_,_c3_,_c4_));
     endcomp;
     rbreak after/dol dul summarize;
     Title3 "Table 8: Financial Variables: Billed Charge vs. Payment";
     footnote3 "In general, billed charge should be greater or equal to payment.";
 run;


************************************************************************;
* Table 9: Financial Variables Analysis--Debit/credit and Duplicate Records    *;
************************************************************************;

proc report data=one nowd box  headline split='*';
     where rd_dat_f='0' and rd_zip_f='0';
     column pnum rf_dbc,records Total;
     define pnum / group width=50 format=$pname. 'PAYER* ' style={just=l cellwidth=37%};
     define rf_dbc/group across format=$dbc_f. width=16 '' style={just=r cellwidth=10%};
     define total /computed format=Comma10. 'TOTAL* ' style={just=r cellwidth=10%};
     define records/format=Comma10. '' ;
     compute total ;
     total=sum(_c2_,_c3_,_c4_,_c5_,_c6_,_c7_);
     endcomp;
     compute after;
     pnum="Total:";
     endcomp;
     rbreak after/dol dul summarize;
     Title3 "Table 9: Financial Variables: Debit/Credit and Duplicate Records";
     footnote3 "Credit/debit pair have + and - payment cancelling each other";
     footnote4 "Duplicate sets are records of the same prescription drug filled more than once in
one day for the same patient";
     footnote5 "Non-debit/credit after adjustment are records in suspect d/c sets that are not
D/C or duplicates";
 run;



************************************************************************;
* Table: 10A. Financial Variables: Check 999s in Billed Charge           *;
************************************************************************;

 proc report data=one nowd box  headline split='*';
     where rd_dat_f='0' and rd_zip_f='0';
     column pnum RBILL,records  Total;
     define pnum / group width=50 format=$pname. 'PAYER' '' style={just=l cellwidth=35%};
     define RBILL / group across format=fmtnine. width=10 '' style={just=r cellwidth=8%};
     define total / computed format=Comma10. 'TOTAL' '' style={just=r cellwidth=8%};
        define records / format=Comma10. '' ;
     compute total ;
          total=sum(_c2_,_c3_,_c4_,_c5_,_c6_,_c7_,_c8_,_c9_);
     endcomp;
     compute after;
       pnum="Total:";
     endcomp;
     rbreak after/dol dul summarize;
     Title3 "Table 10A: Financial Variables: Check 999s in Billed Charge";
        title4 ;
        footnote3 ;
        footnote4 ;

run;


************************************************************************;
* Table: 10B. Financial Variables: Check 999s in Reimbursement Amount    *;
************************************************************************;

 proc report data=one nowd box  headline split='*';
     where rd_dat_f='0' and rd_zip_f='0';
     column pnum RREIMB,records  Total;
```

```
     define pnum / group width=50 format=$pname. 'PAYER' '' style={just=l cellwidth=35%};
     define RREIMB / group across format=fmtnine. width=10 '' style={just=r cellwidth=8%};
     define total / computed format=comma10. 'TOTAL' '' style={just=r cellwidth=8%};
        define records / format=comma10. '' ;
     compute total ;
          total=sum(_c2_,_c3_,_c4_,_c5_,_c6_,_c7_,_c8_);
     endcomp;
     compute after;
       pnum="Total:";
     endcomp;
     rbreak after/dol dul summarize;
     Title3 "Table 10B: Financial Variables: Check 999s in Reimbursement Amount";
        title4 ;
        footnote3 ;
        footnote4 ;
run;


************************************************************************;
* Table: 10C. Financial Variables: Check 999s in Patient Liability       *;
************************************************************************;

 proc report data=one nowd box  headline split='*';
     where rd_dat_f='0' and rd_zip_f='0';
     column pnum rpatliab,records  Total;
     define pnum / group width=50 format=$pname. 'PAYER' '' style={just=l cellwidth=35%};
     define rpatliab / group across format=fmtnine. width=10 '' style={just=r cellwidth=9%};
     define total / computed format=comma10. 'TOTAL' '' style={just=r cellwidth=9%};
        define records / format=comma10. '';
     compute total ;
          total=sum(_c2_,_c3_,_c4_,_c5_,_c6_,_c7_,_c8_);
     endcomp;
     compute after;
       pnum="Total:";
     endcomp;
     rbreak after/dol dul summarize;
     Title3 "Table 10C: Financial Variables: Check 999s in Patient Liability";
        title4 ;
        footnote3 ;
        footnote4 ;

run;


************************************************************************;
* Table: 11A. Financial Variables: Check Values for Billed Charge        *;
************************************************************************;

proc report data=one nowd box  headline split='*';
     where rd_dat_f='0' and rd_zip_f='0';
     column pnum RBILL,records  Total;
     define pnum / group width=46 format=$pname. 'PAYER' '' style={just=l cellwidth=37%};
     define RBILL / group across format=chargel. width=8 '' style={just=r cellwidth=8%};
     define total / computed format=comma10. 'TOTAL' '' style={just=r cellwidth=8%};
        define records / format=comma10. '' ;
     compute total ;
          total=sum(_c2_,_c3_,_c4_,_c5_,_c6_,_c7_,_c8_,_c9_);
     endcomp;
     compute after;
       pnum="Total:";
     endcomp;
     rbreak after/dol dul summarize;
     Title3 "Table 11A: Financial Variables: Check Values for Billed Charge";
        title4 ;
run;


************************************************************************;
* Table: 11B. Financial Variables: Check Values for Reimbursement Amount *;
************************************************************************;
```

```
proc report data=one nowd box  headline split='*';
     where rd_dat_f='0' and rd_zip_f='0';
     column pnum RREIMB,records  Total;
     define pnum / group width=46 format=$pname. 'PAYER' '' style={just=l cellwidth=37%};
     define RREIMB / group across format=chargel. width=8 '' style={just=r cellwidth=8%};
     define total / computed format=comma10. 'TOTAL' '' style={just=r cellwidth=8%};
        define records / format=comma10. '' ;
     compute total ;
          total=sum(_c2_,_c3_,_c4_,_c5_,_c6_,_c7_,_c8_,_c9_);
     endcomp;
     compute after;
       pnum="Total:";
     endcomp;
     rbreak after/dol dul summarize;
     Title3 "Table 11B: Financial Variables: Check Values for Reimbursement Amount";
        title4 ;
run;


************************************************************************;
* Table: 11C. Financial Variables: Check Values for Patient Liability    *;
************************************************************************;

proc report data=one nowd box  headline split='*';
     where rd_dat_f='0' and rd_zip_f='0';
     column pnum rpatliab,records  Total;
     define pnum / group width=46 format=$pname. 'PAYER' '' style={just=l cellwidth=37%};
     define rpatliab / group across format=chargel. width=8 '' style={just=r cellwidth=8%};
     define total / computed format=comma10. 'TOTAL' '' style={just=r cellwidth=8%};
        define records / format=comma10. '';
     compute total ;
          total=sum(_c2_,_c3_,_c4_,_c5_,_c6_,_c7_,_c8_,_c9_);
     endcomp;
     compute after;
       pnum="Total:";
     endcomp;
     rbreak after/dol dul summarize;
     Title3 "Table 11C: Financial Variables: Check Values for Patient Liability";
        title4 ;
run;


************************************************************************;
* Table: 12. Financial Variables: Check Values -All 3 Variables          *;
************************************************************************;

proc freq data=one compress;
where rd_dat_f='0' and rd_zip_f='0';
tables RBILL*RREIMB*rpatliab/list missing ;
format RBILL RREIMB rpatliab charges. ;
Title3 "Table 12:  Financial Variables: Check Values for All 3 variables";
Title4 "Summary for All Payers Combined";
run;


************************************************************************;
* Table 13: Linkage Statistics--Pharmacy and Provider                    *;
************************************************************************;

proc report data=one nowd box missing headline split='*';
     where rd_dat_f='0' and rd_zip_f='0';
     column pnum rf_plink,records pplink Total;
        define pnum / group width=50 format=$pname. 'PAYER* ' style={just=l cellwidth=40%};
     define rf_plink/group across format=$plink. width=10 '' style={just=r outputwidth=9%};
     define total /computed format=comma10. 'TOTAL* ' ;
     define records/format=comma10. '' ;
        define pplink / computed format=percent8.1 '% LINK' ;
     compute total ;
          total=sum(_c2_,_c3_,_c4_,_c5_);
     endcomp;
    compute after;
```

```
       pnum="Total:";
     endcomp;
        compute pplink;
         pplink=sum(_c2_,_c3_,_c4_)/(sum(_c2_,_c3_,_c4_,_c5_));
     endcomp;
     rbreak after/dol dul summarize;
     Title3 "Table 13: Linkage Statistics--Pharmacy and Provider ";
        footnote3 "Pharmacy is linked to the Provider by DEA Number, Provid, or FEDTAXID";
 run;


************************************************************************;
* Table 14: Linkage Statistics--Pharmacy and Provider                  *;
************************************************************************;

proc report data=one nowd  box missing headline split='*';
     where rd_dat_f='0' and rd_zip_f='0';
     column pnum rf_elink,records pelink Total;
        define pnum / group  format=$pname. 'PAYER* ' style={just=l cellwidth=40%};
     define rf_elink/group across format=$elink.  '' style={just=r outputwidth=10%};
     define total /computed format=comma10. 'TOTAL* ' ;
     define records/format=comma10. '' ;
        define pelink / computed format=percent8.1 '% LINK';
     compute total ;
         total=sum(_c2_,_c3_);
     endcomp;
    compute after;
       pnum="Total:";
     endcomp;
        compute pelink;
         pelink=_c2_/(sum(_c2_,_c3_));
     endcomp;
     rbreak after/dol dul summarize;
     Title3 "Table 14: Linkage Statistics--Pharmacy and Encounter";
        footnote3 "Pharmacy is linked to the Encounter by PATID-DOB-SEX";
 run;


************************************************************************;
*Table 15: Number of Services by Payer: Payment Value Status            *;
************************************************************************;

proc report data=one nowd box headline split='*';
     where rd_dat_f='0' and rd_zip_f='0';
     column pnum rf_paym,records Total;
     define pnum / group width=55 format=$pname. 'PAYER* ' style={just=l cellwidth=40%};
     define rf_paym/group across format=$f_paym. width=10 '' style={just=r cellwidth=9%};
     define total /computed format=comma10. 'TOTAL* ' style={just=r cellwidth=9%};
     define records/format=comma10. '' ;
     compute total ;
         total=sum(_c2_,_c3_,_c4_,_c5_);
     endcomp;
    compute after;
       pnum="Total:";
     endcomp;
     rbreak after/dol dul summarize;
     Title3 "Table 15: Financial Variables: Payment Value Status";
     Title4 "Number of services with missing,negative,zero, and positive payments";
     footnote3;
run;


*******************************************************************************;
*Table 16: Financial Variables Analysis--Kept Records Only: Missing and Non-missing *;
*******************************************************************************;

proc report data=one nowd box headline split='*';
     where rd_dat_f='0' and rd_zip_f='0';
     column pnum rf_finms,records Total;
     define pnum / group width=55 format=$pname. 'PAYER* ' style={just=l cellwidth=40%};
     define rf_finms/group across format=$f_finms. width=10 '' style={just=r cellwidth=9%};
```

```
     define total /computed format=comma10. 'TOTAL* ' style={just=r cellwidth=9%};
     define records/  format=comma10. '' ;
     compute total ;
          total=sum(_c2_,_c3_);
     endcomp;
    compute after;
       pnum="Total:";
     endcomp;
     rbreak after/dol dul summarize;
     Title3 "Table 16: Financial Variables: All Values Missing";
     Title4 " Number of Services with vs. without All Variables Missing";

run;


*******************************************************************************************;
* Table 17: Drug Quantity--Kept Records Only                                              *;
*******************************************************************************************;

proc report data=one nowd box  headline split='*';
     where rd_dat_f='0' and rd_zip_f='0';
     column pnum records=fin_n drugqty=fin_mean drugqty=fin_sum
                         drugqty=fin_min drugqty=fin_max drugqty=fin_std ;
     define pnum / group width=55 format=$pname. 'Payer' style={just=l cellwidth=40%};
     define fin_n / analysis n format=comma10. '# Records' style={just=r cellwidth=9%};
     define fin_mean/analysis mean format=comma10. 'Mean*Drug*Quantity' style={just=r
cellwidth=9%};
     define fin_sum/ analysis sum format=comma15. 'Total*Drug*Quantity' style={just=r
cellwidth=10%};
     define fin_min/ analysis min format=comma10. 'Minimum*Drug*Quantity' style={just=r
cellwidth=9%};
     define fin_max/ analysis max format=comma10. 'Maximum*Drug*Quantity' style={just=r
cellwidth=9%};
     define fin_std/ analysis std format=comma10. 'STD*Drug*Quantity' style={just=r
cellwidth=8%};
     compute after;
        pnum="Total:";
     endcomp;
     rbreak after/dol dul summarize;
     Title3 "Table 17: Drug Quantity (Number of Units Dispensed): Summary Statistics";
run;


*******************************************************************************************;
* Table 18: Drug Supply--Kept Records Only                                                *;
*******************************************************************************************;

proc report data=one nowd box  headline split='*';
     where rd_dat_f='0' and rd_zip_f='0';
     column pnum records=fin_n drugspl=fin_mean drugspl=fin_sum
                         drugspl=fin_min drugspl=fin_max drugspl=fin_std ;
     define pnum / group width=55 format=$pname. 'Payer' style={just=l cellwidth=40%};
     define fin_n / analysis n format=comma10. '# Records' style={just=r cellwidth=9%};
     define fin_mean/analysis mean format=comma10. 'Mean*Drug*Supply' style={just=r
cellwidth=9%};
     define fin_sum/ analysis sum format=comma15. 'Total*Drug*Supply' style={just=r
cellwidth=9%};
     define fin_min/ analysis min format=comma10. 'Minimum*Drug*Supply' style={just=r
cellwidth=9%};
     define fin_max/ analysis max format=comma10. 'Maximum*Drug*Supply' style={just=r
cellwidth=9%};
     define fin_std/ analysis std format=comma10. 'STD*Drug*Supply' style={just=r cellwidth=9%};
     compute after;
        pnum="Total:";
     endcomp;
     rbreak after/dol dul summarize;
     Title3 "Table 18: Drug Supply: Summary Statistics";
run;


*********************************************************************;
```

```
*Table 19: Number of Services by Payer: Drug Compound                    *;
*************************************************************************;

proc report data=one missing nowd box headline split='*';
     where rd_dat_f='0' and rd_zip_f='0';
     column pnum drugcmp,records Total;
     define pnum / group width=55 format=$pname. 'PAYER* ' style={just=l cellwidth=35%};
     define drugcmp/group across format=drgcmp. width=13 '' style={just=r cellwidth=11%};
     define total /computed format=comma10. 'TOTAL* ' style={just=r cellwidth=11%};
     define records/  format=comma10. '' ;
     compute total ;
          total=sum(_c2_,_c3_,_c4_,_c5_,_c6_);
     endcomp;
    compute after;
       pnum="Total:";
     endcomp;
     rbreak after/dol dul summarize;
     Title3 "Table 19: Number of Services by Payer: Drug Compound";
     Title4;
      Footnote3 "The only valid values are compound and non-compound, 0,9, and . are invalid";
run;


*************************************************************************;
*Table 20: Number of Services by Payer: Patient Age                      *;
*************************************************************************;

proc report data=one missing nowd box headline split='*';
     where rd_dat_f='0' and rd_zip_f='0';
     column pnum rage12&curyr2,records Total;
     define pnum / group width=55 format=$pname. 'PAYER* ' style={just=l cellwidth=40%};
     define rage12&curyr2/group across format=agecat. width=10 ''  style={just=r cellwidth=9%};
     define total /computed format=comma10. 'TOTAL* '  style={just=r cellwidth=9%};
     define records/  format=comma10. ''  ;
     compute total ;
          total=sum(_c2_,_c3_,_c4_,_c5_);
     endcomp;
    compute after;
       pnum="Total:";
     endcomp;
     rbreak after/dol dul summarize;
     Title3 "Table 20: Number of Services by Payer: Patient Age in Dec &curyr4";
     Title4;
     footnote3;
run;


*************************************************************************;
*Table 21: Number of Services by Payer: Pharmacy ZiP code                *;
*************************************************************************;

proc report data=one missing nowd box headline split='*';
     where rd_dat_f='0' and rd_zip_f='0';
     column pnum  Rf_phZip,records Total;
     define pnum / group width=55 format=$pname. 'PAYER* ' style={just=l cellwidth=40%};
     define  Rf_phZip/group across format=$f_valid. width=10 '' style={just=r cellwidth=9%};
     define total /computed format=comma10. 'TOTAL* ' style={just=r cellwidth=9%};
     define records/  format=comma10. '';
     compute total ;
          total=sum(_c2_,_c3_,_c4_,_c5_);
     endcomp;
    compute after;
       pnum="Total:";
     endcomp;
     rbreak after/dol dul summarize;
     Title3 "Table 21: Number of Services by Payer: Pharmacy Zip Code";
     Title4;
     footnote3 "Valid zip codes means that it matches USPS ZIP code lookup";
run;
```

```
************************************************************************;
*Table 22A: Number of Services by Payer: Pharmacy NCPDP Number           *;
************************************************************************;

proc report data=one missing nowd box headline split='*';
     where rd_dat_f='0' and rd_zip_f='0';
     column pnum  f_ncpdp,records Total;
     define pnum / group width=55 format=$pname. 'PAYER* ' style={just=l cellwidth=40%};
     define  f_ncpdp/group across format=$f_miss. width=13 '' style={just=r cellwidth=9%};
     define total /computed format=comma13. 'TOTAL* ' style={just=r cellwidth=9%};
     define records/  format=comma13. '';
     compute total ;
          total=sum(_c2_,_c3_);
     endcomp;
    compute after;
      pnum="Total:";
     endcomp;
     rbreak after/dol dul summarize;
     Title3 "Table 22A: Number of Services by Payer: Pharmacy NCPDP Number";
     Title4;
     footnote3;
run;


************************************************************************;
* Table: 22B. Pharmacy Location State (based on NCPDP Number)            *;
************************************************************************;

proc report data=one missing nowd box  headline split='*'
         style(header)={font_size=9pt font_weight=bold}
         style (column)={font_size=9pt cellheight=12pt}
         style (summary)={font_weight=bold font_style=roman};
     where rd_dat_f='0' and rd_zip_f='0';
     column pnum NCPDPST,RECORDS PMD Total;
     define pnum / group width=50 format=$pname. 'PAYER' '' style={just=l cellwidth=30%};
     define NCPDPST / group across format=$ssast. width=8 ''  style={just=r outputwidth=7%};
     define RECORDS / format=comma10. ''  ;
     define total / computed format=comma10. 'TOTAL' '' ;
        define PMD / computed format=percent8.2 '% MD' ;
     compute total ;
          total=sum(_c2_,_c3_,_c4_,_C5_,_C6_,_C7_,_C8_);
     endcomp;
        compute PMD;
             PMD = _c4_/(sum(_c2_,_c3_,_c4_,_C5_,_C6_,_C7_,_C8_));
        endcomp;
     compute after;
       pnum="Total:";
     endcomp;
        rbreak after/dol dul summarize;
     Title3 "Table 22B:  Pharmacy Location State (based on NCPDP Number)";
run;


************************************************************************;
* Table: 22C. Mail Order                                                 *;
************************************************************************;

proc report data=one nowd box  headline split='*';
     where rd_dat_f='0' and rd_zip_f='0';
     column pnum rf_morder,RECORDS pmail Total;
     define pnum / group width=55 format=$pname. 'PAYER' '' style={just=l cellwidth=40%};
     define rf_morder / group across format=$m_order. width=10 ''  style={just=r
outputwidth=12%};
     define RECORDS / format=comma10. ''  ;
     define total / computed format=comma10. 'TOTAL' '' ;
        define pmail / computed format=percent8.2 'PERCENT*MAIL ORDER' ;
     compute total ;
          total=sum(_c2_,_c3_,_c4_);
     endcomp;
        compute pmail;
             pmail = _c2_/(sum(_c2_,_c3_));
```

```
        endcomp;
     compute after;
       pnum="Total:";
     endcomp;
        rbreak after/dol dul summarize;
     Title3 "Table 22C: Mail Order";
run;


************************************************************************;
*Table 23A: Number of Services by Payer: Practitioner DEA Number         *;
************************************************************************;

proc report data=one missing nowd box headline split='*';
     where rd_dat_f='0' and rd_zip_f='0';
     column pnum  f_dea,records Total;
     define pnum / group width=55 format=$pname. 'PAYER* ' style={just=l cellwidth=40%};
     define  f_dea/group across format=$f_miss. width=13 '' style={just=r cellwidth=9%};
     define total /computed format=comma10. 'TOTAL* ' style={just=r cellwidth=9%};
     define records/  format=comma10. '' ;
     compute total ;
          total=sum(_c2_,_c3_);
     endcomp;
    compute after;
       pnum="Total:";
     endcomp;
     rbreak after/dol dul summarize;
     Title3 "Table 23A: Number of Services by Payer: Practitioner DEA Number";
     Title4;
     footnote3;
run;



************************************************************************;
*Table 23B: Practitioner Business Activity                               *;
************************************************************************;

proc report data=one nowd box headline split='*';
     where rd_dat_f='0' and rd_zip_f='0';
     column pnum  DEABUS,records Total;
     define pnum / group width=50 format=$pname. 'PAYER* ' style={just=l cellwidth=35%};
     define  DEABUS /group across format=$busnams. width=12 ''  style={just=r cellwidth=9%};
     define records/  format=comma10. '' ;
        define total /computed format=comma10. 'TOTAL* ' style={just=r cellwidth=8%};
     compute total ;
          total=sum(_c2_,_c3_,_c4_,_C5_,_c6_,_C7_);
     endcomp;
     compute after;
       pnum="Total:";
        endcomp;
     rbreak after/dol dul summarize;
     Title3 "Table 23B: Practitioner Business Activity";
     Title4;
     footnote3;
run;


************************************************************************;
*Table 24: Number of Services by Payer: Patient Sex                     *;
************************************************************************;

proc report data=one missing nowd box headline split='*';
     where rd_dat_f='0' and rd_zip_f='0';
     column pnum rsex,records Total;
     define pnum / group width=51 format=$pname. 'PAYER* ' style={just=l cellwidth=40%};
     define rsex/group across format=$sex. width=10 '' style={just=r cellwidth=9%};
     define total /computed format=comma10. 'TOTAL* ' style={just=r cellwidth=9%};
     define records/format=comma10. '';
     compute total ;
          total=sum(_c2_,_c3_,_c4_,_C5_,_c6_);
```

```
     endcomp;
     compute after;
       pnum="Total:";
     endcomp;
     rbreak after/dol dul summarize;
     Title3 "Table 24: Number of Services by Payer: Patient Sex";
     Title4 "Values allowed: 1-Male, 2-Female, 3-Not Coded";
     footnote3;
run;


************************************************************************;
*Table 25: Number of Services by Payer: Specialty Codes                 *;
************************************************************************;

proc report data=one missing nowd box headline split='*';
     where rd_dat_f='0' and rd_zip_f='0';
     column pnum f_spec,records Total;
     define pnum / group width=46 format=$pname. 'PAYER* ' style={just=l cellwidth=40%};
     define f_spec/group across format= spec_f. width=10 '' style={just=r cellwidth=11%};
     define total /computed format=comma10. 'TOTAL*' ''  style={just=r cellwidth=11%};
     define records/format=comma10. '' ;
     compute total ;
          total=sum(_c2_,_c3_,_c4_,_c5_);
     endcomp;
     compute after;
       pnum="Total:";
     endcomp;
     rbreak after/dol dul summarize;
     Title3 "Table 25: Number of Services by Payer: Specialty Codes";
     Title4 "Number of Services with 0, 1, 2, or 3 Specialty Codes";
     footnote3 "Specialty codes are drawn from the Provider file by linking using DEANUM or
PROVID or FEDTAXID key";
run;


************************************************************************;
* Table 26. Statistics for coverage type and delivery system             *;
************************************************************************;

proc report data=one missing nowd box headline split='*'
         style(header)={font_size=9pt font_weight=bold}
         style (column)={font_size=9pt cellheight=12pt}
         style (summary)={font_weight=bold font_style=roman};
     where rd_dat_f='0' and rd_zip_f='0';
     column pnum covtype,records Total;
     define pnum / group width=30 format=$pname. 'PAYER* ' style={just=l cellwidth=27%};
     define covtype/group across format= $ct. width=10 '' style={just=r cellwidth=7%};
     define total /computed format=comma10. 'TOTAL*' ''  style={just=r cellwidth=7%};
     define records/format=comma10. ''  ;
     compute total ;
          total=sum(_c2_,_c3_,_c4_,_c5_,_c6_,_c7_,_c8_,_c9_,_c10_,_c11_);
     endcomp;
     compute after;
       pnum="Total:";
     endcomp;
     rbreak after/dol dul summarize;
     Title3 "Table 26A: Number of Services by Payer:Coverage Type";
     Title4;
    footnote3 "Coverage types are copied from the Encounter file by linking using Patid-DOB-
Sex";run;
run;


proc report data=one missing nowd box headline split='*';
     where rd_dat_f='0' and rd_zip_f='0';
     column pnum delivtyp,records Total;
     define pnum / group width=40 format=$pname. 'PAYER* ' style={just=l cellwidth=35%};
     define delivtyp/group across format= $ds. width=9 '' style={just=r cellwidth=8%};
     define total /computed format=comma10. 'TOTAL*' '' style={just=r cellwidth=8%};
     define records/format=comma10. '' ;
```

```
     compute total ;
          total=sum(_c2_,_c3_,_c4_,_c5_,_c6_,_c7_,_c8_);
     endcomp;
     compute after;
       pnum="Total:";
     endcomp;
     rbreak after/dol dul summarize;
     Title3 "Table 26B: Number of Services by Payer:Delivery systems";
     Title4;
    footnote3 "Delivery systems are copied from the Encounter file by linking using Patid-DOB-
Sex";run;
run;


************************************************************************;
* Table: 27A. Date Variables Analysis--Kept Records Only: Day Filled     *;
************************************************************************;

proc report data=one missing nowd box  headline split='*';
     where rd_dat_f='0' and rd_zip_f='0';
     column pnum filldd,records  Total;
     define pnum / group width=55 format=$pname. 'PAYER' '' style={just=l cellwidth=35%};
     define filldd / group across format=$day. width=10 ''  style={just=r cellwidth=9%};
     define total / computed format=comma10. 'TOTAL' '' style={just=r cellwidth=9%};
        define records / format=comma10. '' ;
     compute total ;
          total=sum(_c2_,_c3_,_c4_);
     endcomp;
     compute after;
       pnum="Total:";
     endcomp;
     rbreak after/dol dul summarize;
     Title3 "Table 27A: Date Variable - Fill Day";
        footnote3 ;
run;


************************************************************************;
* Table: 27B. Date Variables Analysis--Kept Records Only: Month Filled   *;
************************************************************************;

proc report data=one missing nowd box  headline split='*';
     where rd_dat_f='0' and rd_zip_f='0';
     column pnum fillmo,records Total;
     define pnum / group width=55 format=$pname. 'PAYER' '' style={just=l cellwidth=40%};
     define fillmo / group across format=$mth. width=10 '' style={just=r cellwidth=9%};
     define total / computed format=comma10. 'TOTAL' '' style={just=r cellwidth=9%};
        define records / format=comma10. '';
     compute total ;
          total=sum(_c2_,_c3_,_c4_);
     endcomp;
     compute after;
       pnum="Total:";
     endcomp;
     rbreak after/dol dul summarize;
     Title3 "Table 27B: Date Variable - Fill Month";
run;


************************************************************************;
* Table: 27C. Date Variables Analysis--Kept Records Only: Year Filled    *;
************************************************************************;

proc report data=one missing nowd box  headline split='*';
     where rd_dat_f='0' and rd_zip_f='0';
     column pnum fillyr,records Total;
     define pnum / group width=55 format=$pname. 'PAYER' '' style={just=l cellwidth=40%};
     define fillyr / group across format=$f_yr. width=10 '' style={just=r cellwidth=9%};
     define total / computed format=comma10. 'TOTAL' ''  style={just=r cellwidth=9%};
     define records / format=comma10. '' ;
     compute total ;
```

```
          total=sum(_c2_,_c3_,_c4_);
     endcomp;
     compute after;
       pnum="Total:";
     endcomp;
     rbreak after/dol dul summarize;
     Title3 "Table 27C: Date Variable - Fill Year";
run;


************************************************************************;
* Table: 27D. Date Variables Analysis--Kept Records Only: Birth Day      *;
************************************************************************;

proc report data=one missing nowd box  headline split='*';
     where rd_dat_f='0' and rd_zip_f='0';
     column pnum rbirthdd,records  Total;
     define pnum / group width=55 format=$pname. 'PAYER' '' style={just=l cellwidth=40%};
     define rbirthdd / group across format=$b_day. width=10 '' style={just=r cellwidth=9%};
     define total / computed format=comma10. 'TOTAL' ''  style={just=r cellwidth=9%};
        define records / format=comma10. '' ;
     compute total ;
          total=sum(_c2_,_c3_,_c4_);
     endcomp;
     compute after;
       pnum="Total:";
     endcomp;
     rbreak after/dol dul summarize;
     Title3 "Table 27D: Date Variable - Birth Day";
run;


************************************************************************;
* Table: 27E. Date Variables Analysis--Kept Records Only: Birth Month    *;
************************************************************************;

proc report data=one missing nowd box  headline split='*';
     where rd_dat_f='0' and rd_zip_f='0';
     column pnum rbirthmo,records  Total;
     define pnum / group width=55 format=$pname. 'PAYER' '' style={just=l cellwidth=40%};
     define rbirthmo / group across format=$mth. width=10 '' style={just=r cellwidth=9%};
     define total / computed format=comma10. 'TOTAL' '' style={just=r cellwidth=9%};
        define records / format=comma10. '' ;
     compute total ;
          total=sum(_c2_,_c3_,_c4_);
     endcomp;
     compute after;
       pnum="Total:";
     endcomp;
     rbreak after/dol dul summarize;
     Title3 "Table 27E: Date Variable - Birth Month";
run;


************************************************************************;
* Table: 27F. Date Variables Analysis--Kept Records Only: Birth Year     *;
************************************************************************;

proc report data=one missing nowd box  headline split='*';
     where rd_dat_f='0' and rd_zip_f='0';
     column pnum rbirthyr,records  Total;
     define pnum / group width=55 format=$pname. 'PAYER' '' style={just=l cellwidth=40%};
     define rbirthyr / group across format=$yr. width=10 '' style={just=r cellwidth=9%};
     define total / computed format=comma10. 'TOTAL' '' style={just=r cellwidth=9%};
        define records / format=comma10. '';
     compute total ;
          total=sum(_c2_,_c3_,_c4_,_c5_,_c6_,_c7_);
     endcomp;
     compute after;
       pnum="Total:";
     endcomp;
```

```
     rbreak after/dol dul summarize;
     Title3 "Table 27F: Date Variable - Birth Year";
run;


*****************************************************************************;
* Table: 27G. Date Variables Analysis--Kept Records Only: Drug Approval Date *;
*****************************************************************************;
proc report data=one missing nowd box  headline split='*';
     where rd_dat_f='0' and rd_zip_f='0';
     column pnum appr_yr,records  Total;
     define pnum / group width=51 format=$pname. 'PAYER' '' style={just=l cellwidth=35%};
     define appr_yr / group across format=$a_yr. width=10 '' style={just=r cellwidth=9%};
     define total / computed format=comma10. 'TOTAL' '' style={just=r cellwidth=9%};
        define records / format=comma10. '' ;
     compute total ;
          total=sum(_c2_,_c3_,_c4_,_c5_,_c6_);
     endcomp;
     compute after;
       pnum="Total:";
     endcomp;
     rbreak after/dol dul summarize;
     Title3 "Table 27G: Date Variable - Drug Approval Date";
run;


************************************************************************;
* Table: 2X. Patient Level Analysis                                    *;
************************************************************************;

proc summary nway data=one;
class pnum pidbdg;
id aggct;
where rd_dat_f='0' and rd_zip_f='0';
var rpatliab rpayment;
output out=patlev(rename=(_freq_=services _type_=patients)) sum=;
run;


************************************************************************;
* Table: 28. Patient Level Analysis: Number of Prescriptions/Patient     *;
************************************************************************;

proc tabulate data=patlev s={outputwidth=9%};
class pnum;
var patients services;
table  pnum = " " all = "Total:",
       patients*n*[format=comma10.]
       services*mean services*median
       services*std services*p5
       services*p95
       /rts=60 box="Payer";
label services="# of Prescr. per Patient" patients="# of Patients" ;
format pnum $pname.;
     Title3 "Table 28: Patient Level Analysis: Number of Prescription per Unique Patient";
     Footnote2 "This table is generated after applying the exclusion rule";
     Footnote3 "Patients:Number of Unique Patient ID-Birthdate-Sex combination";
     Footnote4 "Prescription: Each record in the database is a prescription";
run;

************************************************************************;
* Table: 29. Patient Level Analysis: Total Payment per Patient           *;
************************************************************************;

proc tabulate data=patlev s={outputwidth=9%};
class pnum;
var patients rpayment;
table pnum = " " all = "Total:",
      patients*n*[format=comma10.]
      rpayment*mean rpayment*median
      rpayment*std rpayment*p5
```

```
      rpayment*p95
      /rts=60 box="Payer";
label rpayment="Payment per Patient" patients="# of Patients";
format pnum $pname.;
     Title3 "Table 29: Patient Level Analysis: Total Payment per Unique Patient";
     Footnote2 "This table is generated after applying the exclusion rule";
     Footnote3 "Patients:Number of Unique Patient ID-Birthdate-Sex combination";
     Footnote4 "Prescriptions: Each record in the database is a prescription";
run;


*************************************************************************;
* Table: 30. Patient Level Analysis: Total Patient liability per Patient *;
*************************************************************************;

proc tabulate data=patlev s={outputwidth=9%};
class pnum;
var patients rpatliab;
table pnum = " " all = "Total:",
      patients*n*[format=comma10.]
      rpatliab*mean rpatliab*median
      rpatliab*std rpatliab*p5
      rpatliab*p95
      /rts=60 box="Payer";
label rpatliab="Liability per Patient" patients="# of Patients" ;
format pnum $pname.;
     Title3 "Table 30: Patient Level Analysis: Total Liability per Unique Patient";
     Footnote2 "This table is generated after applying the exclusion rule";
     Footnote3 "Patients:Number of Unique Patient ID-Birthdate-Sex combination";
     Footnote4 "Prescriptions: Each record in the database is a prescription";
run;


*************************************************************************;
* Table: 31. Patient Level Analysis: Total Patient liability per Patient *;
* For Patients With Comprehensive Standard Health Benefit(Coverage Type=6)*;
*************************************************************************;

proc tabulate data=patlev s={outputwidth=9%};
class pnum;
where aggct='1';
var patients rpatliab;
table pnum = " " all = "Total:",
      patients*n*[format=comma10.]
      rpatliab*mean rpatliab*median
      rpatliab*std rpatliab*p5
      rpatliab*p95
      /rts=60 box="Payer";
label rpatliab="Liability per Patient" patients="# of Patients" ;
format pnum $pname.;
     Title3 "Table 31: Patient Level Analysis: Total Liability per Unique Patient";
     Title4 "For Patients With Comprehensive Standard Health Benefit(Coverage Type=6)";
     Footnote2 "This table is generated after applying the exclusion rule";
     Footnote3 "Patients:Number of Unique Patient ID-Birthdate-Sex combination";
     Footnote4 "Prescriptions: Each record in the database is a prescription";
run;


*************************************************************************;
* Table: 32. Patient Level Analysis: Total Patient liability per Patient *;
* For Patients With Individual Plan and Prv.Emp.Sponsored Plans (Coverage Type=2,3,4)*;
*************************************************************************;

proc tabulate data=patlev s={outputwidth=9%};
class pnum;
where aggct='2';
var patients rpatliab;
table pnum = " " all = "Total:",
      patients*n*[format=comma10.]
      rpatliab*mean rpatliab*median
      rpatliab*std rpatliab*p5
```

```
      rpatliab*p95
      /rts=60 box="Payer";
label rpatliab="Liability per Patient" patients="# of Patients" ;
format pnum $pname.;
     Title3 "Table 32: Patient Level Analysis: Total Liability per Unique Patient";
     Title4 "For Patients With Individual Plan and Prv.Emp.Sponsored Plans (Coverage
Type=2,3,4)";
     Footnote2 "This table is generated after applying the exclusion rule";
     Footnote3 "Patients:Number of Unique Patient ID-Birthdate-Sex combination";
     Footnote4 "Prescriptions: Each record in the database is a prescription";
run;

************************************************************************;
* Table: 33-xx NDC Analysis                                            *;
************************************************************************;

************************************************************************;
* Table 33: Validation of Payer's NDC using Multum NDC                 *;
************************************************************************;

proc report data=one nowd box missing headline split='*';
     where rd_dat_f='0' and rd_zip_f='0';
     column pnum f_ndc,records pndc Total;
        define pnum / group width=55 format=$pname. 'PAYER* ' style={just=l cellwidth=40%};
     define f_ndc/group across format=$f_valid. width=10 '' style={just=r outputwidth=9%};
     define total /computed format=comma10. 'TOTAL* ' ;
     define records/format=comma10. '' ;
        define pndc / computed format=percent8.2 'PERCENT*VALID';
     compute total ;
          total=sum(_c2_,_c3_,_c4_);
     endcomp;
    compute after;
       pnum="Total:";
     endcomp;
        compute pndc;
          pndc=_c4_/(sum(_c2_,_c3_,_c4_));
     endcomp;
     rbreak after/dol dul summarize;
     Title3 "Table 33: NDC Level Analysis: Validation of Payer's NDC using Multum NDC";
     Footnote2 "This table is generated after applying the exclusion rule";
        footnote3 "Valid Means that the payer's 11-digit NDC matches Multum 11-digit NDC";
run;

proc summary  nway
  data=one;
class pnum NDCLABL;
id multdc1 mdcl;
where rd_dat_f='0' and rd_zip_f='0';
var  rpayment;
output out=ndclev1(rename=(_freq_=services ) drop=_type_) sum=;
run;

proc summary nway data=ndclev1;
class pnum;
var services rpayment;
output out=ndclev2(drop=_type_ _freq_) sum=tserv trpay;
run;

data ndclev;
merge ndclev1 ndclev2;
by pnum;
if multdc1="" then multdc1="999";

PCNT=services/tserv;

run;


proc sql;
create table mdclev as
select pnum, mdcl, sum(rpayment) as rpayment, sum(services) as services
```

```
from ndclev
group by pnum, mdcl;
quit;


*****Rank by payer*****;

proc rank data=ndclev out=rankndc descending;
by pnum notsorted ;
where multdc1 ne "999";
var services;
ranks rankserv;
run;

proc sort data=rankndc;
by pnum rankserv;
run;

******Rank Overall******;

proc summary  nway data=ndclev;
class NDCLABL;
id multdc1;
var  rpayment  services;
output out=ndclevall1 sum=;
run;


proc summary nway data=ndclevall1;
var   services rpayment ;
output out=ndclevall2 sum=tserv trpay;
run;

data ndclevall3;
if _n_=1 then set ndclevall2;
set ndclevall1 ;

PCNT=services/tserv;
if multdc1="" then multdc1="999";
run;

proc rank data=ndclevall3 out=rankndcall descending;
where multdc1 ne "999";
var services;
ranks rankserv;
run;

proc sort data=rankndcall;
by  rankserv;
run;


proc print data=rankndcall noobs split='*';
    var rankserv NDCLABL multdc1  rpayment services pcnt;
    where  rankserv<=50 ;
    format rankserv 4. NDCLABL $lbndc. services rpayment comma15. pcnt percent8.2 ;
    label rankserv="Rank" NDCLABL="NDCLABL (1st 9-Digit of NDC)" multdc1="Minor*Drug Category"
          services="Total*Prescriptions" rpayment="Total*Payment" pcnt="% of
Total*Prescriptions";


     Title3 "Table 34: NDC Level Analysis: 50 Most Frequent Drug Prescriptions-All Payers
Combined";
     Footnote2 "This table is generated after applying the exclusion rule";
     Footnote3 "NDCLABL: 1st 9-digit codes of NDC (Labeler+product)";
     Footnote4 "Prescriptions: Each record in the database is a prescription";
run;


proc print data=rankndc noobs split='*';
    var rankserv NDCLABL multdc1  rpayment services pcnt;
```

```
    by pnum;
    pageby pnum;
    where  rankserv<=50 and pnum in ("P020","P030","P130","P131","P160",
                                     "P480","P500","P520","P620","P820");
    format pnum $pname. rankserv 4. NDCLABL $lbndc. services rpayment comma15. pcnt percent8.2 ;
    label rankserv="Rank" NDCLABL="NDCLABL (1st 9-Digit of NDC)" multdc1="Minor*Drug Category"
          services="Total*Prescriptions" rpayment="Total*Payment" pcnt="% of
Total*Prescriptions";

     Title3 "Table 35: NDC Level Analysis: 50 Most Frequent Drug Prescriptions by Payer";
     Footnote2 "This table is generated after applying the exclusion rule";
     Footnote3 "NDCLABL: 1st 9-digit codes of NDC (Labeler+product)";
     Footnote4 "Prescriptions: Each record in the database is a prescription";
run;


proc tabulate data=mdclev order=formatted;
class mdcl;
var services rpayment;
table mdcl=""  all = "Total:",services*sum*[format=comma13.]
             services*pctsum*[format=comma13.1]
             rpayment*sum*[format=comma15.]
             rpayment*pctsum*[format=comma13.1]
             /rts=50 box="Major Drug Category";
label rpayment="Payment" services="Prescriptions";
format pnum $pname. mdcl $mdcl.;
     Title3 "Table 36: NDC Level Analysis: Distribution of Payment by Major Drug Category";
     Title4 "All Payers Combined";
     Footnote2 "This table is generated after applying the exclusion rule";
     Footnote3 "Only NDC Codes matched to the FDA database are included";
     Footnote4 "Prescriptions: Each record in the database is a prescription";
run;

%macro mdclass (pn,pnl);
proc tabulate data=mdclev order=formatted;
class pnum mdcl;
where pnum="&pn";
var services rpayment;
table mdcl=""  all = "Total:",services*sum*[format=comma13.]
             services*pctsum*[format=comma13.1]
             rpayment*sum*[format=comma15.]
             rpayment*pctsum*[format=comma13.1]
             /rts=50 box="Major Drug Category";
label rpayment="Payment" services="Prescriptions";
format pnum $pname. mdcl $mdcl.;
     Title3 "Table 37: NDC Level Analysis: Distribution of Payment by Major drug Category";
     Title4 "&pn : &pnl";
     Footnote2 "This table is generated after applying the exclusion rule";
     Footnote3 "Only NDC Codes matched to the FDA database are included";
     Footnote4 "Prescriptions: Each record in the database is a prescription";
run;
%mend;
%mdclass( P020,Aetna Life & Health Insurance Co.);
%mdclass( P030,Aetna U.S. Healthcare);
%mdclass( P130,Carefirst DC);
%mdclass( P131,Carefirst MD);
%mdclass( P160,CIGNA Healthcare Mid-Atlantic Inc.);
%mdclass( P480,Kaiser Found. Health Plan of M-A);
%mdclass( P500,MAMSI Life and Health Insurance Co.);
%mdclass( P520,MD-Individual Practice Association Inc.);
%mdclass( P620,Optimum Choice Inc.);
%mdclass( P820,United Healthcare Corporation);


************************************************************************;
* Table 38: NDC Level Analysis - Brand or Generic Indicator              *;
************************************************************************;

proc report data=one nowd box missing headline split='*';
     where rd_dat_f='0' and rd_zip_f='0';
     column pnum gbo,records  Total;
```

```
        define pnum / group width=55 format=$pname. 'PAYER* ' style={just=l cellwidth=40%};
     define gbo /group across format=$Gbo. width=10 '' style={just=r cellwidth=9%};
     define total /computed format=comma10. 'TOTAL* ' style={just=r cellwidth=9%};
     define records/format=comma10. '' ;
        compute total ;
          total=sum(_c2_,_c3_,_c4_);
     endcomp;
    compute after;
       pnum="Total:";
     endcomp;
        rbreak after/dol dul summarize;
     Title3 "Table 38: NDC Level Analysis - Brand vs. Generic Drugs";
     Footnote2 ;
        footnote3 ;
 run;


************************************************************************;
* Table 39: NDC Level Analysis - New Vs Old Drugs                      *;
************************************************************************;

proc report data=one nowd box missing headline split='*';
     where rd_dat_f='0' and rd_zip_f='0';
     column pnum RF_NDRUG,records  Total;
        define pnum / group width=55 format=$pname. 'PAYER* ' style={just=l cellwidth=40%};
     define RF_NDRUG /group across format=$Ndrug. width=10 '' style={just=r cellwidth=9%};
     define total /computed format=comma12. 'TOTAL* ' style={just=r cellwidth=9%};
     define records/format=comma12. '';
        compute total ;
          total=sum(_c2_,_c3_,_c4_,_c5_);
     endcomp;
    compute after;
       pnum="Total:";
     endcomp;
        rbreak after/dol dul summarize;
     Title3 "Table 39: NDC Level Analysis - New vs. Old Drugs";
     Footnote2 ;
        footnote3 ;
 run;

ods pdf close;

%jobend;


*** save log and output files;

dm "log;    print file=D:\MCdb\DataCol\RX_Step8\log_out\Step8_RX_Stats.log replace";
dm "output; print file=D:\MCdb\DataCol\RX_Step8\log_out\Step8_RX_Stats.lst replace";
```

```
%jobstart;

*==========================================================*;
* Program:    MCDB_RX_Completeness_Summary.sas            *;
* Programmer: Adrien Ndikumwami/Po-Lun Chou               *;
* Date:       September 12, 2007                          *;
*==========================================================*;
%LET NOBS=max;

%LET PREVYR2=05;
%LET CURRYR2=06;
%LET NXTYR2=07;

options obs = &nobs nodate;

**libname Input is stored;
LIBNAME PREV 'E:\Mcdb\DataCol\RX_Step8\data' ACCESS=READONLY;
LIBNAME CURR 'D:\Mcdb\DataCol\RX_Step8\data' ACCESS=READONLY;
LIBNAME LIBRARY 'D:\MCDB\DATACOL\FORMATS\PERMANENT';

options nosource2;
%include "D:\Mcdb\DataCol\RX_Step8\programs\Template_ODS_Step8.sas";
%include 'D:\Mcdb\DataCol\Formats\mcdbfmt.sas';
%include 'D:\Mcdb\DataCol\Formats\Format_codebook.sas';
%include 'D:\Mcdb\DataCol\Formats\DrugCategories_codebook.sas';
options source2;

options mprint mlogic symbolgen;


***From year to year, we just need to check the cards in the datastep below****;
*** Create a look up table for format used by Alphanumeric variables ***;

data charfmts;
     input @ 1 name $ 10.
           @14 fmt  $ 14.
;

***Flag dates because they use formats and no means is computed on them***;

cards;
DEANUM       $unk.
FILLMO       $monfmt.
GBO          $gbo.
MULTDC1      $mdcat.
NCPDP        $rptmiss.
NDC          $rptmiss.
NDCLABL      $rptmiss.
NDCPACK      $rptmiss.
PATID        $rptmiss.
RBIRTHMO     $monfmt.
RBIRTHYR     $byr.
covtype      $covrtyp.
DELIVTYP     $delvsys.
RF_10K       $rf_10k.
RF_DBC       $dbcfmt.
RF_ELINK     $elink.
RF_FINMS     $ffinms.
RF_MD        $rf_md.
RF_MORDER    $fmorder.
RF_NDC       $ndcfmt.
RF_NDRUG     $f_ndrug.
RF_PAYM      $f_paym.
RF_PLINK     $plink.
RF_YR65      $rf_65yr.
RPATZIP      $rptmiss.
RPHARMGRP    $pgrp.
RSEX         $sex.
RZIP_S       $rptmiss.
RBCHRGE
RREIMAMT
```

```
RPATLIAB
DRUGQTY
DRUGSPL
RAGE
RPAYMENT
;

run;


%macro GenComp(YR=, LIB=);

/*** Get variables automatically using SASHELP Dictionary views ***/

proc sql;
Create table CONTENT as
select  Upcase(A.NAME) as Name, Upcase(A.TYPE) as Type, A.LABEL ,A.LENGTH,B.FMT
from sashelp.vcolumn A, charfmts B
where upcase(memname)="MCDBRX&YR._0"
and libname="&LIB"
and upcase(A.name)=upcase(B.name)
order by Name;
quit;


*****************************************************************;
* 1. Create a dynamic format VARLBL for variable labels, and    *;
*    create a dynamic format VARLEN for variable length         *;
*****************************************************************;

data varfmt;
  set content(drop=type);
  rename name=start;
  type="C";
  fmtname="varlbl";
run;

proc format cntlin=varfmt;
run;

data varfmtl;
  length label $5;
  set content(drop=type label);
  rename name=start;
  type="C";
  label = put(length, 3.);
  fmtname="varlen";
run;

proc format cntlin=varfmtl;
run;


*****************************************************************;
* 2. Create separate macro variables for numeric and alphanumeric *;
*    variables. This is used to dynamically call the variables.   *;
*    Include also a macro variable to store the format name of    *;
*    alpha variable. This needed for value + descriptions for     *;
*    alphanumeric variable.                                       *;
*****************************************************************;

*** Create a macro var to use with Numeric Variables -exclude dates***;

data _null_;
  set content (where=(type="NUM" )) end=last;
  call symput("num"||left(put(_n_,3.)),name);
  if last then call symput("numobs",put(_n_,3.));
run;


*** Create macro vars to use with Alphanumeric Variables ***;
```

```
data _null_;
  set content (where=(type="CHAR")) end=last;
  by name;
  call symput("chr"||left(put(_n_,3.)),name);
  call symput("fmt"||left(put(_n_,3.)),fmt);
  if last then call symput("chrobs",put(_n_,3.));
run;


*******************************************************************;
* 3.Create macro GENSTAT to generate the statistics               *;
*   a)Get the mean for numeric variables and transpose            *;
*   b)Get freq for each alphanumeric variables  and append        *;
*******************************************************************;

  *** a)Get the mean for numeric variables and transpose ***;

  proc means data=&lib..MCDBRX&Yr._0 nway   mean  noprint;
   where fillyr="20&Yr";
    output out=_numstat&Yr(drop=_type_);
        class pnum;
    var %do i=1 %to (%eval(&numobs));
        &&num&i
    %end;
      ;
  run;


proc transpose data=_numstat&Yr (drop=_freq_)
                 out=numstat_&Yr(rename=(_name_=varname _label_=explcode Mean=Mean&Yr));
    id _stat_;
        by pnum;
        where _stat_="MEAN";
run;

data numstat&Yr;
  set numstat_&Yr;
  varname=upcase(varname);
  porder='04';
run;


proc sort data=numstat&Yr;
   by pnum porder varname explcode;
run;

  *** b)Get freq for each alphanumeric variables ***;

  proc freq data=&lib..MCDBRX&Yr._0 noprint;
  where fillyr="20&Yr";
     %do i=1 %to (%eval(&chrobs));
        tables   pnum* &&chr&i / missing out=outp&i;
       format   &&chr&i  &&fmt&i;
    %end;
  run;

  *** Add variable name & format for each dataset corresponding to each variable ***;

  %do i=1 %to (%eval(&chrobs));
     proc sql;
       alter table outp&i
       add varname char(10), fmtlbl char(95);
       update outp&i
       set varname=substr(left("&&chr&i"),1,10) ,
           fmtlbl=put(&&chr&i,&&fmt&i);    *format created by FREQ is temporary;
       quit;
  %end;

  *** Combine all datasets corresponding to char variables ***;
```

```
  data chrstat&Yr;
    set %do i=1 %to (%eval(&chrobs));
           outp&i(drop=&&chr&i)
        %end;
      ;
    length code $25 explcode $70;
    usetype="Character";
    code=scan(fmtlbl,1,"?");      *split fmtlbl into 2;
    explcode=scan(fmtlbl,2,"?");
    if varname in ("RBIRTHYR","FILLYR") and explcode=""
    then explcode="Year = "||left(code);
        Count&Yr=Count;
         porder='03';
  keep porder pnum varname ExplCode code count&Yr ;
  run;

proc sort data=chrstat&Yr;
   by  pnum porder varname explcode;
run;

  ***Number of services and recipients ****;

  proc sql;
  create table Recipients&Yr as
  select '02' as porder, count(distinct(pidbdg)) as  count&Yr, pnum, "RECIPIENTS" as varname,
   "Total Recipients" as explcode
  from  &lib..MCDBRX&Yr._0
  where fillyr="20&Yr"
  group by pnum
  order by pnum;

  ***Number of services and recipients ****;

  proc sql;
  create table Services&Yr as
  select '01' as porder, count(pidbdg) as  count&Yr, pnum, "PRESCRIPTIONS" as varname,
   "Total Prescriptions" as explcode
  from  &lib..MCDBRX&Yr._0
  where fillyr="20&Yr"
  group by pnum
  order by pnum;


data all&Yr;
length explcode $70;
merge services&Yr chrstat&yr recipients&Yr numstat&Yr;
by pnum porder varname explcode;
run;

%mend GenComp;

%GenComp(Yr=05, Lib=PREV);
%GenComp(Yr=06, Lib=CURR);


*************************************************************;
* 4. Merge this year's data with last year's .               *;
*************************************************************;

Data allYr_;
  merge all&CurrYr2 all&PrevYr2 ;
  by pnum porder varname explcode;
  CountPCT=((Count&CurrYr2/Count&PrevYr2)-1)*100;
  MeanPCT=((Mean&CurrYr2/Mean&PrevYr2)-1)*100;
  if varname = "RBCHRGE" then varname = "RBILL";
  if varname = "RREIMAMT" then varname = "RREIMB";
run;

data all (drop = totserv05 totserv06);
  merge allYr_
        services&PrevYr2 (keep=pnum count05 rename=(count05=totserv05))
```

```
        services&CurrYr2 (keep=pnum count06 rename=(count06=totserv06));
  by pnum;
  pct_05 = (count05 / totserv05) * 100;
  pct_06 = (count06 / totserv06) * 100;

  if varname="RECIPIENTS" then do;
     pct_05=.;
        pct_06=.;
  end;
  if varname = 'RF_ELINK' and code = ' ' then do;
     code = '0';
     explcode= 'No Link';
  end;
  if varname = 'RF_PLINK' and code = ' ' then do;
     code = '0';
     explcode= 'No Link';
  end;

run;

data _null_;
 wdate= put(today(),worddate.);
 call symput ('tdate', wdate);
run;

options orientation=landscape missing="-";
ODS PDF STYLE=CUSTSTEP8  BODY="D:\Mcdb\DataCol\RX_STEP8\completeness06.PDF";
ODS ESCAPECHAR = "^" ;


title1 "Pharmacy Data Completeness Summary Report";
title2 "Data Source: 2005 and 2006 Raw Data";
Footnote j=r "Run Date: &tdate";

%macro A1(pnum, pname);

proc report data=all missing nowd box headline split='*';
     where pnum="&pnum";
*    format pnum $pname.;
     column pnum porder Varname code explcode  Count05 pct_05  Count06  pct_06 CountPCT
            Mean05 Mean06 MeanPCT;
        define pnum/ group noprint;
        define porder/ group noprint;
        define code/ group noprint;
     define Varname / group "Variable*Name" style={just=L cellwidth=13%};
     define explcode / "Value*Description"   style={just=L cellwidth=30%};
     define Count06 / "2006*Count" format=comma10. style={just=R cellwidth=8%};
     define pct_06 / "% Distr.*2006" format=comma8.1  style={just=R cellwidth=7%};
     define Count05 / "2005*Count" format=comma10. style={just=R cellwidth=8%};
     define pct_05 / "% Distr.*2005" format=comma8.1  style={just=R cellwidth=7%};
     define CountPCT / "% Change*2005-2006"  format=comma8.1  style={just=R cellwidth=8%};
     define Mean06 / "2006*Mean" format=comma8.1  style={just=R cellwidth=5%};
     define mean05 / "2005*Mean"  format=comma8.1 style={just=R cellwidth=5%};
     define MeanPCT / "% Change*2005-2006"  format=comma8.1 style={just=R cellwidth=8%};
     title4 "&pnum - &pname";
run;

%mend A1;

%A1(pnum = P020, pname = Aetna Life & Health Insurance Co.);
%A1(pnum = P030, pname = Aetna U.S. Healthcare);
%A1(pnum = P070, pname = American Republic Insurance Co.);
%A1(pnum = P130, pname = CareFirst BlueChoice Inc.);
%A1(pnum = P131, pname = CareFirst of Maryland Inc.);
%A1(pnum = P160, pname = CIGNA Healthcare Mid-Atlantic Inc.);
%A1(pnum = P280, pname = Time Insurance Co.(Assurant Health));
%A1(pnum = P320, pname = Golden Rule Insurance Co.);
%A1(pnum = P325, pname = Graphic Arts Benefit Corporation);
%A1(pnum = P330, pname = Great-West Life & Annuity Insurance Co.);
%A1(pnum = P350, pname = Guardian Life Insurance Co. of America);
%A1(pnum = P471, pname = Unicare Life & Health Insurance Co.);
```

```
%A1(pnum = P480, pname = Kaiser Foundation Health Plan of the Mid-Atlantic States Inc.);
%A1(pnum = P500, pname = MAMSI Life and Health Insurance Co.);
%A1(pnum = P510, pname = Maryland Fidelity Insurance Co.);
%A1(pnum = P520, pname = MD-Individual Practice Association Inc.);
%A1(pnum = P530, pname = MEGA Life & Health Insurance Co.);
%A1(pnum = P620, pname = Optimum Choice Inc.);
%A1(pnum = P680, pname = Coventry Healthcare of Delaware Inc.);
%A1(pnum = P760, pname = State Farm Mutual Automobile Insurance Co.);
%A1(pnum = P820, pname = United Healthcare Insurance Co.);
%A1(pnum = P830, pname = Trustmark Insurance Co.);
%A1(pnum = P850, pname = Union Labor Life Insurance Co.);
%A1(pnum = P870, pname = United Healthcare of the Mid-Atlantic Inc.);

ODS PDF close ;


%jobend;

dm "log;    print file=D:\Mcdb\DataCol\RX_STEP8\log_out\MCDB_RX_Completeness_Summary.log
replace";
dm "output; print file=D:\Mcdb\DataCol\RX_STEP8\log_out\MCDB_RX_Completeness_Summary.lst
replace";
```

```
%jobstart;

*********************************************************************;
*                                                                   ;
*  Program Name:  d:\MCdb\DataCol\RX_Step8\RX_Step8_AllPayers_1.SAS  ;
*                                                                   ;
*  Programmer:    Adrien Ndikumwami                                 ;
*                                                                   ;
*  Description:   Scan through the directories for RX files         ;
*                 and concatenate them into one file that will be   ;
*                 for initial statistical analysis.                 ;
*                                                                   ;
*  DATE CREATED: 10/13/2000 by Adrien Ndikumwami                    ;
*                12/26/2000 -Change variables as requested by MHCC  ;
*  DATE UPDATED: 7/26/2005 - moved RF_NDC, MDC1-MDC4 to EXCSERVRX   ;
*                6/30/06   - MOVED MULTDC1-MULTDC4 TO EXCSERVX      ;
*                                                                   ;
*  Modified by:  Po-Lun Chou                                        ;
*  Date:         September 21, 2007                                 ;
*  Purpose:      Modify year (from 05 to 06) and run this program to ;
*                generate MCDB Pharmacy data file for 2006.         ;
*********************************************************************;
options nodate  pageno=1 obs= max; /*ls=148 ps=60*/

libname db "d:\MCDB\Datacol\RX_Step8\Data\";
options nosource2;
LIBNAME LIBRARY "D:\Mcdb\DataCol\Formats\Permanent";
options source2;

****************************************;
*   Create macro for MHCC title         *;
****************************************;

Title1 "MARYLAND HEALTH CARE COMMISION - PHARMACY STEP 8 -- Run on &sysday, &sysdate at
&systime";

*********************************************************************;
*  Steps:                                                           *;
*  1. Find all the directories used by MCDB data and create         *;
*     a libname including all those directories.                    *;
*  2. Find all RX files present in those directories                *;
*  3. Concatenate all the files                                     *;
*                                                                   *;
*********************************************************************;

*********************************************************************;
*  1. Find all the directories used by MCDB data and create         *;
*     a libname including all those directories                     *;
*********************************************************************;

OPTIONS NOXWAIT;

x 'DIR/A:d d:\MCDB\DATACOL\P??? > d:\WORKAREA\TEMP_DIR.TXT';

data one;
  infile  "d:\WORKAREA\TEMP_DIR.TXT" firstobs=6;
  input  @40 payer $4.;
  If upcase(substr(payer,4,1)) = "X" or payer="" then delete;
run;

data _null_;
  set one end=eof;
  call symput('p'||left(_N_), payer);
  if eof then call symput('nobs',_N_);
run;

************Start macro to assign libname*****************;
**********************************************************;
```

```
%macro payrinfo;

  libname rx ( %do i=1 %to &nobs;
                              "d:\MCDB\DATACOL\&&p&i\sasdata",
               %end;
                              "d:\MCDB\DATACOL\&&p&nobs\sasdata");
%mend payrinfo;

%payrinfo;



*********************************************************************;
*  2. Find all RX files present in those directories                 *;
*********************************************************************;

PROC CONTENTS NOPRINT DATA=rx._ALL_
                      OUT=two(KEEP=MEMNAME nobs);
run;

proc sort nodupkey data=two out=three;    *keep 1 record per payer;
   by memname ;
   where substr(memname,5,3)="RX4";
run;

proc print data=three label;
  sum nobs;
  title2 "Number of observations for each payer before concatenation";
  label memname="Payer file" nobs="No. of Records";
run;


data _null_;
  set three  end=eof;
  call symput('pfn'||left(_N_), memname);  * Macro var for payers' filenames;
  if eof then call symput('npayer',_N_);   * Macro var for total number of payers;
run;

*********************************************************************;
*  3. Concatenate all the files                                      *;
*********************************************************************;


******Start macro to append all the files using payer filename********;
***************automatically generated by call symput*****************;
*********While appending variables are renamed to agree with**********;
**************************naming convention***************************;
*********************************************************************;
*options mprint mlogic symbolgen;

%macro appayer;    *to append all payers files;

   data db.mcdbrx06_1(drop= FILLYRX   DRUGCMP_  DRUGQTY_  DRUGSPL_
                            RBCHRGE_  RBIRTHDD
                            RPATLIB_  RREIAMT_  Rf_bcout  Rf_finodd
                            drugcmp   patid_raw
                            lu_limit  neg_fin   pex_ind   rcd_id
                            rf_finzr  NCPDP_ST  Rf_paout  recnum
                            pa_limit  rd_dat_f  dea_payer encserv
                            ncpdp6    payer     xref_link);

  set
   %do j=1 %to &npayer;
     rx.&&pfn&j (rename=(RBCHRGE  = RBILL     NDC2      = NDCPACK   NDC9        = NDCLABL
                         PHAZIP   = RZIP_S    speclty1  = RSPECL1   speclty2    = RSPECL2
                         speclty3 = RSPECL3   rf_multi  = RF_MULTI  rf_plink    = RF_PLINK
                         rscript  = RSCRIPT   rpatliab  = RPATLIAB  rpayment    = RPAYMENT
                         rf_patid = RF_PATID  rf_yr65   = RF_YR65   rf_ndup     = RF_NDUP
                         rf_dbc   = RF_DBC    rf_finms  = RF_FINMS  rd_zip_f    = RD_ZIP_F
                         rf_10k   = RF_10K    rage1206  = RAGE1206  rage        = RAGE
                         apprdate = APPRDATE  gbo       = GBO       drug_id     = DRUG_ID
```

```
                        DELIVTYP = RDELIVTYP pidbdg    = PIDBDG    pnum        = PNUM
                        rF_PAYM  = RF_PAYMD  COVTYPE   = RCOVTYPE  enrolldt    = ENROLLDT
                        RREIMAMT = RREIMB    rf_medsup = RF_MEDSUP disenrolldt = DISENROLLDT))
   %end; ;

   *Change variable labels;

   if RF_PLINK=""  then RF_PLINK="0";
   if RF_ELINK=""  then RF_ELINK="0";

   Label RAGE      = "Patient Age at Service"
         RAGE1206  = "Patient Age as of 12/31/2006"
         RBILL     = "Billed Charge-BC ($)"
         RBIRTHMO  = "Patient Birth Month"
         RBIRTHYR  = "Patient Birth Year"
         RF_FINMS  = "All 3 Financial Vars=0/Missing (1/0)"
         RF_PAYMD  = "Payment Category (1=<0,2=0,3=>0,4=.)"
         RF_YR07   = "Prescription Filled in 2007(1:>=07 0:<07)"
         NCPDP     = "Pharmacy Number (NCPDP) "
         NDC       = "National Drug Code (NDC-11 Chars)"
         NDCPACK   = "Package Code(Chars 10-11 of NDC)"
         NDCLABL   = "Labeler-Product Code(Chars 1-9 of NDC)"
         APPRDATE  = "FDA Approval Date"
         RPATLIAB  = "Patient Liability-PL ($)"
         RPATZIP   = "Patient Residence ZIP Code"
         RPAYMENT  = "Payment = Sum(RA,PL)"
         PNUM      = "Payer Unique ID  "
         DEANUM    = "Practitioner DEA#"
         RREIMB    = "Reimbursement Amount-RA ($)"
         RSEX      = "Patient Sex"
         PATID     = "Patient ID, Encrypted"
         RZIP_S    = "Pharmacy ZIP Code"
         FILLDATE  = "Date Prescription Filled"
         FILLDD    = "Day Prescription Filled"
         FILLMO    = "Month Prescription Filled"
         FILLYR    = "Year Prescription Filled "
         DRUGCMP   = "Drug Compound(1=Compound,2=Non-Cmpnd)"
         DRUGQTY   = "Drug Quantity (No. of Units Dispensed)"
         DRUGSPL   = "Drug Supply (Est. # of Supply Days)"
         RF_DBC    = "Debit/Credit Flag 0=No, 1=Yes, 2,3=Dupe 4=Not D/C after Adj"
         RF_10K    = "Flag for Payment > 10K"
         RF_PATID  = "Flag for Patient ID, 0=OK,1=0s,2=9s,3=Blank"
         RF_YR65   = "Age>65, 0:<65, 1:>=65, 2:Missing"
         GBO       = "Generic/Brand Indicator"
         PIDBDG          = "Unique Patient ID(PNUM/PATID/BRTHYR/BRTHMO/GENDER)"
         MULTDC1   = "Multum Drug Category 1"
         MULTDC2   = "Multum Drug Category 2"
         MULTDC3   = "Multum Drug Category 3"
         MULTDC4   = "Multum Drug Category 4"
         RF_ELINK  = "Encounter Linkage Indicator"
         RF_PLINK  = "Provider Linkage Indicator"
         RF_MD     = "Maryland Residency Indicator (1-MD, 0-non MD)"
         RF_NDUP   = "Duplicate Set W/ at least One Neg. Bill"
         RMSCAT    = "Medical Supply Category"
         RF_MEDSUP = "Medical Supply Indicator(0=No, 1=Yes)";
run;

%mend appayer;

%appayer;

proc contents data=db.mcdbrx06_1;
  title2 "Contents of Data Set MCDBRX06_1";
run;

proc freq data= db.mcdbrx06_1;
  tables PNUM MULTDC1 RMSCAT RF_MEDSUP * RMSCAT / list missing;
  title2 " ";
*title2 "Number of Services per Payer --From the concatenated file--MCDBRX06_1";
  label PNUM="Payer";
  format RMSCAT $MS_CATEGORY.;
```

```
run;

***Delete temporary file;

x 'DEL d:\WORKAREA\TEMP_DIR.TXT';

%jobend;


*** save log and output files;

dm 'log;    print file=D:\MCdb\DataCol\RX_Step8\log_out\RX_Step8_AllPayers_1.log replace';
dm 'output; print file=D:\MCdb\DataCol\RX_Step8\log_out\RX_Step8_AllPayers_1.lst replace';
```

```
proc datasets kill;
run;
quit;


%JOBSTART;

options nodate pageno=1;
title "%sysfunc(putn(%sysfunc(time()), hhmm.)) %sysfunc(putn(%sysfunc(date()), weekdate.))";

*******************************************************************************;
*  Program Name:  D:\MCDB\Pharmacy\Prog\MCDB_RX_06.sas                        *;
*  Programmer:    J May                                                        *;
*  Date Created:  01/04/2005                                                   *;
*-----------------------------------------------------------------------------*;
*  Description:   Screen and flag MCDBRX06.  Create ANALRX06 with RF_ANAL flag.    *;
*-----------------------------------------------------------------------------*;
*  Modified Date: April 21, 2005                                               *;
*  Modified By:   Po-Lun Chou                                                  *;
*  Purpose:       Created a new MCDB Pharmacy data set and a new Analysis Pharmacy *;
*                 data set to rename some variables and to drop some others.  The  *;
*                 new data set will be used in Code Book program.             *;
*  Modified Date: 7/26/05 - J May                                              *;
*                 Replaced Date literals with Macros                          *;
*                 Moved R_NAGE, RF_NDRUG, RF_MORDER AND NCPDP Edits up to EXSERVRX *;
*                 12/27/2005 - Dropped DEABUS and DRUG_ID                     *;
*  Modified Date: 7/7/06 - J May                                               *;
*                 Dropped setting of rage12&curyr2 to .M (done in Exservrx)   *;
*                 Dropped setting of RF_NDUP to zero if missing -             *;
*                 done in Debit_Credit_Analysis                               *;
*                 Moved creation of RF_COV17 to EnctlinkRX                    *;
*  Modified Date: 9/25/2006 - Sireesha                                         *;
*                 Changed the input dataset frm MCDBRX05_2 to MCDBRX05_1      *;
*                 Keep DRUG_ID, Changed the calculation of rf_yr65, RF_DEA    *;
*  Modified Date: 11/03/2006 - Sireesha Thati                                  *;
*                 Merged with Encounter to get the enroll and disenroll dates for  *;
*                 each unique patient id                                      *;
*  Modified Date: 12/07/2006 - Sireesha Thati                                  *;
*                 Dropped Enrolldt and Disenrolldt, renamed DENRDATE to RDENRDATE  *;
*                 and ENRDATE to RENRDATE                                     *;
*                                                                             *;
*  Modified By:   Po-Lun Chou                                                  *;
*  Modified Date: November 15, 2007                                            *;
*  Purpose:       Brough in combined Encounter data file, enc5_enroll_combined,    *;
*                 merged it with Pharmacy data file, MCDBRX06_1, to generate the   *;
*                 RX analysis data files, ANALRX06 & ANMORX06.                *;
*******************************************************************************;

options NOcenter ERRORS=2 OBS= max;

* libname in1 "d:\MCDB\Pharmacy\Data\";

libname in1 "D:\Mcdb\DataCol\RX_Step8\data";
libname in2 "I:\MCdb\MCDB_FN\data";
libname ot1 "D:\MCDB\Pharmacy\data\";
libname ot2 "D:\MCDB\Analysis\data\";
libname library "D:\Mcdb\DataCol\Formats\Permanent";


proc sort data = in1.MCDBRX06_1 (drop = rf_elink rf_yr65 rf_dea rscript rcovtype rdelivtyp

)
          out = mcdbrx06_1;
  by rf_yr07 pidbdg;
run;

proc sort data = in2.enc_rx_link_driver(rename=(f_yr07=rf_yr07) drop=numserv totpay TPatPay
TPatServ)
           out = rx_enc5_1;
  by rf_yr07 pidbdg delivtyp covtype descending TpatPay;
run;
```

```
data rx_enc5;
  set rx_enc5_1;
  by rf_yr07 pidbdg delivtyp covtype descending TpatPay;
  if first.pidbdg ;
  rename covtype=rcovtype  delivtyp=rdelivtyp
         disenroll_p=rdenrdate enroll_p=renrdate;
  run;

DATA OT1.MCDBRX06(COMPRESS=BINARY) OT2.ANALRX06(COMPRESS=BINARY)  ANMORX06(COMPRESS=BINARY);
  length RCOVTYPE  RDELIVTYP $1 RDENRDATE  RENRDATE 8.;
  length RAGE RAGE1206 3;
  merge mcdbrx06_1(in=a) rx_enc5(in=b);
  by rf_yr07 pidbdg;
  if a;
  if a and b then RF_ELINK="1";

  RF_ANAL="1";     *Used for regular analysis;
  RF_MOANAL="1";   *Used for mail order report;

  %LET CURYR2=06;
  %LET CURYR4=2006;
  %let yrEndDt = "31DEC&CURYR4"d;   *for computing age12yy;

********************
Adjusting rf_ndup
********************;
  if rf_ndup = "" then rf_ndup = "0";

** The Calculation of rf_yr65 is done with non-rounded age now instead of rage12&CURYR2 **;
  length RF_YR65 $1;
  if r_nage12&CURYR2 in (., .M) then RF_YR65 = "2";
  else if r_nage12&CURYR2 >= 65 then RF_YR65 = "1";
  else if r_nage12&CURYR2 < 65 then RF_YR65 = "0";

**********************************************
Create new flag, RF_DEA, based on DEA Number
**********************************************;
  length RF_DEA $1;
* RF_DEA = Put(DEANUM, $DEA.);
  RF_DEA = Put(DEANUM, $fdeanum.);
  if RF_DEA ne "1" then do;
       RF_DEA = Put(DEANUM, $lfdeanum.);
  end;

****************************************************************
Add new variable rscript -- based on the definition from claudia
****************************************************************;
  length RSCRIPT 3 ;
  if drugspl <= 30 then RSCRIPT = 1;
    else if 30 < drugspl <= 60 then RSCRIPT = 2;
    else if drugspl > 60 then RSCRIPT = 3;

* SCREEN RECORDS ON FILLDATE;
  IF FILLYR NE "&CURYR4"  THEN DO;
     RF_ANAL  ="0";
     RF_MOANAL="0";
  END;

* SCREEN IF PATIENTS >=65;       *65+ is included in mail-order analysis;
  IF RF_YR65="1" THEN DO;
     RF_ANAL="0";
  END;


* SCREEN IF FINANCIAL VARIABLES MISSING OR ZERO;
  IF RF_FINMS="1" THEN DO;
     RF_ANAL  ="0";
     RF_MOANAL="0";
  END;
```

```
* SCREEN IF RF_DBC=1 -DEBIT/CREDIT;
  IF RF_DBC ="1" THEN DO;
     RF_ANAL  ="0";
     RF_MOANAL="0";
  END;

* SCREEN IF RF_NDUP=1 -DUPLICATE SETS WITH - BILLS;
  IF RF_NDUP ="1" THEN DO;
     RF_ANAL  ="0";
     RF_MOANAL="0";
  END;

* SCREEN OUTLIERS;
  IF RF_10K="1" THEN DO;
     RF_ANAL  ="0";
     RF_MOANAL="0";
  END;

* SCREEN NDCs IF MISSING, ZEROS OR 9s;
  IF RF_NDC IN ("2","3","4") THEN DO;
     RF_ANAL  ="0";
     RF_MOANAL="0";
  END;

* SCREEN IF PATIENT IDS ARE BLANK, ZEROS OR 9s;
  IF RF_PATID NE "0" THEN DO;
     RF_ANAL  ="0";
     RF_MOANAL="0";
  END;

* SCREEN NEGATIVE PAYMENTS;
  IF RF_PAYMD="1" THEN DO;
     RF_ANAL  ="0";
     RF_MOANAL="0";
  END;

* SCREEN IF COVERAGE TYPE IS 1 OR 7; *Medicare covered patients are included in the mail order;
  IF RF_COV17 = "1" THEN DO;
     RF_ANAL = "0";
  END;

* SCREEN IF NO PATIENT ZIPCODE IS IN MARYLAND;
  IF RF_MD="0" THEN DO;
     RF_ANAL  ="0";
     RF_MOANAL="0";
  END;

  label RF_ANAL   = "Analysis Flag(1-Analysis, 0-Exclude)"
        RF_MOANAL = "Mail Order Analysis Flag(1-Analysis, 0-Exclude)"
        RF_PATID  = "Patient ID Flag"
        RF_YR65   = "Age 65+ Flag(0:<65, 1:>=65, 2:Missing)"
        RF_DBC    = "Debit/Credit Flag"
        RF_NDC    = "NDC Flag"
        PIDBDG    = "Unique Patient ID"
               RF_ELINK  = "Plarmacy/Encounter Link Flag"
        RF_PLINK  = "Pharmacy/Provider Link Flag"
        RF_DEA    = "DEA Number(1=Valid, 0=Other, 2=Blank)"
        RSCRIPT   = "Number of Adjusted or 30-day Script"
        RDENRDATE = "Disenrollment Date"
        RENRDATE  = "Enrollment Date";


  DROP /*MDC1-MDC4 DRUG_ID */ RD_ZIP_F DEABUS RF_MULTI DISENROLLDT ENROLLDT;

  if FILLYR in ("2006","2007")then do;
    if RSEX = "0" then RSEX = " ";
    output ot1.mcdbrx06;      *The final DB should have 2006 and 2007 only;
  end;

  if rf_anal = "1" then output ot2.analrx06;
  if rf_moanal = "1" then output anmorx06;
```

```
run;

data anmorx06_1 thirtydays1(keep = PIDBDG MMDC FILLDATE DRUGSPL rec_ID);
  set anmorx06;
  rec_id=put(_n_,z8.); *This ID will be used to apply the 3 30-day consecutive claims;
  output anmorx06_1;    * This file  used later on;

  if DRUGSPL=30 then output thirtydays1; ** Keep only the records with drugspl = 30 ***;
run;

proc sort data = thirtydays1;
  by PIDBDG MMDC FILLDATE;
run;

*******************************************************************************************
Each patient (pidbdg) orders 1 or more MMDC on different fill dates.
For each NDC we have different filldates.
Calculate the number of days between to consecutive fills (DateDiff)
for each MMDC.To account for patients who refill their drug early or late, the
difference between the two fills must be between 20 and 40 to qualify as "consecutive.
*******************************************************************************************
;

DATA thirtydays2 (drop=filldate_old datediff drugspl);
  SET thirtydays1;
  length filldate_old 8 datediff 3 CONSEC3_GRP  3 ;
  by PIDBDG MMDC FILLDATE;
  RETAIN FILLDATE_OLD CONSEC3_GRP  C_DRUGSPL;

  IF FIRST.MMDC THEN DO;    ****For each MMDC, initialize;
     FILLDATE_OLD = FILLDATE;
     C_DRUGSPL=DRUGSPL;
     DATEDIFF = FILLDATE - FILLDATE_OLD;
     CONSEC3_GRP= 1;
  END;
  ELSE DO;
       DATEDIFF = FILLDATE - FILLDATE_OLD;
       IF 20<=DATEDIFF <=40 THEN do;
                C_DRUGSPL=C_DRUGSPL+DRUGSPL;
* Initialize NDGSPL and increment CONSEC3_GRP when 90 days(3 consec. 30-day)are reached;
          IF C_DRUGSPL>90 THEN DO;
             CONSEC3_GRP=CONSEC3_GRP+1;
             C_DRUGSPL=DRUGSPL;
          END;
       END;
       ELSE DO;  **If not within 20 and 40 days, initialize NDGSPL and increment CONSEC3_GRP**;
            CONSEC3_GRP=CONSEC3_GRP+1;
            C_DRUGSPL=DRUGSPL;
       END;
  END;
  OUTPUT;
  FILLDATE_OLD = FILLDATE;
RUN;

proc sort data = thirtydays2;
  by PIDBDG MMDC CONSEC3_GRP descending C_DRUGSPL;
run;

*******************************************************************************************
For any CONSEC3_GRP, if the cumulative drug supply is 90 then all 3 claims are flagged as
rf_30dc="1".
*******************************************************************************************;

data thirtydays3(keep=rec_id rf_30dc CONSEC3_SEQ filldate);
set thirtydays2;
by PIDBDG MMDC CONSEC3_GRP descending C_DRUGSPL;
length rf_30dc $1 consec3_SEQ 3;
retain rf_30dc CONSEC3_SEQ ;

if first.MMDC then consec3_SEQ=0;
if first.CONSEC3_GRP and C_DRUGSPL=90 then do;
```

```
   rf_30dc="1";
   CONSEC3_SEQ=CONSEC3_SEQ+1;
end;

if first.CONSEC3_GRP and C_DRUGSPL ne 90 then do;
    rf_30dc="0";
   CONSEC3_SEQ=0;
end;
run;

/******** Create a format rf_30dc ********/

proc print data =thirtydays3(obs=10);
  where consec3_seq>1;
  format FILLDATE mmddyy10.;
run;

data rf_30dc1;
  length start $8 label $1;
  set thirtydays3(rename=(rec_id=start rf_30dc=label)) end=eof;
  where label="1";
  fmtname="rf30dc";
  type="c";
  output;
  if eof then do;
     fmtname="rf30dc";
     start="other";
     label="0";
     type="c";
     output;
  end;
run;

proc format cntlin=rf_30dc1; run;

data rf_30dc2;
  length start $8 label $1;
  set thirtydays3(rename=(rec_id=start) ) end=eof;
  where rf_30dc="1";
  label=put(CONSEC3_SEQ,1.);
  fmtname="rf30SEQ";
  type="c";
  output;
  if eof then do;
     fmtname="rf30SEQ";
     start="other";
     label="0";
     type="c";
     output;
  end;
run;

proc format cntlin=rf_30dc2; run;

data ot2.anmorx06;
  set anmorx06_1;
  length RF_30DC RF_90D CONSEC3_SEQ $1;
  RF_30DC=put(rec_id,$rf30dc.);
  CONSEC3_SEQ=put(rec_id,$rf30seq.);
  if DRUGSPL=90 then RF_90D="1"; else RF_90D="0";
  Label  RF_30DC="Flag for claims with 3 consecutive 30-day supply"
         CONSEC3_SEQ="Sequence # of 3 consecutive 30-day supply"
         RF_90D = "Flag for claims with 90-day supply";
  drop rec_id CONSEC3_SEQ;
Run;

PROC FREQ DATA=OT1.MCDBRX06;
  WHERE RF_ANAL="1";
  TABLE FILLYR*RF_FINMS*RF_DBC*RF_10K*RF_NDC*RF_PATID*RF_YR65*RF_PAYMD*RF_MD /LIST MISSING;
  TITLE "CROSS FREQ OF VARIABLES WHEN RF_ANAL=1-Regular Analysis file";
RUN;
```

```
PROC FREQ DATA=OT1.MCDBRX06;
  WHERE RF_MOANAL="1";
  TABLE FILLYR*RF_FINMS*RF_DBC*RF_10K*RF_NDC*RF_PATID*RF_YR65*RF_PAYMD*RF_MD /LIST MISSING;
  TITLE "CROSS FREQ OF VARIABLES WHEN RF_MOANAL=1-Mail Order Analysis";
RUN;

PROC FREQ DATA=OT1.MCDBRX06;
  TABLE RF_ANAL  RF_COV17 RAGE     RAGE1206 R_NAGE1206 RF_ELINK RF_PLINK RCOVTYPE * RF_COV17
              FILLYR   RF_YR65  RF_FINMS RF_DBC   RF_NDUP    RF_10K   RF_NDC   RF_PATID
              RF_PAYMD RF_MD / LIST MISSING;
  FORMAT RAGE RAGE1206 R_NAGE1206 AGECAT. RF_ELINK $ELINK.  RF_PLINK $PLINK.   RCOVTYPE $CT.
               FILLYR $YEAR.         RF_YR65 $RF_65YR. RF_FINMS $F_FINMS. RF_DBC $DBCFMT.
               RF_NDUP $NDUPFMT.     RF_10K $RF_10K.   RF_NDC $F_NDC.     RF_PATID $F_PATID.
               RF_PAYMD $F_PAYM.     RF_MD $F_MD.;
  TITLE "OTHER KEY FREQ DISTRIBUTION IN MCDBRX06";
RUN;

PROC FREQ DATA=OT2.ANALRX06;
  TABLE GBO RF_NDRUG RF_MORDER RF_COV17 RF_NDRUG * GBO / LIST MISSING;
  FORMAT GBO $GBO. RF_NDRUG $NDRUG. RF_MORDER $M_ORDER. RF_COV17 $RF_COV.;
  TITLE "FREQ DISTRIBUTION OF GENERIC / BRAND INDICATOR (GBO), NEW DRUG FLAG (RF_NDRUG),
         MAIL ORDER FLAG (RF_MORDER), MEDICARE COVERAGE FLAG (RF_COV17) IN ANALRX06";
RUN;

PROC FREQ DATA=OT2.ANMORX06;
  TABLE GBO RF_NDRUG RF_MORDER RF_COV17 RF_NDRUG * GBO RF_90D RF_30DC RF_90D * RF_30DC/ LIST
MISSING;
  FORMAT GBO $GBO. RF_NDRUG $NDRUG. RF_MORDER $M_ORDER. RF_COV17 $RF_COV. RF_90D $NSC. RF_30DC
$TSC.;
  TITLE "FREQ DISTRIBUTION OF GENERIC / BRAND INDICATOR (GBO), NEW DRUG FLAG (RF_NDRUG),
         MAIL ORDER FLAG (RF_MORDER), MEDICARE COVERAGE FLAG (RF_COV17) IN ANMORX06";
RUN;

PROC CONTENTS DATA=OT1.MCDBRX06;
  TITLE "CONTENTS OF MCDBRX06";
RUN;

PROC CONTENTS DATA=OT2.ANALRX06;
  TITLE "CONTENTS OF ANALRX06-MAIL ORDER ANALYSIS";
RUN;


PROC CONTENTS DATA=OT2.ANMORX06;
  TITLE "CONTENTS OF ANMORX06-MAIL ORDER ANALYSIS";
RUN;


%JOBEND;

dm "log;    print file=D:\MCDB\Pharmacy\log_out\MCDB_RX_06.log replace";
dm "output; print file=D:\MCDB\Pharmacy\log_out\MCDB_RX_06.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 (nonHMO)

Pharmacy Layout:

Reported record count –1,348,142 Actual – 1,348,142.

	Field Name	MHCC COMMENTS
1.	Patient ID (encrypted)	
2.	Patient Sex	
3.	Patient Zip Code	
4.	Patient Date of Birth (CCYYMM00)	
5.	NCPDP Number	
6.	Pharmacy Zip	
7.	Practitioner DEA	Is missing for 8,949 records (0.67%).
8.	NDC Code	
9.	Drug Compound	
10.	Drug Quantity	
11.	Drug Supply	
12.	Date Filled (CCYYMMDD)	
13.	Patient Liability	
14.	Billed Charge	
15.	Reimbursement Amount	

Pharmacy LINK to Encounter Data by PATIENT ID-BIRTHDATE-SEX	
Prescription **Recipients**	83.5%
Prescription **Services**	83.2%

Pharmacy LINK to Provider Data	
No Link	9.2%
Link by Deanum	90.8%
Link by Fedtaxid	0%
Link by Provid	0%

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2006 MCDB

(Data received June 27, 2007)

P030 - AETNA U.S. HEALTHCARE

Pharmacy Layout:

Reported record count –1,667,009 Actual – 1,667,009

	Field Name	MHCC COMMENTS
1.	Patient ID (encrypted)	
2.	Patient Sex	
3.	Patient Zip Code	
4.	Patient Date of Birth (CCYYMM00)	
5.	NCPDP Number	
6.	Pharmacy Zip	
7.	Practitioner DEA	Is missing for 12,022 records (0.72%).
8.	NDC Code	
9.	Drug Compound	
10.	Drug Quantity	
11.	Drug Supply	
12.	Date Filled (CCYYMMDD)	
13.	Patient Liability	
14.	Billed Charge	
15.	Reimbursement Amount	

Pharmacy LINK to Encounter Data by PATIENT ID-BIRTHDATE-SEX	
Prescription **Recipients**	94,3%
Prescription **Services**	98.3%

Pharmacy LINK to Provider Data	
No Link	9.9%
Link by Deanum	90.1%
Link by Fedtaxid	0%
Link by Provid	0%

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2006 MCDB

(Data received June 27, 2007)

P070 - American Republic

Pharmacy Layout:

Reported record count – 8,002 Actual – 8,002

	Field Name	MHCC COMMENTS
1.	Patient ID (encrypted)	
2.	Patient Sex	
3.	Patient Zip Code	
4.	Patient Date of Birth (CCYYMM00)	
5.	NCPDP Number	
6.	Pharmacy Zip	Is missing for 356 records (4.4%).
7.	Practitioner DEA	Is invalid for 4.9% of records (Coded as 'U')
8.	NDC Code	
9.	Drug Compound	
10.	Drug Quantity	
11.	Drug Supply	
12.	Date Filled (CCYYMMDD)	
13.	Patient Liability	
14.	Billed Charge	
15.	Reimbursement Amount	

Pharmacy LINK to Encounter Data by PATIENT ID-BIRTHDATE-SEX	
Prescription **Recipients**	45.3%
Prescription **Services**	55.8%

Pharmacy LINK to Provider Data	
No Link	0%
Link by Deanum	100%
Link by Fedtaxid	0%
Link by Provid	0%

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2006 MCDB

(Data received June 27, 2007)

P130 Carefirst - DC

Pharmacy Layout:

Reported record count – 6,212,172 Actual – 6,212,172

	Field Name	MHCC COMMENTS
1.	Patient ID (encrypted)	
2.	Patient Sex	
3.	Patient Zip Code	
4.	Patient Date of Birth (CCYYMM00)	
5.	NCPDP Number	
6.	Pharmacy Zip	Is invalid for 579,832 records (9.34%).
7.	Practitioner DEA	
8.	NDC Code	
9.	Drug Compound	
10.	Drug Quantity	
11.	Drug Supply	
12.	Date Filled (CCYYMMDD)	41,413 records are duplicates by Patient ID-DOB-NDC-Fill Date.
13.	Patient Liability	
14.	Billed Charge	
15.	Reimbursement Amount	

Pharmacy LINK to Encounter Data by PATIENT ID-BIRTHDATE-SEX	
Prescription **Recipients**	91.8%
Prescription **Services**	96.6%

Pharmacy LINK to Provider Data	
No Link	10.8%
Link by Deanum	89.2%
Link by Fedtaxid	0%
Link by Provid	0%

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2006 MCDB

(Data received June 27, 2007)

P131 Carefirst MD

Pharmacy Layout:

Reported record count – 2,495,115 Actual – 2,495,115

	Field Name	MHCC COMMENTS
1.	Patient ID (encrypted)	
2.	Patient Sex	
3.	Patient Zip Code	
4.	Patient Date of Birth (CCYYMM00)	
5.	NCPDP Number	
6.	Pharmacy Zip	Is invalid for 215,787 (8.65%).
7.	Practitioner DEA	
8.	NDC Code	
9.	Drug Compound	
10.	Drug Quantity	
11.	Drug Supply	
12.	Date Filled (CCYYMMDD)	14,288 records are duplicates by Patient ID-DOB-NDC-Fill Date.
13.	Patient Liability	
14.	Billed Charge	
15.	Reimbursement Amount	

Pharmacy LINK to Encounter Data by PATIENT ID-BIRTHDATE-SEX	
Prescription **Recipients**	85.5%
Prescription **Services**	92.7%

Pharmacy LINK to Provider Data	
No Link	10.5%
Link by Deanum	89.5%
Link by Fedtaxid	0%
Link by Provid	0%

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2006 MCDB

(Data received July 25, 2007)

P160 CIGNA

Pharmacy Layout:

Reported record count – 1,254,002 Actual – 1,254,002.

	Field Name	MHCC COMMENTS
1.	Patient ID (encrypted)	
2.	Patient Sex	
3.	Patient Zip Code	
4.	Patient Date of Birth (CCYYMM00)	
5.	NCPDP Number	
6.	Pharmacy Zip	
7.	Practitioner DEA	
8.	NDC Code	
9.	Drug Compound	
10.	Drug Quantity	
11.	Drug Supply	
12.	Date Filled (CCYYMMDD)	
13.	Patient Liability	
14.	Billed Charge	
15.	Reimbursement Amount	

Pharmacy LINK to Encounter Data by PATIENT ID-BIRTHDATE-SEX	
Prescription **Recipients**	91.0%
Prescription **Services**	96.7%

Pharmacy LINK to Provider Data	
No Link	9.5%
Link by Deanum	90.5%
Link by Fedtaxid	0%
Link by Provid	0%

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2006 MCDB

(Data received July 3, 2007)

P280 Fortis

Pharmacy Layout:

Reported record count – 17,002 Actual – 17,002.

	Field Name	MHCC COMMENTS
1.	Patient ID (encrypted)	
2.	Patient Sex	
3.	Patient Zip Code	
4.	Patient Date of Birth (CCYYMM00)	
5.	NCPDP Number	
6.	Pharmacy Zip	
7.	Practitioner DEA	Is missing for 5,268 records (34%).
8.	NDC Code	
9.	Drug Compound	
10.	Drug Quantity	
11.	Drug Supply	Is zero for 5236 records (34%).
12.	Date Filled (CCYYMMDD)	
13.	Patient Liability	
14.	Billed Charge	
15.	Reimbursement Amount	

Pharmacy LINK to Encounter Data by PATIENT ID-BIRTHDATE-SEX	
Prescription **Recipients**	78.5%
Prescription **Services**	90.1%

Pharmacy LINK to Provider Data	
No Link	41.7%
Link by Deanum	58.3%
Link by Fedtaxid	0%
Link by Provid	0%

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2006 MCDB

(Data received June 28, 2007)

P320 Golden Rule

Pharmacy Layout:

Reported record count – 61,915 Actual – 61,915.

	Field Name	MHCC COMMENTS
1.	Patient ID (encrypted)	
2.	Patient Sex	
3.	Patient Zip Code	
4.	Patient Date of Birth (CCYYMM00)	
5.	NCPDP Number	
6.	Pharmacy Zip	
7.	Practitioner DEA	
8.	NDC Code	
9.	Drug Compound	
10.	Drug Quantity	
11.	Drug Supply	
12.	Date Filled (CCYYMMDD)	
13.	Patient Liability	
14.	Billed Charge	
15.	Reimbursement Amount	

Pharmacy LINK to Encounter Data by PATIENT ID-BIRTHDATE-SEX	
Prescription **Recipients**	75.6%
Prescription **Services**	86.5%

Pharmacy LINK to Provider Data	
No Link	11.7%
Link by Deanum	88.3%
Link by Fedtaxid	0%
Link by Provid	0%

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2006 MCDB

(Last Submission Data Received August 15, 2007)

P325 Graphic Arts

Pharmacy Layout:

Reported record count – 41,833 Actual – 41,833.

	Field Name	MHCC COMMENTS
1.	Patient ID (encrypted)	
2.	Patient Sex	
3.	Patient Zip Code	
4.	Patient Date of Birth (CCYYMM00)	Day of birth should be 00 for all records.
5.	NCPDP Number	
6.	Pharmacy Zip	
7.	Practitioner DEA	Is missing for all records.
8.	NDC Code	
9.	Drug Compound	
10.	Drug Quantity	
11.	Drug Supply	
12.	Date Filled (CCYYMMDD)	
13.	Patient Liability	Average liability=$4. Corrected with resubmission.
14.	Billed Charge	
15.	Reimbursement Amount	

Pharmacy LINK to Encounter Data by PATIENT ID-BIRTHDATE-SEX	
Prescription **Recipients**	63.5%
Prescription **Services**	73.1%

Pharmacy LINK to Provider Data	
No Link	100%
Link by Deanum	0%
Link by Fedtaxid	0%
Link by Provid	0%

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2006 MCDB

(Data received June 28, 2007)

P330 Great West

Pharmacy Layout:

Reported record count – 191,125 Actual – 191,125.

	Field Name	MHCC COMMENTS
1.	Patient ID (encrypted)	
2.	Patient Sex	
3.	Patient Zip Code	
4.	Patient Date of Birth (CCYYMM00)	Day of birth should be 00 for all records.
5.	NCPDP Number	
6.	Pharmacy Zip	
7.	Practitioner DEA	
8.	NDC Code	
9.	Drug Compound	Is zero for 18,015 records (9.43%).
10.	Drug Quantity	
11.	Drug Supply	
12.	Date Filled (CCYYMMDD)	
13.	Patient Liability	
14.	Billed Charge	
15.	Reimbursement Amount	

Pharmacy LINK to Encounter Data by PATIENT ID-BIRTHDATE-SEX	
Prescription **Recipients**	52.8%
Prescription **Services**	72.0%

Pharmacy LINK to Provider Data	
No Link	8.8%
Link by Deanum	91.2%
Link by Fedtaxid	0%
Link by Provid	0%

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2006 MCDB

(Last Submission Data Received August 31, 2007)

P350 Guardian Life

Pharmacy Layout:

Reported record count – 192,256 Actual – 192,256.

	Field Name	MHCC COMMENTS
1.	Patient ID (encrypted)	
2.	Patient Sex	
3.	Patient Zip Code	
4.	Patient Date of Birth (CCYYMM00)	
5.	NCPDP Number	
6.	Pharmacy Zip	
7.	Practitioner DEA	
8.	NDC Code	
9.	Drug Compound	Is zero for 6,832 records (3.63%).
10.	Drug Quantity	
11.	Drug Supply	
12.	Date Filled (CCYYMMDD)	
13.	Patient Liability	
14.	Billed Charge	
15.	Reimbursement Amount	

Pharmacy LINK to Encounter Data by PATIENT ID-BIRTHDATE-SEX	
Prescription **Recipients**	87.0%
Prescription **Services**	95.2%

Pharmacy LINK to Provider Data	
No Link	16.5%
Link by Deanum	83.5%
Link by Fedtaxid	0%
Link by Provid	0%

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2006 MCDB

(Data received June 6, 2007)

P471 Unicare Life

Pharmacy Layout:

Current Reported record count – 128,056 Actual – 128,056.

	Field Name	MHCC COMMENTS
1.	Patient ID (encrypted)	
2.	Patient Sex	Is missing for 53 records.
3.	Patient Zip Code	
4.	Patient Date of Birth (CCYYMM00)	Day of birth should be 00 for all records.
5.	NCPDP Number	
6.	Pharmacy Zip	
7.	Practitioner DEA	
8.	NDC Code	
9.	Drug Compound	Is Zero for 18,154 records (14.18%)
10.	Drug Quantity	
11.	Drug Supply	
12.	Date Filled (CCYYMMDD)	
13.	Patient Liability	
14.	Billed Charge	
15.	Reimbursement Amount	

Pharmacy LINK to Encounter Data by PATIENT ID-BIRTHDATE-SEX	
Prescription **Recipients**	82.1%
Prescription **Services**	91.0%

Pharmacy LINK to Provider Data	
No Link	13.5%
Link by Deanum	86.5%
Link by Fedtaxid	0%
Link by Provid	0%

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2006 MCDB

(Data received June 27, 2007)

P480 Kaiser

Pharmacy Layout:

Current Reported record count – 2,741,080 Actual – 2,741,080.

	Field Name	MHCC COMMENTS
1.	Patient ID (encrypted)	
2.	Patient Sex	Is 'not-coded' for 14,430 records.
3.	Patient Zip Code	
4.	Patient Date of Birth (CCYYMM00)	
5.	NCPDP Number	
6.	Pharmacy Zip	
7.	Practitioner DEA	
8.	NDC Code	
9.	Drug Compound	
10.	Drug Quantity	
11.	Drug Supply	
12.	Date Filled (CCYYMMDD)	
13.	Patient Liability	
14.	Billed Charge	
15.	Reimbursement Amount	

Pharmacy LINK to Encounter Data by PATIENT ID-BIRTHDATE-SEX	
Prescription **Recipients**	97.5%
Prescription **Services**	99.9%

Pharmacy LINK to Provider Data	
No Link	0.04%
Link by Deanum	
Link by Fedtaxid	
Link by Provid	99.96%

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2006 MCDB

(Data received June 21, 2007)

P500 Mamsi

Pharmacy Layout:

Reported record count – 793,031 Actual – 793,031.

	Field Name	MHCC COMMENTS
1.	Patient ID (encrypted)	
2.	Patient Sex	
3.	Patient Zip Code	
4.	Patient Date of Birth (CCYYMM00)	
5.	NCPDP Number	
6.	Pharmacy Zip	Is missing for 24,047 records (3.03%).
7.	Practitioner DEA	
8.	NDC Code	
9.	Drug Compound	
10.	Drug Quantity	
11.	Drug Supply	
12.	Date Filled (CCYYMMDD)	
13.	Patient Liability	
14.	Billed Charge	
15.	Reimbursement Amount	

Pharmacy LINK to Encounter Data by PATIENT ID-BIRTHDATE-SEX	
Prescription **Recipients**	88.7%
Prescription **Services**	95.6%

Pharmacy LINK to Provider Data	
No Link	11.0%
Link by Deanum	89.0%
Link by Fedtaxid	0%
Link by Provid	0%

MARYLAND HEALTH CARE COMMISSION
Quality Review Statement for 2006 MCDB
(Data received June 26, 2007)

P510 Maryland Fidelity

Pharmacy Layout:
Current Reported record count – 281 Actual – 281.

	Field Name	MHCC COMMENTS
1.	Patient ID (encrypted)	
2.	Patient Sex	
3.	Patient Zip Code	
4.	Patient Date of Birth (CCYYMM00)	
5.	NCPDP Number	
6.	Pharmacy Zip	
7.	Practitioner DEA	
8.	NDC Code	
9.	Drug Compound	
10.	Drug Quantity	
11.	Drug Supply	
12.	Date Filled (CCYYMMDD)	
13.	Patient Liability	
14.	Billed Charge	
15.	Reimbursement Amount	

Pharmacy LINK to Encounter Data by PATIENT ID-BIRTHDATE-SEX	
Prescription **Recipients**	71.7%
Prescription **Services**	56.9%

Pharmacy LINK to Provider Data	
No Link	9.6%
Link by Deanum	90.4%
Link by Fedtaxid	0%
Link by Provid	0%

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2006 MCDB

(Data received June 21, 2007)

P520 MD Indiv Practice (Mamsi)

Pharmacy Layout:

Reported record count – 1,462,989 Actual – 1,462,989.

	Field Name	MHCC COMMENTS
1.	Patient ID (encrypted)	
2.	Patient Sex	
3.	Patient Zip Code	
4.	Patient Date of Birth (CCYYMM00)	
5.	NCPDP Number	
6.	Pharmacy Zip	Is missing for 30,515 records (2%).
7.	Practitioner DEA	
8.	NDC Code	
9.	Drug Compound	
10.	Drug Quantity	
11.	Drug Supply	
12.	Date Filled (CCYYMMDD)	
13.	Patient Liability	
14.	Billed Charge	
15.	Reimbursement Amount	

Pharmacy LINK to Encounter Data by PATIENT ID-BIRTHDATE-SEX	
Prescription **Recipients**	91.6%
Prescription **Services**	97.5%

Pharmacy LINK to Provider Data	
No Link	7.8%
Link by Deanum	92.2%
Link by Fedtaxid	0%
Link by Provid	0%

MARYLAND HEALTH CARE COMMISSION
Quality Review Statement for 2006 MCDB
(Data received June 20, 2007)

P530 Mega Life

Pharmacy Layout:
Reported record count – 82,342 Actual – 82,342.

	Field Name	MHCC COMMENTS
1.	Patient ID (encrypted)	
2.	Patient Sex	
3.	Patient Zip Code	
4.	Patient Date of Birth (CCYYMM00)	
5.	NCPDP Number	
6.	Pharmacy Zip	
7.	Practitioner DEA	
8.	NDC Code	
9.	Drug Compound	
10.	Drug Quantity	
11.	Drug Supply	
12.	Date Filled (CCYYMMDD)	
13.	Patient Liability	
14.	Billed Charge	
15.	Reimbursement Amount	

Pharmacy LINK to Encounter Data by PATIENT ID-BIRTHDATE-SEX	
Prescription **Recipients**	78.6%
Prescription **Services**	87.8%

Pharmacy LINK to Provider Data	
No Link	12.2%
Link by Deanum	87.8%
Link by Fedtaxid	0%
Link by Provid	0%

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2006 MCDB

(Data received June 22, 2007)

P620 Optimum Choice (Mamsi)

Pharmacy Layout:

Reported record count – 2,382,713 Actual – 2,382,713.

	Field Name	MHCC COMMENTS
1.	Patient ID (encrypted)	
2.	Patient Sex	
3.	Patient Zip Code	
4.	Patient Date of Birth (CCYYMM00)	
5.	NCPDP Number	
6.	Pharmacy Zip	Is missing for 77,679 records (3.26%).
7.	Practitioner DEA	
8.	NDC Code	
9.	Drug Compound	
10.	Drug Quantity	
11.	Drug Supply	
12.	Date Filled (CCYYMMDD)	
13.	Patient Liability	
14.	Billed Charge	
15.	Reimbursement Amount	

Pharmacy LINK to Encounter Data by PATIENT ID-BIRTHDATE-SEX	
Prescription **Recipients**	87.0%
Prescription **Services**	95.5%

Pharmacy LINK to Provider Data	
No Link	9.2%
Link by Deanum	90.8%
Link by Fedtaxid	0%
Link by Provid	0%

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2005 MCDB

(Last Submission Data Received August 14, 2006)

P680 Coventry Healthcare of Delaware

Pharmacy Layout:

Reported record count – 782,930 Actual – 782,930.

	Field Name	MHCC COMMENTS
1.	Patient ID (encrypted)	
2.	Patient Sex	
3.	Patient Zip Code	
4.	Patient Date of Birth (CCYYMM00)	
5.	NCPDP Number	
6.	Pharmacy Zip	
7.	Practitioner DEA	Is missing for all records. Corrected with data resubmission.
8.	NDC Code	
9.	Drug Compound	
10.	Drug Quantity	
11.	Drug Supply	
12.	Date Filled (CCYYMMDD)	
13.	Patient Liability	
14.	Billed Charge	
15.	Reimbursement Amount	

Pharmacy LINK to Encounter Data by PATIENT ID-BIRTHDATE-SEX	
Prescription **Recipients**	100%
Prescription **Services**	99.8%

Pharmacy LINK to Provider Data	
No Link	9.1%
Link by Deanum	90.9%
Link by Fedtaxid	0%
Link by Provid	0%

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2006 MCDB

(Last Submission Data Received August 27, 2007)

P760 State Farm

Pharmacy Layout:

Reported record count – 50,195 Actual – 50,195.

	Field Name	MHCC COMMENTS
1.	Patient ID (encrypted)	
2.	Patient Sex	Is 'not-coded' for 927 records (2.4%).
3.	Patient Zip Code	
4.	Patient Date of Birth (CCYYMM00)	
5.	NCPDP Number	
6.	Pharmacy Zip	
7.	Practitioner DEA	
8.	NDC Code	
9.	Drug Compound	
10.	Drug Quantity	
11.	Drug Supply	
12.	Date Filled (CCYYMMDD)	No prescription filled in 2007. Corrected with data resubmission.
13.	Patient Liability	
14.	Billed Charge	
15.	Reimbursement Amount	

Pharmacy LINK to Encounter Data by PATIENT ID-BIRTHDATE-SEX	
Prescription **Recipients**	22.4%
Prescription **Services**	32.7%

Pharmacy LINK to Provider Data	
No Link	10.11%
Link by Deanum	89.89%
Link by Fedtaxid	0%
Link by Provid	0%

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2006 MCDB

(Data received June 22, 2007)

P820 United Health Care Ins.

Pharmacy Layout:

Reported record count – 1,037,812 Actual – 1,037,812.

	Field Name	MHCC COMMENTS
1.	Patient ID (encrypted)	
2.	Patient Sex	
3.	Patient Zip Code	
4.	Patient Date of Birth (CCYYMM00)	
5.	NCPDP Number	
6.	Pharmacy Zip	Is missing for all records.
7.	Practitioner DEA	
8.	NDC Code	
9.	Drug Compound	
10.	Drug Quantity	
11.	Drug Supply	
12.	Date Filled (CCYYMMDD)	
13.	Patient Liability	
14.	Billed Charge	
15.	Reimbursement Amount	

Pharmacy LINK to Encounter Data by PATIENT ID-BIRTHDATE-SEX	
Prescription **Recipients**	88.6%
Prescription **Services**	94.1%

Pharmacy LINK to Provider Data	
No Link	11.2%
Link by Deanum	88.8%
Link by Fedtaxid	0%
Link by Provid	0%

MARYLAND HEALTH CARE COMMISSION
Quality Review Statement for 2006 MCDB
(Data received July 2, 2007)

P830 Trustmark

Pharmacy Layout:
Reported record count – 957 Actual – 957.

	Field Name	MHCC COMMENTS
1.	Patient ID (encrypted)	Does not link to the Encounter Patient ID.
2.	Patient Sex	
3.	Patient Zip Code	
4.	Patient Date of Birth (CCYYMM00)	
5.	NCPDP Number	Is missing for 426 records (45%).
6.	Pharmacy Zip	Is missing for 480 records (50%).
7.	Practitioner DEA	Is missing for 428 records (45%).
8.	NDC Code	
9.	Drug Compound	Is 0 for 493 records (51%).
10.	Drug Quantity	Is 0 for 428 records (45%).
11.	Drug Supply	Is 0 for 428 records (44%).
12.	Date Filled (CCYYMMDD)	
13.	Patient Liability	
14.	Billed Charge	
15.	Reimbursement Amount	

Pharmacy LINK to Encounter Data by PATIENT ID-BIRTHDATE-SEX	
Prescription **Recipients**	1.4%
Prescription **Services**	0.4%

Pharmacy LINK to Provider Data	
No Link	48.3%
Link by Deanum	51.7%
Link by Fedtaxid	0%
Link by Provid	0%

MARYLAND HEALTH CARE COMMISSION
Quality Review Statement for 2006 MCDB
(Last Submission Data Received August 13, 2007)

P850 Union Labor

Pharmacy Layout:
Reported record count – 56,068 Actual – 56,068.

	Field Name	MHCC COMMENTS
1.	Patient ID (encrypted)	
2.	Patient Sex	
3.	Patient Zip Code	
4.	Patient Date of Birth (CCYYMM00)	
5.	NCPDP Number	
6.	Pharmacy Zip	
7.	Practitioner DEA	
8.	NDC Code	
9.	Drug Compound	
10.	Drug Quantity	
11.	Drug Supply	
12.	Date Filled (CCYYMMDD)	
13.	Patient Liability	
14.	Billed Charge	Is missing for 6,648 records. Corrected with data resubmission.
15.	Reimbursement Amount	Average reimbursement=$2,816,405. Corrected with data resubmission.

Pharmacy LINK to Encounter Data by PATIENT ID-BIRTHDATE-SEX	
Prescription **Recipients**	62.0%
Prescription **Services**	83.2%

Pharmacy LINK to Provider Data	
No Link	10.1%
Link by Deanum	89.9%
Link by Fedtaxid	0%
Link by Provid	0%

MARYLAND HEALTH CARE COMMISSION

Quality Review Statement for 2006 MCDB

(Data Received June 22, 2007)

P870 United Healthcare Mid Atlantic

Pharmacy Layout:

Reported record count – 70,094 Actual – 70,094.

	Field Name	MHCC COMMENTS
1.	Patient ID (encrypted)	
2.	Patient Sex	
3.	Patient Zip Code	
4.	Patient Date of Birth (CCYYMM00)	
5.	NCPDP Number	
6.	Pharmacy Zip	Is missing for all records.
7.	Practitioner DEA	
8.	NDC Code	
9.	Drug Compound	
10.	Drug Quantity	
11.	Drug Supply	
12.	Date Filled (CCYYMMDD)	
13.	Patient Liability	
14.	Billed Charge	
15.	Reimbursement Amount	

Pharmacy LINK to Encounter Data by PATIENT ID-BIRTHDATE-SEX	
Prescription **Recipients**	91.3%
Prescription **Services**	96.5%

Pharmacy LINK to Provider Data	
No Link	10.6%
Link by Deanum	89.4%
Link by Fedtaxid	0%
Link by Provid	0%

Appendix D

Codebook

2006 Maryland Medical Care Database
MCDB Pharmacy file
MCDBRX06.SAS7BDAT

Contents Output

Data Set Name	IN1.MCDBRX06	**Observations**	23077121
Member Type	DATA	**Variables**	67
Engine	V9	**Indexes**	0
Created	Thursday, March 27, 2008 05:15:41 PM	**Observation Length**	262
Last Modified	Thursday, March 27, 2008 05:15:41 PM	**Deleted Observations**	0
Protection		**Compressed**	BINARY
Data Set Type		**Reuse Space**	NO
Label		**Point to Observations**	YES
Data Representation	WINDOWS_32	**Sorted**	NO
Encoding	wlatin1 Western (Windows)		

Engine/Host Dependent Information	
Data Set Page Size	12288
Number of Data Set Pages	410708
Number of Data Set Repairs	0
File Name	d:\MCDB\Pharmacy\Data\mcdbrx06.sas7bdat
Release Created	9.0101M3
Host Created	XP_PRO

Alphabetic List of Variables and Attributes

#	Variable	Type	Len	Format	Informat	Label
31	APPRDATE	Num	8			FDA Approval Date
55	CHNIND	Char	1			Chain Indicator
24	DEANUM	Char	11			Practitioner DEA#
39	DRUGQTY	Num	4			Drug Quantity (No. of Units Dispensed)
40	DRUGSPL	Num	3			Drug Supply (Est. # of Supply Days)

26	DRUG_ID	Char	6	$6.	$6.	Multum Drug ID
47	FILLDATE	Num	8			Date Prescription Filled
9	FILLDD	Char	2			Day Prescription Filled
10	FILLMO	Char	2			Month Prescription Filled
11	FILLYR	Char	4			Year Prescription Filled
25	GBO	Char	1	$1.	$1.	Generic/Brand Indicator
35	MMDC	Char	5			Main Multum Drug Code
27	MULTDC1	Char	3			Multum Drug Category 1
28	MULTDC2	Char	3			Multum Drug Category 2
29	MULTDC3	Char	3			Multum Drug Category 3
30	MULTDC4	Char	3			Multum Drug Category 4
12	NCPDP	Char	7			Pharmacy Number (NCPDP)
13	NDC	Char	12			National Drug Code (NDC-11 Chars)
33	NDCLABL	Char	9			Labeler-Product Code(Chars 1-9 of NDC)
34	NDCPACK	Char	2			Package Code(Chars 10-11 of NDC)
14	PATID	Char	14			Patient ID, Encrypted
8	PIDBDG	Char	25			Unique Patient ID
41	PNUM	Char	4			Payer Unique ID
5	RAGE	Num	3			Patient Age at Service
6	RAGE1206	Num	3			Patient Age as of 12/31/2006
36	RBILL	Num	8			Billed Charge-BC ($)
16	RBIRTHMO	Char	2			Patient Birth Month
17	RBIRTHYR	Char	4			Patient Birth Year
1	RCOVTYPE	Char	1			Coverage Type
2	RDELIVTYP	Char	1			Delivery System Type
3	RDENRDATE	Num	8	MMDDYY8.		Disenrollment Date
4	RENRDATE	Num	8	MMDDYY8.		Enrollment Date
48	RF_10K	Char	1			Flag for Payment > 10K

63	RF_ANAL	Char	1			Analysis Flag(1-Analysis, 0-Exclude)
51	RF_COV17	Char	1			Medicare Coverage Flag
7	RF_DBC	Char	1			Debit/Credit Flag
66	RF_DEA	Char	1			DEA Number(1=Valid, 0=Other, 2=Blank)
62	RF_ELINK	Char	1			Plarmacy/Encounter Link Flag
21	RF_FINMS	Char	1			All 3 Financial Vars=0/Missing (1/0)
50	RF_MD	Char	1			Maryland Residency Indicator (1-MD, 0-non MD)
32	RF_MEDSUP	Char	1			Medical Supply Indicator(0=No, 1=Yes)
64	RF_MOANAL	Char	1			Mail Order Analysis Flag(1-Analysis, 0-Exclude)
45	RF_MORDER	Char	1			Mail Order Flag(0=No, 1=Yes)
42	RF_NDC	Char	1			NDC Flag
44	RF_NDRUG	Char	1			New Drug Flag
61	RF_NDUP	Char	1			Duplicate Set W/ at least One Neg. Bill
49	RF_PATID	Char	1			Patient ID Flag
22	RF_PAYMD	Char	1			Payment Category (1=<0,2=0,3=>0,4=.)
57	RF_PLINK	Char	1			Pharmacy/Provider Link Flag
53	RF_SELF	Char	1			Self Insurance Coverage Indicator
20	RF_YR07	Char	1			Prescription Filled in 2007(1:>=07 0:<07)
65	RF_YR65	Char	1			Age 65+ Flag(0:<65, 1:>=65, 2:Missing)
43	RMSCAT	Char	5			Medical Supply Category
54	RN_ZIP_S	Char	5			Adjusted Pharmacy Zip Code (Using NACDS data)
38	RPATLIAB	Num	8			Patient Liability-PL ($)
18	RPATZIP	Char	5			Patient Residence ZIP Code
46	RPAYMENT	Num	8			Payment = Sum(RA,PL)
52	RPHARMCTY	Char	3			Pharmacy County
56	RPHARMGRP	Char	1			Pharmacy Group
37	RREIMB	Num	8			Reimbursement Amount-RA ($)
67	RSCRIPT	Num	3			Number of Adjusted or 30-day Script

19	RSEX	Char	1			Patient Sex
58	RSPECL1	Char	3			Practitioner/Supplier Specialty-1
59	RSPECL2	Char	3			Practitioner/Supplier Specialty-2
60	RSPECL3	Char	3			Practitioner/Supplier Specialty-3
15	RZIP_S	Char	5			Pharmacy ZIP Code
23	R_NAGE1206	Num	3			Patient Age as of 12/31/2006 (Not Rounded)