USPS LR-J-112/R2000-1
Revenue, Piece, and Weight Estimates by Mail Characteristics

USPS LR-J-112

Revenue, Piece, and Weight Estimates by Mail Characteristics

Table of Contents

List of Tables

List of Tables, Continued

I. Introduction

USPS LR-J-112 contains First-Class, Priority, Periodicals, Standard, and Package Services mail volumes and pounds by shape, weight increment, and in one case, indicia. This Category 2 library reference is being sponsored by Paul Loetscher (USPS-T-41). This information is used by various witnesses including Smith (USPS-T-15), Mayes (USPS-T-23), Eggleston (USPS-T-25), Miller (USPS-T-22 and USPS-T-24), Keifer (USPS-T-33), and Schenk (USPS-T-43)

This document and its supporting programs and workbooks were prepared under the direction of Paul Loetscher, a Senior Economist at Christensen Associates, in Madison, WI. Other testimony and library references referred to in this library reference include:

- USPS-T-3, Testimony of witness Pafford and supporting library references
- USPS-T-4, Testimony of witness Hunter and supporting library references
- USPS-ST-50, Testimony by witness Talmo, and library references LR-H-105 and LR-H-195 in Docket No. R97-1

Portions of this library reference update previous studies sponsored by USPS witnesses McGrane (USPS-ST-44) and Talmo (USPS-ST-50) in Docket No. R97-1 and witness Daniel (USPS-T-28) and LR-I-102 in Docket No. R00-1.

This library reference documents the methods used to develop First-Class, Periodicals, Standard Mail, and Package Services piece and weight estimates by various piece characteristics. The characteristics are shape, weight increment, and indicia. While shape is detailed for all rate elements in each class, weight increment and indicia detail are not provided for all rate elements. Indicia detail is required only for First-Class Mail single-piece categories. The report describes how postage statement data on presorted mail reported in the PERMIT bulk mail acceptance system are edited for consistency to rate schedules and mailing practices, and how the data are inflated to account for volumes at non-PERMIT

offices. It also describes how single piece estimates are developed from data in the domestic RPW sample.

Official estimates of revenue, pieces, and weight for First-Class, Periodicals, and Standard Mail are developed by the Revenue, Volume, and Performance Measurement group. The primary data source for those estimates is the CBCIS and the domestic RPW sample. The CBCIS draws input from the PERMIT bulk mail acceptance system. These data sources are also used in this analysis although the methods used here are somewhat different. The methodology here focuses on the distribution of mail activity by weight increment and shape. Such distributions are not computed in the official estimates. Since the input data are the same and the methodologies are similar, there is general consistency between the official estimates and those reported here. As a point of emphasis, the estimates reported here are used only as relative percentage keys to distribute the official estimates across mail characteristics.

II. Bulk Entered Mail - Data Sources and Stratification

Estimates of revenue, pieces and weight for presort First-Class, Priority, Periodicals, Standard Mail, and Parcel Post are based on postage statement data from the PERMIT bulk mail entry system. These estimates will be used to develop shape and weight distribution keys to be applied to the reported RPW estimates. Not all post offices report through the PERMIT system. First Class, Periodicals, and Standard Mail data obtained thought the PERMIT system will be stratified by post office size and inflated to represent all mail. Priority and Package Services PERMIT data will be used, uninflated, to develop the necessary distribution keys. The relative importance of First Class, Periodicals, and Standard Mail suggests the use of the more elaborate estimation procedure described in this section.

The PERMIT system is used to record and verify postage at many bulk mail acceptance locations. The system records revenue, pieces, and weight by

individual rate elements, which are identified by a five digit numeric code. These codes are referred to as Volume Information Profile (VIP) codes. VIP code information is available for each postage statement submitted at PERMIT equipped offices.

As mentioned above, the PERMIT system does not report data for all bulk-entered mail. The non-PERMIT offices are small relative to PERMIT offices in terms of total revenue from bulk-entered mail. An examination of the distribution of mail by rate category entered in PERMIT offices indicates the distribution is related to the size of the office as measured by each class's bulk-entered revenue. Therefore, it is more appropriate to represent non-PERMIT offices by PERMIT offices of similar size. For example, large offices tend to have a greater share of carrier presort mail than smaller offices. Consequently, bulk-entered revenue, pieces, and weight are estimated by office size stratum. Such a general stratification scheme is used for First Class, Periodicals, and Standard Mail. Shape distribution keys for Priority and Package Services are developed from the uninflated PERMIT data.

A specific variable must be defined for each class of mail to rank offices by size. Since all Periodicals revenue and 90 percent of Standard Mail revenue comes from permit imprint mail, permit imprint revenue reported in the Trial Balance is used to rank offices in these classes. These data are available for non-PERMIT offices.

In First-Class, metered and stamped revenue accounts for a much larger share of bulk entered revenue. Furthermore, this revenue is not available for non-PERMIT offices. Metered and stamped revenue at non-PERMIT offices is estimated. (See below.) The size ranking variable in First-Class is the sum of Trial Balance permit imprint presort revenue plus PERMIT system metered and stamped revenue (or estimated metered and stamped revenue for non-PERMIT offices).

After offices are ranked, they are grouped into strata of similar revenue size. There is an independent stratification of offices for First-Class, Periodicals inside county, Periodicals outside county, Standard Mail commercial, and Standard Mail nonprofit. The total revenue in each stratum is computed from the Trial Balance, the PERMIT system, and/or estimation methods. The total revenue includes revenue from both PERMIT and non-PERMIT system offices.

In each class or subclass the strata are examined to determine the extent of coverage of PERMIT system offices relative to all offices. Revenue shares by rate groups are sometimes examined to determine whether strata can be grouped, reducing the total number of strata needed for the inflation steps to follow.

In First-Class and Standard Mail, total revenue is divided into 20 equal revenue strata, each containing offices that account for 5 percent of total revenue. For First-Class, the ranking and stratification process was completed several years ago using FY 94 data. Table 1 shows rate element shares of pieces for the original 20 strata in FY 94. (In FY 94 some offices reported mailing statement data in the BRAVIS system. BRAVIS contained the same information as PERMIT. The BRAVIS system has been discontinued.) Based on a general similarity of presort level shares in FY 94, the 20 strata are collapsed to four final strata. Original strata 1 through 14 comprise new stratum 1, original strata 15 and 16 comprise new stratum 2, original strata 17 and 18 comprise new stratum 3, and original strata 19 and 20 comprise new stratum 4. Table 2 shows the PERMIT system revenue coverage for First-Class permit imprint presort mail by the original 20 strata in FY 00.

Table 3 shows the revenue coverage in Standard Mail commercial rates for 20 original revenue strata. Since there are no non-PERMIT offices in the first 17 strata, they are grouped into a single stratum. Since there are only three

remaining strata, each of strata 18 through 20 are retained individually for the inflation process.

Table 4 shows the PERMIT system revenue coverage for Standard Mail nonprofit rates in the original 20 revenue strata. Strata 1 through 16 contain no non-PERMIT offices and so these are grouped into a single stratum. The PERMIT system revenue coverage in strata 17 and 18 is very high, so these two strata are combined to form a second stratum. Finally, strata 19 and 20 form the last two of the four strata in the inflation process.

Two separate stratification schemes are developed for Periodicals – one for outside county and another for inside country rates. Inside and outside county revenues are reported separately in the Trial Balance. In a separate procedure for each of these two revenues, offices are ranked from largest to smallest revenue and allocated to 22 office size strata. The first 18 strata each represent 5 percent of total revenue, while the last 4 strata each represent 2.5 percent of relevant revenue.

Table 5 shows PERMIT coverage by stratum for outside county revenue. There are no non-PERMIT offices in the first 14 strata. These PERMIT offices are grouped into a single stratum. Each of strata 15 through 22 are retained individually as separate strata in the inflation process.

Table 6 shows PERMIT coverage by stratum for inside country revenue. There are no non-PERMIT offices in the first 6 strata so these offices are grouped into a single stratum. Similarly, since there are only 4 non-PERMIT offices in strata 7 through 9, these offices are grouped into a second stratum. The remaining strata are collapsed based on similarity of revenue shares by rate group. Table 7 shows revenue shares for inside county mail in each of the 22 original stratum. Based on general patterns in strata 10 through 22, strata 10 through 13 form a

new third stratum, strata 14 through 16 form a new fourth stratum, strata 17 and 18 form a new fifth stratum, and each of strata 19 through 22 form the new sixth through ninth strata.

In addition to these strata, one final stratum consists of all publications using the Centralized Postage Payment (CPP) program. This program allows postage payment at a single location (in New York) for mail deposited anywhere in the country for the authorized publication. CPP publications are typically among the largest in the country. The CPP stratum accounts for more than 30 percent of all Periodicals revenue. Table 5 shows the revenue coverage of the CPP stratum.

For all classes, the PERMIT transactions in each stratum are inflated to the total revenue in each stratum. The computed revenue control factor is applied to pieces and weight data as well, while maintaining the full array of rate characteristics including rate element, shape, and weight increment. Final results are computed by summing the inflated strata results over all strata.

The revenue control in each stratum is developed separately for permit imprint revenues and metered and stamped revenues in First Class and Standard Mail. Periodicals has no metered and stamped revenue. Permit imprint revenues are reported individually by class and post office in the Trial Balance. There is general consistency between Trial Balance permit imprint revenues and PERMIT system revenues. For the permit imprint portion of the revenue control, Trial Balance permit imprint revenues are used.

The Trial Balance does not uniquely identify metered and stamped revenue from bulk transactions. These revenues are reported together with metered and stamped revenue of many mail classes. As such, the Trial Balance cannot be used for the metered and stamped portion of the revenue control. At PERMIT offices, the metered and stamped portion of the revenue control comes directly

from the PERMIT system. At non-PERMIT offices the metered and stamped portion is estimated.

III. Estimating Metered and Stamped Revenue at Non-PERMIT Offices

First-Class and Standard Mail metered and stamped revenue estimates at non-PERMIT offices are obtained from a linear regression model. In First Class these estimates are used to assign non-PERMIT offices to the appropriate office size stratum. (Size strata for Standard Mail is determined solely from Trial Balance permit imprint revenue.) The estimates also contribute to the revenue control total in each stratum for inflating the PERMIT office data.

PERMIT bulk revenue is regressed on various Trial Balance revenue variables, by office. Separate equations are estimated for each postal quarter. It is assumed that the estimated relation between Trial Balance revenues and bulk metered and stamped revenues from PERMIT offices can be used to estimate non-PERMIT office bulk metered and stamped revenue. The regression parameters are multiplied by their respective Trial Balance revenue from each non-PERMIT office to yield metered and stamped revenue estimates for these offices. Table 8 shows the First-Class regression parameters estimated on FY 95 data.

The regression procedures for Standard Mail are similar to those for First-Class. The models are estimated using FY 96 data. Separate models are estimated for commercial and nonprofit rates. For commercial rates, two separate regressions are estimated based on office size. The size range of the non-PERMIT offices suggests that larger PERMIT sites be used to represent larger non-PERMIT offices. For the large office regression, only PERMIT offices in the first 18 original strata are used. (Stratification is based on FY 96 revenues.) The small office regression uses data from PERMIT offices in original strata 19 and 20. In addition, the largest 100 offices in both commercial and nonprofit rates are not included in the model estimation since these offices are considerably larger than

any of the non-PERMIT offices to be estimated. The regression parameters for Standard Mail are shown in Table 9.

IV. Postage Statement Data Processing

Transaction records from the PERMIT system are aggregated into a set of arrays by accounting period. The arrays contain revenue, pieces (and copies for Periodicals), and weight and are indexed by the rate category of mail, the weight increment, the stratum of the office where the mail was entered, the processing category (letters, flats, or parcels), and the indicia (permit imprint or metered and stamped). These arrays reduce the large quantity of transaction level data to the minimum detail required to produce the final estimates.

A. First-Class

The aggregation programs perform several checks on each transaction. The transactions, which pass all the integrity checks, are assigned to one of three quality categories:

Group 1: Identical piece transactions with valid weight information;

Group 2: Non-identical piece transactions with valid weight information; or

Group 3: Transactions with valid revenue and piece data but without valid weight information.

Observations that cannot meet the standards for any of these three groups are discarded. Their revenue, if valid, is retained for the inflation steps to follow.

Each VIP code in a transaction is checked for weight and rate validity. The weight must be greater than a minimum feasible weight and be less than the maximum allowable weight for each rate element. For example, First-Class cards and nonstandard pieces must be less than 1 ounce and all pieces must be no greater than 13 ounces. The data editing rules are depend on the information contained in the PERMIT record. See Appendix A.

B. Periodicals

Periodicals transactions are checked for internal consistency with respect to the various postage determinants. All Periodicals transactions must contain identical copies. Consequently, the data cleaning procedure for Periodicals is less complicated than for First-Class. It is clear in which weight increment each copy in a transaction belongs. If a transaction fails one of the consistency checks, it is discarded. Its revenue, however, is retained for the subsequent inflation steps.

C. Standard Mail

Pieces in Standard Mail transactions are also somewhat simpler to assign to weight increments than First-Class. First, various internal consistency checks are conducted for each transaction. Next, non-identical weight pieces all paying minimum rate within a transaction are assigned to the weight increment based on their average weight. Then, for pieces paying pound rates, the pound rate revenue is used to determine the weight within that presort/entry level rate. The ratio of this weight to the respective pieces is used to assign the pieces to a weight increment.

V. Inflation Process

Revenues in each First-Class and Standard Mail stratum are separated into permit imprint revenue and metered and stamped revenue. There is no separation in Periodicals since all revenue is permit imprint. For each separation in each stratum, the ratio of a revenue control total to observed PERMIT revenue is computed. These ratios are used to control observed revenue, pieces, and weight of all transactions by stratum. The computation of these ratios in First-Class and Standard Mail is completed in two stages.

In First-Class and Standard Mail observed permit imprint revenue from the PERMIT data is first controlled to the Trial Balance permit imprint revenue of PERMIT offices by stratum and quarter. This first stage controls for any missing data in the PERMIT system. In the second stage, revenue is controlled to the

total Trial Balance permit imprint revenue by stratum and quarter. This second control inflates the estimates to account for non-PERMIT offices.

The first stage metered and stamped revenue controls are handled differently for First-Class and Standard Mail. For First-Class, missing PERMIT system data are filled in from the CBCIS data files if they exist. If CBCIS is also missing data, then average revenues for the office are computed over the available accounting periods in the year. Once missing revenue data are filled in, a revenue control factor is computed and pieces and weight are adjusted accordingly. For Standard Mail, the first stage control factors for permit imprint revenue (explained above) are used to control for missing metered and stamped revenue.

The second stage control factors for First-Class and Standard Mail are computed identically. Each stratum is controlled to the total stratum revenue (including estimates of metered and stamped revenue from non-PERMIT offices). Finally, the delineation of the three data quality groups of First-Class transactions is maintained during the inflation process.

For Periodicals, since there is no metered or stamped mail, PERMIT system revenues by stratum are inflated directly to the Trial Balance revenue of all offices in the stratum.

VI. First-Class Missing Weight and Nonidentical Mail

After the strata are inflated, two more steps are needed to complete the First-Class estimates. First, the weight of non-identical mail in quality group 2 must be estimated. In this case there is valid information on the weight increment of the mail although the exact weight information is missing. The average weight of mail with good weight information is used to impute the weight of the missing weight mail by ounce increment.

The second case concerns the pieces and weight estimates of nonidentical piece transactions in quality group 3. Here only total pieces and revenue have been estimated. Neither the weight of these pieces nor their distribution by weight increment is known. These pieces must be redistributed to ounce increments so that total revenue in each rate category does not change. If the pieces were redistributed using the distribution of identical pieces by rate category, revenue would not remain unchanged. Also, mail in nonidentical piece transactions tends to be heavier than mail in identical piece transactions.

Redistribution of nonidentical pieces uses the weight distribution of pieces in identical weight mailings as a starting point. For identical pieces, the ounce increment with the largest proportion of mail is determined in each rate category. This ounce increment is used as a pivot point. The share of pieces from the identical piece distribution is computed for ounce increments up to and including the pivot ounce increment. Similar shares are computed for ounce increments above the pivot. The nonidentical pieces are distributed to these two groups and, in turn, to individual ounce increments based on the identical piece shares within each group. Nonidentical pieces are distributed to the two groups in a proportion that yields the known revenue for these pieces. This method provides a solution only if the average revenue of the nonidentical pieces falls between the average revenues of the two groups (above and below the pivot ounce increment) in the identical piece distributions. If this condition does not hold, the pivot ounce increment is moved one increment at a time until the condition is met.

VII. Allocation of First-Class Estimates to Half-ounce Increments.

The final results will show volumes by half-ounce increment up to 4 ounces, then by full ounce increment up to 13, the maximum weight. For Periodicals and Standard Mail, transaction data are of sufficiently high quality to immediately assign pieces to half-ounce increments. In the procedures described above, Periodicals and Standard Mail estimates contain detail by half-ounce increment

up to 4 ounces. For First-Class, estimates to this point are made only to whole ounce increments.

To produce half-ounce increment estimates for First-Class requires a second aggregation and inflation of transaction data. In this second inflation, only identical piece transactions with plausible weight information are used. Half ounce increments can be identified within this set of transactions. The resulting piece shares of half ounces within each full ounce increment are applied to the respective pieces by whole ounce increment from the first inflation. This procedure is applied by ounce increment up to 4 ounces, by shape, and by rate category. Weight and weight per piece by half ounce increment is also computed for the set of identical piece transactions. The identical piece transaction weight per piece times the distributed piece estimates provides the new estimate for total weight by half ounce increment.

VIII. Single Piece Mail

Single piece information by weight increment is needed for First-Class mail. Single piece input data are contained in the final edited domestic RPW sample data files. These data files include the proper sample inflation factors for each mail piece. After weight per piece is computed, pieces are assigned to their particular ounce increment. For pieces weighing four ounces or less, the intervals are defined in half ounce increments. Otherwise, the intervals are defined in whole ounce increments. All records are then inflated using the proper inflation factors and aggregated to mail category code, shape, indicia, and ounce increment.

IX. Revenue, Piece, and Weight Estimates by Mail Characteristics

Tables 10 through 21 and Table 24 contain results by mail class using the methods discussed above. The level of detail provided in each table is determined by the input needs of the various models that use these results. The estimates in each mail class can be converted to a set of distribution keys to be

applied to published RPW estimates. These keys are generally applied to RPW estimates of detailed rate elements in each Postal quarter.

First Class

In First-Class, in each quarter, the presort estimates by rate element are converted to shape and weight distributions keys which are then applied to corresponding RPW estimates. For single piece, the keys are also computed by indicia before distributing the RPW estimates. Revenue, pieces, and weight are distributed independently.

Periodicals

The Periodicals control to RPW is more elaborate due to the piece, weight, and discount portions of the rate structure. The following control steps are applied by subclass and quarter. First, pieces from the piece portion and weight from the weight portion are controlled to RPW pieces and weight, respectively. These same factors are applied to piece portion revenue and weight portion revenue, respectively. These adjusted piece and weight portion revenues plus the revenue discounts are then controlled to the RPW revenue total. Piece portion copies are controlled by the same factor as pieces. Weight portion copies are then controlled to these resulting piece portion copies. Similarly, and lastly, piece portion weight is controlled to weight portion weight.

Standard Mail

The results of the inflation procedure for Standard Mail transactions in the PERMIT system are used to develop a shape distribution key for each rate element. Keys are developed by Postal quarter, for each of revenue, pieces, and weight. The distribution keys for two small rate categories, Standard Mail paid at First Class rates and Standard Mail paid at Priority rates, are developed from the First Class and Priority mail transactions, respectively. The First Class estimation procedure is described above. The Priority distribution keys are derived from the uninflated Priority PERMIT transactions.

Package Services

Shape distribution keys for bulk entered rate elements of Package Services are developed from uninflated PERMIT transactions. Single piece rate elements are distributed to shape based on ODIS data. The keys are developed by Postal quarter.

PFY and GFY Estimates

In all classes, postal fiscal year (PFY) estimates by any characteristic are the sum of the four quarterly estimates. Government fiscal year (GFY) estimates by characteristic are derived by forming distribution keys from the PFY estimates and applying them to the reported RPW GFY estimates.

X. Updating the Mail Entry Point Profile from Witness Talmo (USPS-ST-50) in Docket No. R97-1 (LR-H-105 and R97-1 LR-H-195)

The Standard Mail Mail Characteristics Surveys sponsored in Docket No. R97-1 by USPS witness Talmo (USPS-ST-50/R97-1) and documented in LR-H-105 and LR-H-195 asked for detailed information on the volume of mail entered at various facility types, the volume of mail destinating in the service area for BMEU entry, volumes receiving entry discounts for drop shipment, and volumes plant loaded. The PERMIT system is used to control the entry profile distribution of Standard Mail volumes estimated in Docket No. R97-1 USPS-ST-50 to be consistent with FY 2000 volumes by entry discount. This information is used to help model cost of transportation between postal facilities and non-transportation distance related costs of mail processing.

In Docket No. R97-1 USPS-ST-50 the mail entry point profile questions were divided into three sections. The first section concerns only BMEU accepted transactions. Here, for each transaction, the total volume and the volume destinating in the office's service area were recorded. The second section

concerns transactions that are drop shipments. These volumes were distinguished by the type of facility to which the mail is first transported (BMCs, ASFs, SCFs, and AOs). The volume of mail receiving an entry discount for each of these facilities' service areas was also recorded. The third section concerns mail that was plant loaded. These volumes were distinguished by the type of facility to which the mail was first transported (BMCs, ASFs, SCFs, and AOs). The volume of mail destinating in each of these facilities' service territories was also recorded.

These data were controlled to the FY 00 distribution by applying the ratio of FY 00 volume to FY 96 volume by entry discount (None, DBMC, DSCF and DDU).

Tables 22 and 23 show mail volumes and weight by container type, origin or destination entry, and facility type for commercial and nonprofit subclasses of Standard Mail, respectively. For BMEU entry, the facility type is that of the sampled office. For drop shipment and plant load transactions, the facility type is based on the facilities to where the mail will first be transported. BMEU entry and drop shipment mail volumes are summed together in the columns labeled "Drop Shipped", and plant load mail volumes appear in the column labeled "Plant Load".

XI. Tables

Table 1

First-Class Mail, PFY 1994
PERMIT/BRAVIS System Volume Shares by Presort Level
(Percent)

Stratum	Single Piece	Nonpresort Z4 & BC	Presort Residuals	3/5 Digit Presort	Presort Barcoded	Carrier Route	Total Pieces (Thousands)
1	2.2	0.1	2.8	16.6	75.4	2.8	1,985,410
2	4.2	0.6	4.0	20.5	47.9	22.7	1,678,904
3	2.5	0.4	4.6	24.2	60.8	7.5	1,551,063
4	4.1	1.4	6.3	23.8	54.9	9.6	1,977,468
5	3.6	0.3	5.5	22.6	56.0	12.0	1,692,104
6	2.9	0.7	4.6	28.1	52.1	11.8	1,632,434
7	2.8	0.3	7.5	22.5	61.1	5.8	1,916,945
8	2.5	0.3	6.7	29.7	52.4	8.4	1,660,877
9	3.3	0.7	6.2	29.3	48.9	11.6	1,664,790
10	3.0	0.3	6.8	29.5	55.0	5.3	1,836,516
11	2.1	0.3	6.5	20.0	59.6	11.5	1,760,145
12	3.7	0.7	6.3	25.2	57.6	6.5	1,851,898
13	4.9	1.4	7.4	31.2	49.5	5.5	1,882,490
14	2.5	1.1	8.4	31.4	50.7	5.9	1,665,845
15	6.6	1.3	8.3	31.0	45.0	7.8	1,646,204
16	4.6	2.2	8.2	31.9	43.1	10.0	1,675,541
17	6.6	2.3	8.6	37.9	33.3	11.2	1,393,397
18	9.1	2.6	8.9	49.0	22.2	8.1	1,036,945
19	15.3	3.9	7.7	60.5	7.2	5.4	596,534
20	30.2	2.4	5.8	53.4	3.5	4.7	244,317
Total	4.2	1.0	6.5	28.1	51.3	8.9	31,349,828

Table 2

First-Class Mail Total Revenue Coverage
PFY 2000

	All Offices		PERMIT Offices		Other Offices		
Stratum	Number of Offices	Total Revenue	Number of Offices	Total Revenue	Number of Offices	Total Revenue	Percent Unreported
1	2	879,742,892	2	879,742,892	0	0	0.0%
2	3	538,495,489	3	538,495,489	0	0	0.0%
3	3	578,199,680	3	578,199,680	0	0	0.0%
4	4	761,776,264	4	761,776,264	0	0	0.0%
5	4	637,598,199	4	637,598,199	0	0	0.0%
6	4	521,762,823	4	521,762,823	0	0	0.0%
7	6	793,954,548	6	793,954,548	0	0	0.0%
8	5	573,731,915	5	573,731,915	0	0	0.0%
9	6	496,118,019	6	496,118,019	0	0	0.0%
10	8	769,509,012	8	769,509,012	0	0	0.0%
11	10	473,892,810	10	473,892,810	0	0	0.0%
12	13	736,311,871	13	736,311,871	0	0	0.0%
13	14	667,270,502	14	667,270,502	0	0	0.0%
14	19	615,656,772	18	615,470,737	1	186,035	0.0%
15	31	872,419,372	31	872,419,372	0	0	0.0%
16	51	787,693,251	49	786,486,972	2	1,206,279	0.2%
17	99	630,700,631	98	628,845,149	1	1,855,482	0.3%
18	251	813,848,946	230	786,839,557	21	27,009,389	3.3%
19	865	659,410,221	646	555,440,395	219	103,969,826	15.8%
20	9,269	949,058,583	1,014	435,652,552	8,255	513,406,031	54.1%
Total	10,667	13,757,151,800	2,168	13,109,518,758	8,499	647,633,042	4.7%

Table 3

Standard Mail Commercial Rate PERMIT System Revenue Coverage by Stratum, FY 2000

	All Offices		PERMIT Offices		Other Offices		
Stratum	Number of Offices	Total Revenue	Number of Offices	Total Revenue	Number of Offices	Total Revenue	Percent Unreported
1	1	514,785,259	1	514,785,259	0	0	0.0%
2	3	666,255,683	3	666,255,683	0	0	0.0%
3	4	682,599,861	4	682,599,861	0	0	0.0%
4	4	535,165,612	4	535,165,612	0	0	0.0%
5	6	643,857,435	6	643,857,435	0	0	0.0%
6	6	585,146,966	6	585,146,966	0	0	0.0%
7	7	611,151,290	7	611,151,290	0	0	0.0%
8	8	620,319,645	8	620,319,645	0	0	0.0%
9	9	597,192,810	9	597,192,810	0	0	0.0%
10	12	656,644,647	12	656,644,647	0	0	0.0%
11	13	593,258,758	13	593,258,758	0	0	0.0%
12	15	586,196,923	15	586,196,923	0	0	0.0%
13	19	621,456,498	19	621,456,498	0	0	0.0%
14	23	618,626,408	23	618,626,408	0	0	0.0%
15	29	608,601,628	29	608,601,628	0	0	0.0%
16	39	595,418,226	39	595,418,226	0	0	0.0%
17	62	616,548,513	62	616,548,513	0	0	0.0%
18	117	608,418,855	116	603,341,935	2	5,076,920	0.8%
19	368	607,682,357	349	579,115,303	29	28,567,054	4.7%
20	14,937	608,643,492	1,520	314,949,563	13,528	293,693,928	48.3%

Table 4

Standard Mail Nonprofit Rate PERMIT System Revenue Coverage by Stratum, FY 2000

	All Offices		PERMIT Offices		Other Offices		
Stratum	Number of Offices	Total Revenue	Number of Offices	Total Revenue	Number of Offices	Total Revenue	Percent Unreported
1	2	55,219,238	26	55,219,238	0	0	0.0%
2	4	80,859,119	52	80,859,119	0	0	0.0%
3	4	67,286,684	52	67,286,684	0	0	0.0%
4	4	54,754,412	52	54,754,412	0	0	0.0%
5	7	74,229,156	91	74,229,156	0	0	0.0%
6	7	59,421,249	91	59,421,249	0	0	0.0%
7	9	64,898,688	117	64,898,688	0	0	0.0%
8	11	67,342,520	143	67,342,520	0	0	0.0%
9	13	66,506,417	169	66,506,417	0	0	0.0%
10	16	64,864,645	208	64,864,645	0	0	0.0%
11	21	65,743,010	273	65,743,010	0	0	0.0%
12	26	65,755,238	338	65,755,238	0	0	0.0%
13	34	64,980,956	442	64,980,956	0	0	0.0%
14	47	65,664,418	611	65,664,418	0	0	0.0%
15	75	65,397,939	975	65,397,939	0	0	0.0%
16	121	65,336,049	1,539	64,013,118	4	1,322,931	2.0%
17	219	65,522,324	2,780	64,124,095	6	1,398,229	2.1%
18	471	65,559,114	5,166	56,477,714	79	9,081,400	13.9%
19	1,245	65,485,610	8,235	35,824,960	628	29,660,651	45.3%
20	14,933	65,508,633	6,937	7,507,318	14,434	58,001,315	88.5%

Table 5

Outside-County Periodicals PERMIT System Coverage by Stratum, FY 2000

Stratum	PERMIT Offices	PERMIT Revenue	Trial Balance Offices	Trial Balance Revenue	Revenue Coverage
1	1	80,319,039	1	80,256,039	100%
2	1	55,647,805	1	56,324,444	99%
3	2	60,668,314	2	60,702,332	100%
4	2	57,042,852	2	57,413,519	99%
5	3	79,599,199	3	78,857,215	101%
6	2	47,265,609	2	47,458,177	100%
7	4	81,637,634	4	80,511,175	101%
8	3	52,162,687	3	52,091,982	100%
9	5	74,414,958	5	71,685,749	104%
10	5	65,650,465	5	65,305,321	101%
11	6	61,839,367	6	62,037,624	100%
12	8	64,038,535	8	64,414,892	99%
13	11	64,205,495	11	64,537,249	99%
14	16	65,644,472	16	65,934,858	100%
15	20	57,638,232	22	63,636,003	91%
16	34	67,261,174	34	64,185,658	105%
17	63	63,498,301	63	64,925,605	98%
18	142	63,766,648	143	65,157,909	98%
19	151	26,098,238	187	32,364,767	81%
20	302	21,153,901	459	32,344,418	65%
21	437	11,856,347	1,182	32,359,561	37%
22	670	4,842,012	5,083	32,373,795	15%
Total	1,888	1,226,251,283	7,242	1,294,878,293	95%
CPP*	1	773,770,883	1	764,398,315	101%

* Centralized Postage Payment - CPP revenues include both inside county and outside county revenues since the Trial Balance does not separate these amounts.

Table 6

Inside-County Periodicals PERMIT System Coverage by Stratum, FY 2000

Stratum	PERMIT Offices	PERMIT Revenue	Trial Balance Offices	Trial Balance Revenue	Revenue Coverage
1	2	4,062,708	2	4,003,447	101%
2	2	1,421,105	2	2,289,332	62%
3	5	3,488,063	5	3,494,430	100%
4	7	3,220,199	7	3,229,391	100%
5	9	3,184,906	9	3,208,024	99%
6	13	3,269,527	13	3,301,685	99%
7	18	3,077,506	19	3,228,080	95%
8	23	2,951,787	25	3,173,017	93%
9	38	3,179,401	39	3,237,511	98%
10	47	2,883,244	54	3,273,860	88%
11	55	2,588,735	69	3,210,538	81%
12	69	2,403,346	93	3,242,313	74%
13	92	2,437,024	124	3,257,040	75%
14	93	1,826,665	160	3,246,571	56%
15	104	1,601,797	214	3,241,633	49%
16	138	1,607,207	284	3,235,595	50%
17	152	1,277,256	387	3,246,044	39%
18	185	1,087,766	546	3,238,347	34%
19	103	443,082	377	1,624,578	27%
20	117	380,168	499	1,618,920	23%
21	113	276,889	717	1,622,313	17%
22	414	612,264	1,836	1,620,252	38%
Total	1,799	47,280,645	5,481	64,842,921	73%

Table 7

Inside County Periodicals Revenue Shares by Rate Group and Stratum, FY 2000

Stratum	Basic Auto	Basic Non-Auto	3/5 Auto	3/5 Non-Auto	Carrier Route	High Density	Saturation
1	0.00	0.04	0.13	0.11	0.72	0.00	0.00
2	0.00	0.01	0.03	0.06	0.28	0.57	0.05
3	0.00	0.11	0.12	0.16	0.46	0.04	0.11
4	0.01	0.10	0.04	0.18	0.65	0.02	0.00
5	0.00	0.06	0.08	0.14	0.68	0.00	0.05
6	0.00	0.05	0.08	0.14	0.63	0.08	0.02
7	0.00	0.09	0.07	0.16	0.63	0.00	0.04
8	0.00	0.06	0.04	0.11	0.50	0.08	0.20
9	0.00	0.08	0.07	0.17	0.60	0.06	0.02
10	0.00	0.08	0.04	0.16	0.52	0.09	0.11
11	0.00	0.05	0.05	0.16	0.57	0.06	0.10
12	0.00	0.09	0.05	0.17	0.53	0.10	0.07
13	0.00	0.07	0.05	0.17	0.60	0.08	0.04
14	0.00	0.11	0.05	0.20	0.53	0.09	0.02
15	0.00	0.08	0.04	0.20	0.58	0.07	0.02
16	0.00	0.10	0.04	0.23	0.55	0.05	0.03
17	0.00	0.14	0.04	0.25	0.50	0.04	0.01
18	0.00	0.18	0.05	0.26	0.47	0.03	0.01
19	0.01	0.23	0.04	0.33	0.38	0.01	0.00
20	0.00	0.17	0.07	0.36	0.37	0.03	0.01
21	0.00	0.26	0.08	0.32	0.26	0.03	0.04
22	0.01	0.16	0.04	0.22	0.52	0.05	0.00
Total	0.00	0.08	0.06	0.16	0.57	0.07	0.05

Table 8

First-Class Mail Regression Estimates

Dependent Variable: Bulk Entered First Class Metered and Stamped Revenue

Independent Variables	PQ1	PQ2	PQ3	PQ4
Permit Imprint Presort Revenue	0.187064	0.248894	0.254393	0.161755
	(0.0462)	(0.0466)	(0.0502)	(0.0510)
All Metered Revenue	0.234259	0.244983	0.240415	0.261537
	(0.0103)	(0.0110)	(0.0110)	(0.0113)
All Metered Revenue Squared (in millions)	-0.004364	-0.00415	-0.004106	-0.003343

Table 9

Standard Mail Regression Estimates

Regular and ECR Large Office Estimate
Dependent Variable: Bulk Entered Metered and Stamped Revenue

Independent Variables	PQ1	PQ2	PQ3	PQ4
Permit Imprint Revenue	0.097314	0.103178	0.13003	0.115697
	(.0142)	(.0140)	(.0151)	(.0136)
All Metered Revenue	0.023253	0.019722	0.016672	0.015445
	(.0058)	(.0051)	(.0053)	(.0048)
All Metered Revenue Squared (in millions)	-0.000104	-0.00011	-0.000048	-0.000045
	(.0001)	(.0001)	(.0001)	(.00004)
R-Squared	0.21	0.21	0.28	0.23
Number of Observations	312	331	345	343

Regular and ECR Small Office Estimate
Dependent Variable: Bulk Entered Metered and Stamped Revenue

Independent Variables	PQ1	PQ2	PQ3	PQ4
Permit Imprint Revenue	0.086807	0.100748	0.103331	0.121616
	(.0218)	(.0259)	(.0199)	(.0206)
All Metered Revenue	0.009477	0.011302	0.006696	0.005279
	(.0034)	(.0035)	(.0028)	(.0028)
All Metered Revenue Squared (in millions)	-0.000336	-0.000123	-0.000523	-0.000422
	(.0004)	(.0004)	(.0003)	(.0002)
R-Squared	0.11	0.09	0.11	0.11
Number of Observations	779	910	1024	1045

Nonprofit and Nonprofit ECR Estimate
Dependent Variable: Bulk Entered Metered and Stamped Revenue

Independent Variables	PQ1	PQ2	PQ3	PQ4
Permit Imprint Revenue	0.272776	0.165223	0.146355	0.170342
	(.0218)	(.0195)	(.0218)	(.0211)
All Metered Revenue	-0.001207	0.001546	0.0025	0.00101
All Metered Revenue Squared (in millions)	-0.000035	-0.000096	-0.000102	-0.000052
	(.0001)	(.00004)	(.00004)	(.00004)
R-Squared	0.18	0.10	0.07	0.08
Number of Observations	937	1051	1168	1212

Table 10

First Class Single Piece Volume and Weight By Shape and Indicia, FY 2000

(Thousands)

Shape	Indicia	Pieces	Weight
Letters	Stamped	25,322,888	640,139
	Metered	18,372,991	648,908
	Permit Imprint	2,744,762	93,902
	Stamped Envelope	166,095	4,671
	Precanceled Stamp	5,241	192
	Stamped - Govt	9,460	356
	Metered - Govt	220,176	9,387
	Penalty - All Other	527	47
	Permit Imprint - Govt	598	16
	Stmpd Envlop - Govt	8,517	248
	Franked - Govt	-	-
	Penalty - USPS	-	-
	None	40,201	1,081
	Permit Imprint - USPS	-	-
	Postal Validation Imprint	141,650	7,133
	Total	47,033,105	1,406,078
Flats	Stamped	648,925	116,561
	Metered	3,659,079	806,403
	Permit Imprint	378,059	80,542
	Stamped Envelope	386	55
	Precanceled Stamp	1,013	215
	Stamped - Govt	1,855	387
	Metered - Govt	60,482	12,228
	Penalty - All Other	100	14
	Permit Imprint - Govt	209	56
	Stmpd Envlop - Govt	794	159
	Franked - Govt	-	-
	Penalty - USPS	-	-
	None	5,131	1,098
	Permit Imprint - USPS	-	-
	Postal Validation Imprint	88,667	22,282
	Total	4,844,701	1,040,001

Shape	Indicia	Pieces	Weight
Parcels	Stamped	52,788	13,157
	Metered	243,352	83,950
	Permit Imprint	152,658	41,835
	Stamped Envelope	36	13
	Precanceled Stamp	18	4
	Stamped - Govt	98	39
	Metered - Govt	844	336
	Penalty - All Other	9	4
	Permit Imprint - Govt	2	1
	Stmpd Envlop - Govt	8	3
	Franked - Govt	-	-
	Penalty - USPS	-	-
	None	2,320	977
	Permit Imprint - USPS	-	-
	Postal Validation Imprint	39,596	14,984
	Total	491,729	155,302
Cards	Stamped	1,111,237	6,945
	Metered	703,911	4,399
	Permit Imprint	482,479	3,021
	Stamped Envelope	3,471	22
	Precanceled Stamp	401,108	2,507
	Stamped - Govt	564	4
	Metered - Govt	2,846	18
	Penalty - All Other	74	0
	Permit Imprint - Govt	51	0
	Stmpd Envlop - Govt	206	1
	Franked - Govt	-	-
	Penalty - USPS	-	-
	None	3,022	19
	Permit Imprint - USPS	-	-
	Postal Validation Imprint	10,331	65
	Total	2,719,298	17,001

Table 11

First Class Single Piece Volume by Shape, Ounce Increment, and Indicia and Weight by Shape and Ounce Increment, FY 2000 (Thousands)

Volume, All Indicias

	0-0.5	0.5-1.0	1.0-1.5	1.5-2.0	2.0-2.5	2.5-3.0	3.0-3.5	3.5-4.0	4.0-5.0	5.0-6.0	6.0-7.0	7.0-8.0	8.0-9.0	9.0-10.0	10.0-11.0	11.0-12.0	12.0-13.0	Total
Letters	35,268,121	9,373,618	1,454,408	488,607	219,155	102,915	53,226	27,081	26,128	10,291	4,954	2,139	1,188	509	521	166	79	47,033,105
Flats	42,241	314,660	791,793	710,792	552,005	411,459	345,864	270,925	407,244	278,810	199,612	151,911	117,863	87,962	69,051	53,462	39,047	4,844,701
Parcels	33,729	10,413	26,852	32,770	35,900	31,632	28,038	27,429	51,013	40,183	37,013	30,355	30,270	23,920	20,403	17,622	14,187	491,729
Total	35,344,092	9,698,691	2,273,053	1,232,169	807,061	546,006	427,128	325,435	484,384	329,284	241,579	184,405	149,321	112,390	89,975	71,250	53,313	52,369,535

Volume, Stamped

	0-0.5	0.5-1.0	1.0-1.5	1.5-2.0	2.0-2.5	2.5-3.0	3.0-3.5	3.5-4.0	4.0-5.0	5.0-6.0	6.0-7.0	7.0-8.0	8.0-9.0	9.0-10.0	10.0-11.0	11.0-12.0	12.0-13.0	Total
Letters	20,791,501	3,869,282	440,118	147,707	46,610	20,010	10,679	4,748	4,028	1,163	1,112	362	219	38	7	2	1	25,337,589
Flats	9,919	50,377	125,198	109,960	74,994	64,767	50,485	36,343	46,816	27,383	18,519	14,122	8,829	5,862	3,597	2,997	1,625	651,793
Parcels	2,664	2,311	5,384	6,074	4,787	4,890	3,633	3,257	4,796	3,639	2,827	2,343	1,993	1,557	1,085	898	765	52,904
Total	20,804,084	3,921,970	570,700	263,740	126,391	89,666	64,798	44,348	55,641	32,186	22,458	16,827	11,041	7,456	4,690	3,898	2,391	26,042,285

Volume, Stamped Envelope

	0-0.5	0.5-1.0	1.0-1.5	1.5-2.0	2.0-2.5	2.5-3.0	3.0-3.5	3.5-4.0	4.0-5.0	5.0-6.0	6.0-7.0	7.0-8.0	8.0-9.0	9.0-10.0	10.0-11.0	11.0-12.0	12.0-13.0	Total
Letters	136,150	34,699	3,106	441	78	13	2	1	123	0	0	0	0	0	0	0	0	174,612
Flats	66	127	354	112	42	93	35	36	117	28	58	35	23	42	6	3	3	1,180
Parcels	0	1	1	1	3	3	2	4	1	6	6	7	2	4	1	1	0	44
Total	136,216	34,828	3,461	553	124	109	39	41	241	34	64	42	25	47	7	4	3	175,837

Volume, Metered

	0-0.5	0.5-1.0	1.0-1.5	1.5-2.0	2.0-2.5	2.5-3.0	3.0-3.5	3.5-4.0	4.0-5.0	5.0-6.0	6.0-7.0	7.0-8.0	8.0-9.0	9.0-10.0	10.0-11.0	11.0-12.0	12.0-13.0	Total
Letters	12,695,319	4,382,672	884,267	302,718	158,893	76,116	37,525	20,632	20,369	8,089	3,087	1,641	927	223	476	147	65	18,593,167
Flats	24,336	225,039	599,935	536,279	425,761	304,326	261,996	208,126	321,581	224,790	161,466	122,889	96,610	72,437	56,403	44,567	33,021	3,719,561
Parcels	10,105	4,108	11,638	15,464	14,858	14,729	13,845	13,836	24,737	21,838	19,609	17,324	17,015	13,742	12,265	10,431	8,650	244,196
Total	12,729,760	4,611,819	1,495,840	854,460	599,513	395,172	313,366	242,593	366,688	254,718	184,162	141,854	114,552	86,402	69,144	55,145	41,737	22,556,924

Weight, All Indicia

	0-0.5	0.5-1.0	1.0-1.5	1.5-2.0	2.0-2.5	2.5-3.0	3.0-3.5	3.5-4.0	4.0-5.0	5.0-6.0	6.0-7.0	7.0-8.0	8.0-9.0	9.0-10.0	10.0-11.0	11.0-12.0	12.0-13.0	Total
Letters	773,433	382,381	115,068	53,665	31,230	17,795	10,864	6,374	7,249	3,535	2,011	1,008	628	304	349	122	62	1,406,078
Flats	869	17,069	65,676	77,058	79,733	70,702	71,435	63,629	114,856	96,112	81,439	71,526	62,847	52,439	45,477	38,545	30,590	1,040,001
Parcels	159	659	2,233	3,697	5,134	5,526	5,781	6,522	14,499	13,923	15,145	14,345	16,170	14,268	13,454	12,674	11,114	155,302
Total	774,460	400,109	182,977	134,419	116,097	94,024	88,079	76,525	136,604	113,569	98,596	86,879	79,645	67,011	59,280	51,341	41,766	2,601,381

Table 12

First Class Presort Volume By Shape, Ounce Increment, and Rate Group and Weight by Shape and Ounce Increment, FY 2000 (Thousands)

Volume, Non-Automated

	Ounces																	
Shape	0-0.5	0.5-1.0	1.0-1.5	1.5-2.0	2.0-2.5	2.5-3.0	3.0-3.5	3.5-4.0	4.0-5.0	5.0-6.0	6.0-7.0	7.0-8.0	8.0-9.0	9/0-10.0	10.0-11.0	11.0-12.0	12.0-13.0	Grand Total
Letters	1,762,487	2,168,970	85,531	38,935	15,992	27,714	5,034	7,160	5,444	396	146	35	85	18	36	21	1	4,118,006
Flats	6,446	30,992	34,253	56,007	29,630	58,367	9,824	32,954	13,595	5,718	3,565	3,715	3,520	1,877	1,820	1,890	821	294,996
Parcels	41	6,370	145	342	173	979	345	122	395	343	223	254	68	95	55	30	0	9,980
Total	1,768,975	2,206,332	119,928	95,285	45,795	87,061	15,203	40,236	19,434	6,456	3,934	4,005	3,673	1,991	1,912	1,941	823	4,422,981

Volume, Automated Non-Carrier

	Ounces																	
Shape	0-0.5	0.5-1.0	1.0-1.5	1.5-2.0	2.0-2.5	2.5-3.0	3.0-3.5	3.5-4.0	4.0-5.0	5.0-6.0	6.0-7.0	7.0-8.0	8.0-9.0	9/0-10.0	10.0-11.0	11.0-12.0	12.0-13.0	Grand Total
Letters	14,968,554	23,786,724	625,299	283,630	39,746	25,417	8,919	0	0	0	0	0	0	0	0	0	0	39,738,290
Flats	18,677	82,712	72,218	75,518	37,071	39,061	18,048	13,335	21,659	13,327	8,593	10,631	7,498	7,052	5,814	4,822	2,833	438,867
Parcels	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Total	14,987,231	23,869,436	697,517	359,149	76,817	64,478	26,967	13,335	21,659	13,327	8,593	10,631	7,498	7,052	5,814	4,822	2,833	40,177,157

Volume, Automated Carrier

	Ounces																	
Shape	0-0.5	0.5-1.0	1.0-1.5	1.5-2.0	2.0-2.5	2.5-3.0	3.0-3.5	3.5-4.0	4.0-5.0	5.0-6.0	6.0-7.0	7.0-8.0	8.0-9.0	9/0-10.0	10.0-11.0	11.0-12.0	12.0-13.0	Grand Total
Letters	345,739	687,382	29,803	8,107	3,025	1,131	147	0	0	0	0	0	0	0	0	0	0	1,075,333
Flats	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Parcels	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Total	345,739	687,382	29,803	8,107	3,025	1,131	147	0	0	0	0	0	0	0	0	0	0	1,075,333

Volume, Cards

	Ounces																	
Rate Group	0-0.5	0.5-1.0	1.0-1.5	1.5-2.0	2.0-2.5	2.5-3.0	3.0-3.5	3.5-4.0	4.0-5.0	5.0-6.0	6.0-7.0	7.0-8.0	8.0-9.0	9/0-10.0	10.0-11.0	11.0-12.0	12.0-13.0	Grand Total
Non-Auto	682,588	4,031	383	1,105	0	0	13	0	8	0	0	0	0	0	0	0	0	688,129
Automated	1,961,179	23,278	351	48	10	18	0	0	0	0	0	0	0	0	0	0	0	1,984,884
Auto-Carrier	88,214	147	12	14	0	7	0	0	0	0	0	0	0	0	0	0	0	88,394
Total	2,731,981	27,456	746	1,167	10	26	13	0	8	0	0	0	0	0	0	0	0	2,761,407

Weight

	Ounces																	
Shape	0-0.5	0.5-1.0	1.0-1.5	1.5-2.0	2.0-2.5	2.5-3.0	3.0-3.5	3.5-4.0	4.0-5.0	5.0-6.0	6.0-7.0	7.0-8.0	8.0-9.0	9.0-10.0	10.0-11.0	11.0-12.0	12.0-13.0	Total
Letters	345,900	1,177,019	52,464	34,457	7,987	9,064	2,673	1,545	1,425	125	55	15	41	10	21	14	1	1,632,817
Flats	627	5,802	7,926	13,931	8,832	15,830	5,285	10,409	9,360	6,100	4,608	6,313	5,441	4,984	4,721	4,563	2,668	117,400
Parcels	1	357	10	36	23	157	67	26	103	106	83	111	34	51	34	20	0	1,218
Cards	20,750	1,262	60	119	1	4	3	0	3	0	0	0	0	0	0	0	0	22,202
Total	367,278	1,184,439	60,461	48,544	16,843	25,056	8,027	11,981	10,891	6,331	4,745	6,438	5,516	5,045	4,776	4,597	2,669	1,773,637

Table 13

Priority Pieces by Mail Category and Shape, FY 2000
(Thousands)

Code	*Mail Category*	Letters	Flats	Parcels
7500	1-C PRIORITY MAIL	14,553	290,829	787,627
7501	AGEN 1-C PRIORITY MAIL	13	252	678
7520	1-C FLAT RATE ENVELOPE PRIORITY MAIL	0	108,633	12,880
7550	1-C SP KEYS AND IDENTIFICATION DEVICES PRIORITY MAIL	0	0	117
	TOTAL	14,565	399,715	801,302

Table 14

Periodicals Pieces and Copies by Subclass, Presort Level, and Shape, FY 2000
(Thousands)

		Letters		Flats		Parcels		Total	
Subclass	Presort Level	Pieces	Copies	Pieces	Copies	Pieces	Copies	Pieces	Copies
In-County	Basic Auto	891	891	1,217	1,223	0	0	2,108	2,114
In-County	Basic Non-Auto	22,022	22,038	66,031	74,000	114	116	88,167	96,153
In-County	3-Digit Auto	5,283	5,283	4,212	4,244	0	0	9,495	9,527
In-County	3-Digit Non-Auto	6,573	6,576	36,392	36,976	2	2	42,966	43,554
In-County	5-Digit Auto	5,272	5,334	41,355	41,477	0	0	46,628	46,811
In-County	5-Digit Non-Auto	21,212	21,217	102,839	103,816	1	1	124,052	125,034
In-County	Basic Carrier Route	15,532	15,557	471,876	473,399	5	5	487,413	488,961
In-County	High Density	91	91	54,466	54,567	0	0	54,557	54,658
In-County	Saturation	70	70	41,613	41,696	0	0	41,683	41,767
In-County	Enclosures	0	0	0	0	0	0	0	0
Regular	Basic Auto	48,290	48,361	123,133	124,742	0	0	171,423	173,103
Regular	Basic Non-Auto	5,358	7,948	353,063	474,012	1,125	3,328	359,546	485,288
Regular	3-Digit Auto	40,484	40,484	952,349	956,518	0	0	992,833	997,002
Regular	3-Digit Non-Auto	3,897	4,543	372,464	423,644	373	445	376,735	428,632
Regular	5-Digit Auto	1,140	1,140	2,017,319	2,019,630	0	0	2,018,459	2,020,770
Regular	5-Digit Non-Auto	1,920	2,406	420,764	463,016	389	410	423,073	465,832
Regular	Basic Carrier Route	6,078	6,080	2,852,961	2,872,133	5	5	2,859,044	2,878,218
Regular	High Density	1	1	23,955	23,977	1	2	23,957	23,980
Regular	Saturation	31	31	25,247	25,248	0	0	25,278	25,279
Regular	Enclosures	0	0	0	0	0	0	0	0
Nonprofit	Basic Auto	20,971	20,971	23,132	23,506	0	0	44,103	44,477
Nonprofit	Basic Non-Auto	13,907	198,477	71,323	165,056	434	11,455	85,664	374,989
Nonprofit	3-Digit Auto	40,657	40,657	171,475	172,724	0	0	212,132	213,381
Nonprofit	3-Digit Non-Auto	7,440	7,573	69,382	90,502	115	2,309	76,936	100,384
Nonprofit	5-Digit Auto	1,257	1,273	451,253	453,757	0	0	452,510	455,029
Nonprofit	5-Digit Non-Auto	6,799	6,916	97,473	109,395	45	1,269	104,316	117,580
Nonprofit	Basic Carrier Route	30,969	30,970	1,055,863	1,058,610	1	34	1,086,833	1,089,614
Nonprofit	High Density	52	52	73,068	73,078	0	0	73,120	73,130
Nonprofit	Saturation	89	89	17,697	17,698	0	0	17,786	17,787
Nonprofit	Enclosures	0	0	0	0	0	0	0	0
Classroom	Basic Auto	163	163	1,906	1,941	0	0	2,068	2,103
Classroom	Basic Non-Auto	17	17	8,179	224,195	3	58	8,199	224,271
Classroom	3-Digit Auto	170	170	11,453	11,523	0	0	11,624	11,693
Classroom	3-Digit Non-Auto	85	85	6,071	72,241	0	1	6,156	72,327
Classroom	5-Digit Auto	9	9	14,437	14,730	0	0	14,446	14,739
Classroom	5-Digit Non-Auto	9	9	3,957	81,240	0	2	3,966	81,251
Classroom	Basic Carrier Route	290	290	16,824	20,580	0	0	17,114	20,871
Classroom	High Density	0	0	4	4	0	0	4	4
Classroom	Saturation	0	0	391	391	0	0	391	391
Classroom	Enclosures	0	0	0	0	0	0	0	0

Table 15

Periodicals Copies and Pounds By Shape and Ounce Increment, FY 2000
(Thousands)

Regular Rate Copies

Shape/Ounce	0-0.5	0.5-1.0	1.0-1.5	1.5-2.0	2.0-2.5	2.5-3.0	3.0-3.5	3.5-4.0	4.0-5.0	5.0-6.0	6.0-7.0	7.0-8.0	8.0-9.0	9.0-10.0	10.0-11.0	11.0-12.0	12.0-13.0	13.0-14.0	14.0-15.0	15.0-16.0	> 16.0	Total
Letters	17,067	33,400	29,601	18,982	8,148	1,680	1,219	381	325	54	8	41	12	1	11	3	4	1	12	39	3	110,993
Flats	60,100	22,994	64,794	139,032	123,032	184,024	223,977	244,731	599,111	1,000,319	1,036,670	680,814	548,974	387,056	369,930	277,654	239,049	204,226	178,235	166,388	631,811	7,382,921
Parcels	169	1,251	623	274	223	30	21	10	103	48	46	39	38	21	48	27	24	37	24	25	1,110	4,190
Total	77,336	57,645	95,017	158,288	131,403	185,734	225,218	245,122	599,539	1,000,422	1,036,725	680,894	549,024	387,077	369,989	277,684	239,077	204,264	178,271	166,452	632,923	7,498,104

Nonprofit Rate Copies

Shape/Ounce	0-0.5	0.5-1.0	1.0-1.5	1.5-2.0	2.0-2.5	2.5-3.0	3.0-3.5	3.5-4.0	4.0-5.0	5.0-6.0	6.0-7.0	7.0-8.0	8.0-9.0	9.0-10.0	10.0-11.0	11.0-12.0	12.0-13.0	13.0-14.0	14.0-15.0	15.0-16.0	> 16.0	Total
Letters	34,071	20,200	231,468	11,783	6,971	1,870	352	116	117	5	8	5	1	1	1	1	0	0	0	0	7	306,978
Flats	35,708	56,398	164,188	302,870	419,120	145,940	138,119	141,647	169,860	89,758	133,927	121,862	35,749	24,682	27,693	26,040	32,672	25,626	35,204	13,860	23,403	2,164,326
Parcels	732	557	747	1,790	474	192	179	417	2,374	2,507	1,171	2,283	0	0	0	163	305	0	0	0	1,174	15,066
Total	70,511	77,156	396,403	316,443	426,565	148,003	138,650	142,181	172,351	92,269	135,106	124,150	35,750	24,683	27,694	26,204	32,977	25,626	35,204	13,860	24,584	2,486,370

Classroom Rate Copies

Shape/Ounce	0-0.5	0.5-1.0	1.0-1.5	1.5-2.0	2.0-2.5	2.5-3.0	3.0-3.5	3.5-4.0	4.0-5.0	5.0-6.0	6.0-7.0	7.0-8.0	8.0-9.0	9.0-10.0	10.0-11.0	11.0-12.0	12.0-13.0	13.0-14.0	14.0-15.0	15.0-16.0	> 16.0	Total
Letters	72	28	635	9	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	744
Flats	169,367	132,151	28,562	23,591	11,915	6,023	8,981	7,311	20,422	5,184	5,439	2,495	4,296	175	58	199	61	69	31	0	517	426,845
Parcels	0	0	1	4	16	1	10	3	9	0	16	0	0	0	0	0	0	0	0	0	0	61
Total	169,439	132,179	29,198	23,604	11,931	6,025	8,991	7,313	20,431	5,184	5,455	2,495	4,296	175	58	199	61	69	31	0	517	427,650

Regular Rate Pounds

Shape/Ounce	0-0.5	0.5-1.0	1.0-1.5	1.5-2.0	2.0-2.5	2.5-3.0	3.0-3.5	3.5-4.0	4.0-5.0	5.0-6.0	6.0-7.0	7.0-8.0	8.0-9.0	9.0-10.0	10.0-11.0	11.0-12.0	12.0-13.0	13.0-14.0	14.0-15.0	15.0-16.0	> 16.0	Total
Letters	454	1,623	2,234	2,023	1,104	281	252	88	82	18	3	17	6	1	8	2	3	1	9	37	4	8,248
Flats	477	1,148	5,196	15,230	17,245	31,711	44,670	56,682	167,016	341,403	414,770	313,369	286,203	225,718	239,260	196,732	183,898	169,762	158,749	158,429	774,038	3,801,704
Parcels	4	71	45	28	30	5	4	3	27	17	18	18	20	12	31	19	19	31	22	24	2,489	2,936
Total	935	2,842	7,475	17,280	18,378	31,997	44,926	56,772	167,125	341,438	414,791	313,404	286,229	225,731	239,299	196,753	183,919	169,793	158,780	158,490	776,531	3,812,889

Nonprofit Rate Pounds

Shape/Ounce	0-0.5	0.5-1.0	1.0-1.5	1.5-2.0	2.0-2.5	2.5-3.0	3.0-3.5	3.5-4.0	4.0-5.0	5.0-6.0	6.0-7.0	7.0-8.0	8.0-9.0	9.0-10.0	10.0-11.0	11.0-12.0	12.0-13.0	13.0-14.0	14.0-15.0	15.0-16.0	> 16.0	Total
Letters	675	970	18,277	1,257	950	309	72	27	31	2	3	2	1	0	0	1	0	0	0	0	12	22,590
Flats	628	2,737	13,027	34,405	56,186	25,274	28,183	33,485	47,083	31,004	55,451	56,925	19,022	14,611	18,212	18,995	26,021	21,751	32,047	13,209	33,989	582,245
Parcels	8	26	60	193	68	34	38	101	622	897	442	1,093	0	0	0	116	230	0	0	0	1,488	5,417
Total	1,311	3,733	31,364	35,855	57,204	25,616	28,293	33,614	47,736	31,902	55,895	58,020	19,023	14,612	18,213	19,112	26,251	21,751	32,048	13,209	35,489	610,252

Classroom Rate Pounds

Shape/Ounce	0-0.5	0.5-1.0	1.0-1.5	1.5-2.0	2.0-2.5	2.5-3.0	3.0-3.5	3.5-4.0	4.0-5.0	5.0-6.0	6.0-7.0	7.0-8.0	8.0-9.0	9.0-10.0	10.0-11.0	11.0-12.0	12.0-13.0	13.0-14.0	14.0-15.0	15.0-16.0	> 16.0	Total
Letters	1	1	56	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	59
Flats	3,795	6,091	2,201	2,581	1,647	1,009	1,847	1,707	5,604	1,751	2,193	1,139	2,287	103	37	139	46	60	27	0	789	35,053
Parcels	0	0	0	0	2	0	2	1	3	0	6	0	0	0	0	0	0	0	0	0	0	15
Total	3,796	6,092	2,257	2,583	1,649	1,009	1,849	1,707	5,606	1,751	2,199	1,139	2,287	103	37	139	46	60	27	0	789	35,127

Table 16

Standard Mail Commercial Rate Pieces by Mail Category and Shape, FY 2000
(Thousands)

Mail Category	Letters	Flats	Parcels	Total
BASIC NONAUTOMATION	1,192,320	659,668	189,364	2,041,353
BASIC AUTOMATION	3,744,552	346,136	0	4,090,689
3/5 NONAUTOMATION	2,102,037	1,077,871	492,574	3,672,482
3/5 AUTO FLATS	0	11,752,204	0	11,752,204
5-DIGIT AUTOMATION	9,268,195	0	0	9,268,195
3-DIGIT AUTOMATION	12,065,235	0	0	12,065,235
ECR BASIC	4,230,918	11,873,753	14,185	16,118,856
ECR BASIC AUTO LETTERS	1,975,998	0	0	1,975,998
ECR HIGH DENSITY	421,733	1,524,327	142	1,946,202
ECR SATURATION	3,705,651	9,028,655	637	12,734,943
TOTAL (EXCLUDES PAID AT FIRST AND PRIORITY)	38,706,641	36,262,615	696,902	75,666,158

Table 17

Standard Mail Commercial Rate Pieces by Shape, Ounce Increment, and Rate Group, FY 2000
(Thousands)

	Ounces 0-0.5	0.5-1.0	1.0-1.5	1.5-2.0	2.0-2.5	2.5-3.0	3.0-3.5	3.5-4.0	4.0-5.0	5.0-6.0	6.0-7.0	7.0-8.0	8.0-9.0	9.0-10.0	10.0-11.0	11.0-12.0	12.0-13.0	13.0-14.0	14.0-15.0	15.0-16.0	Total
Auto																					
Letters	8,941.0	11,081.5	2,732.9	1,001.0	503.0	522.4	296.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	25,078.0
Flats	107.4	544.5	933.8	1,076.1	1,026.8	1,383.7	1,591.8	1,272.7	1,718.9	916.1	533.3	301.1	186.0	123.4	112.3	96.1	62.6	50.6	33.1	28.1	12,098.3
Parcels	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Non-Auto (includes Std A pieces at First-Class rates)																					
Letters	1,504.4	1,027.9	476.1	192.4	78.9	52.4	28.7	17.9	14.2	3.9	1.5	0.8	0.6	0.3	0.2	0.2	0.2	0.2	0.1	0.1	3,401.0
Flats	45.7	153.1	179.6	164.3	149.9	150.2	160.1	132.0	188.0	118.4	83.4	68.2	45.6	30.6	25.6	19.4	16.4	12.6	10.1	8.4	1,761.7
Parcels	0.3	1.6	4.2	12.1	23.7	13.4	8.8	9.2	21.2	36.2	41.7	56.3	49.1	59.8	83.7	93.7	109.5	43.8	13.1	10.4	691.9
ECR - Basic																					
Letters	2,114.4	1,223.1	418.1	270.2	115.3	69.4	15.3	1.1	2.1	0.7	0.4	0.3	0.2	0.0	0.2	0.1	0.0	0.0	0.1	0.0	4,230.9
Flats	333.4	921.5	900.7	834.4	966.0	1,204.1	1,530.7	1,373.0	1,795.7	872.9	466.0	229.6	184.5	91.7	47.3	62.7	26.8	16.0	7.1	9.5	11,873.8
Parcels	0.0	0.4	0.0	1.5	0.0	9.1	0.2	0.8	0.2	1.6	0.1	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	14.2
ECR - High Density																					
Letters	161.4	69.0	14.1	46.4	62.5	60.1	3.7	2.2	0.8	0.3	0.3	0.4	0.1	0.1	0.1	0.0	0.0	0.0	0.0	0.0	421.7
Flats	18.0	66.3	119.7	185.2	219.4	184.1	89.8	63.8	164.6	157.9	105.8	63.6	35.6	22.7	11.7	7.5	3.9	2.5	1.4	0.6	1,524.3
Parcels	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1
ECR - Saturation																					
Letters	1,487.2	954.9	361.9	329.1	348.3	141.8	35.5	34.5	10.0	1.6	0.3	0.1	0.2	0.1	0.1	0.0	0.0	0.0	0.0	0.0	3,705.7
Flats	194.2	607.5	848.2	1,368.0	1,903.4	1,110.9	388.1	410.7	1,073.2	671.6	251.1	118.4	43.1	20.3	8.8	4.8	2.9	1.7	1.0	0.8	9,028.7
Parcels	0.0	0.0	0.0	0.1	0.0	0.0	0.1	0.0	0.1	0.0	0.1	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.6
ECR - Auto																					
Letters	858.0	768.6	144.4	54.1	33.1	69.3	48.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1,976.0
Flats	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Parcels	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table 18

Standard Mail Commercial Rate Pounds by Shape, Ounce Increment, and Rate Group, FY 2000
(Thousands)

	Ounces 0-0.5	0.5-1.0	1.0-1.5	1.5-2.0	2.0-2.5	2.5-3.0	3.0-3.5	3.5-4.0	4.0-5.0	5.0-6.0	6.0-7.0	7.0-8.0	8.0-9.0	9.0-10.0	10.0-11.0	11.0-12.0	12.0-13.0	13.0-14.0	14.0-15.0	15.0-16.0	Total
Auto																					
Letters	191.3	483.3	203.1	106.2	69.6	89.5	56.9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1,200.0
Flats	2.7	26.8	74.7	117.2	145.5	241.9	323.1	297.6	479.4	312.1	215.3	140.5	98.4	73.3	73.9	68.8	48.9	42.7	29.9	27.2	2,840.0
Parcels	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Non-Auto (includes Std A pieces at First-Class rates)																					
Letters	28.9	44.7	37.5	20.7	10.8	8.9	5.7	4.2	3.9	1.3	0.6	0.4	0.3	0.2	0.1	0.2	0.2	0.1	0.1	0.1	168.9
Flats	1.1	7.3	14.0	17.8	20.9	25.8	32.5	31.2	53.1	40.9	34.2	32.4	24.5	18.4	17.0	14.1	13.1	10.8	9.3	8.3	426.6
Parcels	0.0	0.1	0.3	1.3	3.3	2.3	1.8	2.2	6.1	12.6	17.2	26.6	26.2	35.8	55.5	67.1	85.9	37.2	12.0	10.3	403.8
ECR																					
Letters	88.2	128.7	70.0	76.1	76.5	57.2	19.9	8.7	3.5	0.9	0.4	0.4	0.2	0.1	0.2	0.1	0.0	0.0	0.1	0.0	531.1
Flats	13.4	76.4	150.9	273.1	461.7	443.2	405.1	430.9	841.1	577.1	328.9	190.8	138.3	79.3	43.9	53.3	26.3	16.9	8.5	10.5	4,569.8
Parcels	0.0	0.0	0.0	0.2	0.0	1.5	0.1	0.2	0.1	0.6	0.1	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.9

Table 19

Standard Nonprofit Mail Pieces by Mail Category and Shape, FY 2000
(Thousands)

Mail Category	Letters	Flats	Parcels	Total
BASIC NONAUTOMATION	829,619	186,964	12,342	1,028,925
BASIC AUTOMATION	1,485,888	78,682	0	1,564,569
3/5 NONAUTOMATION	1,588,670	309,263	7,538	1,905,471
3/5 AUTO FLATS	0	1,336,907	0	1,336,907
5-DIGIT AUTOMATION	1,954,219	0	0	1,954,219
3-DIGIT AUTOMATION	3,535,567	0	0	3,535,567
ECR BASIC	473,247	1,005,329	198	1,478,775
ECR BASIC AUTO LETTERS	298,898	0	0	298,898
ECR HIGH DENSITY	76,236	11,349	112	87,697
ECR SATURATION	710,003	347,415	1,851	1,059,268
TOTAL	10,952,345	3,275,909	22,041	14,250,295

Table 20

Standard Mail Nonprofit Rate Pieces by Shape, Ounce Increment, and Rate Group, FY 2000
(Thousands)

	Ounces																			
	0-0.5	0.5-1.0	1.0-1.5	1.5-2.0	2.0-2.5	2.5-3.0	3.0-3.5	3.5-4.0	4.0-5.0	5.0-6.0	6.0-7.0	7.0-8.0	8.0-9.0	9.0-10.0	10.0-11.0	11.0-12.0	12.0-13.0	13.0-14.0	14.0-15.0	15.0-16.0	Total
Auto																					
Letters	1,701.8	3,432.3	1,323.2	342.7	95.1	63.8	16.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	6,975.7
Flats	14.4	221.4	207.1	212.9	147.6	103.3	118.7	109.9	116.8	61.7	41.7	26.9	12.2	6.0	4.9	3.5	2.1	1.2	1.9	1.3	1,415.6
Parcels	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Non-Auto																					
Letters	1,108.4	926.0	254.4	75.8	25.7	13.6	7.1	2.9	2.2	0.8	0.5	0.3	0.2	0.1	0.1	0.0	0.0	0.0	0.0	0.0	2,418.3
Flats	18.3	100.3	83.1	73.3	43.3	33.6	29.7	26.8	27.1	16.9	15.3	8.4	5.1	4.1	3.0	2.3	1.8	1.6	1.2	1.1	496.2
Parcels	0.0	0.3	0.6	1.1	1.6	0.7	0.4	0.7	1.3	1.6	1.4	1.5	1.3	1.4	1.3	1.1	1.0	1.1	1.1	0.5	19.9
ECR - Basic																					
Letters	185.3	187.0	58.2	22.5	10.1	5.3	3.3	0.4	0.7	0.2	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	473.2
Flats	20.6	500.3	93.8	95.4	72.2	39.4	47.1	39.0	47.5	25.5	11.4	9.3	1.4	0.8	0.8	0.5	0.2	0.1	0.2	0.1	1,005.3
Parcels	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2
ECR - High Density																					
Letters	57.7	12.7	3.5	0.9	0.4	0.2	0.3	0.0	0.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	76.2
Flats	0.8	1.8	2.7	2.0	0.8	0.5	0.3	0.5	1.0	0.3	0.2	0.0	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	11.3
Parcels	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1
ECR - Saturation																					
Letters	372.1	153.6	59.0	55.6	29.5	24.9	10.3	2.6	2.2	0.1	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	710.0
Flats	13.3	65.3	46.8	54.9	36.3	36.0	25.6	18.9	28.0	8.6	6.9	3.1	1.5	1.4	0.7	0.1	0.0	0.0	0.0	0.0	347.4
Parcels	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.9	0.9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.9
ECR - Auto																					
Letters	107.2	130.2	47.8	10.1	1.8	1.3	0.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	298.9
Flats	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Parcels	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table 21

Standard Mail Nonprofit Rate Pounds by Shape, Ounce Increment, and Rate Group, *FY 2000*
(Thousands)

	Ounces																			
	0-0.5	0.5-1.0	1.0-1.5	1.5-2.0	2.0-2.5	2.5-3.0	3.0-3.5	3.5-4.0	4.0-5.0	5.0-6.0	6.0-7.0	7.0-8.0	8.0-9.0	9.0-10.0	10.0-11.0	11.0-12.0	12.0-13.0	13.0-14.0	14.0-15.0	15.0-16.0	Total
Auto																					
Letters	37.2	161.7	94.0	33.0	12.8	10.7	2.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	352.0
Flats	0.4	10.6	16.4	22.5	19.7	18.2	23.9	25.2	33.8	20.8	16.8	12.0	5.7	2.8	2.6	1.9	1.2	0.7	1.2	0.9	237.4
Parcels	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Non-Auto																					
Letters	20.8	40.8	18.1	7.2	3.3	2.1	1.3	0.6	0.6	0.3	0.2	0.1	0.1	0.1	0.0	0.0	0.0	0.0	0.0	0.0	95.7
Flats	0.5	4.9	6.4	7.6	5.7	5.5	5.3	7.3	8.1	5.8	5.5	3.6	2.4	2.0	1.6	1.3	1.0	1.0	0.8	0.7	77.1
Parcels	0.0	0.0	0.0	0.1	0.2	0.1	0.1	0.2	0.4	0.7	0.6	0.7	0.7	0.8	0.8	0.6	0.6	0.8	0.8	0.4	8.8
ECR																					
Letters	13.0	22.1	12.7	8.2	5.2	5.0	2.3	0.8	0.8	0.1	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	70.3
Flats	1.2	24.1	12.1	16.3	13.7	12.5	13.4	13.5	21.2	12.4	8.2	4.8	1.6	1.0	0.8	0.3	0.1	0.1	0.1	0.0	157.5
Parcels	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.4	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.9

Table 22

Standard Mail Commercial Rate
Mail Entry Profile Controlled to FY 2000 PERMIT System Volumes

Pieces (Thousands)

	Trays on Pallets		Loose Trays		Bundles or Sacks on Pallets		Loose Sacks		
Entry Type	Drop Shipped	Plant Load	Drop Shipped	Plant Load	Drop Shipped	Plant Load	Drop Shipped	Plant Load	Total
Origin Delivery Unit	28,668	0	1,793,655	662,054	130	19,937	519,544	26,602	3,050,589
Origin SCF	309,572	56,163	5,016,255	94,962	924,120	52,805	1,086,966	66,190	7,607,031
Origin BMC	1,548,256	1,979,088	646,554	4,289,656	176,577	1,023,819	141,978	1,636,715	11,442,642
Destination BMC	7,240,531	205,242	3,149,482	193,848	5,638,961	67,540	1,465,617	71,856	18,033,079
Destination SCF	6,468,098	59,210	3,928,705	90,328	15,291,673	415,328	811,001	0	27,064,343
Destination Delivery Unit	251,449	0	693,398	1,467	5,896,770	0	928,487	0	7,771,571

Weight (Thousands of Pounds)

	Trays on Pallets		Loose Trays		Bundles or Sacks on Pallets		Loose Sacks		
Entry Type	Drop Shipped	Plant Load	Drop Shipped	Plant Load	Drop Shipped	Plant Load	Drop Shipped	Plant Load	Total
Origin Delivery Unit	1,078	0	77,252	35,796	18	1,789	189,053	2,297	307,283
Origin SCF	10,367	2,255	202,598	2,473	155,879	5,673	261,272	19,274	659,790
Origin BMC	63,297	78,374	16,205	280,597	35,215	291,362	6,714	316,871	1,088,635
Destination BMC	289,455	11,413	261,918	10,797	1,220,874	14,283	296,257	18,382	2,123,379
Destination SCF	290,683	2,415	223,010	1,102	3,358,888	41,114	154,262	0	4,071,474
Destination Delivery Unit	3,255	0	31,715	72	1,257,010	0	185,165	0	1,477,217

Table 23

Standard Mail Nonprofit Rate
Mail Entry Profile Controlled to FY 2000 PERMIT System Volumes

Pieces (Thousands)

	Trays on Pallets		Loose Trays		Bundles or Sacks on Pallets		Loose Sacks		
Entry Type	Drop Shipped	Plant Load	Drop Shipped	Plant Load	Drop Shipped	Plant Load	Drop Shipped	Plant Load	Total
Origin Delivery Unit	160,658	0	602,082	71,710	0	0	57,428	0	891,878
Origin SCF	64,805	2,001	2,207,334	257,057	39,641	0	550,702	50,782	3,172,321
Origin BMC	0	274,161	449,904	2,064,110	36,602	96,402	264,568	276,031	3,461,779
Destination BMC	1,291,813	32,542	734,800	103,913	355,558	10,848	442,929	16,084	2,988,487
Destination SCF	127,795	0	2,094,943	28,988	327,819	0	581,393	1,481	3,162,418
Destination Delivery Unit	0	0	383,730	0	0	0	167,641	0	551,372

Weight (Thousands of Pounds)

	Trays on Pallets		Loose Trays		Bundles or Sacks on Pallets		Loose Sacks		
Entry Type	Drop Shipped	Plant Load	Drop Shipped	Plant Load	Drop Shipped	Plant Load	Drop Shipped	Plant Load	Total
Origin Delivery Unit	6,640	0	24,093	3,576	0	0	4,596	0	38,904
Origin SCF	3,286	92	108,557	12,875	2,902	0	109,876	11,478	249,066
Origin BMC	0	13,467	19,225	108,829	7,237	15,219	51,925	29,675	245,576
Destination BMC	53,364	2,035	49,743	6,139	61,987	3,548	24,742	1,666	203,225
Destination SCF	11,019	0	79,876	731	37,682	0	89,878	170	219,356
Destination Delivery Unit	0	0	14,415	0	0	0	19,365	0	33,779

Table 24

Package Services Pieces by Mail Category by Shape, FY 2000
(Thousands)

Mail Code	Mail Category	Flats	Parcels	Total
4100	STD B INTER-BMC MACH PARCEL POST	1,155	32,546	33,702
4101	AGEN STD B INTER-BMC MACH PARCEL POST	82	2,023	2,105
4102	CONGR FRANK STD B INTER-BMC MACH PARCEL POST	0	0	0
4105	STD B INTRA-BMC PARCEL POST	978	27,564	28,542
4115	STD B BCODE INTER-BMC MACH PARCEL POST	98	2,862	2,959
4120	STD B BCODE INTRA-BMC PARCEL POST	66	1,676	1,741
4127	STD B ORIGIN BMC PRES INTER-BMC MACH PARCEL POST	47	601	648
4130	STD B ORIGIN BMC PRES BCODE INTER-BMC MACH PARCEL POST	22	2,316	2,337
4135	STD B ORIGIN BMC PRES INTER-BMC NONMACH PARCEL POST	0	60	60
4140	STD B BMC PRES INTER-BMC MACH PARCEL POST	22	1,845	1,867
4145	STD B BMC PRES BCODE INTER-BMC MACH PARCEL POST	6	2,590	2,596
4150	STD B INTER-BMC NONMACH PARCEL POST	42	1,215	1,258
4160	STD B DESTINATION BMC PARCEL POST	4	49,205	49,209
4161	AGEN STD B DESTINATION BMC PARCEL POST	0	1	1
4162	CONGR FRANK STD B DESTINATION BMC PARCEL POST	0	0	0
4165	STD B BCODE DESTINATION BMC PARCEL POST	23	152,108	152,131
4170	STD B DESTINATION SCF PARCEL POST	44	4,824	4,868
4175	STD B DESTINATION DELIV UNIT PARCEL POST	5	38,061	38,067
4180	STD B BMC PRES INTER-BMC NONMACH PARCEL POST	1	153	153
27165	STANDARD (B) INTRA-BMC ALASKA BYPASS PARCEL POST	0	1,923	1,923
4200	STD B SP BOUND PRINTED MATTER	11,359	14,114	25,473
4205	STD B SP BCODE BOUND PRINTED MATTER	783	991	1,774
4220	STD B BASIC PRES BOUND PRINTED MATTER	145,202	112,522	257,724
4221	AGEN STD B BASIC PRES BOUND PRINTED MATTER	1,037	992	2,029
4222	CONGR FRANK STD B BASIC PRES BOUND PRINTED MATTER	0	0	0
4227	STD B BASIC PRES BCODE BOUND PRINTED MATTER	8,869	134,296	143,165
4230	STD B CRT BOUND PRINTED MATTER	89,630	40,423	130,052
4300	STD B SP SPECIAL STD	27,673	112,697	140,370
4305	STD B SP BCODE SPECIAL STD	1,461	5,884	7,345
4320	STD B PRES SPECIAL STD	1,453	60,500	61,953
4321	AGEN STD B PRES SPECIAL STD	39	1,714	1,753
4322	CONGR FRANK STD B PRES SPECIAL STD	0	0	0
4327	STD B PRES BCODE SPECIAL STD	21	4,490	4,511
4400	STD B SP LIBRARY MAIL	9,797	17,709	27,506
4401	AGEN STD B SP LIBRARY MAIL	0	0	0
4405	STD B SP BCODE LIBRARY MAIL	46	76	122
4420	STD B PRES LIBRARY MAIL	91	390	481
4427	STD B PRES BCODE LIBRARY MAIL	1	2	3
	TOTAL	300,059	828,372	1,128,431

Appendix A
First-Class Data Editing Algorithm

Data from each transaction are edited at the VIP code level. Consistency of revenue, pieces, and weight for each VIP code in each transaction is checked. The transaction piece weight field and a direct recomputation of the transaction average weight per piece are also used in the editing algorithm. The assignment of VIP elements to data quality groups depends on a set of logical conditions. The following scheme describes these conditions. Each condition is represented by an "If" statement, nested according to the indentations.

I. If Transaction piece Weight is Non-Zero
    If VIP weight is valid
        If rate paid equals rate schedule at weight increment
            Assign to quality group 1
        If rate paid does not equal rate schedule at weight increment
            Discard piece and weight data, retain revenue data for inflation process
    If VIP weight is not valid
        If rate paid equals rate schedule at some weight increment
            Assign to quality group 2
        If rate paid does not equal rate schedule at any weight increment
            Discard piece and weight data, retain revenue data for inflation process

II. If Transaction Piece Weight is Zero
    If rate is valid and an automation rate
        If rate paid equals 1, 2, 3, or 4 ounce rate
            If more than one VIP element in transaction
                Assign to quality group 2
            If only one VIP element in transaction
                If not valid weight
                    Assign to quality group 2
                If valid weight
                    If VIP weight matches transaction average weight per piece
                        Assign to quality group 1
                    If VIP weight does not match transaction average weight per piece
                        Assign to quality group 2
        If rate paid does not equal 1, 2, 3, or 4 ounce rate
            Assign to quality group 3
    If rate is valid and not an automation rate
        If rate matches 1 ounce rate
            If only one VIP element in transaction
                If not valid weight
                    Assign to quality group 2
                If valid weight
                    If VIP weight matches transaction average weight per piece
                        Assign to quality group 1
                    If VIP weight does not match transaction average weight per piece
                        Assign to quality group 2
            If more than one VIP element in transaction
                Assign to quality group 2
        If rate paid does not match 1 ounce rate
            Assign to quality group 3
    If rate is not valid
        Discard piece and weight data, retain revenue data for inflation process

The following rule attempts to reassign VIP elements from quality group 2 to group 1.

III. If Transaction Piece Weight is Zero and Quality Group Equals 2
    If all VIP elements (more than one) in transaction match rate
    at 1, 2, 3, or 4 ounce increment
        Reassign to quality group 1

Appendix B
Program Documentation

Introduction

This appendix describes the computer programs used to produce volume estimates by rate, weight, and shape for First-Class, Priority, Periodicals, Standard, and Package Services Mail.

Each quarter the San Mateo Data Center supplies two tapes containing the PERMIT system transaction records. Data are organized by nodes that are groups of post offices that report their PERMIT data together. Periodicals records are stored in separate files by node from First-Class and Standard mail. The files are in a VAX variable length record format, with padded blocks. This is not compatible with the Data General operating system that processes the data. It is necessary to translate each block of data by stripping off the padding characters.

Records for non-mail transactions (deposits, refunds, etc.) are skipped. Unused information, such as clerk's initials, supervisor's initials, and account balance information, is removed from the record. Each data record contains several packed decimal format fields and several trailing overpunched fields that are converted into ordinary numbers. Reversed transactions are paired with the corresponding original entry and deleted from the files. The files are sorted by accounting period and finance number, all nodes are grouped together, and they are written as thirteen accounting period files for each class of mail: First-Class, Priority, Periodicals, Standard Commercial, Standard Nonprofit, and Package Services.

Each record in the transaction file represents a postage statement. The records are comprised of a fixed length header and a variable length portion. The record header contains transaction information such as finance number, date, permit number, and total transaction revenue, pieces, and weight. The different presort/entry levels within the postage statement form the variable portion of the transaction record. Each level in the mailing is represented by a group of fields in the transaction record, and labeled with a Volume Information Profile (VIP) code to indicate presort/entry level.

Program Listing

All files with ".f" extension are Fortran programs. Files with ".sm" extension are Sort/Merge programs. Files with ".xls" extension are Excel workbooks. Any reference to {ap} below refers to the two digit accounting period from "01" to "13".

For Standard Mail data processing, only the commercial rate (Regular and ECR) programs are documented here. Programs for the nonprofit subclasses of Standard mirror the commercial rate process. Nonprofit data processing programs are supplied in electronic form and are identified by "np" characters in the program names.

Office Stratification Programs
First-Class

mateo_NCTB.f Read in year-to-date Trial Balance data downloaded from San Mateo data center.
- input: all_finnames.dat
  tb00{ap}.dat
- output: NCTB.AP{ap}

diffNCTB.f Create files for each AP containing First-Class stamped, metered, and permit imprint presort revenue by finance numbers.
- input: NCTB.AP{ap}
- output: findata.ap{ap}

make_s41416.f Creates a file with First-Class permit imprint presort revenues by AP by finance number.
input: all_finnames.dat
findata.ap{ap}
output: s41416

pnctbrev.f Creates a file with First-Class permit imprint presort revenue by quarter for PERMIT offices.
input: finstrata.pmt
s41416
output: pnctbstr.dat

tnctbrev.f Creates a file with First-Class permit imprint presort revenue by quarter for all offices.
input: finstrata.new.00
s41416
output: tnctbstr.dat

mergeq.f Rolls up the First-Class stamped and metered estimates for non-PERMIT offices from AP to quarter.
input: finstrata.pmt
finstrata.new.00
nonpbfit.{ap}.rev
output: nonpb.rev

buildconnew.f Creates stratified control revenue files for First-Class stamped and metered presort mail. There is a file created for PERMIT office revenue, and one for PERMIT office revenue plus the estimated stamped and metered revenue from non-PERMIT offices.
input: finstrata.new.00
finrev.{ap}
averev.dat
finrev.1st.{ap}
nonpb.rev
output: trevenue.dat
prevenue.dat
sm_rev.dat

sm_data.f Creates a file with a count by strata of non-PERMIT offices with estimated First-Class revenues.
input: finstrata.new.00
finrev.{ap}
averev.dat
finrev.1st.{ap}
nonpb.rev
output: sm_rev.dat

Summarization of Data
First-Class

perrolln.f — Aggregates and checks the validity of the PERMIT transaction records.

input:
- finstrata.new.00
- vipmap.1st.00
- ratetable.00
- permit.1st.{ap}

output:
- perrolln.1st.{ap}.rpt
- perrolln.1st.{ap}
- perrolln.trans.1st.{ap}
- finrev.1st.{ap}

perhalf.f — Aggregates and checks the validity of identical weight PERMIT transaction records. Maintains detail by half ounce increment up to 4 ounces.

input:
- finstrata.new.00
- vipmap.1st.00
- ratehalf.00
- permit.1st.{ap}

output:
- perhalf.1st.{ap}
- *perhalf.1st.{ap}.rpt*

nrollup.f — Aggregates and checks the validity of the CBCIS records.

input:
- finstrata.new.00
- vipmap.1st.00
- ratetable.00
- avewt.99.dat
- dist.99.dat
- cbc00{ap}.dat

output:
- finrev.{ap}

ave.f — Creates average stamped and metered revenue and average permit imprint revenue by finance number using CBCIS data.

input:
- finstrata.new.00
- finrev.{ap}

output:
- averev.dat

fit.nonp.f — Estimates First-Class stamped and metered presort and non-presort automation revenue by AP for non-PERMIT offices.

input:
- averev.dat
- findata.ap{ap}

output:
- nonpbfit.{ap}.rev

control_shape_half.f Inflates PERMIT data of identical piece transactions using the stratification revenue files. Creates a file by VIP code and ounce increment (including half ounces up to 4 ounces) for revenue, pieces, and weight.

input: trevenue.dat
prevenue.dat
tnctbstr.dat
pnctbstr.dat
newrates.half.00
perhalf.1st.{ap}

output: halfoz00x1.csv

control_shape.f Inflates the PERMIT data using the stratification revenue files. Creates a file by VIP code and full ounce increment for revenue, pieces, and weight. Estimates and distributes weight for unknown weight transactions.

input: trevenue.dat
prevenue.dat
tnctbstr.dat
pnctbstr.dat
ratetable.reclass
perrolln.1st.xx

output: shape00.csv.q4

first_qdist98_detail.xls Distributes pieces and weight to half ounce increment up to 4 ounces and controls to published RPW piece totals.

input: shape98.csv
halfoz98.csv

Job08AX8 SAS job to isolate First-Class records; inflate revenue, pieces, and weight; and assign proper ounce increments, maintaining shape and indicia detail.

input: HSI.RPW.HQ044D01.FY00PQ1
HSI.RPW.HQ044D01.FY00PQ2
HSI.RPW.HQ044D01.FY00PQ3
HSI.RPW.HQ044D01.FY00PQ4

output: H38139.RPW.FY00.FCM.DATA(PQB1)
H38139.RPW.FY00.FCM.DATA(PQB2)
H38139.RPW.FY00.FCM.DATA(PQB3)
H38139.RPW.FY00.FCM.DATA(PQB4)

1SP DPS00.xls Distributes single piece volumes to weight step, shape, and indicia.

first_qdist00_yearshr.xls Distributes presort volumes to weight step and shape.

RPWshape_First.xls Distributes RPW estimates by rate element to shape and indicia.

Summarization of Data
Priority

priority.f Extracts Priority Mail records from PERMIT transaction data.
input: permit.00.other
output: priority.{ap}

rollpriority.f Verifies and rolls up PERMIT system Priority revenue and pieces by processing category and VIP.
input: vipmap.pri.00
priority.{ap}
output: priroll.00

Priority by shp00 LR.xls Distributes volumes to shape.

Initial Data Processing Programs
Periodicals

reverser UNIX shell script that accesses the following input files and 5 programs to remove the reversed transactions from the data. This program is run using data from all 4 quarters.

input: nodes.org – list of all the nodes used in the year.
This file is created with the following UNIX commands:
cat Rev.Fin.FY00.q* > rev.all &
cut –c1-6 rev.all | sort –u > nodes.org &

gzcat UNIX command to unzip the data files. It removes the extension 'gz' from the filename.
input: q{q}.<nodename>.data.gz, where {q} is the Postal quarter, 1 to 4.
output: unpacked.s

sortpub.sm Sorts the data by rate type, finance number, publication number, and revenue.
input: unpacked.s
output: unpacked

chkforzero.f Checks data for zeros and removes useless records.
input: unpacked
output: none – records are removed only if they are bad. Good records are funneled directly into the next program, reverse_v2.f

reverse_v2.f Checks the data for reverse transactions and attempts to match to original transactions.
input: unpacked
output: goodtrans – this file is renamed S.reversed.<nodename> in the UNIX shell script

gzip UNIX command that zips up the data files.
input: S.reversed.<nodename>
output: S.reversed.<nodename>.gz

sepdata.f Reads in all the reversed data and writes out 2 files, one containing the data from prior to the 1999 rate change, and one containing data following the rate change.

gzcat UNIX command to unzip and combine all the reverse files into one large file.
Input: S.reversed.<nodename).gz
Output: S.reversed.all

Input: seccon.950101.950931
seccon.960701.961005
seccon.961006.971004
seccon.971005.present – these 4 files contain rates for each VIP code and are used to check rate consistency in the data. Rates are valid for transactions falling within the dates in the filename. The third file is valid through the end of FY00. While no FY 2000 transactions use rates in the first file, this file also includes flags used to categorize VIP codes as pertaining to pieces or pounds, or whether revenue should be positive or negative (discounts).
Output: Postchange.data – File containing only the PERMIT transactions following the rate change.
Prechange.data – File containing only the PERMIT transactions prior to the rate change.

bin2nd2.f Reads all of the S.reversed.<nodename> files, checks the records for consistency by calling checktrn.f, and writes the records out to one data set for records prior to the 1999 rate change.

subroutine: checktrn.f Checks for consistency in the data passed through from bin2nd.f. It compares rates with multiple rate tables used as check files, and removes data if an inconsistency exists. Information on the inconsistent transactions is written to the standard output file (bin2nd.out).
jul_to_txt.f a subroutine that converts a Julian date to the "yymmdd" format.
leap.f a logical function called from jul_to_txt.f that identifies leap years.

input: prechange.data – File containing only the PERMIT transactions prior to the rate change.
fin.map – list of all the finance numbers used during the year. This file is created using the following UNIX commands:
cat Rev.Fin.FY00.q* > rev.all &
cut –c7-12 rev.all | sort –u > fin.map &
seccon.950101.950931
seccon.960701.961005
seccon.961006.971004
seccon.971005.present – these 4 files contain rates for each VIP code and are used to check rate consistency in the data. Rates are valid for transactions falling within the dates in the filename. The third file is valid through the end of FY00. While no FY 2000 transactions use rates in the first file, this file also includes flags used to categorize VIP codes as pertaining to pieces or pounds, or whether revenue should be positive or negative (discounts).

output: olddata.data – File of FY00 PERMIT system periodical data for transactions prior to the 1999 rate change.

bin2nd2_new.f Reads all of the S.reversed.<nodename> files, checks the records for consistency by calling checktrn.f, and writes the records out to one data set for records following the 1999 rate change.

subroutine: checktrn.f Checks for consistency in the data passed through from bin2nd.f. It compares rates with multiple rate tables used as check files, and removes data if an inconsistency exists. Information on the inconsistent transactions is written to the standard output file (bin2nd.out).

jul_to_txt.f a subroutine that converts a Julian date to the "yymmdd" format.

leap.f a logical function called from jul_to_txt.f that identifies leap years.

input: postchange.data – File containing only the PERMIT transactions following the rate change.

fin.map – list of all the finance numbers used during the year. This file is created using the following UNIX commands:

cat Rev.Fin.FY00.q* > rev.all &

cut –c7-12 rev.all | sort –u > fin.map &

seccon.950101.950931

seccon.960701.961005

seccon.961006.971004

seccon.971005.present – these 4 files contain rates for each VIP code and are used to check rates consistency in the data. Rates are valid for transactions falling within the dates in the filename. The third file is valid through the end of FY00. While no FY 2000 transactions use rates in the first file, this file also includes flags used to categorize VIP codes as pertaining to pieces or pounds, or whether revenue should be positive or negative (discounts).

output: newdata.data – File of FY00 PERMIT system periodical data for transactions following the 1999 rate change.

Office Stratification Programs
Periodicals and Standard

doextract Korn shell script containing the following program:

extract.sm Extracts data for accounts.
input: Trial Balance tape
output: nctbext.00{ap} – extract of Trial Balance accounts

mapfin.sm Sort/Merge program that creates list of Trial Balance finance numbers
input: nctbext.00{ap}
output: mapfin

revaccts_byap.q4.f Converts data from binary format and breaks out revenue by AP.
input: nctbext.00{ap}
mapfin – list of Trial Balance finance numbers
output: A{ac}.txt - where ac are relevant Trial Balance account numbers

dostrata Korn shell script containing the following program:

strata_00.f groups offices into 20 strata by revenue size.
input: A{ac}.txt
output: strata.{ac} – finance numbers stratified by permit imprint revenue

Office Stratification Programs
Periodicals

Strata_99_22.f Sums the total FY00 Trial Balance outside county revenue by stratum and quarter; divides the revenue into PERMIT and non-PERMIT site totals; divides the offices into 22 strata with the first 18 each containing 5 percent of revenue and the last 4 each containing 2.5 percent of revenue.
input: strata41310 – Original Trial Balance data for account number 41310.
output: strata41310.n22 – Trial Balance data with its strata number.

Strata_99_22_41316.f Sums the total FY00 Trial Balance inside county revenue by stratum and quarter; divides the revenue into PERMIT and non-PERMIT site totals; divides the offices into 22 strata with the first 18 each containing 5 percent of revenue and the last 4 each containing 2.5 percent of revenue.
input: strata41316 - Original Trial Balance data for account number 41316.
output: strata41316.n22 – Trial Balance data with strata number.

fixmap.sm Sorts the outside county revenue file by finance number.
Input: strata41310.n22 – Trial Balance data sorted by revenue
Output: strata41310.n22.srt – Trial Balance data sorted by Finance Number.

Fix41316.sm Sorts the In-County revenue file by finance number.
Input: strata41316.n22 – Trial Balance data sorted by revenue.
Output: strata41316.n22.srt – Trial Balance data sorted by Finance Number.

mkstr.f Rolls up data for the outside county Trial Balance to quarterly and annual totals.
input: strata.41310.n22.srt – Trial Balance data sorted by Finance Number.
fin.map
output: key00_oc.2nd – Quarterly and yearly revenue totals by finance number.

mkstr41316.f Rolls up data for the Inside county Trial Balance to quarterly and annual totals.
input: strata.41316.n22.srt – Trial Balance data sorted by Finance Number.
fin.map
output: key00_ic.2nd – Quarterly and yearly revenue totals by finance number.

mkmap.f Makes a map of each finance number with its inside county and outside county strata.
Input: strata.41316.n22.srt – Trial Balance data sorted by Finance Number for In county.
Strata.41310.n22.srt – Trial Balance data sorted by Finance Number for Outside County.
Fin.map
Output: fin_strata.map – Map containing each finance number and its inside county and outside county strata.

Summarization of Data
Periodicals

sepgov.f Creates a file of government revenue by finance number. This is needed to adjust the inflation process. Government revenues are reported in RPW, but not in the Trial Balance revenues.

input: newdata.data
fin.map
seccon.9501010.950931
olddata.data

output: gov_ic.rev
gov_oc.rev
nongov_ic.rev
nongov_oc.rev

control.f Reads PERMIT revenue file, sums revenue by stratum, and calculates inflation factors.

input: Fin_strata.map
key00_ic.2nd
key00_oc.2nd
gov_oc.rev
nongov_oc.rev
nongov_ic.rev
strata.41320 – Account number for CPP Trial Balance revenues.

output: Control.in.NCTB.NonCPP
Control.out.NCTB.NonCPP
Control.Matrix.NCTB.CPP
Control.Matrix.RPW

Roll2nd_99_inf_weight.f Rolls up the Periodicals PERMIT data by weight increment, VIP code, subclass, and shape.

input: fin_strata.map
seccon.950101.950931
Control.in.NCTB.NonCPP
Control.out.NCTB.NonCPP
Control.Matrix.RPW
Control.Matrix.NCTB.CPP
Newdata.data
Olddata.data
Pubs.map – Map of Publication Numbers with their names.

output: Rolled3.inf.00.weight – PERMIT data rolled up by VIP code, subclass and shape
Rev3.chk.wei.weight – Finance Numbers with their total revenue.

roll_bill_wgtinc.f Rolls up over VIP codes in previous program's results to presort level, subclass, weight increment, and shape.

input: Rolled3.inf.00.weight
seccon.950101.950931

output: pcinfo_we – Revenue, Pieces, Weight, and copies by presort level, subclass, weight, increment, and shape from the piece portion of the postage statement.
wtinfo_we – Revenue, Pieces, Weight, and copies by presort level, subclass, weight increment, and shape from the weight portion of the postage statement.
disinfo_we – Revenue from the discount portion of the postage statement.

cps_wgt_inc_LR.xls Distributes copies to weight increment and shape.

Detail_shape_00_LR.xls Distributes RPW estimates to subclass, presort level, and shape.

wgt_wgt_inc_LR.xls Distributes weight to weight increment and shape.

Office Stratification Programs
Standard Mail

pmtstrata.f Creates strata map for PERMIT roll-up.
input: finno.pmt - map of PERMIT finance numbers sorted version of fins.pmt.00
strata.41411
strata.41414
strata.41416
output: finstrata.00 - PERMIT finance numbers and stratum for each class

revcov.f Compares PERMIT and Trial Balance Revenues and creates maps for further processing.
input: finstrata.00
PIrevbyacct.00.ye
strata.41411
strata.41414
strata.41416
output: to_be_fit.41411 – map of finance number and an indicator for permit status for Regular and ECR Standard
to_be_fit.41414 – map of finance number and an indicator for permit status for Nonprofit and Nonprofit ECR Standard
finsbyap.pmt.00 – map of PERMIT finance numbers and indicators for activity in permit by accounting period

Summarization of Data
Standard Mail

stdadisk CSHELL that executes the following program

stda_roll.f FORTRAN program that checks data for internal consistency.
subroutine: check_stda.f
input: vip_stda99.new - map of VIP codes and rates
finstrata.00 - map of PERMIT finance numbers and strata
error.codes - print titles for transaction error codes
permit.00.stda.{ap} – PERMIT transaction files
output: permit.new.stda.{ap} – Cleaned PERMIT transaction database.

wgt_std_roll2.f Aggregates transaction database by VIP, weight increment, stratum, and shape.
include: permit_stda.h
permit_read_stda.h
input: permit.new.stda.{ap}
vip_stda99.new
finstrata.00
ouput: pmt_stda.wi.{ap} – aggregated PERMIT transaction data by VIP, stratum, and shape.

fitdat.f Organizes Trial Balance data for further processing.

input: revfile – strata.41411 sorted by finance number
tb00{ap}.dat – Stamped and Metered revenue by finance number extracted form Trial Balance tapes.

output: nctb_fit_ye.00 – Trial Balance revenue for permit imprint Standard, stamped revenue and metered revenue by finance number.

fit_stda.f Applies regression parameters to generate estimated stamped and metered Standard revenue for non-PERMIT offices by stratum.

input: to_be_fit.41411 – list of finance numbers and an indicator of PERMIT system status.
nctb_fit_ye.00 – Trial Balance revenue for permit imprint Standard, stamped revenue and metered revenue by finance number.

output: fit_stda_ye.00 – estimated stamped and metered revenues for non-PERMIT offices by stratum and quarter.

eststda_20.f Applies revenue controls and produces estimates of pieces by rate element, shape, and weight increment.

input: vip_stda99.new - map of VIP codes and rates
finsbyap.pmt.00 - map of PERMIT finance numbers and accounting periods they were active in PERMIT
pmt_stda.wi.{ap} – PERMIT Standard volumes, revenues and weight by VIP code, stratum, transaction type, weight increment and shape
strata.41411 - Trial Balance Standard permit imprint revenue and finance numbers stratified by revenue
fit_stda_ye.00 - estimated stamped and metered revenues for non-PERMIT offices by stratum and quarter.

output: stda.csv

weststda_20.f Applies revenue controls and produces estimates of pieces by rate element, shape, and weight increment.

input: vip_stda99.new – map of VIP codes and rates
finsbyap.pmt.00 – map of PERMIT finance numbers and accounting periods they were active in PERMIT
pmt_stda.wi.{ap} – PERMIT Standard volumes, revenues, and weight by VIP code, stratum, transaction type, weight increment and shape
strata.41411 - Trial Balance Standard permit imprint revenue and finance numbers stratified by revenue
fit_stda_ye.00 -- estimated stamped and metered revenues for non-PERMIT offices by stratum and quarter.

output: wtstda.csv

stda98c.xls Controls output of previous program to published RPW piece and weight totals.

Rmepp.xls Controls 1996 Standard mail characteristics survey (mail entry point profile) results to FY 1998 entry discount volumes.

links: stda98c.xls

RPW_Shape_{pq}.xls Distributes RPW estimates to shape for each Postal quarter {pq}.

RPW_Shape_{pq}_ounce.xls	Distributes RPW piece estimates to ounce increment for each Postal quarter {pq}.
RPW_Shape_{pq}_ounce_w.xls	Distributes RPW weight estimates to ounce increment for each Postal quarter {pq}.
RPW_Shape_PFY_GFY.xls	Combines quarterly RPW estimates by shape to compute PFY and GFY RPW estimates by shape.
RPW_Shape_PFY_GFY_ounce.xls	Combines quarterly RPW piece estimates by ounce increment to compute PFY and GFY RPW piece estimates by ounce increment.
RPW_Shape_PFY_GFY_ounce_w.xls	Combines quarterly RPW weight estimates by ounce increment to compute PFY and GFY RPW weight estimates by ounce increment.

Package Services Programs

Permit5.f — This program reads in the raw PERMIT data and writes it to a file with one line for each line of VIP detail. This program is run for each quarter.
- Input: datafile{quarter} – PERMIT data for quarter {quarter}. Where {quarter} = {01.02.03,04}
- Output: dataq{quarter}_4 – Valid PERMIT transactions for {quarter} with each VIP as a separate line.

Roll_vip.f — This program rolls up the PERMIT data by quarter, shape, and VIP code.
- Input: dataq{quarter} _4
  Vipmap – List of all the VIP codes used in 2000 Standard B.
- Output: viprpw_4 – PERMIT data summarized by quarter, shape, and VIP code.

Odis.f — Summarizes ODIS volume data by class, subclass, shape, and indicia.
- Input: ODIS-560 data
  subclassmap
- Output: hardcopy.{qfy}

stdb_{pq}2000.xls	Distributes RPW estimates by shape for Postal quarter {pq}.
stdb_GFY_2000.xls	Combines quarterly RPW estimates by shape to compute GFY estimates by shape.

Program Source Code Files and Input Data

Program source code files and miscellaneous input files are included on the CD-ROM disk accompanying this library reference. The source PERMIT transaction files and Trial Balance files are not included. Instead, prepared PERMIT and Trial Balance files are included which redact the identity of individual mailers and post offices. In addition, throughout this library reference, data files that identify individual finance numbers are redacted by substituting a random number for the finance number. This allows the data processing to be replicated while protecting the identity of individual post offices and mailers.

PERMIT files are supplied in a compressed format. These files can be uncompressed with the Microsoft "WinZip" utility. The PERMIT files supplied on the CD-ROM disk for each class of mail are:

First-Class: FC_Permit_data.zip
This compressed file contains PERMIT data in 13 files, one for each AP of FY 2000.

These files are named permit.1st.{ap}, where {ap} represents the two-digit accounting period number from 01 to 13. These files are created by program permitbyap.f.

Priority: permit.other.00

This compressed file contains PERMIT data for each AP of FY 2000. These files are created by program permitbyap.f.

Periodicals: data.adjusted.zip

This compressed file is created by program extract.f and contains PERMIT data for all of FY 2000. Some programs use the binary form of this file as input. (See bin2nd.data above.)

Regular and ECR Standard: Reg_Permit_data.zip

This compressed file contains PERMIT data in 13 files, one for each AP of FY 2000. These files are named permit.stda.{ap}, where {ap} represents the two-digit accounting period number from 01 to 13. These files are created by program permitbyap.f.

Nonprofit and Nonprofit ECR Standard: NP_Permit_data.zip

This compressed file contains PERMIT data in 13 files, one for each AP of FY 2000. These files are named permit.stdanp.{ap}, where {ap} represents the two-digit accounting period number from 01 to 13. These files are created by program permitbyap.f.

Package Services

These files are created by program permitbyap.f.

Appendix C
Program Listings

Office Stratification Programs
First-Class

Summarization of Data
First-Class

Summarization of Data
Priority

Initial Data Processing Programs
Periodicals

Office Stratification Programs
Periodicals and Standard

Office Stratification Programs
Periodicals

Summarization of Data
Periodicals

Office Stratification Programs
Standard

Summarization of Data
Standard

Summarization of Data Package services

```
      PROGRAM mateo_NCTB

C     DESCRIPTION: Read extract of AP NCTB by Finance Number tape from
C                  San Mateo.  Store all 1C PPI, Stamped, and Metered Revenue
C                  by Finno
C
C
C     CREATED BY:        ta
C     DATE:              12-12-95
C
C     LAST MODIFIED BY: ta
C     DATE:              10-16-97;  update for 98
C                         YTD PI as of ap6. No longer need the add on - see code.
C                        10-06-98; update for 99

      IMPLICIT NONE

C       **** Parameters ****
      integer*4   nfin
      parameter   (nfin = 28946)            ! update

C       **** Input variables ****
      integer*4       ier       ! Error code
      character*6     fin       ! Finance number from input
      real*8          acct411x, acct412x, acct41416x
      integer*4       countrec
      character*6     finx(nfin)

C       **** Output variables ****

      real*8           acct411(1:nfin)/nfin*0.0/
      real*8           acct412(1:nfin)/nfin*0.0/
      real*8           acct41416(1:nfin)/nfin*0.0/
      real*8           tot411, tot412, tot41416

C       **** More variables ****

      integer*4   i, cnt, ifin,iap
      integer*4   searchc
      character*2 cap

C     Read AP from command line

      CALL getarg(1,cap)
      READ(cap,'(I2)') iap

      OPEN(40,file='all_finnames.dat')
41    format(a6)

      ier=0
      cnt = 0
      do while(ier.eq.0)
         do i = 1, nfin
            read(40,41,iostat=ier,end=42)  finx(i)
            cnt = cnt + 1
         end do
      end do
42    print *,'finished reading finnames.dat  ier= ',ier
      print*,'read ',cnt,' lines'


C OPEN data INPUT FILE:


      open(60,file='tb00'//cap//'.dat',organization='line')
61    format(a6,3f14.2)     ! check format

      ier = 0
      tot411 = 0.0
      tot412 = 0.0
      tot41416 = 0.0
      countrec = 1

      do while (ier.eq.0)
            read(60,61,iostat=ier,end=220) fin,acct41416x,acct411x,
     +          acct412x

            if (fin(1:1).eq.' ')  fin(1:1) = '0'
            ifin=searchc(finx,nfin,fin)
            if (ifin.gt.0) then
```

```
            acct411(ifin) = acct411x
            acct412(ifin) = acct412x
            acct41416(ifin) = acct41416x

            tot411 = tot411 + acct411x
            tot412 = tot412 + acct412x
            tot41416 = tot41416 + acct41416x

         else
            print *, 'FINANCE NUMBER NOT FOUND ',fin
            write(*,'(3f15.2)') acct411x, acct412x, acct41416x
         end if
      countrec = countrec + 1
   end do

220 print *,' finished reading file  ier = ', ier
   print *,' record count is ', countrec
   print*,' tot411 = ',tot411
   print*,' tot412 = ',tot412
   print*,' tot41416 = ',tot41416


   open(70,file='NCTB.AP'//cap)
71 format(a6,3f15.2)

   do i = 1, nfin
      write(70,71) finx(i), acct411(i), acct412(i), acct41416(i)
   end do

   STOP
   END
```

```
      PROGRAM diffNCTB

C     DESCRIPTION: Take differences between AP NCTB files.
C
C     CREATED BY:        Bill Humphries
C     DATE:              Thu Mar 30 10:03:44 CST 1995
C
C     LAST MODIFIED BY: Bill Humphries
C     DATE:              Mon Oct  2 14:38:23 CDT 1995
C
C        ** NOTE; if any current fins are not found they must be added to
C      the previous NCTB data file.


      IMPLICIT NONE

      INTEGER*4 nap,nfin
      PARAMETER (nap=13,nfin=28946)      ! lines in all_finnames.dat

      INTEGER*4 ifin,ier,iap,i,j,found,ncur,nprv,searchc
      INTEGER*4 zerofin(nfin)
      REAL*8    currentap(nfin,3),prevap(nfin,3)
      REAL*8    data(nfin,3)
      CHARACTER*6 currentfin(nfin),prevfin(nfin),finno
      CHARACTER*2 cap
      CHARACTER*2 aplist(nap)/'01','02','03','04','05','06','07','08',
     +   '09','10','11','12','13'/

      do i = 1,nfin
         zerofin(i) = 0
         do j = 1,3
            data(i,j) = 0.0
         end do
      end do

      CALL getarg(1,cap)
      READ(cap,'(I2)') iap

      IF (iap.eq.1) THEN
         PRINT *, "You do not need to take differences in AP01."
         STOP
      ELSE
         PRINT *, "Current : AP",cap
         PRINT *, "Previous: AP",aplist(iap-1)
         PRINT *
      END IF

C     Read current and previous AP's NCTB

      OPEN(10,FILE='NCTB.AP'//cap)
      OPEN(20,FILE='NCTB.AP'//aplist(iap-1))

11    FORMAT(A6,3F15.2)

      ier = 0
      i = 1
      DO WHILE (ier.eq.0)
         READ(10,11,IOSTAT=ier,END=12) currentfin(i),(currentap(i,j),j=1,3)
         i = i + 1
      END DO
12    PRINT *, "Finished reading NCTB.AP",cap," with ier = ",ier
      ncur = i - 1
      PRINT *, "Read ",ncur," Finance Numbers"

      ier = 0
      i = 1
      DO WHILE (ier.eq.0)
         READ(20,11,IOSTAT=ier,END=22) prevfin(i),(prevap(i,j),j=1,3)
         i = i + 1
      END DO
22    PRINT *, "Finished reading NCTB.AP",aplist(iap-1)," with ier = ",ier
      nprv = i - 1
      PRINT *, "Read ",nprv," Finance Numbers"

C     Loop through current AP and take first differences

      found = 0

      DO i=1,ncur
         finno = currentfin(i)
```

```
      ifin = searchc(prevfin,nprv,finno)
      IF (ifin.le.0) THEN          ! current fin is new
         PRINT *, "Can't find finance number ",currentfin(i),
     +      " in ap ",aplist(iap-1)," ;add to previous ap file."
      ELSE
         DO j=1,3
            IF (currentap(i,j).eq.0.and.prevap(ifin,j).ne.0) THEN
               zerofin(i) = 1
            END IF
            data(i,j) = currentap(i,j) - prevap(ifin,j)
         END DO
         found = found + 1
      END IF
   END DO

   PRINT *, "Found ",found," of ",ncur," finance numbers."

C  Write differenced data

   OPEN(30,FILE='findata.ap'//cap)
   OPEN(40,FILE='zerofin.ap'//cap)

   DO i=1,ncur
      WRITE(30,11) currentfin(i),(data(i,j),j=1,3)
      IF (zerofin(i).eq.1) THEN
         WRITE(40,11) currentfin(i),(currentap(i,j),j=1,3)
      END IF
   END DO

   PRINT *, "Wrote differenced file."

   STOP
   END
```

```
      PROGRAM make_s41416
C
C       PURPOSE: puts NCTB PI presort rev for each ap into one file
C
C       AUTHOR: MM
C       DATE   : 15 JUL 93
C       MODIFIED : amr 10/3/94 for 94 processing
C                  ta  11/14/95 for 96 processing
C                  ta  09/13/96 - read in all_finnames.dat instead of IMF
C                  ta  10/30/96 - update for 97
C                  ta  01/15/98 - update for FY98
C                  ta  01/25.99 - update for FY99
C                  ts  02/01/00 - update for FY00
C                  ts  11/16/00 - update for new compiler


      IMPLICIT NONE

      integer*4   nfins, naps
      parameter   (nfins=28946)  ! fins in all_finnames
      parameter   (naps=13)       ! ytd aps

      integer*4   ind, inx, fin(1:nfins), ier, i,j, totfin, ap, loc
      integer*4   finx, acctx, finpos, totact
      integer*4   searchi, act, inz, a, f,nmap,newfins
      integer*4   foundfins/0/

      real*8      rev(1:13,1:nfins)
      real*8      revamt
      real*8      revtot(1:nfins)
      integer*4   map(1:13000),smap(1:13000)

      character*2 aps(1:13), apx


C  initialize arrays to 0.

      do j = 1, nfins
         revtot(j) = 0.0
         do i = 1,13
            rev(i,j) = 0.0
         end do
      end do

C  READ MAP OF ALL FINANCE NUMBERS   update

      open(16,file='all_finnames.dat',readonly)
17    format(i6)

      ind = 1
      ier = 0
      do while (ier.eq.0)
         read(16,17,iostat=ier,end=101) fin(ind)
         ind = ind + 1
      end do

101   totfin = ind-1
      print*, 'Read  fins file: ',totfin ,'lines'
      print*, 'check is ',totfin - nfins
      print*, 'exit code read as  ', ier


C  WRITE OUT DATA FOR EACH AP

C  set up arrays

      aps(1) = '01'
      aps(2) = '02'
      aps(3) = '03'
      aps(4) = '04'
      aps(5) = '05'
      aps(6) = '06'
      aps(7) = '07'
      aps(8) = '08'
      aps(9) = '09'
      aps(10) = '10'
      aps(11) = '11'
      aps(12) = '12'
      aps(13) = '13'
```

```
C       open NCTBEXT files by ap

      ap = 0
      inx = 0

61    format(I6,30X,F15.2)

      do i = 1,naps
         apx = aps(i)
         ap = ap + 1
C        This file based on all_finnames
         open(60,file='findata.ap'//apx,readonly)
         ier = 0
         do while (ier.eq.0)
               read(60,61,iostat=ier) finx,revamt
               inx = searchi(fin,totfin,finx)
               if (inx.gt.0) then
                  f=inx
                  rev(ap,f) = -revamt
               else
                  print*,'fin not found ',finx,' rev is ',revamt
               end if
         end do
         print*,' read exit code =',ier
         close(60)
      end do

C     Sum over the amount of revenue for each AP.

      do finpos = 1,totfin
         do ap=1, naps
               revtot(finpos)= revtot(finpos) + rev(ap,finpos)
         end do
      end do

C  Now write the data to file

      open(47,file='s41416')
41    format(i6.6,3x,f15.2,13(f12.2))

      do loc = 1,totfin
         if (revtot(loc).gt.0) then
            write(47,41) fin(loc),revtot(loc),
     +         (rev(ap,loc),ap=1,13)
            foundfins = foundfins + 1
         end if
      end do

      print*,'fins with rev ', foundfins

      close(47)
      end

C     ------------------------------------------------------------------------
```

```
      program pnctbrev

C     Date: DATE
C     Programmer: mcb.weh
C
C     Last Modified: 10/30/96  ta  ; 97 update
C
C     Purpose: Calculate NCTB revenue for PERMIT offices by strata.
C              Change the map of finance numbers.

      IMPLICIT NONE

      INTEGER*4 nstrata,nap,nelm,nbfin,npfin,nbad,nq

      PARAMETER (nstrata=20,nap=13,npfin=2203,nq=4) ! update npfin
      PARAMETER (nelm=3*nstrata*nq)

      REAL*8 tabrev(3,nstrata,nq)/nelm*0.0/

      CHARACTER*6 pfins(npfin)
      INTEGER*4 searchc,tstrata(npfin)

      INTEGER*4 noffice(3,nstrata)/60*0/

      REAL*8 rev(nap)/nap*0.0/
      INTEGER*4 strata, check

      INTEGER*4 nlst,nsites
      INTEGER*4 apq(nap)/1,1,1,2,2,2,3,3,3,4,4,4,4/

      INTEGER*4 i,j,k,ier,idx, nrec

      CHARACTER*6 finno

C     Read the list of permit sites.

      OPEN(10,FILE='finstrata.pmt',readonly)
      ier = 0
      nsites = 1
      DO WHILE (ier.EQ.0)
         READ(10,'(A6,I4)',IOSTAT=ier,END=19) pfins(nsites),tstrata(nsites)
         nsites = nsites + 1
      END DO
 19   nsites = nsites - 1
      check = nsites - npfin
      print*,'check is ', check
      PRINT *, 'Read ',nsites,' offices. ier =', ier


C     Read Strata lists
 21   format(a6,i3,15x,13f12.2)
      OPEN(20,FILE='s41416',readonly)   ! NCTB PI data
      ier = 0
      nlst = 1
      DO WHILE (ier.EQ.0)
         READ(20,21,IOSTAT=ier,END=27) finno, strata, rev    ! 13 aps
         idx = searchc(pfins,npfin,finno)
         IF (idx.GT.0) THEN
            strata = tstrata(idx)
            do k=1,13
               tabrev(1,strata,apq(k))=tabrev(1,strata,apq(k))+rev(k)
            end do
            noffice(1,strata)=noffice(1,strata)+1
           nlst = nlst+1
         END IF
      END DO
 27   nlst = nlst - 1
      PRINT *, 'Read ',nlst,' valid PERMIT 1st Class PI sites. ier = ', ier
      CLOSE(20)

C     Write report

      OPEN(40,FILE='pnctbstr.dat')
 44   FORMAT(4F15.0,I4)
      DO j=1,1
         DO strata = 1, 20
            WRITE(40,44) (tabrev(j,strata,k),k=1,nq),noffice(j,strata)
         END DO
      END DO
```

```
      END
C-----------------------------------------------------------------------
```

```
      program tnctbrev

C     Date: DATE
C     Programmer: mcb.weh
C
C     Last Modified: 10/30/96  ta
C                    08/22/97  ta; place unmapped fins in 20
C
C     Purpose: Calculate NCTB PI presort revenue by strata.


      IMPLICIT NONE

      INTEGER*4 nstrata,nap,nelm,ntfin,nbfin,npfin,nbad,nq

      PARAMETER (nstrata=20,nap=13,ntfin=10815,nq=4) ! update ntfin
      PARAMETER (nelm=3*nstrata*nq)

      REAL*8 tabrev(3,nstrata,nq)/nelm*0.0/
      REAL*8 totrev
      REAL*8 unmaprev/0.0/

      CHARACTER*6 tfins(ntfin)
      INTEGER*4 searchc,tstrata(ntfin), check

      INTEGER*4 noffice(3,nstrata)/60*0/
      INTEGER*4 unmap/0/

      REAL*8 rev(nap)/nap*0.0/
      real*8 total(4)/4*0.0/

      INTEGER*4 strata,finstrata(ntfin)

      INTEGER*4 nlst,nsites,nmaps
      INTEGER*4 apq(nap)/1,1,1,2,2,2,3,3,3,4,4,4,4/

      INTEGER*4 i,j,k,ier,idx, nrec

      CHARACTER*6 finno

C     Read the list of PI sites.

      OPEN(10,FILE='finstrata.new.00',readonly)   ! update
      ier = 0
      nsites = 1
      DO WHILE (ier.EQ.0)
         READ(10,'(A6,I4)',IOSTAT=ier,END=19) tfins(nsites),tstrata(nsites)
         nsites = nsites + 1
      END DO
 19   nsites = nsites - 1
      check = nsites - ntfin
      PRINT *, 'Read ',nsites,' offices. ier =', ier

C     Read Strata lists
 21   format(a6,f18.2,13f12.2)
      OPEN(20,FILE='s41416',readonly)   ! NCTB PI data
      ier = 0
      nlst = 1
      DO WHILE (ier.EQ.0)
         READ(20,21,IOSTAT=ier,END=27) finno, totrev, rev   ! 13 aps
         idx = searchc(tfins,ntfin,finno)
         IF (idx.GT.0) THEN
            strata = tstrata(idx)
            do k=1,13
               tabrev(1,strata,apq(k))=tabrev(1,strata,apq(k))+rev(k)
            end do
            noffice(1,strata)=noffice(1,strata)+1
           nlst = nlst+1
         ELSE        ! put unmapped fins in strata 20
            IF (totrev.gt.0) then
               strata = 20
               do k=1,13
                  tabrev(1,strata,apq(k))=tabrev(1,strata,apq(k))+rev(k)
               end do
               noffice(1,strata)=noffice(1,strata)+1
               nlst = nlst+1
               unmap = unmap + 1
               unmaprev = unmaprev + totrev
            END IF
         END IF
```

```
      END DO
 27   nlst = nlst - 1
      PRINT *, 'Read ',nlst,' valid 1st Class PI sites. ier = ', ier
      print *, 'unmapped fins ', unmap,' with rev of ',unmaprev
      CLOSE(20)

C     Write report

      OPEN(40,FILE='tnctbstr.dat')
 44   FORMAT(4F15.0,I4)
      DO j=1,1
         DO strata = 1, 20
            WRITE(40,44) (tabrev(j,strata,k),k=1,nq),noffice(j,strata)
         END DO
      END DO

      DO j=1,1
         DO strata = 1, 20
            do k = 1,4
               total(k) = total(k) + tabrev(j,strata,k)
            end do
         END DO
      END DO

      DO j = 1,4
         print*,'total NCTB PI q',j,' = ', total(j)
      END DO

      END
C-----------------------------------------------------------------------------
```

```
      PROGRAM mergeq

C     DESCRIPTION  Sum the fitted non-pb revenue from the ap estimation
C                  to the quarters.
C
C     CREATED BY:        Bill Humphries
C     DATE:              Fri Apr 21 14:16:03 CDT 1995
C
C     LAST MODIFIED BY:  ta
C     DATE:              02-20-96
C                        10-08-96  ta; remove reference to P/B check.
C                        08-15-97  ta; add back Permit office check.
C                        01-15-98  ta; update for FY98
C                        01-25-99  ta; update for FY99
C                        02-01-00  ts; update for FY00

      IMPLICIT NONE

      INTEGER*4 npfin,ntfin,nap,nq
      PARAMETER (ntfin=10815,npfin=2203, nap=13,nq=4)  ! update all

      INTEGER*4 iap,ier,ifin,searchc,iq,ipfin,i,j
      CHARACTER*6 fins(ntfin),pfins(npfin),fin
      CHARACTER*2 cap(nap)/'01','02','03','04','05','06','07','08','09','10','11','12','13'/ !update
      INTEGER*4 ap_to_q(nap)/1,1,1,2,2,2,3,3,3,4,4,4,4/     ! update; rem q4
      REAL*8  rev(nq,ntfin),smrev(nq,ntfin),pirev(nq,ntfin)
      REAL*8  sm,meter,stamp,pi

C     Initialize variables
      do j = 1, ntfin
         do i = 1, nq
            rev(i,j) = 0.0
            smrev(i,j) = 0.0
            pirev(i,j) = 0.0
         end do
      end do

7     format(a6)
      OPEN(8,FILE='finstrata.new.00',readonly)  ! update
      DO ifin=1,ntfin
         READ(8,7) fins(ifin)
      END DO

      OPEN(8,FILE='finstrata.pmt',readonly)
      DO ifin=1,npfin
         READ(8,7) pfins(ifin)
      END DO

9     FORMAT(A6,1X,F20.10,3F12.0)
      DO iap=1,nap
         iq = ap_to_q(iap)
C     These files only contains non-PERMIT fins that needed rev fit for the ap.
         OPEN(10,FILE='nonpbfit.'//cap(iap)//'.rev',readonly)
         ier = 0
         DO WHILE (ier.eq.0)
            READ(10,9,IOSTAT=ier,END=11) fin,sm,meter,stamp,pi   ! sm = smhat
            ipfin = searchc(pfins,npfin,fin)
            IF (ipfin.le.0) THEN
               ifin = searchc(fins,ntfin,fin)
               IF (ifin.gt.0) THEN
                  rev(iq,ifin) = rev(iq,ifin) + sm + pi
                  smrev(iq,ifin) = smrev(iq,ifin) + sm
                  pirev(iq,ifin) = pirev(iq,ifin) + pi
               ELSE
                  print*,'fin not found ', fin
               END IF
            END IF
         END DO

11       CONTINUE
         print*,'Finished ap ', iap
      END DO

12    FORMAT(A6,I1,3F20.10)
      OPEN(20,FILE='nonpb.rev')
      DO iq=1,nq
         DO ifin=1,ntfin
            IF (rev(iq,ifin).gt.0) THEN
               WRITE(20,12) fins(ifin),iq,rev(iq,ifin),smrev(iq,ifin),
     +            pirev(iq,ifin)
```

```
      END IF
   END DO
END DO
print*,'Wrote new file through qtr ',nq

STOP
END
```

```
      PROGRAM buildconnew

C     DESCRIPTION  generate the control revenue for SM
C
C     CREATED BY:       Bill Humphries
C     DATE:             Fri Oct 21 16:06:19 CDT 1994
C
C     LAST MODIFIED BY: ta
C     DATE:             11-15-95
C                       19-apr-96 ; set file size of finrev.1st based on qtr
C                       22-apr-96 ; write out diagnostics by quarter
C                       18-sep-96 ; update for reclass
C                       30-oct-96 ; update for fy97, this program copied from
C                                   buildconnew_RC.f in 96
C                       15-jan-98 ; update for FY98
C                       25-jan-99 ; update for FY99
C                       01-feb-00 ; update for FY00

C     **** IMPORTANT; for new qtr need to add code for finrev reads ****

      IMPLICIT NONE

      INTEGER*4 nstrata,ntfin,nap,nq
      PARAMETER(nstrata=20,ntfin=10815)            ! update
      parameter (nap=13,nq=4)                       ! update

      CHARACTER*6 fin,tfins(ntfin)

      REAL*8    pbtotalq(nq),ntotalq(nq),pirev,tpiq(nq),tavesmq(nq),nsmtotq(nq)
      REAL*8    rev,smbad,pibad,npitotq(nq),tsmq(nq),smrev,totalq(nq)
      REAL*8    cbcispi(nap,ntfin),cbcissm(nap,ntfin),tcbcissmq(nq)
      REAL*8    trevenueq(nq,nstrata), smrawq(nq)
      REAL*8    tarevenue(nstrata)
      REAL*8    arevenue(nq,nstrata)
      REAL*8    ppirevenue(nq,nstrata)
      REAL*8    psmrevenue(nq,nstrata)
      REAL*8    npirevenue(nq,nstrata)
      REAL*8    pirevenue(nq,nstrata),smrevenue(nq,nstrata)
      REAL*8    prevenue(nq,nstrata),nrevenue(nq,nstrata)
      REAL*8    revenue(nq,nstrata),ppirev(ntfin,nap)
      REAL*8    bsmrev(ntfin,nap),avesmrev
      REAL*8    prev(ntfin,nap),psmrev(ntfin,nap),avepi(ntfin)
      REAL*8    tpsmrev(ntfin),tprev(ntfin),tppirev(ntfin),avesm(ntfin)
      REAL*8    averev,avepirev,tpinq(nq),ptotalq(nq)
      REAL*8    nsmrevenue(nq,nstrata),prawq(nq,nstrata)
      REAL*8    pcraw(ntfin),check1,check2,scale(1100),total

      INTEGER*4 taps(nstrata)/nstrata*0/
      INTEGER*4 tstrata(ntfin), j
      INTEGER*4 i,ifin,iap,searchc,istrata,ifinb,iq,ier,nfound
      INTEGER*4 npnon,ipfin,tap, nrec, qfins
      INTEGER*4 npbaps(ntfin,nap),npaps(ntfin,nap)

      CHARACTER*2 cap(nap)/'01','02','03','04','05','06','07','08','09','10','11','12','13'/ ! update
      INTEGER*4 quartermap(nap)/1,1,1,2,2,2,3,3,3,4,4,4,4/     ! update

C     INITIALIZATIONS

      npnon = 0
      total = 0.0

      do i = 1,nq
         ntotalq(i) = 0.0
         ptotalq(i) = 0.0
         npitotq(i) = 0.0
         nsmtotq(i) = 0.0
         tpinq(i) = 0.0
         tpiq(i) = 0.0
         tsmq(i) = 0.0
         smrawq(i) = 0.0
         tavesmq(i) = 0.0
         tcbcissmq(i) = 0.0
         totalq(i) = 0.0
         do j = 1,nstrata
            nrevenue(i,j) = 0.0
            nsmrevenue(i,j) = 0.0
            npirevenue(i,j) = 0.0
            psmrevenue(i,j) = 0.0
            prevenue(i,j) = 0.0
            ppirevenue(i,j) = 0.0
```

```
            psmrevenue(i,j) = 0.0
            arevenue(i,j) = 0.0
            trevenueq(i,j) = 0.0
            revenue(i,j) = 0.0
            pirevenue(i,j) = 0.0
            smrevenue(i,j) = 0.0
            prawq(i,j) = 0.0
         end do
      end do

      do i = 1,nstrata
         tarevenue(i) = 0.0
         taps(i) = 0
      end do

      do i = 1,ntfin
         tpsmrev(i) = 0.0
         tprev(i) = 0.0
         tppirev(i) = 0.0
         pcraw(i) = 0.0
         do j = 1, nap
            npbaps(i,j) = 0
            npaps(i,j) = 0
         end do
      end do

      do i = 1,nap
         do j = 1,ntfin
            cbcispi(i,j) = 0.0
            cbcissm(i,j) = 0.0
            ppirev(j,i) = 0.0
            prev(j,i) = 0.0
            psmrev(j,i) = 0.0
         end do
      end do

2     FORMAT(a6,1x,i3)

      OPEN(1,FILE='finstrata.new.00',readonly)     ! update

      DO i = 1, ntfin
         READ(1,2) tfins(i),tstrata(i)
      END DO

      PRINT *, "Read Strata Maps."

      DO iap=1,nap
         ier = 0
         OPEN(1,FILE='finrev.'//cap(iap),readonly)
         nrec = 0
         DO WHILE (ier.eq.0)
            READ(1,'(A6,3F10.0)',IOSTAT=ier,END=9) fin,pirev,smrev
            ifin = searchc(tfins,ntfin,fin)
            IF (ifin.gt.0) THEN
               nrec = nrec + 1
               cbcispi(iap,ifin) = pirev
               cbcissm(iap,ifin) = smrev
            END IF
         END DO
9        CLOSE(1)
         PRINT *, "Read AP",cap(iap)," CBCIS revenues. ier = ",ier,"nrec = ",nrec
         nrec = 0
      END DO

      OPEN(1,FILE='averev.dat',readonly)
97     format(a6,2f20.5)

      nrec = 0
      ier = 0
      DO WHILE (ier.eq.0)
         READ(1,97,IOSTAT=ier,END=19) fin,pirev,smrev
         ifin = searchc(tfins,ntfin,fin)
         IF (ifin.gt.0) THEN
            avepi(ifin) = pirev
            avesm(ifin) = smrev
            nrec = nrec + 1
         END IF
      END DO

19    PRINT *, "Read average stamped and metered for CBCIS offices. ier = ",ier
```

```
      print*,' total recs where fin in ave.rev was found = ',nrec

C     MAIN BODY

21    FORMAT(F20.5,1X,A6,4F20.5)

      DO iap=1,nap
         iq = quartermap(iap)
         OPEN(20,file='finrev.1st.'//cap(iap),readonly)     ! PERMIT data

         DO i = 1,ntfin
            READ(20,21) rev,fin,pirev,smrev
            ifin = searchc(tfins,ntfin,fin)
            IF (ifin.gt.0) THEN
               IF ((pirev+smrev).gt.0) THEN
                  smrawq(iq) = smrawq(iq) + smrev
                  istrata = tstrata(ifin)
                  prawq (iq,istrata) = prawq(iq,istrata) + pirev + smrev
                  tping(iq) = tping(iq) + pirev + smrev
                  ppirev(ifin,iap) = pirev + ppirev(ifin,iap)
                  prev(ifin,iap) = pirev + smrev + prev(ifin,iap)
                  psmrev(ifin,iap) = smrev + psmrev(ifin,iap)
                  tpsmrev(ifin) = tpsmrev(ifin) + smrev
                  tprev(ifin) = tprev(ifin) + pirev + smrev
                  tppirev(ifin) = tppirev(ifin) + pirev
                  npaps(ifin,iap) = 1
               END IF
            ELSE
C                PRINT *, "Can't Find: ",fin
            END IF
         END DO
         CLOSE(20)
         PRINT *, "Read AP",cap(iap)," PERMIT."
      END DO

      DO ifin=1,ntfin
            averev = avepi(ifin)+avesm(ifin)
            avepirev = avepi(ifin)
            avesmrev = avesm(ifin)
            DO iap = 1,nap
               iq = quartermap(iap)

C              If no data for this ap then use cbcis or ave cbcis.

               IF (prev(ifin,iap).eq.0.0) THEN
                  IF (cbcispi(iap,ifin).eq.0.and.cbcissm(iap,ifin).eq.0) THEN
                     prev(ifin,iap) = averev
                     ppirev(ifin,iap) = avepirev
                     psmrev(ifin,iap) = avesmrev
                     tavesmq(iq) = tavesmq(iq) + avesmrev
                  ELSE
                     prev(ifin,iap) = cbcispi(iap,ifin) + cbcissm(iap,ifin)
                     ppirev(ifin,iap) = cbcispi(iap,ifin)
                     psmrev(ifin,iap) = cbcissm(iap,ifin)
                     tcbcissmq(iq) = tcbcissmq(iq) + cbcissm(iap,ifin)
                     print*,'CBCIS fin ',tfins(ifin),' data used'
                  END IF
               END IF
            END DO
            istrata = tstrata(ifin)
            DO iap=1,nap
               iq = quartermap(iap)
               psmrevenue(iq,istrata) = psmrevenue(iq,istrata) +
     +            psmrev(ifin,iap)
               prevenue(iq,istrata) = prevenue(iq,istrata) + prev(ifin,iap)
               ppirevenue(iq,istrata) = ppirevenue(iq,istrata) +
     +            ppirev(ifin,iap)
            END DO

      END DO


C     get totals

      DO istrata=1,nstrata
         DO iq=1,nq
            ptotalq(iq) = ptotalq(iq) + prevenue(iq,istrata)
         END DO
      END DO
```

```
C     Distribute Revenue from Non PERMIT Offices

      OPEN(10,FILE='nonpb.rev',readonly)    ! from mergeq.f
3     FORMAT(A6,I1,3F20.10)
      ier = 0
      i = 0
      nfound = 0
      DO WHILE (ier.eq.0)
         READ(10,3,IOSTAT=ier,END=4) fin,iq,rev,smrev,pirev
         ifin = searchc(tfins,ntfin,fin)
         IF (ifin.gt.0) THEN
            istrata = tstrata(ifin)
            nsmrevenue(iq,istrata) = nsmrevenue(iq,istrata) +
     +         smrev
            nrevenue(iq,istrata) = nrevenue(iq,istrata) +
     +         rev
            npirevenue(iq,istrata) = npirevenue(iq,istrata) +
     +            pirev
            ntotalq(iq) = ntotalq(iq) + rev
            nsmtotq(iq) = nsmtotq(iq) + smrev
            npitotq(iq) = npitotq(iq) + pirev
            nfound = nfound + 1
         END IF
         i = i + 1
      END DO
4     PRINT *, "Read ",i," records."
      PRINT *, "Found ",nfound," non P/B offices."

      DO istrata=1,nstrata
         DO iq=1,nq
            revenue(iq,istrata) = prevenue(iq,istrata) + nrevenue(iq,istrata)
            pirevenue(iq,istrata) =
     +         ppirevenue(iq,istrata) + npirevenue(iq,istrata)
            smrevenue(iq,istrata) =
     +         psmrevenue(iq,istrata) + nsmrevenue(iq,istrata)
         END DO
      END DO

      DO istrata=1,nstrata
         DO iq = 1, nq
            trevenueq(iq,istrata) = prevenue(iq,istrata)
     +         + nrevenue(iq,istrata)
            total = total + trevenueq(iq,istrata)
            totalq(iq) = totalq(iq) + trevenueq(iq,istrata)
         END DO
      END DO

      OPEN(10,FILE='trevenue.dat')  ! includes fitted rev
      OPEN(20,FILE='prevenue.dat')

5     FORMAT(4F15.0)
      DO istrata=1,nstrata
         WRITE(10,5) (revenue(iq,istrata),iq=1,nq)
         WRITE(10,5) (pirevenue(iq,istrata),iq=1,nq)
         WRITE(10,5) (smrevenue(iq,istrata),iq=1,nq)
         WRITE(20,5) (prevenue(iq,istrata),iq=1,nq)
         WRITE(20,5) (ppirevenue(iq,istrata),iq=1,nq)
         WRITE(20,5) (psmrevenue(iq,istrata),iq=1,nq)
      END DO

C     get totals

      DO iq = 1,nq
         DO istrata = 1, nstrata
            tpiq(iq) = tpiq(iq) + pirevenue(iq,istrata)
            tsmq(iq) = tsmq(iq) + smrevenue(iq,istrata)
         END DO
      END DO

      write(6,'("Total revenue          = ",4f12.0)') (totalq(j),j = 1,nq)
      WRITE(6,'("Total revenue Non P    = ",4f12.0)') (ntotalq(j),j = 1,nq)
      WRITE(6,'("Total Non P PI         = ",4f12.0)') (npitotq(j),j = 1,nq)
      WRITE(6,'("Total Non P SM         = ",4f12.0)') (nsmtotq(j),j = 1,nq)
      WRITE(6,'("Total PERMIT           = ",4f12.0)') (ptotalq(j),j = 1,nq)
      WRITE(6,'("Total Raw PERMIT       = ",4f12.0)') (tpinq(j),j = 1,nq)
      WRITE(6,'("Total PI Presort       = ",4f12.0)') (tpiq(j),j = 1,nq)
      WRITE(6,'("Raw SM                 = ",4f12.0)') (smrawq(j),j = 1,nq)
      WRITE(6,'("Total SM               = ",4f12.0)') (tsmq(j),j = 1,nq)
```

```
WRITE(6,'("Total SM filled w/Ave = ",4f12.0)') (tavesmq(j),j = 1,nq)
WRITE(6,'("Total SM filled w/CBC = ",4f12.0)') (tcbcissmq(j),j = 1,nq)
PRINT *, " "

DO iq = 1, nq
   PRINT *,'QUARTER ',iq
   PRINT *,'          trevenue      prevenue      nrevenue'
   DO i=1,nstrata
      WRITE(6,'("Strata: ",I2,1X,3F12.0)') i,trevenueq(iq,i),
+        prevenue(iq,i), nrevenue(iq,i)
   END DO
   Print *,'              praw       prevenue      '
   DO i=1,nstrata
      WRITE(6,'("Strata: ",I2,1X,2F12.0)') i,prawq(iq,i),prevenue(iq,i)
   END DO
END DO

STOP
END
```

```
      PROGRAM sm_data

C     DESCRIPTION  generate the control revenue for SM
C
C     CREATED BY:        ta
C     DATE:              nov 16, 1999
C     PROJECT:           mcb/task13/data00/strata
C
C     LAST MODIFIED BY: ta

C     **** IMPORTANT; for new qtr need to add code for finrev reads ****

      IMPLICIT NONE

      INTEGER*4 nstrata,ntfin,nap,nq
      PARAMETER(nstrata=20,ntfin=10826)            ! update
      parameter (nap=13,nq=4)                       ! update

      CHARACTER*6 fin,tfins(ntfin)

      REAL*8    pbtotalq(nq),ntotalq(nq),pirev,tpiq(nq),tavesmq(nq),nsmtotq(nq)
      REAL*8    rev,smbad,pibad,npitotq(nq),tsmq(nq),smrev,totalq(nq)
      REAL*8    cbcispi(nap,ntfin),cbcissm(nap,ntfin),tcbcissmq(nq)
      REAL*8    trevenueq(nq,nstrata), smrawq(nq)
      REAL*8    tarevenue(nstrata)
      REAL*8    arevenue(nq,nstrata)
      REAL*8    ppirevenue(nq,nstrata)
      REAL*8    psmrevenue(nq,nstrata)
      REAL*8    npirevenue(nq,nstrata)
      REAL*8    pirevenue(nq,nstrata),smrevenue(nq,nstrata)
      REAL*8    prevenue(nq,nstrata),nrevenue(nq,nstrata)
      REAL*8    revenue(nq,nstrata),ppirev(ntfin,nap)
      REAL*8    bsmrev(ntfin,nap),avesmrev
      REAL*8    prev(ntfin,nap),psmrev(ntfin,nap),avepi(ntfin)
      REAL*8    tpsmrev(ntfin),tprev(ntfin),tppirev(ntfin),avesm(ntfin)
      REAL*8    averev,avepirev,tpinq(nq),ptotalq(nq)
      REAL*8    nsmrevenue(nq,nstrata),prawq(nq,nstrata)
      REAL*8    pcraw(ntfin),check1,check2,scale(1100),total

      INTEGER*4 taps(nstrata)/nstrata*0/
      INTEGER*4 tstrata(ntfin), j
      INTEGER*4 i,ifin,iap,searchc,istrata,ifinb,iq,ier,nfound
      INTEGER*4 npnon,ipfin,tap, nrec, qfins
      INTEGER*4 npbaps(ntfin,nap),npaps(ntfin,nap)
      integer*4 psmct(nap,nstrata)/260*0/
      integer*4 nonpsmct(nq,nstrata)/80*0/

      CHARACTER*2 cap(nap)/'01','02','03','04','05','06','07','08','09',
     +                     '10','11','12','13'/                ! update
      INTEGER*4 quartermap(nap)/1,1,1,2,2,2,3,3,3,4,4,4,4/    ! update

C     INITIALIZATIONS

      do istrata = 1,nstrata
         tarevenue(istrata) = 0.0
         taps(istrata) = 0
         do iq = 1,nq
            nrevenue(iq,istrata) = 0.0
            nsmrevenue(iq,istrata) = 0.0
            npirevenue(iq,istrata) = 0.0
            psmrevenue(iq,istrata) = 0.0
            prevenue(iq,istrata) = 0.0
            ppirevenue(iq,istrata) = 0.0
            arevenue(iq,istrata) = 0.0
            trevenueq(iq,istrata) = 0.0
            revenue(iq,istrata) = 0.0
            pirevenue(iq,istrata) = 0.0
            prawq(iq,istrata) = 0.0
            smrevenue(iq,istrata) = 0.0
         end do
      end do

      do ipfin =1 ,ntfin
         tprev(ipfin) = 0.0
         tppirev(ipfin) = 0.0
         tpsmrev(ipfin) = 0.0
         pcraw(ipfin) = 0.0
         do iap = 1,nap
            cbcispi(iap,ipfin) = 0.0
            cbcissm(iap,ipfin) = 0.0
```

```
            psmrev(ipfin,iap) = 0.0
            prev(ipfin,iap) = 0.0
            ppirev(ipfin,iap) = 0.0
            npbaps(ipfin,iap) = 0
            npaps(ipfin,iap) = 0
         end do
      end do


      pirev = 0.0
      smrev = 0.0
      total = 0.0
      npnon = 0
      do iq =1 ,nq
         ntotalq(iq) = 0.0
         ptotalq(iq) = 0.0
         npitotq(iq) = 0.0
         nsmtotq(iq) = 0.0
         tpinq(iq) = 0.0
         tpiq(iq) = 0.0
         tsmq(iq) = 0.0
         smrawq(iq) = 0.0
         tavesmq(iq) = 0.0
         tcbcissmq(iq) = 0.0
         totalq(iq) = 0.0
      end do

2     FORMAT(a6,1x,i3)

      OPEN(1,FILE='finstrata.new.00',readonly)     ! update

      DO i = 1, ntfin
         READ(1,2) tfins(i),tstrata(i)
      END DO

      PRINT *, "Read Strata Maps."

      DO iap=1,nap
         ier = 0
         OPEN(1,FILE='/u/mcb/task13/CBCIS00/finrev.'//cap(iap),readonly)
         nrec = 0
         DO WHILE (ier.eq.0)
            READ(1,'(A6,3F10.0)',IOSTAT=ier,END=9) fin,pirev,smrev
            ifin = searchc(tfins,ntfin,fin)
            IF (ifin.gt.0) THEN
               nrec = nrec + 1
               cbcispi(iap,ifin) = pirev
               cbcissm(iap,ifin) = smrev
            END IF
         END DO
9        CLOSE(1)
         PRINT *, "Read AP",cap(iap)," CBCIS revenues. ier = ",ier,"nrec = ",nrec
         nrec = 0
      END DO

      OPEN(1,FILE='/u/mcb/task13/CBCIS00/averev.dat',readonly)
97     format(a6,2f20.5)

      nrec = 0
      ier = 0
      DO WHILE (ier.eq.0)
         READ(1,97,IOSTAT=ier,END=19) fin,pirev,smrev
         ifin = searchc(tfins,ntfin,fin)
         IF (ifin.gt.0) THEN
            avepi(ifin) = pirev
            avesm(ifin) = smrev
            nrec = nrec + 1
         END IF
      END DO

19    PRINT *, "Read average stamped and metered for CBCIS offices. ier = ",ier
      print*,' total recs where fin in ave.rev was found = ',nrec

C     MAIN BODY

21    FORMAT(F20.5,1X,A6,4F20.5)

      DO iap=1,nap
         iq = quartermap(iap)
         OPEN(20,file='../pmt/matrices/finrev.1st.'//cap(iap),readonly)     ! PERMIT data
```

```
      if ((iap.ge.1).and.(iap.le.13)) qfins = 10826   !!! add for next qtr

      DO i = 1,qfins
         READ(20,21) rev,fin,pirev,smrev
         ifin = searchc(tfins,ntfin,fin)
         IF (ifin.gt.0) THEN
            IF ((pirev+smrev).gt.0) THEN
               smrawq(iq) = smrawq(iq) + smrev
               istrata = tstrata(ifin)
               prawq (iq,istrata) = prawq(iq,istrata) + pirev + smrev
               tpinq(iq) = tpinq(iq) + pirev + smrev
               ppirev(ifin,iap) = pirev + ppirev(ifin,iap)
               prev(ifin,iap) = pirev + smrev + prev(ifin,iap)
               psmrev(ifin,iap) = smrev + psmrev(ifin,iap)
               tpsmrev(ifin) = tpsmrev(ifin) + smrev
               tprev(ifin) = tprev(ifin) + pirev + smrev
               tppirev(ifin) = tppirev(ifin) + pirev
               npaps(ifin,iap) = 1
            END IF
         ELSE
C             PRINT *, "Can't Find: ",fin
         END IF
      END DO
      CLOSE(20)
      PRINT *, "Read AP",cap(iap)," PERMIT."
   END DO

   DO ifin=1,ntfin
         averev = avepi(ifin)+avesm(ifin)
         avepirev = avepi(ifin)
         avesmrev = avesm(ifin)
         DO iap = 1,nap
            iq = quartermap(iap)

C           If no data for this ap then use cbcis or ave cbcis.

            IF (prev(ifin,iap).eq.0.0) THEN
               IF (cbcispi(iap,ifin).eq.0.and.cbcissm(iap,ifin).eq.0) THEN
                  prev(ifin,iap) = averev
                  ppirev(ifin,iap) = avepirev
                  psmrev(ifin,iap) = avesmrev
                  tavesmq(iq) = tavesmq(iq) + avesmrev
               ELSE
                  prev(ifin,iap) = cbcispi(iap,ifin) + cbcissm(iap,ifin)
                  ppirev(ifin,iap) = cbcispi(iap,ifin)
                  psmrev(ifin,iap) = cbcissm(iap,ifin)
                  tcbcissmq(iq) = tcbcissmq(iq) + cbcissm(iap,ifin)
               END IF
            END IF
         END DO
         istrata = tstrata(ifin)
         DO iap=1,nap
            iq = quartermap(iap)
            if (psmrev(ifin,iap).ne.0.0) psmct(iap,istrata) =  psmct(iap,istrata)+1
            psmrevenue(iq,istrata) = psmrevenue(iq,istrata) +
     +         psmrev(ifin,iap)
            prevenue(iq,istrata) = prevenue(iq,istrata) + prev(ifin,iap)
            ppirevenue(iq,istrata) = ppirevenue(iq,istrata) +
     +         ppirev(ifin,iap)
         END DO

   END DO


C  get totals

   DO istrata=1,nstrata
      DO iq=1,nq
         ptotalq(iq) = ptotalq(iq) + prevenue(iq,istrata)
      END DO
   END DO



C  Distribute Revenue from Non PERMIT Offices

   OPEN(10,FILE='nonpb.rev',readonly)    ! from mergeq.f
1  FORMAT(A6,I1,3F20.10)
   ier = 0
```

```
      i = 0
      nfound = 0
      DO WHILE (ier.eq.0)
         READ(10,3,IOSTAT=ier,END=4) fin,iq,rev,smrev,pirev
         ifin = searchc(tfins,ntfin,fin)
         IF (ifin.gt.0) THEN
            istrata = tstrata(ifin)
            nsmrevenue(iq,istrata) = nsmrevenue(iq,istrata) +
     +         smrev
            nonpsmct(iq,istrata) = nonpsmct(iq,istrata) + 1
            nrevenue(iq,istrata) = nrevenue(iq,istrata) +
     +         rev
            npirevenue(iq,istrata) = npirevenue(iq,istrata) +
     +            pirev
            ntotalq(iq) = ntotalq(iq) + rev
            nsmtotq(iq) = nsmtotq(iq) + smrev
            npitotq(iq) = npitotq(iq) + pirev
            nfound = nfound + 1
         END IF
         i = i + 1
      END DO
4     PRINT *, "Read ",i," records."
      PRINT *, "Found ",nfound," non P/B offices."

      DO istrata=1,nstrata
         DO iq=1,nq
            revenue(iq,istrata) = prevenue(iq,istrata) + nrevenue(iq,istrata)
            pirevenue(iq,istrata) =
     +         ppirevenue(iq,istrata) + npirevenue(iq,istrata)
            smrevenue(iq,istrata) =
     +         psmrevenue(iq,istrata) + nsmrevenue(iq,istrata)
         END DO
      END DO

      DO istrata=1,nstrata
         DO iq = 1, nq
            trevenueq(iq,istrata) = prevenue(iq,istrata)
     +         + nrevenue(iq,istrata)
            total = total + trevenueq(iq,istrata)
            totalq(iq) = totalq(iq) + trevenueq(iq,istrata)
         END DO
      END DO

      OPEN(10,FILE='sm_rev.dat')


5     FORMAT(4F15.0)
7     format(i9)
      DO istrata=1,nstrata
         WRITE(10,5) (psmrevenue(iq,istrata),iq=1,nq)
      end do
      DO istrata=1,nstrata
         WRITE(10,5) (nsmrevenue(iq,istrata),iq=1,nq)
      end do
      DO istrata=1,nstrata
         WRITE(10,7) psmct(13,istrata)
      end do
      DO istrata=1,nstrata
         WRITE(10,7) nonpsmct(4,istrata)
      end do


      get totals

      DO iq = 1,nq
         DO istrata = 1, nstrata
            tpiq(iq) = tpiq(iq) + pirevenue(iq,istrata)
            tsmq(iq) = tsmq(iq) + smrevenue(iq,istrata)
         END DO
      END DO

      write(6,'("Total revenue          = ",4f12.0)') (totalq(j),j = 1,nq)
      WRITE(6,'("Total revenue Non P    = ",4f12.0)') (ntotalq(j),j = 1,nq)
      WRITE(6,'("Total Non P PI         = ",4f12.0)') (npitotq(j),j = 1,nq)
      WRITE(6,'("Total Non P SM         = ",4f12.0)') (nsmtotq(j),j = 1,nq)
      WRITE(6,'("Total PERMIT           = ",4f12.0)') (ptotalq(j),j = 1,nq)
      WRITE(6,'("Total Raw PERMIT       = ",4f12.0)') (tpinq(j),j = 1,nq)
      WRITE(6,'("Total PI Presort       = ",4f12.0)') (tpiq(j),j = 1,nq)
      WRITE(6,'("Raw SM                 = ",4f12.0)') (smrawq(j),j = 1,nq)
      WRITE(6,'("Total SM               = ",4f12.0)') (tsmq(j),j = 1,nq)
```

```
WRITE(6,'("Total SM filled w/Ave = ",4f12.0)') (tavesmq(j),j = 1,nq)
WRITE(6,'("Total SM filled w/CBC = ",4f12.0)') (tcbcissmq(j),j = 1,nq)
PRINT *, " "

DO iq = 1, nq
   PRINT *,'QUARTER ',iq
   PRINT *,'          trevenue      prevenue      nrevenue'
   DO i=1,nstrata
      WRITE(6,'("Strata: ",I2,1X,3F12.0)') i,trevenueq(iq,i),
+        prevenue(iq,i), nrevenue(iq,i)
   END DO
   Print *,'              praw       prevenue      '
   DO i=1,nstrata
      WRITE(6,'("Strata: ",I2,1X,2F12.0)') i,prawq(iq,i),prevenue(iq,i)
   END DO
END DO

STOP
END
```

```
      PROGRAM perrolln

C     DESCRIPTION: verifies and rolls up first class PERMIT system records
C                  to strata, mailing size, VIP, weight increment, & shape
C
C     CREATED BY:        Bill Humphries
C     DATE:              Thu Sep 15 10:06:39 CDT 1994
C
C     LAST MODIFIED BY: Tom Ayen
C     DATE:
C
C              4/26/95         Keep revenue from bad records in separate array
C
C              5/1/95          Decided to Keep records with inconsistent VIP/
C                                   record header data
C              3/19/96         revise checking for wt incr in pcwt = 0 loop
C                              in verify routine
C              9/11/96         Update for new VIPs due to reclass
C             10/24/96         edit nonstnd surcharge vip section; it appears
C                              that this vip has been eliminated with reclass.
C             11/1/96          Update for FY97.  Remove reference to 'old'
C                              VIPs.  This program created from perrolln_RC.f
C                              from 96.
C             01/08/97         finish updating for FY97.
C             01/10/97         update for new record layout for FY97.
C             08/15/97         update for new aviion with divide by zero check
C                              and common block fix.
C             08/28/97         check for Government mail records and exclude
C             01/15/97         update for 98 and correct NI wt icrement
C             08/07/98         correct for records mixing auto parcels with priority
C             01/25/99         update for FY99
C             03/31/99         update for new rates; use old program for aps 1-3
C             05/12/99         merge SP vips into PI and SM due to definitional change
C                              to the 41416 account which now includes PI SP.
C                              Use the two other programs for q1 and q2 data.
C             11/15/99         write out recs of heavy flats to diagnostic file
C             01/13/00         update for FY00 - edit for priority and prior rates
C             11/15/00         update code for new compiler for Q4 run

C
C     NOTE : update nfin below AND in function mail_size. If any other params
C               updated check the functions also.

      IMPLICIT NONE

C     Program Parameters
      INTEGER*4 maxvip,maxerr,nvip,nfin,nmap,nrpw,ntrp,nnewcls,size1
      PARAMETER (maxvip=20,maxerr=11,nvip=102,
     +   nfin=10815,nrpw=3,ntrp=2,nnewcls=34)      ! update nfin
      INTEGER*4 nshp,nwt,nsize,nstrata,nodes,ntype,nind,iap,nwtold
      PARAMETER (nshp=3,nwt=13,nsize=3,nstrata=20,nodes=34,ntype=12,nind=2,nwtold=11)
      REAL*8    zero
      PARAMETER (zero=0.0)
      PARAMETER (size1=nind*nstrata*nsize)

C     Maps
      CHARACTER*6 fins(nfin)
      INTEGER*4 stratamap(nfin)

C     Indicies and Counters
      INTEGER*4 ifin,istrata,iwt,isize,ivip,ier,nstd,irpw,itrp,iind,cx
      INTEGER*4 i,j,k,error(maxvip),icls
      INTEGER*4 finreass/0/,xminind
      INTEGER*4 minind(1:20)
      LOGICAL   vipsvalid,presort,imprint,firstokay
      CHARACTER*30 errorname(maxerr)
      CHARACTER*30 vipname(nodes)
      CHARACTER*30 tname(ntype)
      CHARACTER*2 cap

C     Functions
      INTEGER*4 searchc,mail_size,weight_incr,indicia

C     Data read from PERMIT record
      CHARACTER*5 ind
      CHARACTER*5 vip
      CHARACTER*6 finno,lfinno
      CHARACTER*30 unique
      CHARACTER*960 vipfields
      REAL*8 rev,pcs,wt,pcwt,fees,viprev,vippcs,viprate,pcsout
```

```
      INTEGER*4 ishp,lines,year,day,seq,trandate
      CHARACTER*1 psflag
      CHARACTER*5 pmtnum
      character*1024 record1,record2,record3,record4,record5
      character*5120 record

C     Verification Common Block
      INTEGER*4 recvip(maxvip),reciwt(maxvip),vipcount(nodes),tcount(ntype)
      REAL*8    recrev(maxvip),recpcs(maxvip),recwt(maxvip),recrte(maxvip)
      REAL*8    rate(nvip,3),ratetable(nvip,nwt),comprate(maxvip)
      REAL*8    sizerpt(nind,nsize,nsize)
      INTEGER*4 vip_to_rate(nvip),maxwt(nvip),recind(maxvip),recnew(maxvip)
      INTEGER*4 ishape(maxvip)
      CHARACTER*5 vips(nvip)
      COMMON /verify/recvip,reciwt,recrev,ishape,
     +   recpcs,recwt,recrte,rate,
     +   vips,ratetable,vipcount,tcount,
     +   vip_to_rate,comprate,maxwt,recnew
      COMMON /trantype/sizerpt
      REAL*8 freeresids,nfresid,freenstdres,nfnstd,freezip4,nfzip4
      COMMON /freeflt/freeresids,nfresid,freenstdres,nfnstd,freezip4,nfzip4,
     +   seq,psflag

C     Rate Maps
      REAL*8    newratetable(nvip,nwt)

C     Error Tracking
      REAL*8 errrev(maxerr)/maxerr*0.0/,errpcs(maxerr)/maxerr*0.0/
      REAL*8 errwt(maxerr)/maxerr*0.0/
      INTEGER*4 errcount(maxerr)/maxerr*0/


C     Running Totals
      INTEGER*4 recin/0/,okrec/0/,badtran/0/,nvipvaild/0/,nvipinvalid/0/
      INTEGER*4 govct/0/
      REAL*8 revin/0.0/,pcsin/0.0/,wtin/0.0/
      REAL*8 okrev/0.0/,okpcs/0.0/,okwt/0.0/
      REAL*8 igrevout/0.0/,ibrevout/0.0/,ngrevout/0.0/,nbrevout/0.0/
      REAL*8 igpcsout/0.0/,ibpcsout/0.0/,ngpcsout/0.0/,nbpcsout/0.0/
      REAL*8 igwtout/0.0/,ngwtout/0.0/,ttrans/0.0/
      REAL*8 trev/0.0/,tpcs/0.0/
      REAL*8 stratarev(nstrata+1,2)/42*0.0/
      REAL*8 stratapcs(nstrata+1,2)/42*0.0/
      REAL*8 finrev(nfin),ppirev(nfin),smrev(nfin),sprev(nfin)
      REAL*8 clspcs(nnewcls)/nnewcls*0.0/
      REAL*8 tsmout/0.0/,tpiout/0.0/,bsmout/0.0/,tspout/0.0/

C     RPW and Transaction Arrays
C     Arrays which accumulate by VIP, weight and shape use the indexing:
C       irpw = 1: Revenue, 2: Pieces, 3: Weight
C     Arrays which accumulate by size and shape of mailing use the indexing:
C       irpw = 1: Transactions, 2: Revenue, 3: Pieces

      REAL*8    rpw(nrpw,nnewcls,nwt,nshp,nstrata,nind)
      REAL*8    brpw(nrpw,nnewcls,nwt,nshp,nstrata,nind)
      REAL*8    mrpw(nrpw,nnewcls,nwt,nshp,nstrata,nind)
      REAL*8    trp(ntrp,nshp,nsize,nstrata,nind)
      INTEGER*4 trans(nsize,nstrata,nind)/size1*0/
      REAL*8    badrev(nfin,nind)

      CHARACTER*8 timestr
      CHARACTER*24 clstitle(nnewcls)

C     INITIALIZATIONS

      CALL time(timestr)
      PRINT *,"Start: ",timestr

C     Read AP from Command Line
      CALL getarg(1,cap)
      READ (cap,'(I2)') iap

      OPEN(7,FILE='titles.pmt')

C     Set Captions for Decision Tree Counter

      DO i=1,34
         READ(7,'(A30)') vipname(i)
      END DO
```

```
      DO i=1,nodes
         vipcount(i) = 0
      END DO

C     Error Captions

      DO i=1,11
         READ(7,'(A30)') errorname(i)
      END DO

      DO i=1,maxerr
         errcount(i) = 0
      END DO

C     Transaction Captions

      DO i=1,12
         READ(7,'(A30)') tname(i)
      END DO

      DO i=1,ntype
         tcount(i) = 0
      END DO

      nstd = 0

      OPEN(UNIT=90,FILE='ig.bad.pmt.1st.'//cap)
      OPEN(UNIT=91,FILE='ni.pmt.1st.'//cap)
      OPEN(UNIT=92,FILE='ig.rate.pmt.1st.'//cap)
      OPEN(UNIT=93,FILE='ib.rate.pmt.1st.'//cap)
      OPEN(UNIT=94,FILE='ni.rate.pmt.1st.'//cap)
      OPEN(UNIT=95,FILE='vip.pmt.1st.'//cap)
      OPEN(UNIT=96,FILE='finno.pmt.1st.'//cap)
      OPEN(UNIT=97,FILE='rev.pmt.1st.'//cap)
      OPEN(UNIT=98,FILE='pcs.pmt.1st.'//cap)
      OPEN(UNIT=99,FILE='nonstd.pmt.1st.'//cap)

      PRINT *, 'Opened Error Files.'

C     Read List of Valid Finance Numbers and associated Strata

      OPEN(1,FILE='finstrata.new.00') ! update

2     FORMAT(a6,1x,3i3)
      DO i = 1, nfin
         READ(1,2) fins(i), stratamap(i)
      END DO

      PRINT *, "Read Strata Maps."

C     Read List of Valid VIP Codes and Map to Rate Elements

      OPEN(7,FILE='vipmap.1st.00')
6     FORMAT(4X,A5,3X,I9)
      DO i=1,nvip
         READ(7,6) vip,j
         IF (vip(1:1).eq.' ') vip(1:1) = '0'
         vip_to_rate(i) = j
         vips(i) = vip
      END DO
      CLOSE(7)

      PRINT *, "Read VIP Map."


C     Read List of NEW Rates for VIP

      OPEN(9,FILE='ratetable.00')
10    format(6x,i2,1x,13f6.3)

      DO i=1,nvip
         read(9,10) maxwt(i), (ratetable(i,iwt), iwt = 1, nwt)
      END DO
      PRINT *, "read New rate list."

      CLOSE(9)

C     Initialize RPW Arrays
C     Treating each array separately for effeciency.
```

```
      do irpw = 1, nrpw
         do icls = 1, nnewcls
            do iwt = 1, nwt
              do ishp =1, nshp
               do istrata = 1,nstrata
                  do iind = 1, nind
                     rpw(irpw,icls,iwt,ishp,istrata,iind) = 0.0
                     brpw(irpw,icls,iwt,ishp,istrata,iind) = 0.0
                     mrpw(irpw,icls,iwt,ishp,istrata,iind) = 0.0
                  end do
               end do
              end do
             end do
          end do
       end do

      do irpw = 1, ntrp
         do ishp = 1, nshp
            do isize = 1, nsize
               do istrata = 1,nstrata
                  do iind = 1, nind
                     trp(irpw,ishp,isize,istrata,iind) = 0.0
                  end do
               end do
              end do
          end do
       end do

       do i = 1,nfin
          finrev(i) = 0.0
          ppirev(i) = 0.0
          smrev(i) = 0.0
          sprev(i) = 0.0
          do j = 1,nind
             badrev(i,j) = 0.0
          end do
       end do

      PRINT *, "Initialized Arrays, Starting Main Loop."

C     MAIN LOOP
C     Read a complete PERMIT record. Assign the VIP, revenue, pieces & weight
C     from each VIP field to the record array. Determine if record is an
C     identical or non-identical transaction. Assign weight increment, shape.

C      PERMIT AP File now read from STDOUT stream of gzcat process with perloop.

11    FORMAT(A6,A2,A30,1X,F12.2,2X,F12.2,12X,A1,I1,1X,F8.4,F12.0,F14.4,20X,I3,
     +   A960)
12    FORMAT(6X,A5,1X,F6.3,F12.0,F18.7)
13    FORMAT(A6,1X,F12.2,F12.2,1X,I1,1X,F8.4,F12.0,F14.4,1X,I3,1X,A48,F6.3)
14    format(a1024)

      open(15,file='permit.1st.tmp')
      ier = 0
      lfinno = 'xxxxxx'

C Read a PERMIT system record. Add the total revenue, pieces and weight
C to the running totals.

C      open(28,file='test',recl=5008)
      open(65,file='badvips.'//cap,recl=2000)
      open(75,file='f1_11_13.'//cap,recl=2000)

      DO WHILE (ier.eq.0)
         READ(15,11,iostat=ier,end=99) finno,ind,unique,rev,fees,psflag,
     +      ishp,pcwt,pcs,wt,lines,vipfields
         READ(unique,'(I4,I3,6X,I2,2X,A5,I8)') year,day,seq,pmtnum,trandate
         trev = 0.0
         tpcs = 0.0
         pcsout = 0.0
         imprint = .FALSE.
         firstokay = .TRUE.
         recin = recin + 1
         rev = rev - fees       ! subtract difference for non-qualifying pieces
         revin = revin + rev
         wtin  = wtin + wt
C   edit for new avilon
         if (rev.eq.0.0) rev = 1.0
```

```
         if (pcs.eq.0.0) pcs = 1.0

         IF (lines.gt.maxvip) THEN
            PRINT *, "Warning: maxvip exceeded with value: ",lines,
     +         " at record ",recin,"."
            STOP
         END IF

         IF (pmtnum(1:1).eq.'G') THEN        ! government mail
            govct = govct + 1
            go to 200
         END IF

C Compare the current to the previously read finance number. If it's new,
C locate the finance number in the strata map. If not found, then look for
C it in a list of remappings. If you still can't find the finance number,
C set istrata to zero and let the verification subroutine handle the
C problem.

         IF (lfinno.ne.finno) THEN
            ifin = searchc(fins,nfin,finno)
            IF (ifin.le.0) THEN
               finreass = finreass + 1
               ifin = searchc(fins,nfin,finno)
            END IF
            IF (ifin.gt.0) THEN
               istrata = stratamap(ifin)
            ELSE
               istrata = 0
               PRINT *, 'Did not find Finance Number: ',finno
            END IF
         END IF
         lfinno = finno
C         PRINT *, "Fin #: ",finno," ifin = ",ifin," istrata = ",istrata,
C     +      " vips = ",lines
         vipsvalid = .TRUE.

C Read each VIP field of the transaction. Check to see if it is a valid
C VIP. If it isn't, set the invalid VIP flag for the transaction to false.
C Let the verification subroutine handle the error.

         DO i=1,lines
            read(vipfields((i-1)*48+1:(i-1)*48+48),12) vip,viprate,vippcs,
     +         viprev
            IF (vip(1:1).eq.' '.and.vip.ne.'    0') THEN
               vip(1:1) = '0'
               imprint = .TRUE.
            END IF
            ivip = searchc(vips,nvip,vip)
C            PRINT *, "VIP : ",vip," rate = ",viprate," pcs = ",vippcs,
C     +         " rev = ",viprev," ivip = ",ivip," i = ",i
            IF (ivip.le.0.and.vip.ne.'    0') THEN  ! probably 'old' vip
               vipsvalid = .FALSE.
               nvipinvalid = nvipinvalid + 1
               recvip(i) = 0
               recrev(i) = 0
               recpcs(i) = 0
               recrte(i) = 0
C               print *,'Invalid vip ',vip,'  rev is ', viprev,'  fin is ',finno
               write(65,11) finno,ind,unique,rev,fees,psflag,
     +         ishp,pcwt,pcs,wt,lines,vipfields

            ELSE IF (ivip.le.0.and.vip.eq.'    0') THEN !should not exist after 96
               viprev = 0.0
               vippcs = 0.0
               viprate = 0.0
               nstd = nstd + 1
               nvipvaild = nvipvaild + 1
               recvip(i) = 0
               recrev(i) = 0
               recpcs(i) = 0
               recrte(i) = 0
               print *,'NS surcharge vip '

            ELSE IF (ivip.gt.0) THEN
               trev = trev + viprev
               tpcs = tpcs + vippcs
               nvipvaild = nvipvaild + 1
               recvip(i) = ivip
               recrev(i) = viprev
```

```
                recpcs(i) = vippcs
                recrte(i) = viprate
             END IF
          END DO

C Send record to verification routine. Routine returns '0' if record is a
C valid, identical, good weight transaction.

          pcsin = pcsin + tpcs

C          IF ((year.eq.2000).or.(year.eq.1999)) THEN
C             ratetable = newratetable
C          ELSE
C             print*,'year error ', year
C             stop
C          END IF

C    test records with 1st vip 1574 and others priority.  This began in ap7 98
C    for finno 419044.  Keep the first vip and ignore the rest.  Consider this
C    a good transaction. Reset for verification routine.

                if (lines.gt.1) then
                   if (recvip(1).eq.11.and.recvip(2).eq.0) then
                      lines = 1
                      vipsvalid = .TRUE.
                      pcs = tpcs    ! for verification
                      print*,'switched lines to 1'
                   end if
                end if

          CALL ver1st(error,istrata,vipsvalid,rev,pcs,wt,
     +       trev,tpcs,lines,ishp,pcwt)

C         This function assigning shape and creating recnew(i)
          isize = mail_size(pcs,lines,ishp)


C Assign revenue, pieces, and weight from valid transactions to matrices.

                xminind = 1      ! check for all valid vips in transaction
                DO i=1,lines
                   IF (recvip(i).ne.0) then
                      iind = indicia(vips(recvip(i)),recnew(i))
                      minind(i) = iind       ! assign indicia
                      firstokay = .FALSE.
                   ELSE
                      xminind = 0    ! bad vip
                   END IF
                END DO


          IF (.not.firstokay.and.xminind.ne.0) THEN       ! count transactions
             trans(isize,istrata,minind(1)) =
     +          trans(isize,istrata,minind(1)) + 1
          END IF

          IF (xminind.eq.0) THEN       ! bad vip transactions
             badtran = badtran + 1
          END IF

          DO i=1,lines
             IF (error(i).le.3.and.xminind.ne.0) THEN
                iwt = reciwt(i)
                pcsout = pcsout + recpcs(i)
                IF (minind(i).lt.3) THEN
                   finrev(ifin) = finrev(ifin) + recrev(i)
                END IF
                IF (minind(i).eq.1) THEN
                   ppirev(ifin) = ppirev(ifin) + recrev(i)
                ELSE IF (minind(i).eq.2) THEN
                   smrev(ifin) = smrev(ifin) + recrev(i)
                ELSE IF (minind(i).eq.3) THEN
                   sprev(ifin) = sprev(ifin) + recrev(i)
                END IF
                trp(1,ishape(i),isize,istrata,minind(i)) =
     +             trp(1,ishape(i),isize,istrata,minind(i)) + recrev(i)
                trp(2,ishape(i),isize,istrata,minind(i)) =
     +             trp(2,ishape(i),isize,istrata,minind(i)) + recpcs(i)
C               print*,recnew(i)," ",ishape(i)," ",istrata," ",minind(i)
                IF (error(i).eq.1) THEN
```

```
                  rpw(1,recnew(i),iwt,ishape(i),istrata,minind(i)) =
     +               rpw(1,recnew(i),iwt,ishape(i),istrata,minind(i)) + recrev(i)
                  rpw(2,recnew(i),iwt,ishape(i),istrata,minind(i)) =
     +               rpw(2,recnew(i),iwt,ishape(i),istrata,minind(i)) + recpcs(i)
                  rpw(3,recnew(i),iwt,ishape(i),istrata,minind(i)) =
     +               rpw(3,recnew(i),iwt,ishape(i),istrata,minind(i)) + recwt(i)
               ELSE IF (error(i).eq.2) THEN
                  brpw(1,recnew(i),iwt,ishape(i),istrata,minind(i)) =
     +               brpw(1,recnew(i),iwt,ishape(i),istrata,minind(i)) + recrev(i)
                  brpw(2,recnew(i),iwt,ishape(i),istrata,minind(i)) =
     +               brpw(2,recnew(i),iwt,ishape(i),istrata,minind(i)) + recpcs(i)
                  brpw(3,recnew(i),iwt,ishape(i),istrata,minind(i)) =
     +               brpw(3,recnew(i),iwt,ishape(i),istrata,minind(i)) + recwt(i)
               ELSE IF (error(i).eq.3) THEN
                  mrpw(1,recnew(i),iwt,ishape(i),istrata,minind(i)) =
     +               mrpw(1,recnew(i),iwt,ishape(i),istrata,minind(i))
     +               + recrev(i)
                  mrpw(2,recnew(i),iwt,ishape(i),istrata,minind(i)) =
     +               mrpw(2,recnew(i),iwt,ishape(i),istrata,minind(i))
     +               + recpcs(i)
                  mrpw(3,recnew(i),iwt,ishape(i),istrata,minind(i)) =
     +               mrpw(3,recnew(i),iwt,ishape(i),istrata,minind(i))
     +               + recwt(i)
               END IF
            ELSE IF (xminind.ne.0) THEN
                  badrev(ifin,minind(i)) = badrev(ifin,minind(i)) + recrev(i)
            END IF
            errcount(error(i)) = errcount(error(i)) + 1
            errrev(error(i)) = errrev(error(i)) + recrev(i)
            errpcs(error(i)) = errpcs(error(i)) + recpcs(i)
            errwt(error(i)) = errwt(error(i)) + recwt(i)
            IF (error(i).gt.1) THEN
               WRITE(88+error(i),13) finno,rev,fees,ishp,pcwt,pcs,wt,
     +            lines,vipfields((i-1)*48+1:(i-1)*48+48),comprate(i)
            END IF
C        Temp for diagnostic of heavy flats
            IF (error(i).lt.3.and.iwt.gt.10.and.(recnew(i).eq.2.or.recnew(i).eq.6.
     +      or.recnew(i).eq.16.or.recnew(i).eq.25.or.recnew(i).eq.26)) Then
               write(75,11) finno,ind,unique,rev,fees,psflag,
     +      ishp,pcwt,pcs,wt,lines,vipfields
            END IF

         END DO

C Okay, you're done with that record.

200   continue

      END DO                    ! Main Loop

C Write a Report file.

99    PRINT *, "Finished Reading AP",cap," with ier = ",ier
      OPEN(80,FILE='perrolln.lst.'//cap//'.rpt')
      WRITE(80,'("Report for new PERMIT by Size AP",A2," PFY 2000")') cap
81    FORMAT("     Disposition                        VIPs    Revenue",
     +   "     Pieces     Weight")
83    FORMAT("     Group                             Revenue",
     +   "     Pieces     Weight")
82    FORMAT("        Records Read : ",I10,3F12.0)
84    FORMAT(A30," : ",I10,3F12.0)
      WRITE(80,81)
      WRITE(80,'(70("-"))')
      DO i=1,maxerr
         WRITE(80,84) errorname(i),errcount(i),errrev(i),errpcs(i),errwt(i)
      END DO
      WRITE(80,'(70("-"))')
      WRITE(80,'("  Valid Vips : ",I10)') nvipvaild
      WRITE(80,'("Invalid Vips : ",I10)') nvipinvalid
      WRITE(80,'(70("-"))')
      WRITE(80,'("Numbers of VIPs Routed at each node:")')
      WRITE(80,'(70("-"))')
      DO i=1,nodes
         WRITE(80,84) vipname(i),vipcount(i)
      END DO
      WRITE(80,'(70("-"))')
      DO i=1,ntype
         WRITE(80,84) tname(i),tcount(i)
      END DO
      WRITE(80,'(70("-"))')
```

```
C     Write Roll-up File

      OPEN(50,FILE='perrolln.1st.'//cap)
51    FORMAT(I3,I2,I1,I2,I1,"11",3F20.10)
52    FORMAT(I3,I2,I1,I2,I1,"10",3F20.10)
53    FORMAT(I3,I2,I1,I2,I1,"00",3F20.10)
      OPEN(60,FILE='perrolln.trans.1st.'//cap)
61    FORMAT(I2,I1,I2,I1,3F20.10)

C     Write VIP Matrices
      DO iind=1,nind
         DO istrata=1,nstrata
            DO ishp=1,nshp
               DO iwt=1,nwt
                  DO icls=1,nnewcls
                     IF (rpw(1,icls,iwt,ishp,istrata,iind).ne.0) THEN
                        WRITE(50,51) icls,iwt,ishp,istrata,iind,
     +                     (rpw(irpw,icls,iwt,ishp,istrata,iind),irpw=1,nrpw)
                        igrevout = igrevout+rpw(1,icls,iwt,ishp,istrata,iind)
                        igpcsout = igpcsout+rpw(2,icls,iwt,ishp,istrata,iind)
                        igwtout = igwtout+rpw(3,icls,iwt,ishp,istrata,iind)
                        stratarev(istrata,1) =
     +                     rpw(1,icls,iwt,ishp,istrata,iind)
     +                     + stratarev(istrata,1)
                        stratapcs(istrata,1) =
     +                     rpw(2,icls,iwt,ishp,istrata,iind)
     +                     + stratapcs(istrata,1)
                        clspcs(icls) = clspcs(icls) +
     +                     rpw(2,icls,iwt,ishp,istrata,iind)
                     END IF
                     IF (brpw(1,icls,iwt,ishp,istrata,iind).ne.0) THEN
                        WRITE(50,52) icls,iwt,ishp,istrata,iind,
     +                     (brpw(irpw,icls,iwt,ishp,istrata,iind),irpw=1,nrpw)
                        ibrevout =ibrevout+brpw(1,icls,iwt,ishp,istrata,iind)
                        ibpcsout =ibpcsout+brpw(2,icls,iwt,ishp,istrata,iind)
                     stratarev(istrata,1) =
     +                     brpw(1,icls,iwt,ishp,istrata,iind)
     +                     + stratarev(istrata,1)
                     stratapcs(istrata,1) =
     +                     brpw(2,icls,iwt,ishp,istrata,iind)
     +                     + stratapcs(istrata,1)
                     clspcs(icls) = clspcs(icls) +
     +                        brpw(2,icls,iwt,ishp,istrata,iind)
                     END IF
                     IF (mrpw(1,icls,iwt,ishp,istrata,iind).ne.0) THEN
                        WRITE(50,53) icls,iwt,ishp,istrata,iind,
     +                     (mrpw(irpw,icls,iwt,ishp,istrata,iind),irpw=1,nrpw)
                        nbrevout =nbrevout+mrpw(1,icls,iwt,ishp,istrata,iind)
                        nbpcsout =nbpcsout+mrpw(2,icls,iwt,ishp,istrata,iind)
                     stratarev(istrata,1) =
     +                     mrpw(1,icls,iwt,ishp,istrata,iind)
     +                     + stratarev(istrata,1)
                     stratapcs(istrata,1) =
     +                     mrpw(2,icls,iwt,ishp,istrata,iind)
     +                     + stratapcs(istrata,1)
                     clspcs(icls) = clspcs(icls) +
     +                     mrpw(2,icls,iwt,ishp,istrata,iind)
                     END IF
                  END DO
               END DO
            END DO
         END DO
      END DO

      OPEN(7,FILE='clstitles.dat')
      PRINT *, "Pieces by Class:"
      DO icls=1,nnewcls
         READ(7,'(A24)') clstitle(icls)
         WRITE (6,'(I2,1X,A24,1X,F15.0)') icls,clstitle(icls),clspcs(icls)
      END DO

C     Write Transaction Matrices

      DO iind=1,nind
         DO istrata=1,nstrata
            DO isize=1,nsize
               DO ishp=1,nshp
                  IF (trp(1,ishp,isize,istrata,iind).gt.0)
     +                WRITE(60,61) isize,ishp,istrata,iind,
```

```
     +                (trp(itrp,ishp,isize,istrata,iind),itrp=1,ntrp)
                   stratarev(istrata,2) = trp(1,ishp,isize,istrata,iind)
     +                + stratarev(istrata,2)
                   stratapcs(istrata,2) = trp(2,ishp,isize,istrata,iind)
     +                + stratapcs(istrata,2)
               END DO
            END DO
         END DO
      END DO

      DO isize = 1,nsize
         DO istrata = 1,nstrata
            DO iind = 1, nind
                  ttrans = ttrans + trans(isize,istrata,iind)
            END DO
         END DO
      END DO

C     Rollup checks by strata
      DO istrata=1,nstrata
         DO i=1,2
            stratarev(nstrata+1,i) = stratarev(nstrata+1,i) +
     +         stratarev(istrata,i)
            stratapcs(nstrata+1,i) = stratapcs(nstrata+1,i) +
     +         stratapcs(istrata,i)
         END DO
      END DO

C     Write rollup totals to report file

      WRITE(80,83)
      WRITE(80,'(70("-"))')
86    FORMAT(A30," : ",3F12.0)
      WRITE(80,86) errorname(1),igrevout,igpcsout,igwtout
      WRITE(80,86) errorname(2),ibrevout,ibpcsout
      WRITE(80,86) errorname(3),nbrevout,nbpcsout
      WRITE(80,'(70("-"))')
C                  12345678901234567890123456789012345678901234567890
      WRITE(80,'("Strata    Rev by VIP   Rev by Size")')
      DO istrata=1,nstrata+1
         WRITE(80,'(I2,8X,2F12.0)') istrata,(stratarev(istrata,i),i=1,2)
      END DO
      WRITE(80,'(70("-"))')
      WRITE(80,'("Strata    Pcs by VIP   Pcs by Size")')
      DO istrata=1,nstrata+1
         WRITE(80,'(I2,8X,2F12.0)') istrata,(stratapcs(istrata,i),i=1,2)
      END DO
      WRITE(80,'(70("-"))')
      WRITE(80,'("Total Transactions: ",F10.0)') ttrans
      WRITE(80,'("GOVERNMENT Transactions: ",i10)') govct


C     Write Finrev Files
C     Write Bad Revenue

      OPEN(70,FILE='finrev.lst.'//cap)
71    FORMAT(F20.5,":",A6,4F20.5)
      DO ifin=1,nfin
         WRITE(70,71) finrev(ifin),fins(ifin),ppirev(ifin)+badrev(ifin,1),
     +    smrev(ifin)+badrev(ifin,2)
         tsmout = tsmout + smrev(ifin) + badrev(ifin,2)
         tpiout = tpiout + ppirev(ifin) + badrev(ifin,1)
         bsmout = bsmout + badrev(ifin,2)
      END DO
      CLOSE(70)

      WRITE(80,'("Total SM Revenue: ",F20.5)') tsmout
      WRITE(80,'("Total Bad SM Rev: ",F20.5)') bsmout
      WRITE(80,'("Total PI Revenue: ",F20.5)') tpiout

      CLOSE(80)

      PRINT *, "Nonsurchrge VIP count = ",nstd
      PRINT *, "Transactions with bad vips = ", badtran
      PRINT *, "All done with AP",cap,"."

      CALL time(timestr)
      PRINT *,"End: ",timestr

      STOP
```

```
      END

C-----------------------------------------------------------------------

      SUBROUTINE ver1st(error,istrata,vipsvalid,rev,pcs,wt,trev,
     +   tpcs,lines,ishp,pcwt)

      INTEGER*4 istrata,lines,ishp,maxvip,nvip,i,weight_incr,iwt,cx
      INTEGER*4 nodes,ntype,apply_rates,nwt,maxerr,searchc
      PARAMETER (maxvip=20,nvip=102,nwt=13,nodes=34,ntype=12,maxerr=11)
      REAL*8    rtol,minltr,minflt
      PARAMETER (rtol=0.002,minltr=0.0063,minflt=0.0125)
      INTEGER*4 error(maxvip)
      REAL*8    rev,pcs,wt,trev,tpcs,pcwt,testrate,nextrate,tpcwt
      LOGICAL   vipsvalid,weight,rateok,notcard,nonstd,idmail,found,same
      CHARACTER*5 tmpvip

C     Verification Common Block
      CHARACTER*5 vips(nvip)
      INTEGER*4 recvip(maxvip),reciwt(maxvip),vipcount(nodes),tcount(ntype)
      REAL*8    recrev(maxvip),recpcs(maxvip),recwt(maxvip),recrte(maxvip)
      REAL*8    rate(nvip,3),ratetable(nvip,nwt),comprate(maxvip)
      REAL*8    newratetable(nvip,nwt)
      INTEGER*4 vip_to_rate(nvip),maxwt(nvip),recnew(maxvip), ishape(maxvip)
      COMMON /verify/recvip,reciwt,recrev,ishape,
     +   recpcs,recwt,recrte,rate,
     +   vips,ratetable,vipcount,tcount,
     +   vip_to_rate,comprate,maxwt,recnew

      weight = .TRUE.
      rateok = .TRUE.
      notcard = .TRUE.
      nonstd = .FALSE.
      idmail = .TRUE.
      tpcwt = wt/pcs

      DO i=1,maxvip
         error(i) = 0
      END DO

      IF (vipsvalid) THEN

         IF (istrata.gt.0) THEN

C     First Determine Shape Information    REVISED for reclass

            DO i=1,lines
               IF (recvip(i).ne.0) THEN
                  IF  (vips(recvip(i))(5:5).eq.'3')THEN     ! cards
                     ishp = 1                 ! may have 0 on tape
                     notcard = .FALSE.
                  END IF
                  IF (vips(recvip(i))(2:5).eq.'1642'.or.
     +      vips(recvip(i))(2:5).eq.'1652'.or.vips(recvip(i))(2:5).eq.'1674'.or.
     +      vips(recvip(i))(2:5).eq.'1861'.or.vips(recvip(i))(2:5).eq.'1871'.or.
     +      vips(recvip(i))(2:5).eq.'1862'.or.vips(recvip(i))(2:5).eq.'1872'.or.
     +      vips(recvip(i))(2:5).eq.'1864'.or.vips(recvip(i))(2:5).eq.'1874'.or.
     +      vips(recvip(i))(2:5).eq.'1881'.or.vips(recvip(i))(2:5).eq.'1882'.or.
     +      vips(recvip(i))(2:5).eq.'1884')
     +            THEN

                     nonstd = .TRUE.
                  END IF
               END IF
            END DO

C     Now Determine if Total Revenue and VIP revenue agree within 5%

            IF (ABS(trev/rev-1).le.0.05) THEN ! Revenue sum okay
               IF (ABS(trev/rev-1).gt.0.0.and.ABS(trev/rev-1).le.0.05)
     +            rev = trev     ! If error within tol., switch rev.
            ELSE
               DO i=1,lines
C                 We will keep these records 5/1/95
C                 error(i) = 9 ! Inconsistent Revenue Error
C                 comprate(i) = recrev(i)/recpcs(i)
                  vipcount(31) = vipcount(31) + 1
               END DO
            END IF
```

```
C     Next compare VIP pieces to total pieces

            IF (ABS(tpcs-pcs).gt.0.01) THEN
               DO i=1,lines
                  error(i) = 10 ! Inconsistent Pieces Error
                  if (recpcs(i).ne.0.0) then
                     comprate(i) = recrev(i)/recpcs(i)
                  else
                     comprate(i) = 0
                  end if
                  vipcount(32) = vipcount(32) + 1
               END DO
            END IF

         ELSE
            DO i=1,lines
               error(i) = 8     ! Invalid Finance Number Error
               vipcount(30) = vipcount(30) + 1
            END DO
         END IF

      ELSE      ! bad vip
         DO i=1,lines
            error(i) = 7        ! Invalid VIP Error; entire tran is ignored
            vipcount(29) = vipcount(29) + 1
         END DO
      END IF

C      Now Determine if the mailing is identical or non-identical

      IF (error(1).eq.0) THEN      ! Is the mailing valid so far?

C        Test for invalid weight information

         IF (weight_incr(wt,pcs,pcwt).le.0.or.
     +       weight_incr(wt,pcs,pcwt).gt.13) THEN
            weight = .FALSE.
         ELSE
            IF (nonstd) THEN
               weight = .TRUE.
            ELSE IF (wt/pcs.lt.minltr.and.ishp.eq.1.and.notcard) THEN
               weight = .FALSE.
            ELSE IF (wt/pcs.lt.minflt.and.ishp.gt.1) THEN
               weight = .FALSE.
            END IF
         END IF

         IF (pcwt.ne.0.0) THEN             ! identical wt mailing
            tcount(1) = tcount(1) + 1
            IF (nonstd) THEN
               tcount(5) = tcount(5) + 1
               tcount(2) = tcount(2) + 1
            ELSE IF (.not.notcard) THEN
               tcount(4) = tcount(4) + 1
               tcount(2) = tcount(2) + 1
            ELSE IF (weight) THEN
               tcount(3) = tcount(3) + 1
               tcount(2) = tcount(2) + 1
            ELSE
               tcount(6) = tcount(6) + 1
            END IF

            DO i=1,lines
               IF (recvip(i).ne.0) THEN
                  vipcount(1) = vipcount(1) + 1 ! Node #1
                  IF (weight) THEN ! Good Weight
                     vipcount(2) = vipcount(2) + 1 ! Node #2
                     recwt(i) = recpcs(i)*pcwt
                     reciwt(i) = weight_incr(wt,pcs,pcwt)
C                    Check the rate paid against the rate for that VIP.
                     testrate = ratetable(recvip(i),reciwt(i))
                     if (recpcs(i).ne.0.0) then
                        comprate(i) = recrev(i)/recpcs(i)
                     else
                        comprate(i) = 0
                     end if
                     IF (ABS(comprate(i)-testrate).gt.rtol) THEN
c                       Rate doesn't match
                        error(i) = 4
                        vipcount(4) = vipcount(4) + 1
```

```
                  PRINT *," Bad Rate: VIP =",vips(recvip(i)),
     +               " iwt = ",reciwt(i)," actual = ",
     +               comprate(i)," computed = ",testrate
C                   print*," recrev = ",recrev(i),"  recpcs = ",recpcs(i)
                 ELSE
                    error(i) = 1
                    vipcount(3) = vipcount(3) + 1
                 END IF
              ELSE              ! Bad Weight
                 vipcount(5) = vipcount(5) + 1 ! Node #5
                 if (recpcs(i).ne.0.0) then
                    comprate(i) = recrev(i)/recpcs(i)
                 else
                    comprate(i) = 0
                 end if
                 found = .FALSE.
                 iwt = 1
                 DO WHILE (iwt.le.13.and.(.not.found))
                    testrate = ratetable(recvip(i),iwt)
                    IF (ABS(comprate(i)-testrate).le.rtol) THEN
                       found = .TRUE.
                    ELSE
                       iwt = iwt + 1
                    END IF
                 END DO
                 IF (found) THEN
                    vipcount(7) = vipcount(7) + 1
                    error(i) = 2
                    reciwt(i) = iwt
                 ELSE
                    vipcount(6) = vipcount(6) + 1
                    error(i) = 5
                 END IF
              END IF
           ELSE              ! Nonstandard Surcharge VIP Junk
              vipcount(33) = vipcount(33) + 1 ! Node #30
              error(i) = 11
           END IF
        END DO ! Loop over VIPS

     ELSE                    ! Piece Weight = 0.0000, test if identical

        tcount(7) = tcount(7) + 1
        IF (nonstd) THEN
           tcount(11) = tcount(11) + 1
           tcount(8) = tcount(8) + 1
        ELSE IF (.not.notcard) THEN
           tcount(10) = tcount(10) + 1
           tcount(8) = tcount(8) + 1
        ELSE IF (weight) THEN
           tcount(9) = tcount(9) + 1
           tcount(8) = tcount(8) + 1
        ELSE
           tcount(12) = tcount(12) + 1
        END IF

        DO i=1,lines
           IF (recvip(i).gt.0) THEN
              vipcount(8) = vipcount(8) + 1
              if (recpcs(i).ne.0.0) then
                 comprate(i) = recrev(i)/recpcs(i)
              else
                 comprate(i) = 0
              end if
              found = .FALSE.
              iwt = 1
              cx = 0

              IF (((vips(recvip(i))(5:5).eq.'3').or.
     +             (vips(recvip(i))(2:5).eq.'1642'.or.
     +        vips(recvip(i))(2:5).eq.'1652'.or.vips(recvip(i))(2:5).eq.'1674'.or.
     +        vips(recvip(i))(2:5).eq.'1861'.or.vips(recvip(i))(2:5).eq.'1871'.or.
     +        vips(recvip(i))(2:5).eq.'1862'.or.vips(recvip(i))(2:5).eq.'1872'.or.
     +        vips(recvip(i))(2:5).eq.'1881'.or.vips(recvip(i))(2:5).eq.'1882'.or.
     +        vips(recvip(i))(2:5).eq.'1884'.or.
     +        vips(recvip(i))(2:5).eq.'1864'.or.vips(recvip(i))(2:5).eq.'1874')))
     +              THEN        ! card or NS

                 testrate = ratetable(recvip(i),1)   ! check 1 oz rate
                 IF (comprate(i).ge.(testrate-rtol).and.
```

```
+                  comprate(i).le.(testrate+rtol)) THEN

                   reciwt(i) = 1    ! assign as 1 oz
                   found = .TRUE.
                END IF
             ELSE                    ! not card or NS
C               * look for 'highest' wt increment
                DO WHILE (iwt.le.13.and.(cx.ne.2))
                   IF (iwt.le.12) THEN
                      IF (iwt.eq.1) THEN
                testrate = ratetable(recvip(i),1)    ! check 1 oz rate
                         IF (comprate(i).ge.(testrate-rtol).and.
+                           comprate(i).le.(testrate+rtol)) THEN
                            reciwt(i) = 1
                            cx = 2
                            found = .TRUE.
                         END IF
                      ELSE
                         testrate = ratetable(recvip(i),iwt-1)
                         nextrate = ratetable(recvip(i),iwt)
                         IF (comprate(i).ge.(testrate-rtol).and.
+                           comprate(i).lt.(nextrate+rtol)) THEN
                            reciwt(i) = iwt
                            cx = cx + 1
                            found = .TRUE.
                         END IF
                      END IF
                      iwt = iwt + 1
                   ELSE       ! 13 oz check
                      testrate = ratetable(recvip(i),13)  ! NOTE
                      cx = 2
                      IF (comprate(i).ge.(testrate-rtol).and.
+                        comprate(i).le.(testrate+rtol)) THEN
                         reciwt(i) = 13
                         found = .TRUE.
                      END IF
                   END IF
                END DO
             END IF

             IF (found) THEN ! found a rate region match
                vipcount(10) = vipcount(10) + 1
                IF ((vip_to_rate(recvip(i)).ge.21.and. ! AUTO
+                  vip_to_rate(recvip(i)).le.28).or.
+                  (vip_to_rate(recvip(i)).ge.31.and.
+                  vip_to_rate(recvip(i)).le.34)) THEN
                   vipcount(20) = vipcount(20) + 1
                   IF ((reciwt(i).le.4.and.comprate(i).ge.
+                     (ratetable(recvip(i),reciwt(i))-rtol)).and.
+                     (comprate(i).le.
+                     (ratetable(recvip(i),reciwt(i))+rtol))) THEN
                      vipcount(22) = vipcount(22) + 1
                      IF (lines.gt.1) THEN
                         vipcount(23) = vipcount(23) + 1
                         error(i) = 2
                      ELSE
                         vipcount(24) = vipcount(24) + 1
                         IF (.not.weight) THEN
                            vipcount(25) = vipcount(25) + 1
                            error(i) = 2
                         ELSE
                            vipcount(26) = vipcount(26) + 1
                            IF (weight_incr(wt,pcs,tpcwt).
+                              eq.reciwt(i)) THEN
                               recwt(i) = recpcs(i)*tpcwt
                               error(i) = 1
                               vipcount(27) = vipcount(27) + 1
                            ELSE
                               error(i) = 2
                               vipcount(28) = vipcount(28) + 1
                            END IF
                         END IF
                      END IF
                   ELSE
                      vipcount(21) = vipcount(21) + 1
                      error(i) = 3
                   END IF
                ELSE            ! Not AUTO
                   vipcount(11) = vipcount(11) + 1
                   IF (comprate(i).ge.(ratetable(recvip(i),1)-rtol).and.
```

```
     +                    comprate(i).le.(ratetable(recvip(i),1)+rtol)) THEN
                          vipcount(13) = vipcount(13) + 1
                          IF (lines.eq.1) THEN
                             vipcount(15) = vipcount(15) + 1
                             IF (.not.weight) THEN
                                vipcount(16) = vipcount(16) + 1
                                error(i) = 2
                             ELSE
                                vipcount(17) = vipcount(17) + 1
                                IF (weight_incr(wt,pcs,tpcwt).
     +                             eq.reciwt(i)) THEN
                                   recwt(i) = recpcs(i)*tpcwt
                                   error(i) = 1
                                   vipcount(18) = vipcount(18) + 1
                                ELSE
                                   vipcount(19) = vipcount(19) + 1
                                   error(i) = 2
                                END IF
                             END IF
                          ELSE
                             vipcount(14) = vipcount(14) + 1
                             error(i) = 2
                          END IF
                       ELSE
                          vipcount(12) = vipcount(12) + 1
                          error(i) = 3
                       END IF
                    END IF
                 ELSE
C                     PRINT *, "Cannot Match Rate: ",
C     +                  vips(recvip(i))," rev/pcs: ",comprate(i),
C     +                  " min: ",ratetable(recvip(i),1),
C     +                  " max: ",ratetable(recvip(i),maxwt(recvip(i))),
C     +                  " oz: ",wt/pcs*16.0
                    vipcount(9) = vipcount(9) + 1
                    error(i) = 6
                 END IF
              ELSE              ! bad VIP
                 vipcount(33) = vipcount(33) + 1
                 error(i) = 11
              END IF
           END DO
C Check to see if all VIPs paid exactly the same ounce rate.
C If so, then make them identical good. If not, leave the error vector
C unchanged
           IF (lines.gt.1.and.reciwt(1).le.4.and.
     +        (comprate(1).ge.(ratetable(recvip(1),reciwt(1))-rtol).
     +        and.comprate(1).le.(ratetable(recvip(1),reciwt(1))+rtol))) THEN
              i = 2
              same = .TRUE.
              DO WHILE (i.le.lines.and.same)
                 IF (recvip(i).ne.0) THEN
                    IF (comprate(i).ge.(ratetable(recvip(i),reciwt(1))-rtol).
     +                 and.comprate(i).le.
     +                 (ratetable(recvip(i),reciwt(1))+rtol)) THEN
                       same = .TRUE.
                    ELSE
                       same = .FALSE.
                    END IF
                 END IF
                 i = i + 1
              END DO
              IF (same) THEN
                 IF (weight) THEN
                    IF (weight_incr(wt,pcs,tpcwt).eq.reciwt(1)) THEN
                       DO i=1,lines
                          IF (recvip(i).ne.0) THEN
                                recwt(i) = tpcwt*recpcs(i)
                                error(i) = 1
                                vipcount(34) = vipcount(34) + 1
                             IF ((vip_to_rate(recvip(i)).ge.21.and. ! AUTO
     +                          vip_to_rate(recvip(i)).le.28).or.
     +                          (vip_to_rate(recvip(i)).ge.31.and.
     +                          vip_to_rate(recvip(i)).le.34)) THEN
                                vipcount(23) = vipcount(23) - 1
                             ELSE
                                IF (reciwt(i).gt.1) THEN
                                   vipcount(12) = vipcount(12) - 1
                                ELSE
                                   vipcount(14) = vipcount(14) - 1
```

```
                              END IF
                            END IF
                          END IF
                        END DO
                      END IF
                    END IF
                  END IF
                END IF
              END IF
            END IF
          END IF

          RETURN
          END

C---  ------------------------------------------------------------------------

          INTEGER*4 FUNCTION mail_size(pieces,lines,ishp)

C         Purpose: Given the number of pieces in a mailing, returns the
C                  the size index for the mailing.  Also assigns shape and
C                   recnew array.

C         Program Parameters
          INTEGER*4 maxvip,maxerr,nvip,nfin,nmap,nnewcls,nrpw,ntrp
          PARAMETER (maxvip=20,maxerr=11,nvip=102,nnewcls=34,
         +   nfin=10815,nrpw=3,ntrp=3)
          INTEGER*4 nshp,nwt,nsize,nstrata,nodes,ntype,nind
          PARAMETER (nshp=3,nwt=13,nsize=3,nstrata=20,nodes=34,ntype=12,nind=2)

          INTEGER*4 lines,i,ishp,irate,seq

C         Verification Common Block
          INTEGER*4 recvip(maxvip),reciwt(maxvip),vipcount(nodes),tcount(ntype)
          REAL*8    recrev(maxvip),recpcs(maxvip),recwt(maxvip),recrte(maxvip)
          REAL*8    rate(nvip,3),ratetable(nvip,nwt),comprate(maxvip)
          REAL*8    sizerpt(nind,nsize,nsize)
          REAL*8    pieces
          INTEGER*4 vip_to_rate(nvip),maxwt(nvip),recind(maxvip),recnew(maxvip)
          INTEGER*4 ishape(maxvip)
          CHARACTER*5 vips(nvip)
          CHARACTER*1 psflag

          COMMON /verify/recvip,reciwt,recrev,ishape,
         +   recpcs,recwt,recrte,rate,
         +   vips,ratetable,vipcount,tcount,
         +   vip_to_rate,comprate,maxwt,recnew
          COMMON /trantype/sizerpt


          IF (pieces.le.5000) THEN
             mail_size = 1
          ELSE IF (pieces.le.10000) THEN
             mail_size = 2
          ELSE
             mail_size = 3
          END IF

C         Assign shape

          DO i=1,lines
             IF (recvip(i).ne.0) THEN
                irate = vip_to_rate(recvip(i))
                IF (irate.eq.1) THEN
                   recnew(i) = 1
                   ishape(i) = 1
                ELSE IF (irate.eq.2) THEN
                   recnew(i) = 2
                   ishape(i) = 2
                ELSE IF (irate.eq.3) THEN
                   recnew(i) = 3
                   ishape(i) = 3
                ELSE IF (irate.eq.4) THEN
                   recnew(i) = 4
                   ishape(i) = 3
                ELSE IF (irate.eq.5) THEN
                   recnew(i) = 5
                   ishape(i) = 1
                ELSE IF (irate.eq.6) THEN
                   recnew(i) = 6
                   ishape(i) = 2
```

```
ELSE IF (irate.eq.7) THEN
   recnew(i) = 7
   ishape(i) = 3
ELSE IF (irate.eq.8) THEN
   recnew(i) = 8
   ishape(i) = 1
ELSE IF (irate.eq.9) THEN
   recnew(i) = 9
   ishape(i) = 2
ELSE IF (irate.eq.10) THEN
   recnew(i) = 10
   ishape(i) = 3
ELSE IF (irate.eq.11) THEN
   recnew(i) = 11
   ishape(i) = 3
ELSE IF (irate.eq.12) THEN
   recnew(i) = 12
   ishape(i) = 1
ELSE IF (irate.eq.13) THEN
   recnew(i) = 13
   ishape(i) = 2
ELSE IF (irate.eq.14) THEN
   recnew(i) = 14
   ishape(i) = 3
ELSE IF (irate.eq.15) THEN
   recnew(i) = 15
   ishape(i) = 1
ELSE IF (irate.eq.16) THEN
   recnew(i) = 16
   ishape(i) = 2
ELSE IF (irate.eq.17) THEN
   recnew(i) = 17
   ishape(i) = 3
ELSE IF (irate.eq.18) THEN
   recnew(i) = 18
   ishape(i) = 1
ELSE IF (irate.eq.19) THEN
   recnew(i) = 19
   ishape(i) = 2
ELSE IF (irate.eq.20) THEN
   recnew(i) = 20
   ishape(i) = 3
ELSE IF (irate.eq.21) THEN
   recnew(i) = 21
   ishape(i) = 1
ELSE IF (irate.eq.22) THEN
   recnew(i) = 22
   ishape(i) = 1
ELSE IF (irate.eq.23) THEN
   recnew(i) = 23
   ishape(i) = 1
ELSE IF (irate.eq.24) THEN
   recnew(i) = 24
   ishape(i) = 1
ELSE IF (irate.eq.25) THEN
   recnew(i) = 25
   ishape(i) = 2
ELSE IF (irate.eq.26) THEN
   recnew(i) = 26
   ishape(i) = 2
ELSE IF (irate.eq.27) THEN
   recnew(i) = 27
   ishape(i) = 2
ELSE IF (irate.eq.28) THEN
   recnew(i) = 28
   ishape(i) = 2
ELSE IF (irate.eq.29) THEN
   recnew(i) = 29
   ishape(i) = 1
ELSE IF (irate.eq.30) THEN
   recnew(i) = 30
   ishape(i) = 1
ELSE IF (irate.eq.31) THEN
   recnew(i) = 31
   ishape(i) = 1
ELSE IF (irate.eq.32) THEN
   recnew(i) = 32
   ishape(i) = 1
ELSE IF (irate.eq.33) THEN
   recnew(i) = 33
```

```
            ishape(i) = 1
         ELSE IF (irate.eq.34) THEN
            recnew(i) = 34
            ishape(i) = 1

         END IF
      END IF
   END DO

   RETURN
   END

C--------------------------------------------------------------------------------
   INTEGER*4 FUNCTION weight_incr(weight,pieces,pcwt)

   INTEGER*4 iwt,nwt,ioz
   PARAMETER (nwt=13)
   REAL*8 weight,pieces,pcwt,ozs,tpcwt

   weight_incr = 0
   IF (pcwt.eq.0) THEN
      tpcwt = weight/pieces
   ELSE
      tpcwt = pcwt
   END IF
   ozs = tpcwt*16.0
   ioz = int(ozs)
   IF ((ozs-dble(ioz)).gt.0.0) ioz = ioz + 1
   IF ((ioz.ge.1).and.(ioz.le.13)) THEN
      weight_incr = ioz
   ELSE
      weight_incr = 0
   END IF

   RETURN
   END
C--------------------------------------------------------------------------------
   INTEGER*4 FUNCTION indicia(vip,icls)

   CHARACTER*5 vip
   INTEGER*4 icls

      IF (vip(1:1).eq.'1'.or.vip(1:1).eq.'2') THEN
         indicia = 2       ! SM
      ELSE IF (vip(1:1).eq.'0') THEN
         indicia = 1       ! PI
      ELSE
         PRINT *, "NEW VIP ",vip," doesn't correspond to known indicia."
         STOP
      END IF


   RETURN
   END
```

```
      PROGRAM perhalf

C     DESCRIPTION: verifies and rolls up first class PERMIT system records
C                  to strata, mailing size, VIP, weight increment, & shape
C
C     CREATED BY:        Bill Humphries
C     DATE:              Thu Sep 15 10:06:39 CDT 1994
C
C     LAST MODIFIED BY: Tom Ayen
C     DATE:
C
C     This program only deals with records in which the pcwt > 0 for the
C      PERMIT record.  It compiles data by half oz increments to 4 oz.
C
C
C     NOTE : update nfin below AND in function mail_size. If any other params
C               updated check the functions also.

      IMPLICIT NONE

C     Program Parameters
      INTEGER*4 maxvip,maxerr,nvip,nfin,nmap,nrpw,ntrp,nnewcls,size1
      PARAMETER (maxvip=20,maxerr=11,nvip=102,
     +   nfin=10815,nrpw=3,ntrp=2,nnewcls=34)      ! update nfin
      INTEGER*4 nshp,nwt,nsize,nstrata,nodes,ntype,nind,iap,nwtold
      PARAMETER (nshp=3,nwt=17,nsize=3,nstrata=20,nodes=34,ntype=12,nind=2,nwtold=15)
      REAL*8    zero
      PARAMETER (zero=0.0)
      parameter (size1=nind*nstrata*nsize)

C     Maps
      CHARACTER*6 fins(nfin)
      INTEGER*4 stratamap(nfin)

C     Indicies and Counters
      INTEGER*4 ifin,istrata,iwt,isize,ivip,ier,nstd,irpw,itrp,iind,cx
      INTEGER*4 i,j,k,error(maxvip),icls
      INTEGER*4 finreass/0/,xminind
      INTEGER*4 minind(1:20)
      LOGICAL   vipsvalid,presort,imprint,firstokay
      CHARACTER*30 errorname(maxerr)
      CHARACTER*30 vipname(nodes)
      CHARACTER*30 tname(ntype)
      CHARACTER*2 cap

C     Functions
      INTEGER*4 searchc,mail_size,weight_incr,indicia

C     Data read from PERMIT record
      CHARACTER*5 ind
      CHARACTER*5 vip
      CHARACTER*6 finno,lfinno
      CHARACTER*30 unique
      CHARACTER*960 vipfields
      REAL*8 rev,pcs,wt,pcwt,fees,viprev,vippcs,viprate,pcsout
      INTEGER*4 ishp,lines,year,day,seq,trandate
      CHARACTER*1 psflag
      CHARACTER*5 pmtnum

C     Verification Common Block
      INTEGER*4 recvip(maxvip),reciwt(maxvip),vipcount(nodes),tcount(ntype)
      REAL*8    recrev(maxvip),recpcs(maxvip),recwt(maxvip),recrte(maxvip)
      REAL*8    rate(nvip,3),ratetable(nvip,nwt),comprate(maxvip)
      REAL*8    ratetable1(nvip,nwt),ratetable2(nvip,nwt)
      REAL*8    sizerpt(nind,nsize,nsize)
      INTEGER*4 vip_to_rate(nvip),maxwt(nvip),recind(maxvip),recnew(maxvip)
      INTEGER*4 ishape(maxvip)
      CHARACTER*5 vips(nvip)
      COMMON /verify/recvip,reciwt,recrev,ishape,
     +   recpcs,recwt,recrte,rate,
     +   vips,ratetable,vipcount,tcount,
     +   vip_to_rate,comprate,maxwt,recnew
      COMMON /trantype/sizerpt
      REAL*8 freeresids,nfresid,freenstdres,nfnstd,freezip4,nfzip4
      COMMON /freeflt/freeresids,nfresid,freenstdres,nfnstd,freezip4,nfzip4,
     +   seq,psflag

C     Rate Maps
      REAL*8    newratetable(nvip,nwt)
```

```
C     Error Tracking
      REAL*8 errrev(maxerr)/maxerr*0.0/,errpcs(maxerr)/maxerr*0.0/
      REAL*8 errwt(maxerr)/maxerr*0.0/
      INTEGER*4 errcount(maxerr)/maxerr*0/


C     Running Totals
      INTEGER*4 recin/0/,okrec/0/,badtran/0/,nvipvaild/0/,nvipinvalid/0/
      INTEGER*4 govct/0/
      REAL*8 revin/0.0/,pcsin/0.0/,wtin/0.0/
      REAL*8 okrev/0.0/,okpcs/0.0/,okwt/0.0/
      REAL*8 igrevout/0.0/,ibrevout/0.0/,ngrevout/0.0/,nbrevout/0.0/
      REAL*8 igpcsout/0.0/,ibpcsout/0.0/,ngpcsout/0.0/,nbpcsout/0.0/
      REAL*8 igwtout/0.0/,ngwtout/0.0/,ttrans/0.0/
      REAL*8 trev,tpcs
      REAL*8 stratarev(nstrata+1,2)/42*0.0/
      REAL*8 stratapcs(nstrata+1,2)/42*0.0/
      REAL*8 finrev(nfin),ppirev(nfin),smrev(nfin),sprev(nfin)
      REAL*8 clspcs(nnewcls)/nnewcls*0.0/
      REAL*8 tsmout/0.0/,tpiout/0.0/,bsmout/0.0/,tspout/0.0/

C     RPW and Transaction Arrays
C     Arrays which accumulate by VIP, weight and shape use the indexing:
C       irpw - 1: Revenue, 2: Pieces, 3: Weight
C     Arrays which accumulate by size and shape of mailing use the indexing:
C       irpw = 1: Transactions, 2: Revenue, 3: Pieces

      REAL*8    rpw(nrpw,nnewcls,nwt,nshp,nstrata,nind)
      REAL*8    brpw(nrpw,nnewcls,nwt,nshp,nstrata,nind)
      REAL*8    mrpw(nrpw,nnewcls,nwt,nshp,nstrata,nind)
      REAL*8    trp(ntrp,nshp,nsize,nstrata,nind)
      INTEGER*4 trans(nsize,nstrata,nind)/size1*0/
      REAL*8    badrev(nfin,nind)

      CHARACTER*8 timestr
      CHARACTER*24 clstitle(nnewcls)

C     INITIALIZATIONS

      CALL time(timestr)
      PRINT *,"Start: ",timestr

C     Read AP from Command Line
      CALL getarg(1,cap)
      READ (cap,'(I2)') iap

      OPEN(7,FILE='titles.pmt',readonly)

C     Set Captions for Decision Tree Counter

      DO i=1,34
         READ(7,'(A30)') vipname(i)
      END DO

      DO i=1,nodes
         vipcount(i) = 0
      END DO

C     Error Captions

      DO i=1,11
         READ(7,'(A30)') errorname(i)
      END DO

      DO i=1,maxerr
         errcount(i) = 0
      END DO

C     Transaction Captions

      DO i=1,12
         READ(7,'(A30)') tname(i)
      END DO

      DO i=1,ntype
         tcount(i) = 0
      END DO

      nstd = 0
```

```
C     Read List of Valid Finance Numbers and associated Strata

      OPEN(1,FILE='finstrata.new.00',readonly) ! update

2     FORMAT(a6,1x,3i3)
      DO i = 1, nfin
         READ(1,2) fins(i), stratamap(i)
      END DO

      PRINT *, "Read Strata Maps."

C     Read List of Valid VIP Codes and Map to Rate Elements

      OPEN(7,FILE='vipmap.1st.00',readonly)
6     FORMAT(4X,A5,3X,I9)
      DO i=1,nvip
         READ(7,6) vip,j
         IF (vip(1:1).eq.' ') vip(1:1) = '0'
         vip_to_rate(i) = j
         vips(i) = vip
      END DO
      CLOSE(7)

      PRINT *, "Read VIP Map."


C     Read List of NEW Rates for VIP

      OPEN(9,FILE='ratehalf.00',readonly)
10    format(6x,i2,1x,17f6.3)

      DO i=1,nvip
         read(9,10) maxwt(i), (ratetable(i,iwt), iwt = 1, nwt)
      END DO
      PRINT *, "read New rate list."

      CLOSE(9)


C     Initialize RPW Arrays
C     Treating each array separately for effeciency.

      do irpw = 1, nrpw
         do icls = 1, nnewcls
            do iwt = 1, nwt
              do ishp =1, nshp
               do istrata = 1,nstrata
                  do iind = 1, nind
                     rpw(irpw,icls,iwt,ishp,istrata,iind) = 0.0
                     brpw(irpw,icls,iwt,ishp,istrata,iind) = 0.0
                     mrpw(irpw,icls,iwt,ishp,istrata,iind) = 0.0
                  end do
               end do
              end do
            end do
         end do
       end do

      do irpw = 1, ntrp
         do ishp = 1, nshp
            do isize = 1, nsize
               do istrata = 1,nstrata
                  do iind = 1, nind
                     trp(irpw,ishp,isize,istrata,iind) = 0.0
                  end do
               end do
            end do
         end do
       end do

       do i = 1,nfin
          finrev(i) = 0.0
          ppirev(i) = 0.0
          smrev(i) = 0.0
          sprev(i) = 0.0
          do j = 1,nind
             badrev(i,j) = 0.0
          end do
       end do
```

```
      PRINT *, "Initialized Arrays, Starting Main Loop."

C     MAIN LOOP
C     Read a complete PERMIT record. Assign the VIP, revenue, pieces & weight
C     from each VIP field to the record array. Determine if record is an
C     identical or non-identical transaction. Assign weight increment, shape.

C      PERMIT AP File now read from STDOUT stream of gzcat process with perloop.

11    FORMAT(A6,A2,A30,1X,F12.2,2X,F12.2,12X,A1,I1,1X,F8.4,F12.0,F14.4,20X,I3,
     +   A960)
12    FORMAT(6X,A5,1X,F6.3,F12.0,F18.7)
13    FORMAT(A6,1X,F12.2,F12.2,1X,I1,1X,F8.4,F12.0,F14.4,1X,I3,1X,A48,F6.3)

      open(15,file='permit.1st.tmp',readonly)
      ier = 0
      lfinno = 'xxxxxx'

C Read a PERMIT system record. Add the total revenue, pieces and weight
C to the running totals.

C      open(28,file='test',recl=5008,iointent='input')
C      open(65,file='badvips.'//cap,recl=2000,iointent='output')
C      open(75,file='fl_11_13.'//cap,recl=2000,iointent='output')

      DO WHILE (ier.eq.0)
         READ(15,11,IOSTAT=ier,END=99) finno,ind,unique,rev,fees,psflag,
     +      ishp,pcwt,pcs,wt,lines,vipfields
         READ(unique,'(I4,I3,6X,I2,2X,A5,I8)') year,day,seq,pmtnum,trandate
         trev = 0.0
         tpcs = 0.0
         pcsout = 0.0
         imprint = .FALSE.
         firstokay = .TRUE.
         recin = recin + 1
         rev = rev - fees       ! subtract difference for non-qualifying pieces
         revin = revin + rev
         wtin  = wtin + wt
C   edit for new aviion
         if (rev.eq.0.0) rev = 1.0
         if (pcs.eq.0.0) pcs = 1.0

         IF (lines.gt.maxvip) THEN
            PRINT *, "Warning: maxvip exceeded with value: ",lines,
     +         " at record ",recin,"."
            STOP
         END IF

         IF (pmtnum(1:1).eq.'G') THEN       ! government mail
            govct = govct + 1
            go to 200
         END IF

C Compare the current to the previously read finance number. If it's new,
C locate the finance number in the strata map. If not found, then look for
C it in a list of remappings. If you still can't find the finance number,
C set istrata to zero and let the verification subroutine handle the
C problem.

         IF (lfinno.ne.finno) THEN
            ifin = searchc(fins,nfin,finno)
            IF (ifin.le.0) THEN
               finreass = finreass + 1
               ifin = searchc(fins,nfin,finno)
            END IF
            IF (ifin.gt.0) THEN
               istrata = stratamap(ifin)
            ELSE
               istrata = 0
               PRINT *, 'Did not find Finance Number: ',finno
            END IF
         END IF
         lfinno = finno
C         PRINT *, "Fin #: ",finno," ifin = ",ifin," istrata = ",istrata,
C     +      " vips = ",lines
         vipsvalid = .TRUE.

C Read each VIP field of the transaction. Check to see if it is a valid
C VIP. If it isn't, set the invalid VIP flag for the transaction to false.
```

```
C Let the verification subroutine handle the error.

         DO i=1,lines
            read(vipfields((i-1)*48+1:(i-1)*48+48),12) vip,viprate,vippcs,
     +         viprev
            IF (vip(1:1).eq.' '.and.vip.ne.'    0') THEN
               vip(1:1) = '0'
               imprint = .TRUE.
            END IF
            ivip = searchc(vips,nvip,vip)
C            PRINT *, "VIP : ",vip," rate = ",viprate," pcs = ",vippcs,
C     +         " rev = ",viprev," ivip = ",ivip," i = ",i
            IF (ivip.le.0.and.vip.ne.'    0') THEN  ! probably 'old' vip
               vipsvalid = .FALSE.
               nvipinvalid = nvipinvalid + 1
               recvip(i) = 0
               recrev(i) = 0
               recpcs(i) = 0
               recrte(i) = 0
C               print *,'Invalid vip ',vip,'  rev is ', viprev,'  fin is ',finno
C               write(65,11) finno,ind,unique,rev,fees,psflag,
C     +      ishp,pcwt,pcs,wt,lines,vipfields

            ELSE IF (ivip.le.0.and.vip.eq.'    0') THEN !should not exist after 96
               viprev = 0.0
               vippcs = 0.0
               viprate = 0.0
               nstd = nstd + 1
               nvipvaild = nvipvaild + 1
               recvip(i) = 0
               recrev(i) = 0
               recpcs(i) = 0
               recrte(i) = 0
               print *,'NS surcharge vip '

            ELSE IF (ivip.gt.0) THEN
               trev = trev + viprev
               tpcs = tpcs + vippcs
               nvipvaild = nvipvaild + 1
               recvip(i) = ivip
               recrev(i) = viprev
               recpcs(i) = vippcs
               recrte(i) = viprate
            END IF
         END DO

C Send record to verification routine. Routine returns '0' if record is a
C valid, identical, good weight transaction.

         pcsin = pcsin + tpcs

C          IF (((year.eq.1999).and.(day.lt.10)).or.(year.eq.1998)) THEN
C             ratetable = oldratetable
C          ELSE IF (year.eq.1999) THEN        ! see section below for inconsistencies
C             ratetable = newratetable
C          ELSE
C             print*,'year error ', year
C             stop
C          END IF

C   Set variables for Special check for rate change inconsistent with date
C   done in the verification routine.

C          ratetable1 = oldratetable
C          ratetable2 = newratetable

C          print*,oldratetable,"  ",newratetable

C   test records with 1st vip 1574 and others priority.  This began in ap7 98
C   for finno 419044.  Keep the first vip and ignore the rest.  Consider this
C   a good transaction. Reset for verification routine.

               if (lines.gt.1) then
                  if (recvip(1).eq.11.and.recvip(2).eq.0) then
                     lines = 1
                     vipsvalid = .TRUE.
                     pcs = tpcs    ! for verification
                     print*,'switched lines to 1'
                  end if
               end if
```

```
      CALL verlst(error,istrata,vipsvalid,rev,pcs,wt,
     +     trev,tpcs,lines,ishp,pcwt)

C     This function assigning shape and creating recnew(i)
      isize = mail_size(pcs,lines,ishp)


C Assign revenue, pieces, and weight from valid transactions to matrices.

            xminind = 1      ! check for all valid vips in transaction
            DO i=1,lines
               IF (recvip(i).ne.0) then
                  iind = indicia(vips(recvip(i)),recnew(i))
                  minind(i) = iind      ! assign indicia
                  firstokay = .FALSE.
               ELSE
                  xminind = 0    ! bad vip
               END IF
            END DO


      IF (.not.firstokay.and.xminind.ne.0) THEN      ! count transactions
         trans(isize,istrata,minind(1)) =
     +      trans(isize,istrata,minind(1)) + 1
      END IF

      IF (xminind.eq.0) THEN      ! bad vip transactions
         badtran = badtran + 1
      END IF

      DO i=1,lines
         IF (error(i).eq.1.and.xminind.ne.0.and.reciwt(i).gt.0) THEN
            iwt = reciwt(i)
            pcsout = pcsout + recpcs(i)
            IF (minind(i).lt.3) THEN
               finrev(ifin) = finrev(ifin) + recrev(i)
            END IF
            IF (minind(i).eq.1) THEN
               ppirev(ifin) = ppirev(ifin) + recrev(i)
            ELSE IF (minind(i).eq.2) THEN
               smrev(ifin) = smrev(ifin) + recrev(i)
            ELSE IF (minind(i).eq.3) THEN
               sprev(ifin) = sprev(ifin) + recrev(i)
            END IF
            trp(1,ishape(i),isize,istrata,minind(i)) =
     +         trp(1,ishape(i),isize,istrata,minind(i)) + recrev(i)
            trp(2,ishape(i),isize,istrata,minind(i)) =
     +         trp(2,ishape(i),isize,istrata,minind(i)) + recpcs(i)
C            print*,recnew(i)," ",ishape(i)," ",istrata," ",minind(i)
            IF (error(i).eq.1) THEN
               rpw(1,recnew(i),iwt,ishape(i),istrata,minind(i)) =
     +            rpw(1,recnew(i),iwt,ishape(i),istrata,minind(i)) + recrev(i)
               rpw(2,recnew(i),iwt,ishape(i),istrata,minind(i)) =
     +            rpw(2,recnew(i),iwt,ishape(i),istrata,minind(i)) + recpcs(i)
               rpw(3,recnew(i),iwt,ishape(i),istrata,minind(i)) =
     +            rpw(3,recnew(i),iwt,ishape(i),istrata,minind(i)) + recwt(i)
            END IF
         ELSE IF (xminind.ne.0) THEN
               badrev(ifin,minind(i)) = badrev(ifin,minind(i)) + recrev(i)
         END IF
         if (error(i).gt.0) then
            errcount(error(i)) = errcount(error(i)) + 1
            errrev(error(i)) = errrev(error(i)) + recrev(i)
            errpcs(error(i)) = errpcs(error(i)) + recpcs(i)
            errwt(error(i)) = errwt(error(i)) + recwt(i)
         end if
         IF (error(i).gt.1) THEN
            WRITE(88+error(i),13) finno,rev,fees,ishp,pcwt,pcs,wt,
     +         lines,vipfields((i-1)*48+1:(i-1)*48+48),comprate(i)
         END IF
C       Temp for diagnostic of heavy flats
C            IF (error(i).lt.3.and.iwt.gt.10.and.(recnew(i).eq.2.or.recnew(i).eq.6.
C     +      or.recnew(i).eq.16.or.recnew(i).eq.25.or.recnew(i).eq.26)) Then
C               write(75,11) finno,ind,unique,rev,fees,psflag,
C     +      ishp,pcwt,pcs,wt,lines,vipfields
C            END IF

      END DO
```

```
C Okay, you're done with that record.

200   continue

      END DO                    ! Main Loop

C Write a Report file.

99    PRINT *, "Finished Reading AP",cap," with ier = ",ier
      OPEN(80,FILE='perhalf.1st.'//cap//'.rpt')
      WRITE(80,'("Report for new PERMIT by Size AP",A2," PFY 2000")') cap
81    FORMAT("     Disposition                        VIPs    Revenue",
     +   "     Pieces     Weight")
83    FORMAT("     Group                                 Revenue",
     +   "     Pieces     Weight")
82    FORMAT("         Records Read : ",I10,3F12.0)
84    FORMAT(A30," : ",I10,3F12.0)
      WRITE(80,81)
      WRITE(80,'(70("-"))')
      DO i=1,maxerr
         WRITE(80,84) errorname(i),errcount(i),errrev(i),errpcs(i),errwt(i)
      END DO
      WRITE(80,'(70("-"))')
      WRITE(80,'("  Valid Vips : ",I10)') nvipvaild
      WRITE(80,'("Invalid Vips : ",I10)') nvipinvalid
      WRITE(80,'(70("-"))')
      WRITE(80,'("Numbers of VIPs Routed at each node:")')
      WRITE(80,'(70("-"))')
      DO i=1,nodes
         WRITE(80,84) vipname(i),vipcount(i)
      END DO
      WRITE(80,'(70("-"))')
      DO i=1,ntype
         WRITE(80,84) tname(i),tcount(i)
      END DO
      WRITE(80,'(70("-"))')

C     Write Roll-up File

      OPEN(50,FILE='perhalf.1st.'//cap)
51    FORMAT(I3,I2,I1,I2,I1,"11",3F20.10)
52    FORMAT(I3,I2,I1,I2,I1,"10",3F20.10)
53    FORMAT(I3,I2,I1,I2,I1,"00",3F20.10)
      OPEN(60,FILE='perhalf.trans.1st.'//cap)
61    FORMAT(I2,I1,I2,I1,3F20.10)

C     Write VIP Matrices
      DO iind=1,nind
         DO istrata=1,nstrata
            DO ishp=1,nshp
               DO iwt=1,nwt
                  DO icls=1,nnewcls
                     IF (rpw(1,icls,iwt,ishp,istrata,iind).ne.0) THEN
                        WRITE(50,51) icls,iwt,ishp,istrata,iind,
     +                  (rpw(irpw,icls,iwt,ishp,istrata,iind),irpw=1,nrpw)
                        igrevout = igrevout+rpw(1,icls,iwt,ishp,istrata,iind)
                        igpcsout = igpcsout+rpw(2,icls,iwt,ishp,istrata,iind)
                        igwtout = igwtout+rpw(3,icls,iwt,ishp,istrata,iind)
                        stratarev(istrata,1) =
     +                     rpw(1,icls,iwt,ishp,istrata,iind)
     +                     + stratarev(istrata,1)
                        stratapcs(istrata,1) =
     +                     rpw(2,icls,iwt,ishp,istrata,iind)
     +                     + stratapcs(istrata,1)
                        clspcs(icls) = clspcs(icls) +
     +                     rpw(2,icls,iwt,ishp,istrata,iind)
                     END IF
                  END DO
               END DO
            END DO
         END DO
      END DO

      OPEN(7,FILE='clstitles.dat',readonly)
      PRINT *, "Pieces by Class:"
      DO icls=1,nnewcls
         READ(7,'(A24)') clstitle(icls)
         WRITE (6,'(I2,1X,A24,1X,F15.0)') icls,clstitle(icls),clspcs(icls)
      END DO
```

```
C     Write Transaction Matrices

      DO iind=1,nind
         DO istrata=1,nstrata
            DO isize=1,nsize
               DO ishp=1,nshp
                  IF (trp(1,ishp,isize,istrata,iind).gt.0)
     +                WRITE(60,61) isize,ishp,istrata,iind,
     +                (trp(itrp,ishp,isize,istrata,iind),itrp=1,ntrp)
                  stratarev(istrata,2) = trp(1,ishp,isize,istrata,iind)
     +                + stratarev(istrata,2)
                  stratapcs(istrata,2) = trp(2,ishp,isize,istrata,iind)
     +                + stratapcs(istrata,2)
               END DO
            END DO
         END DO
      END DO

      DO isize = 1,nsize
         DO istrata = 1,nstrata
            DO iind = 1, nind
                  ttrans = ttrans + trans(isize,istrata,iind)
            END DO
         END DO
      END DO

C     Rollup checks by strata
      DO istrata=1,nstrata
         DO i=1,2
            stratarev(nstrata+1,i) = stratarev(nstrata+1,i) +
     +         stratarev(istrata,i)
            stratapcs(nstrata+1,i) = stratapcs(nstrata+1,i) +
     +         stratapcs(istrata,i)
         END DO
      END DO

C     Write rollup totals to report file

      WRITE(80,83)
      WRITE(80,'(70("-"))')
86    FORMAT(A30," : ",3F12.0)
      WRITE(80,86) errorname(1),igrevout,igpcsout,igwtout
      WRITE(80,86) errorname(2),ibrevout,ibpcsout
      WRITE(80,86) errorname(3),nbrevout,nbpcsout
      WRITE(80,'(70("-"))')
C                1234567890123456789012345678901234567890
      WRITE(80,'("Strata    Rev by VIP   Rev by Size")')
      DO istrata=1,nstrata+1
         WRITE(80,'(I2,8X,2F12.0)') istrata,(stratarev(istrata,i),i=1,2)
      END DO
      WRITE(80,'(70("-"))')
      WRITE(80,'("Strata    Pcs by VIP   Pcs by Size")')
      DO istrata=1,nstrata+1
         WRITE(80,'(I2,8X,2F12.0)') istrata,(stratapcs(istrata,i),i=1,2)
      END DO
      WRITE(80,'(70("-"))')
      WRITE(80,'("Total Transactions: ",F10.0)') ttrans
      WRITE(80,'("GOVERNMENT Transactions: ",i10)') govct


      WRITE(80,'("Total SM Revenue: ",F20.5)') tsmout
      WRITE(80,'("Total Bad SM Rev: ",F20.5)') bsmout
      WRITE(80,'("Total PI Revenue: ",F20.5)') tpiout

      CLOSE(80)

      PRINT *, "Nonsurchrge VIP count = ",nstd
      PRINT *, "Transactions with bad vips = ", badtran
      PRINT *, "All done with AP",cap,"."

      CALL time(timestr)
      PRINT *,"End: ",timestr

      STOP
      END

C-----------------------------------------------------------------------

      SUBROUTINE verlst(error,istrata,vipsvalid,rev,pcs,wt,trev,
     +   tpcs,lines,ishp,pcwt)
```

```
      INTEGER*4 istrata,lines,ishp,maxvip,nvip,i,weight_incr,iwt,cx
      INTEGER*4 nodes,ntype,apply_rates,nwt,maxerr,searchc
      PARAMETER (maxvip=20,nvip=102,nwt=17,nodes=34,ntype=12,maxerr=11)
      REAL*8    rtol,minltr,minflt
      PARAMETER (rtol=0.002,minltr=0.0063,minflt=0.0125)
      INTEGER*4 error(maxvip)
      REAL*8    rev,pcs,wt,trev,tpcs,pcwt,testrate,testrate1,testrate2,nextrate,tpcwt
      LOGICAL   vipsvalid,weight,rateok,notcard,nonstd,idmail,found,same
      CHARACTER*5 tmpvip

C     Verification Common Block
      CHARACTER*5 vips(nvip)
      INTEGER*4 recvip(maxvip),reciwt(maxvip),vipcount(nodes),tcount(ntype)
      REAL*8    recrev(maxvip),recpcs(maxvip),recwt(maxvip),recrte(maxvip)
      REAL*8    rate(nvip,3),ratetable(nvip,nwt),oldratetable(nvip,nwt),comprate(maxvip)
      REAL*8    newratetable(nvip,nwt),ratetable1(nvip,nwt),ratetable2(nvip,nwt)
      INTEGER*4 vip_to_rate(nvip),maxwt(nvip),recnew(maxvip), ishape(maxvip)
      COMMON /verify/recvip,reciwt,recrev,ishape,
     +   recpcs,recwt,recrte,rate,
     +   vips,ratetable,vipcount,tcount,
     +   vip_to_rate,comprate,maxwt,recnew

      weight = .TRUE.
      rateok = .TRUE.
      notcard = .TRUE.
      nonstd = .FALSE.
      idmail = .TRUE.
      tpcwt = wt/pcs

      DO i=1,maxvip
         error(i) = 0
      END DO

      IF (vipsvalid) THEN

         IF (istrata.gt.0) THEN

C     First Determine Shape Information    REVISED for reclass

            DO i=1,lines
               IF (recvip(i).ne.0) THEN
                  IF  (vips(recvip(i))(5:5).eq.'3')THEN     ! cards
                     ishp = 1                 ! may have 0 on tape
                     notcard = .FALSE.
                  END IF
                  IF (vips(recvip(i))(2:5).eq.'1642'.or.
     +      vips(recvip(i))(2:5).eq.'1652'.or.vips(recvip(i))(2:5).eq.'1674'.or.
     +      vips(recvip(i))(2:5).eq.'1861'.or.vips(recvip(i))(2:5).eq.'1871'.or.
     +      vips(recvip(i))(2:5).eq.'1862'.or.vips(recvip(i))(2:5).eq.'1872'.or.
     +      vips(recvip(i))(2:5).eq.'1864'.or.vips(recvip(i))(2:5).eq.'1874'.or.
     +      vips(recvip(i))(2:5).eq.'1881'.or.vips(recvip(i))(2:5).eq.'1882'.or.
     +      vips(recvip(i))(2:5).eq.'1884')
     +            THEN

                     nonstd = .TRUE.
                  END IF
               END IF
            END DO

C     Now Determine if Total Revenue and VIP revenue agree within 5%

            IF (ABS(trev/rev-1).le.0.05) THEN ! Revenue sum okay
               IF (ABS(trev/rev-1).gt.0.0.and.ABS(trev/rev-1).le.0.05)
     +            rev = trev    ! If error within tol., switch rev.
            ELSE
               DO i=1,lines
C                 We will keep these records 5/1/95
C                 error(i) = 9 ! Inconsistent Revenue Error
C                 comprate(i) = recrev(i)/recpcs(i)
                  vipcount(31) = vipcount(31) + 1
               END DO
            END IF

C     Next compare VIP pieces to total pieces

            IF (ABS(tpcs-pcs).gt.0.01) THEN
               DO i=1,lines
                  error(i) = 10 ! Inconsistent Pieces Error
                  if (recpcs(i).ne.0.0) then
```

```
                  comprate(i) = recrev(i)/recpcs(i)
               else
                  comprate(i) = 0
               end if
               vipcount(32) = vipcount(32) + 1
            END DO
         END IF

      ELSE
         DO i=1,lines
            error(i) = 8     ! Invalid Finance Number Error
            vipcount(30) = vipcount(30) + 1
         END DO
      END IF

   ELSE      ! bad vip
      DO i=1,lines
         error(i) = 7        ! Invalid VIP Error; entire tran is ignored
         vipcount(29) = vipcount(29) + 1
      END DO
   END IF

C     Now Determine if the mailing is identical or non-identical

   IF (error(1).eq.0) THEN      ! Is the mailing valid so far?

C     Test for invalid weight information

      IF (weight_incr(wt,pcs,pcwt).le.0.or.
     +    weight_incr(wt,pcs,pcwt).gt.17) THEN
         weight = .FALSE.
      ELSE
         IF (nonstd) THEN
            weight = .TRUE.
         ELSE IF (wt/pcs.lt.minltr.and.ishp.eq.1.and.notcard) THEN
            weight = .FALSE.
         ELSE IF (wt/pcs.lt.minflt.and.ishp.gt.1) THEN
            weight = .FALSE.
         END IF
      END IF

      IF (pcwt.ne.0.0) THEN            ! identical wt mailing
         tcount(1) = tcount(1) + 1
         IF (nonstd) THEN
            tcount(5) = tcount(5) + 1
            tcount(2) = tcount(2) + 1
         ELSE IF (.not.notcard) THEN
            tcount(4) = tcount(4) + 1
            tcount(2) = tcount(2) + 1
         ELSE IF (weight) THEN
            tcount(3) = tcount(3) + 1
            tcount(2) = tcount(2) + 1
         ELSE
            tcount(6) = tcount(6) + 1
         END IF

         DO i=1,lines
            IF (recvip(i).ne.0) THEN
               vipcount(1) = vipcount(1) + 1 ! Node #1
               IF (weight) THEN ! Good Weight
                  vipcount(2) = vipcount(2) + 1 ! Node #2
                  recwt(i) = recpcs(i)*pcwt
                  reciwt(i) = weight_incr(wt,pcs,pcwt)
C                 Check the rate paid against the rate for that VIP.
                  testrate = ratetable(recvip(i),reciwt(i))
                  if (recpcs(i).ne.0.0) then
                     comprate(i) = recrev(i)/recpcs(i)
                  else
                     comprate(i) = 0
                  end if
                  IF (ABS(comprate(i)-testrate).gt.rtol) THEN
C                    Neither old or new rates match
                     error(i) = 4
                     vipcount(4) = vipcount(4) + 1
                     PRINT *," Bad Rate: VIP =",vips(recvip(i)),
     +                  " iwt = ",reciwt(i)," actual = ",
     +                  comprate(i)," computed = ",testrate
C                     print*," recrev = ",recrev(i),"  recpcs = ",recpcs(i)
                  ELSE
                     error(i) = 1
```

```
                  vipcount(3) = vipcount(3) + 1
               END IF
            ELSE              ! Bad Weight
               vipcount(5) = vipcount(5) + 1 ! Node #5
               if (recpcs(i).ne.0.0) then
                  comprate(i) = recrev(i)/recpcs(i)
               else
                  comprate(i) = 0
               end if
               found = .FALSE.
               iwt = 1
               DO WHILE (iwt.le.17.and.(.not.found))
                  testrate = ratetable(recvip(i),iwt)
                  IF (ABS(comprate(i)-testrate).le.rtol) THEN
                     found = .TRUE.
                  ELSE
                     iwt = iwt + 1
                  END IF
               END DO
               IF (found) THEN
                  vipcount(7) = vipcount(7) + 1
                  error(i) = 2
                  reciwt(i) = iwt
               ELSE
                  vipcount(6) = vipcount(6) + 1
                  error(i) = 5
               END IF
            END IF
         ELSE              ! Nonstandard Surcharge VIP Junk
            vipcount(33) = vipcount(33) + 1 ! Node #30
            error(i) = 11
         END IF
      END DO ! Loop over VIPS
   END IF
END IF

RETURN
END

C-------------------------------------------------------------------------------

      INTEGER*4 FUNCTION mail_size(pieces,lines,ishp)

C     Purpose: Given the number of pieces in a mailing, returns the
C              the size index for the mailing.  Also assigns shape and
C               recnew array.

C     Program Parameters
      INTEGER*4 maxvip,maxerr,nvip,nfin,nmap,nnewcls,nrpw,ntrp
      PARAMETER (maxvip=20,maxerr=11,nvip=102,nnewcls=34,
     +   nfin=10815,nrpw=3,ntrp=3)                    ! update nfin
      INTEGER*4 nshp,nwt,nsize,nstrata,nodes,ntype,nind
      PARAMETER (nshp=3,nwt=17,nsize=3,nstrata=20,nodes=34,ntype=12,nind=2)

      INTEGER*4 lines,i,ishp,irate,seq

C     Verification Common Block
      INTEGER*4 recvip(maxvip),reciwt(maxvip),vipcount(nodes),tcount(ntype)
      REAL*8    recrev(maxvip),recpcs(maxvip),recwt(maxvip),recrte(maxvip)
      REAL*8    rate(nvip,3),ratetable(nvip,nwt),comprate(maxvip)
      REAL*8    sizerpt(nind,nsize,nsize)
      REAL*8    pieces
      INTEGER*4 vip_to_rate(nvip),maxwt(nvip),recind(maxvip),recnew(maxvip)
      INTEGER*4 ishape(maxvip)
      CHARACTER*5 vips(nvip)
      CHARACTER*1 psflag

      COMMON /verify/recvip,reciwt,recrev,ishape,
     +   recpcs,recwt,recrte,rate,
     +   vips,ratetable,vipcount,tcount,
     +   vip_to_rate,comprate,maxwt,recnew
      COMMON /trantype/sizerpt


      IF (pieces.le.5000) THEN
         mail_size = 1
      ELSE IF (pieces.le.10000) THEN
         mail_size = 2
      ELSE
         mail_size = 3
```

```
      END IF

C     Assign shape

      DO i=1,lines
         IF (recvip(i).ne.0) THEN
            irate = vip_to_rate(recvip(i))
            IF (irate.eq.1) THEN
               recnew(i) = 1
               ishape(i) = 1
            ELSE IF (irate.eq.2) THEN
               recnew(i) = 2
               ishape(i) = 2
            ELSE IF (irate.eq.3) THEN
               recnew(i) = 3
               ishape(i) = 3
            ELSE IF (irate.eq.4) THEN
               recnew(i) = 4
               ishape(i) = 3
            ELSE IF (irate.eq.5) THEN
               recnew(i) = 5
               ishape(i) = 1
            ELSE IF (irate.eq.6) THEN
               recnew(i) = 6
               ishape(i) = 2
            ELSE IF (irate.eq.7) THEN
               recnew(i) = 7
               ishape(i) = 3
            ELSE IF (irate.eq.8) THEN
               recnew(i) = 8
               ishape(i) = 1
            ELSE IF (irate.eq.9) THEN
               recnew(i) = 9
               ishape(i) = 2
            ELSE IF (irate.eq.10) THEN
               recnew(i) = 10
               ishape(i) = 3
            ELSE IF (irate.eq.11) THEN
               recnew(i) = 11
               ishape(i) = 3
            ELSE IF (irate.eq.12) THEN
               recnew(i) = 12
               ishape(i) = 1
            ELSE IF (irate.eq.13) THEN
               recnew(i) = 13
               ishape(i) = 2
            ELSE IF (irate.eq.14) THEN
               recnew(i) = 14
               ishape(i) = 3
            ELSE IF (irate.eq.15) THEN
               recnew(i) = 15
               ishape(i) = 1
            ELSE IF (irate.eq.16) THEN
               recnew(i) = 16
               ishape(i) = 2
            ELSE IF (irate.eq.17) THEN
               recnew(i) = 17
               ishape(i) = 3
            ELSE IF (irate.eq.18) THEN
               recnew(i) = 18
               ishape(i) = 1
            ELSE IF (irate.eq.19) THEN
               recnew(i) = 19
               ishape(i) = 2
            ELSE IF (irate.eq.20) THEN
               recnew(i) = 20
               ishape(i) = 3
            ELSE IF (irate.eq.21) THEN
               recnew(i) = 21
               ishape(i) = 1
            ELSE IF (irate.eq.22) THEN
               recnew(i) = 22
               ishape(i) = 1
            ELSE IF (irate.eq.23) THEN
               recnew(i) = 23
               ishape(i) = 1
            ELSE IF (irate.eq.24) THEN
               recnew(i) = 24
               ishape(i) = 1
            ELSE IF (irate.eq.25) THEN
```

```
            recnew(i) = 25
            ishape(i) = 2
         ELSE IF (irate.eq.26) THEN
            recnew(i) = 26
            ishape(i) = 2
         ELSE IF (irate.eq.27) THEN
            recnew(i) = 27
            ishape(i) = 2
         ELSE IF (irate.eq.28) THEN
            recnew(i) = 28
            ishape(i) = 2
         ELSE IF (irate.eq.29) THEN
            recnew(i) = 29
            ishape(i) = 1
         ELSE IF (irate.eq.30) THEN
            recnew(i) = 30
            ishape(i) = 1
         ELSE IF (irate.eq.31) THEN
            recnew(i) = 31
            ishape(i) = 1
         ELSE IF (irate.eq.32) THEN
            recnew(i) = 32
            ishape(i) = 1
         ELSE IF (irate.eq.33) THEN
            recnew(i) = 33
            ishape(i) = 1
         ELSE IF (irate.eq.34) THEN
            recnew(i) = 34
            ishape(i) = 1

         END IF
      END IF
   END DO

   RETURN
   END

C--------------------------------------------------------------------------------
   INTEGER*4 FUNCTION weight_incr(weight,pieces,pcwt)

   INTEGER*4 iwt,nwt,ioz
   PARAMETER (nwt=17)
   REAL*8 weight,pieces,pcwt,ozs,tpcwt

   weight_incr = 0
   IF (pcwt.eq.0) THEN
      tpcwt = weight/pieces
   ELSE
      tpcwt = pcwt
   END IF
   ozs = tpcwt*16.0

   IF (ozs.gt.0.0.and.ozs.le.0.5) then
      weight_incr = 1
   ELSE IF (ozs.le.1.0) then
      weight_incr = 2
   ELSE IF (ozs.le.1.5) then
      weight_incr = 3
   ELSE IF (ozs.le.2.0) then
      weight_incr = 4
   ELSE IF (ozs.le.2.5) then
      weight_incr = 5
   ELSE IF (ozs.le.3.0) then
      weight_incr = 6
   ELSE IF (ozs.le.3.5) then
      weight_incr = 7
   ELSE IF (ozs.le.4.0) then
      weight_incr = 8
   ELSE IF (ozs.le.5.0) then
      weight_incr = 9
   ELSE IF (ozs.le.6.0) then
      weight_incr = 10
   ELSE IF (ozs.le.7.0) then
      weight_incr = 11
   ELSE IF (ozs.le.8.0) then
      weight_incr = 12
   ELSE IF (ozs.le.9.0) then
      weight_incr = 13
   ELSE IF (ozs.le.10.0) then
      weight_incr = 14
```

```
      ELSE IF (ozs.le.11.0) then
         weight_incr = 15
      ELSE IF (ozs.le.12.0) then
         weight_incr = 16
      ELSE IF (ozs.le.13.0) then
         weight_incr = 17
      ELSE
         print*,' weight_incr = 0 '
         weight_incr = 0
      END IF

      RETURN
      END
C-------------------------------------------------------------------------------
      INTEGER*4 FUNCTION indicia(vip,icls)

      CHARACTER*5 vip
      INTEGER*4 icls

         IF (vip(1:1).eq.'1'.or.vip(1:1).eq.'2') THEN
            indicia = 2       ! SM
         ELSE IF (vip(1:1).eq.'0') THEN
            indicia = 1       ! PI
         ELSE
            PRINT *, "NEW VIP ",vip," doesn't correspond to known indicia."
            STOP
         END IF


      RETURN
      END
```

```
      PROGRAM nrollup

C     DESCRIPTION: Clean and Roll CBCIS reports
C
C     CREATED BY:       Bill Humphries
C     DATE:             Mon Jan 23 16:08:30 CST 1995
C
C     LAST MODIFIED BY: ta
C     DATE:             sep 24 -96; revise for FY97 processing.
C                       oct 22  96; update to skip 'old' VIPs
C                       dec 13  96; skip 'total' priority recs
C                       jun 27  97; update common block for new compiler
C                       oct  6  98; update for FY99
C                       jan 15, 99; update for jan 10, 1999 rate changes
C                       feb 25, 99; update for new vip codes from rate changes
C                       apr 08, 99; move SP data into PI and SM due to change in 41416
C
C
C        NOTE: >>>> THIS program for ap 7 and beyond
C

      IMPLICIT NONE

C     Declare Constants

      INTEGER*4 nfin,nvip,nstrata,nerrs,nrpw,ncls,ntype,nwt
      PARAMETER (nfin=10815)      ! update - lines in finstrata.new
      PARAMETER (nstrata=20,nvip=102,nerrs=11)
      PARAMETER (nrpw=3,ncls=34,ntype=2,nwt=13)
      REAL*8    tol
      PARAMETER  (tol=30.0)

C     Maps

      CHARACTER*20 errnames(nerrs)
      CHARACTER*6  fins(nfin)
      CHARACTER*5  vips(nvip)
      INTEGER*4    stratamap(nfin)
      INTEGER*4    vip_to_rate(nvip)
      INTEGER*4    maxwt(nvip)
      REAL*8       ratetable(nvip,nwt)
      REAL*8       ave(ncls,nwt),dist(ncls,nwt)

C     Indicies

      INTEGER*4 ivip,ifin,istrata,irpw,icls,i,j,k,l,ier,searchc,error,itype
      INTEGER*4 iap,iwt

C     Data Read from Record

      CHARACTER*6 fin
      CHARACTER*5 vip
      REAL*8      rev,pcs,wt,erev,epcs,ewt,eavewt,eaverev
      REAL*8      averev,avewt,first,heavy,last

C     Storage

      REAL*8     okay(nrpw),bad(nrpw),total(nrpw)
      REAL*8     data(nrpw,nvip,nstrata,ntype)
      REAL*8     dataredist(nrpw,nvip,nstrata,ntype)
      REAL*8     bystrata(nstrata,ntype,nrpw)
      REAL*8     bytype(nfin,ntype)
      REAL*8     priority(nstrata)

C     Report

      INTEGER*4  fintot,viptot,neg,finerr,viperr
      REAL*8     errors(nerrs,nrpw)
      INTEGER*4  numerr(nerrs)
      REAL*8     byclass(ncls,nrpw)
      REAL*8     errbycls(ncls,nrpw)
      INTEGER*4  tviperr(nvip)
      INTEGER*4  viperror(nvip,nerrs)
      REAL*8     reverrvip(nvip)
      REAL*8     viprev(nvip)
      INTEGER*4  fincount(nfin),tfinerr(nfin)
      INTEGER*4  finerror(nfin,nerrs)
      REAL*8     reverrfin(nfin)
      REAL*8     finrev(nfin)
      REAL*8     out(nrpw)
```

```
      INTEGER*4  nwtfix,nrevfix, ct
      REAL*8     negrev,negpcs,negwt,wtfixrev,wtfixpcs,wtfixwt,
     +   revfixrev,revfixpcs,revfixwt,
     +   ltrrev,ltrpcs,ltrwt,cdrev,cdpcs,cdwt,fltrev,fltpcs,fltwt,
     +   fcdrev,fcdpcs,fcdwt,mvrev,mvwt,mvpcs,wterr

      CHARACTER*2 ap

C     Functions

      INTEGER*4 verify,type

      COMMON /map/vips,ave,ratetable,dist,
     +   maxwt,vip_to_rate
      COMMON /ver/averev,avewt,first,heavy,last,
     +   nwtfix,wtfixrev,wtfixpcs,wtfixwt,nrevfix,revfixrev,revfixpcs,revfixwt,
     +   mvrev,mvwt,mvpcs,wterr
      COMMON /red/
     +   ltrrev,ltrpcs,ltrwt,cdrev,cdpcs,cdwt,
     +   dataredist,
     +   fltrev,fltpcs,fltwt,fcdrev,fcdpcs,fcdwt

C     Get Argument

      CALL getarg(1,ap)
      PRINT *, "Working on AP",ap
      READ(ap,'(I2)') iap

C     Open and Read Map of Finance Numbers and Their Strata

      OPEN(1,FILE='finstrata.new.00')

2     FORMAT(a6,1x,i3)
      DO i = 1, nfin
         READ(1,2) fins(i), stratamap(i)
      END DO

      PRINT *, "Read Strata Maps."

C     Read List of Valid VIP Codes and Map to Rate Elements

      OPEN(7,FILE='vipmap.1st.00')
6     FORMAT(4X,A5,3X,I9)
      i = 0
      ier = 0
      DO WHILE (ier.eq.0)
         READ(7,6,IOSTAT=ier,END=7) vip,j
         i = i + 1
         IF (vip(1:1).eq.' ') vip(1:1) = '0'
         vip_to_rate(i) = j
         vips(i) = vip
      END DO
      CLOSE(7)
7     PRINT *, "Read VIP map with ier = ",ier
      PRINT *, "Read ",i," VIPs."


C     Read List of Rates for Each VIP by ounce increment
      OPEN(7,FILE='ratetable.00')

8     format(5x,1x,i2,1x,13f6.3)

      DO i=1,nvip
         READ(7,8) maxwt(i), (ratetable(i,iwt),iwt=1,nwt)
      END DO
      PRINT *, "Read Rate List."
      CLOSE(7)

      do j = 1, nwt
         do i = 1,ncls
            ave(i,j) = 0.0
         end do
      end do

C     Read Average Weight by Ounce (in lbs.) and distribution from PERMIT data
      OPEN(7,FILE='avewt.99.dat')
      OPEN(8,FILE='dist.99.dat')
      DO i=1,ncls
         READ(7,'(11(F11.5))') (ave(i,iwt),iwt=1,11)
         READ(8,'(11(F11.5))') (dist(i,iwt),iwt=1,11)     ! note
```

```
      END DO
      DO i=1,ncls          ! set these to zero until they can be estimated
         dist(i,12) = 0
         dist(i,13) = 0
C   only valid ave will be used in program
         ave(i,12) = .719    ! midpoints
         ave(i,13) = .781
      END DO


      PRINT *, "Read Average Wt. by ounce and class"
      CLOSE(7)

      DO i=1,ncls
         DO iwt=1,nwt
            ave(i,iwt) = ave(i,iwt)*16.0       ! Scale to Ounces
         END DO
      END DO

C     Error Names and Files

20    FORMAT(A6,1X,A5,6F9.0,F4.0)

C                      12345678901234567890
      errnames(1) = '(1) Missing Data  : '
      OPEN(21,FILE='error.data.'//ap)
      errnames(2) = '(2) Bad VIP       : '
      OPEN(22,FILE='error.vip.'//ap)
      errnames(3) = '(3) Bad Finance # : '
      OPEN(23,FILE='error.finno.'//ap)
      errnames(4) = '(4) Revenue Error : '
      OPEN(24,FILE='error.revenue.'//ap)
      errnames(5) = '(5) Weight Error  : '
      OPEN(25,FILE='error.weight.'//ap)
      errnames(6) = '(6) Redist Cd/Ltr : '
      OPEN(26,FILE='error.redist.'//ap)
      errnames(7) = '(7) Cannot Redist : '
      OPEN(27,FILE='error.fail.redist.'//ap)
      errnames(8) = '(8) Wt/Rev Incon. : '
      OPEN(28,FILE='error.wtrev.'//ap)
      errnames(9) = '(9) Priority      : '
      OPEN(29,FILE='priority.'//ap)
      errnames(10) ='(10) Heavy Nonstd : '
      OPEN(30,FILE='error.heavy.nonstd.'//ap)
      errnames(11) ='(11) Heavy Auto.  : '
      OPEN(31,FILE='error.heavy.auto.'//ap)

      do i = 1, nfin
         fincount(i) = 0
         tfinerr(i) = 0
         reverrfin(i) = 0.0
         finrev(i) = 0.0
         do k = 1, ntype
            bytype(i,k) = 0.0
         end do
         do j = 1, nerrs
            finerror(i,j) = 0
         end do
      end do

      do j = 1, nstrata
         priority(j) = 0.0
         do i = 1, ntype
            do k = 1, nvip
               do l = 1, nrpw
                  dataredist(l,k,j,i) = 0.0
                  data(l,k,j,i) = 0.0
               end do
            end do
         end do
      end do

      do j = 1, nerrs
         numerr(j) = 0
         do i = 1, nrpw
            errors(j,i) = 0.0
         end do
      end do

      do j = 1, nrpw
```

```
         out(j) = 0.0
         do i = 1, ncls
            byclass(i,j) = 0.0
            errbycls(i,j) = 0.0
         end do
      end do

      do i = 1, nvip
         viprev(i) = 0.0
      end do

      ier = 0
      viptot = 0
      viperr = 0
      finerr = 0
      fintot = 0
      neg = 0
      negrev = 0.0
      negpcs = 0.0
      negwt = 0.0
      ct = 0

11    FORMAT(5X,A6,22X,A5,F10.2,F10.0,F12.2)

      OPEN(10,FILE='cbc00'//ap//'.dat')

      DO WHILE (ier.eq.0)
         ct =ct + 1
c         print*, ct
         READ (10,11,IOSTAT=ier,END=12) fin,vip,rev,pcs,wt
         error = 0
         viptot = viptot + 1
         total(1) = total(1) + rev
         total(2) = total(2) + pcs
         total(3) = total(3) + wt
         IF (fin(1:1).eq.' ') fin(1:1) = '0'
         IF (vip(1:1).eq.' ') vip(1:1) = '0'
         ifin = searchc(fins,nfin,fin)
         IF (ifin.le.0) THEN
            ifin = searchc(fins,nfin,fin)
            IF (ifin.le.0) THEN
               error = 3
               istrata = 0
            END IF
         END IF
         IF (error.eq.0) THEN
            finrev(ifin) = finrev(ifin) + rev
            istrata = stratamap(ifin)
            fincount(ifin) = 1
            ivip = searchc(vips,nvip,vip)
            IF (ivip.le.0) THEN
C              PRINT *, "VIP ",vip," index = ",ivip
                  ivip = 0
                  error = 2
                  IF (vip(2:2).eq.'7') then !.and.vip(2:5).ne.'7000') THEN   ! Priority
                     error = 9
                     total(1) = total(1) - rev
                     total(2) = total(2) - pcs
                     total(3) = total(3) - wt
                     IF (vip(1:2).eq.'07'.and.vip(4:4).eq.'9') THEN ! presorted
                        priority(stratamap(ifin)) =
     +                     priority(stratamap(ifin)) + rev
                     END IF
                  ELSE          ! old or bad VIP
                       icls = 35
                       numerr(2) = numerr(2) + 1
                       WRITE(22,'(A6,1X,A5,1X,I2,3F10.0)') fin,vip,icls,rev,pcs,wt
                  END IF
            END IF
            IF (ivip.gt.0) THEN
               viprev(ivip) = viprev(ivip) + rev
               icls = vip_to_rate(ivip)
            END IF
         END IF
         IF (error.eq.0) THEN
            itype = type(vip,icls)
            error = verify(ivip,icls,rev,pcs,wt,istrata,itype)
         END IF
         IF (error.gt.0.and.icls.ne.35) THEN
            erev = rev
```

```
            epcs = pcs
            ewt = wt
            IF (epcs.gt.0) THEN
               eavewt = ewt/epcs
               eaverev = erev/epcs
            ELSE
               eavewt = 0.0
               eaverev = 0.0
            END IF
            CALL repair(error,ivip,icls,rev,pcs,wt,istrata,itype)
            WRITE(20+error,'(A6,1X,A5,1X,I2,3F10.0,5F7.3)')
     +         fin,vip,icls,erev,epcs,ewt,eaverev,eavewt*16.0,
     +         ave(icls,1),first,last
            errbycls(icls,1) = errbycls(icls,1) + rev
            errbycls(icls,2) = errbycls(icls,2) + pcs
            errbycls(icls,3) = errbycls(icls,3) + wt
            IF (error.ne.9) then        ! not priority
               bad(1) = bad(1) + rev
               bad(2) = bad(2) + pcs
               bad(3) = bad(3) + wt
            END IF
            IF (ifin.gt.0) THEN
               finerror(ifin,error) = finerror(ifin,error) + 1
               reverrfin(ifin) = reverrfin(ifin) + rev
            END IF
            IF (ivip.gt.0) THEN
               viperror(ivip,error) = viperror(ivip,error) + 1
               reverrvip(ivip) = reverrvip(ivip) + rev
            END IF
            errors(error,1) = errors(error,1) + rev
            errors(error,2) = errors(error,2) + pcs
            errors(error,3) = errors(error,3) + wt
            numerr(error) = numerr(error) + 1
         ELSE    ! error = 0
            okay(1) = okay(1) + rev
            okay(2) = okay(2) + pcs
            okay(3) = okay(3) + wt
         END IF

         IF (error.ne.2.and.error.ne.3.and.error.ne.9) THEN ! not vip,fin,pri
            bytype(ifin,itype) = bytype(ifin,itype) + rev
            IF (error.ne.6.and.error.ne.7) THEN            ! not redist error
               byclass(icls,1) = byclass(icls,1) + rev
               byclass(icls,2) = byclass(icls,2) + pcs
               byclass(icls,3) = byclass(icls,3) + wt
               data(1,ivip,istrata,itype) = data(1,ivip,istrata,itype) + rev
               data(2,ivip,istrata,itype) = data(2,ivip,istrata,itype) + pcs
               data(3,ivip,istrata,itype) = data(3,ivip,istrata,itype) + wt
            END IF
         END IF
      END DO

12    PRINT *, "Finished Reading AP",ap," CBCIS File, ier = ",ier

      WRITE(6,'("CBCIS File Report")')
1     FORMAT(A17,3F15.0)
      WRITE(6,13) 'Total excl Pri : ',total
      WRITE(6,13) 'Okay  excl Pri : ',okay
      WRITE(6,13) 'Bad   excl Pri : ',bad

      PRINT *, "Total Records Read ", viptot

      DO ifin = 1,nfin
         fintot = fintot + fincount(ifin)
         DO error=1,nerrs
            IF (finerror(ifin,error).gt.0) THEN
               tfinerr(ifin) = 1
            END IF
         END DO
         finerr = finerr + tfinerr(ifin)
      END DO

      DO ivip = 1,nvip
         DO error=1,nerrs
            IF (viperror(ivip,error).gt.0) THEN
               tviperr(ivip) = 1
            END IF
         END DO
         viperr = viperr + tviperr(ivip)
      END DO
```

```
      PRINT *, "Unique VIPs with Errs : ",viperr
      PRINT *, "Total Fins Read       : ",fintot
      PRINT *, "Total Fins With Errors: ",finerr
      PRINT *, "Total Weight in Error : ",wterr
      PRINT *
      PRINT *, "Reason For Error:"

14    FORMAT(A20,I5,3F15.0)
      DO i=1,nerrs
         WRITE(6,14) errnames(i),numerr(i),(errors(i,irpw),irpw=1,nrpw)
      END DO

      PRINT *, "Filled in Missing Data:"
      WRITE(6,14) "Fixed Weight",nwtfix,wtfixrev,wtfixpcs,wtfixwt
      WRITE(6,14) "Fixed Revenue",nrevfix,revfixrev,revfixpcs,revfixwt
      PRINT *, "Redistributing Mixed Letter/Card VIPs:"
      WRITE(6,'(A25,3F15.0)') "As Letters:",ltrrev,ltrpcs,ltrwt
      WRITE(6,'(A25,3F15.0)') "As Cards:",cdrev,cdpcs,cdwt
      PRINT *, "Could Not Redistribute, Forced Pieces:"
      WRITE(6,'(A25,3F15.0)') "As Letters:",fltrev,fltpcs,fltwt
      WRITE(6,'(A25,3F15.0)') "As Cards:",fcdrev,fcdpcs,fcdwt
      PRINT *, "Attempted to Repair Pieces and Weight:"
      WRITE(6,'(A25,3F15.0)') "            ",mvrev,mvpcs,mvwt

      OPEN(20,FILE='redist.'//ap//'.dat')
15    FORMAT(I3,I2,I1,3F15.0,I3,F6.3)
      DO itype=1,ntype
         DO istrata=1,nstrata
            DO ivip=1,nvip
               IF(dataredist(1,ivip,istrata,itype).gt.0)
     +            WRITE(20,15) ivip,istrata,itype,
     +            (dataredist(irpw,ivip,istrata,itype),irpw=1,nrpw)
            END DO
         END DO
      END DO

      OPEN(20,FILE='rollup.'//ap//'.dat')
      DO itype=1,ntype
         DO istrata=1,nstrata
            DO ivip=1,nvip
               IF((data(1,ivip,istrata,itype)+
     +            dataredist(1,ivip,istrata,itype)).gt.0) THEN
     +            WRITE(20,15) ivip,istrata,itype,
     +               ((data(irpw,ivip,istrata,itype)+
     +               dataredist(irpw,ivip,istrata,itype)),irpw=1,nrpw),
     +               vip_to_rate(ivip),
     +               ((data(1,ivip,istrata,itype)+
     +               dataredist(1,ivip,istrata,itype))/
     +               (data(2,ivip,istrata,itype)+
     +               dataredist(2,ivip,istrata,itype)))
                  DO irpw=1,nrpw
                     bystrata(istrata,itype,irpw) =
     +                  bystrata(istrata,itype,irpw) +
     +                  data(irpw,ivip,istrata,itype) +
     +                  dataredist(irpw,ivip,istrata,itype)
                     out(irpw) = out(irpw) +
     +                  data(irpw,ivip,istrata,itype) +
     +                  dataredist(irpw,ivip,istrata,itype)
                  END DO
               END IF
            END DO
         END DO
      END DO

      PRINT *,"Revenue, Pieces and Weight Written:"
      WRITE(6,'(3F20.0)') out

      OPEN(20,FILE='bystrata.'//ap//'.dat')
      DO itype=1,ntype
         DO istrata=1,nstrata
            WRITE(20,'(I1,I2,3F15.0)') itype,istrata,
     +         (bystrata(istrata,itype,irpw),irpw=1,nrpw)
         END DO
      END DO

      DO itype=1,ntype
         DO istrata=1,nstrata
            DO ivip=1,nvip
               IF (dataredist(1,ivip,istrata,itype).gt.0) THEN
```

```
                DO irpw=1,nrpw
                   byclass(vip_to_rate(ivip),irpw) =
     +                dataredist(irpw,ivip,istrata,itype) +
     +                byclass(vip_to_rate(ivip),irpw)
                END DO
             END IF
          END DO
       END DO
    END DO

    OPEN(20,FILE='finrev.'//ap)
    DO ifin=1,nfin
       WRITE(20,'(A6,3F10.0)') fins(ifin),(bytype(ifin,itype),itype=1,ntype)
    END DO

    OPEN(20,FILE='prirev.'//ap)
    DO istrata=1,nstrata
       WRITE(20,'(F10.0)') priority(istrata)
    END DO

    OPEN(20,FILE='byclass.'//ap//'.txt')
    OPEN(30,FILE='errbycls.'//ap//'.txt')
16  FORMAT(I2,1X,3F15.0)
    DO icls=1,ncls
       WRITE(20,16) icls,(byclass(icls,irpw),irpw=1,nrpw)
       WRITE(30,16) icls,(errbycls(icls,irpw),irpw=1,nrpw)
    END DO

    OPEN(30,FILE='errbyfin.'//ap//'.txt')
    DO ifin=1,nfin
       IF (tfinerr(ifin).gt.0) THEN
          WRITE(30,'(A6,11(1X,I3),2F10.0,F7.2)') fins(ifin),(
     +       finerror(ifin,error),
     +       error=1,nerrs),reverrfin(ifin),finrev(ifin),
     +       reverrfin(ifin)/finrev(ifin)*100
       END IF
    END DO

    OPEN(30,FILE='errbyvip.'//ap//'.txt')
    DO ivip=1,nvip
       IF (tviperr(ivip).gt.0) THEN
          WRITE(30,'(A5,11(1X,I3),2F10.0,F7.2)') vips(ivip),
     +       (viperror(ivip,error),
     +       error=1,nerrs),reverrvip(ivip),viprev(ivip),
     +       reverrvip(ivip)/viprev(ivip)*100
       END IF
    END DO

    STOP
    END

C   ------------------------------------------------------------------------

    INTEGER*4 FUNCTION verify(ivip,icls,rev,pcs,wt,istrata,itype)

    INTEGER*4 ncls,nwt,nvip
    REAL*8    tol,wtol
    PARAMETER (ncls=34,nwt=13,tol=0.01,wtol=.01,nvip=102)

    CHARACTER*5 vips(nvip)
    INTEGER*4 ivip,icls,nwtfix,nrevfix,istrata,itype
    REAL*8    rev,pcs,wt,averev,avewt,first,heavy,addoz,last
    REAL*8    wtres,rvres,rvlst
    REAL*8    wtfixrev,wtfixpcs,wtfixwt,mvrev,mvwt,mvpcs
    REAL*8    revfixrev,revfixpcs,revfixwt
    REAL*8    ratetable(nvip,nwt)
    REAL*8    ave(ncls,nwt),dist(ncls,nwt),wterr
    INTEGER*4 maxwt(nvip),vip_to_rate(nvip)
    PARAMETER (addoz=0.22)

    COMMON /map/vips,ave,ratetable,dist,
     +   maxwt,vip_to_rate
    COMMON /ver/averev,avewt,first,heavy,last,
     +   nwtfix,wtfixrev,wtfixpcs,wtfixwt,nrevfix,revfixrev,revfixpcs,revfixwt,
     +   mvrev,mvwt,mvpcs,wterr

    averev = rev/pcs
    avewt = wt/pcs
    first = ratetable(ivip,1)
    last = ratetable(ivip,maxwt(ivip))
```

```
      verify = 0
      IF (rev.eq.0.or.pcs.eq.0.or.wt.eq.0) THEN
         verify = 1
      END IF
      IF (verify.eq.0) THEN
         IF (icls.eq.8.or.icls.eq.9.
     +      or.icls.eq.10.or.icls.eq.11.or.icls.eq.18.or.icls.eq.19.
     +      or.icls.eq.20.or.icls.eq.27.or.icls.eq.28) THEN   ! Nonstandard
            IF ((avewt*16.0).gt.1.01) THEN
               verify = 10       ! Weight Error
               PRINT *,"NS wt error, vip: ",vips(ivip)," icls: ",
     +            icls," avewt: ",avewt*16.0
            END IF
         ELSE IF (icls.gt.28)        THEN                ! Cards
            IF (ABS(averev-first).gt.tol) THEN
                  verify = 4     ! Revenue Error    ! BLOCK ONLY FOR AP 5
            END IF
         ELSE                     ! Letters, Flats and Parcels
            IF (averev.le.(first-tol).or.averev.gt.(last+tol)) THEN
               verify = 4        ! Revenue Error    ! BLOCK ONLY FOR AP 5
            END IF
            IF (verify.eq.0) THEN
               IF (icls.ge.21.and.icls.le.24) THEN    ! auto letter
                  IF (avewt*16.0.gt.3.31) THEN
                     verify = 5
                     PRINT *,"HEAVY vip: ",vips(ivip)," icls: ",
     +                  icls," avewt: ",avewt*16.0
                  END IF
               ELSE
                  IF (avewt*16.0.gt.13.01) THEN
                     verify = 5
                     PRINT *,(avewt*16.0-13.0)
                     PRINT *,"HEAVY over 13, vip: ",vips(ivip)," icls: ",
     +                  icls," avewt: ",avewt*16.0
                  END IF
               END IF
               IF (verify.eq.0) THEN
                  IF (avewt*16.0.gt.2.0.and.(icls.lt.29)) THEN  !  not cards
                     first = ratetable(ivip,3) - 0.44
                  END IF
                  rvlst = pcs * first
                  rvres = rev - rvlst
                  wtres = (rvres/addoz)/16.0
                  IF ((wtres+pcs/16.0).lt.(wt - 1)) THEN
                     verify = 8
C+++                     WRITE(6,'(I2,2F12.2,F5.3,F10.2,F6.3)')
C+++     +                  icls,rev,pcs,first,rvres,rev/pcs
C+++                     WRITE(6,'(2X,4F12.2,2F8.3,F10.2)')
C+++     +                  wtres,pcs/16.0,wtres+pcs/16.0,wt,
C+++     +                  (wtres+pcs/16.0)/wt,wt/pcs*16.0,wt-(wtres+pcs/16.0)
                     wterr = (wt-(wtres+pcs/16.0)) + wterr
                  END IF
               END IF
            END IF
         END IF
      END IF

C     check for heavy cards
      if ((icls.gt.28).and.(avewt.gt.0.3).and.(pcs.gt.500)) then
            print *,'*heavy card for;'
77          format(a6,1x,a5)
            write(6,77) vips(ivip)
            print *, rev, pcs, wt
      end if

      RETURN
      END

C-------  ------------------------------------------------------------------

      SUBROUTINE repair(error,ivip,icls,rev,pcs,wt,istrata,itype)

      INTEGER*4 ncls,nwt,nvip,redist,error
      REAL*8    tol,wtol
      PARAMETER (ncls=34,nwt=13,tol=0.01,wtol=.01,nvip=102)

      CHARACTER*5 vips(nvip),newvip
      INTEGER*4 ivip,icls,iwt,nwtfix,nrevfix,istrata,itype
      REAL*8    rev,pcs,wt,averev,avewt,first,heavy,last
```

```
 REAL*8     wtfixrev,wtfixpcs,wtfixwt,mvrev,mvwt,mvpcs
 REAL*8     revfixrev,revfixpcs,revfixwt
 REAL*8     ratetable(nvip,nwt)
 REAL*8     ave(ncls,nwt),dist(ncls,nwt),wterr
 INTEGER*4 maxwt(nvip),vip_to_rate(nvip),searchc

 COMMON /map/vips,ave,ratetable,dist,
+   maxwt,vip_to_rate
 COMMON /ver/averev,avewt,first,heavy,last,
+   nwtfix,wtfixrev,wtfixpcs,wtfixwt,nrevfix,revfixrev,revfixpcs,revfixwt,
+   mvrev,mvwt,mvpcs,wterr

 IF (error.eq.1) THEN     ! missing data
    IF (rev.eq.0.and.wt.gt.0.and.pcs.gt.0) THEN
       IF (avewt*16.0.le.1.0) THEN
          rev = ratetable(ivip,1)*pcs
       ELSE IF ((avewt*16.0).gt.1.0) THEN
          DO iwt=1,maxwt(ivip)
             rev = rev + ratetable(ivip,iwt)*pcs*dist(icls,iwt)
          END DO
       END IF
       nrevfix = nrevfix + 1
       revfixrev = revfixrev + rev
       revfixpcs = revfixpcs + pcs
       revfixwt = revfixwt + wt
    ELSE IF (wt.eq.0) THEN
       IF (averev.le.(ratetable(ivip,1)+tol)) THEN
          wt = pcs*ave(icls,1)/16.0
       ELSE
          DO iwt=1,maxwt(ivip)
             wt = wt + pcs*dist(icls,iwt)*ave(icls,iwt)/16.0
          END DO
       END IF
       nwtfix = nwtfix + 1
       wtfixrev = wtfixrev + rev
       wtfixpcs = wtfixpcs + pcs
       wtfixwt = wtfixwt + wt
    END IF
 END IF

 IF (error.eq.4) THEN       ! revenue error
    IF (icls.gt.28) THEN    ! cards
       IF (averev.gt.ratetable(ivip,1)) THEN
          error = redist(icls,first,ivip,rev,pcs,wt,istrata,itype)  ! C ONLY AP 5
          continue
       END IF
    ELSE                    ! not cards
       IF ((averev.lt.ratetable(ivip,1)).and.(icls.eq.1.or.
+          icls.eq.15.or.(icls.ge.21.and.icls.le.24))) THEN   ! letters
          error = redist(icls,first,ivip,rev,pcs,wt,istrata,itype)  ! C ONLY AP 5
          continue
       ELSE
          PRINT *, "Attempting Repair for averev gt 1 oz. rate :"
          WRITE(6,'("icls: ",I2," rev: ",F11.0," pcs: ",F11.0," wt: ",
+            F11.0)') icls,rev,pcs,wt
          pcs = 0.0
          DO iwt=1,maxwt(ivip)
             pcs = pcs + rev*dist(icls,iwt)/ratetable(ivip,iwt)
          END DO
          PRINT *, "After Repair:"
          WRITE(6,'("icls: ",I2," rev: ",F11.0," pcs: ",F11.0," wt: ",
+            F11.0)') icls,rev,pcs,wt
          mvrev = mvrev + rev
          mvpcs = mvpcs + pcs
          mvwt = mvwt + wt
       END IF
    END IF
 END IF

 IF ((error.eq.5).and.(avewt*16.0.le.13.01)) THEN ! wt error; convert to F
    IF (icls.ge.21.and.icls.le.24) THEN     ! auto letter
       error = 11
       newvip = vips(ivip)
       if (icls.eq.21) then
          icls = 25
          newvip(2:5) = '1552'
          ivip = 9
       else if (icls.eq.22) then
          icls = 26
          newvip(2:5) = '1542'
```

```
            ivip = 7
         else if (icls.eq.17) then
            icls = 26
            newvip(2:5) = '1542'
            ivip = 7
         else if (icls.eq.24) then
            icls = 26
            newvip(2:5) = '1542'
            ivip = 7
         end if

      END IF
   END IF

   IF ((error.eq.5).and.(avewt*16.0.gt.13.01)) THEN      ! wt error > 13 oz
      PRINT *, "Repairing Weight Problem over 13 oz. : icls =",icls
      WRITE(6,'("Before: Rev = ",F11.2," Pcs = ",F11.2," Wt = ",F11.2)')
     +    rev,pcs,wt
      wt = 0.0
      rev = 0.0
      DO iwt=1,maxwt(ivip)
         rev = rev + pcs*dist(icls,iwt)*ratetable(ivip,iwt)
         wt = wt + pcs*dist(icls,iwt)*ave(icls,iwt)/16.0
      END DO
      WRITE(6,'("After : Rev = ",F11.2," Pcs = ",F11.2," Wt = ",F11.2)')
     +    rev,pcs,wt
   END IF

   IF (error.eq.8) THEN    ! revise wt
      PRINT *, "Repairing Inconsistent Revenue and Weight: vip = ",
     +    vips(ivip)
      WRITE(6,'("Before: Rev = ",F11.2," Pcs = ",F11.2," Wt = ",F11.2)')
     +    rev,pcs,wt
      wt = (pcs + (rev-pcs*ratetable(ivip,1))/0.22)/16.0
      WRITE(6,'("After : Rev = ",F11.2," Pcs = ",F11.2," Wt = ",F11.2)')
     +    rev,pcs,wt
   END IF

   IF (error.eq.10) THEN     ! NS > 1 oz
      PRINT *, "Repairing NS > 1 oz. wt : vip = ",vips(ivip)
      newvip = vips(ivip)
         if (newvip(3:3).eq.'6')  newvip(3:3) = '5'
         if (newvip(3:3).eq.'8')  newvip(3:3) = '7'
         ivip = searchc(vips,nvip,newvip)
         IF (ivip.gt.0) THEN
            icls = vip_to_rate(ivip)
            PRINT *, "Moved to vip: ",newvip," icls = ",icls
         ELSE
            PRINT *, "Can't repair this rec;"
            print *,rev,' ',pcs,' ',wt
      .     END IF
   END IF

   RETURN
   END
C  ------------------------------------------------------------------------
   INTEGER*4 FUNCTION redist(icls,first,ivip,rev,pcs,wt,istrata,itype)

   INTEGER*4 nvip,ncls,searchc,nrpw,nstrata,ntype,nwt
   PARAMETER (nvip=102,ncls=34,nrpw=3,nstrata=20,ntype=2,nwt=13)

   INTEGER*4 icls,ltrcls,cdcls,icdvip,iltrvip,ivip
   INTEGER*4 vip_to_rate(nvip),istrata,itype,maxwt(nvip)
   CHARACTER*5 vip,ltrvip,cdvip,vips(nvip)
   REAL*8 rev,pcs,wt,ratetable(nvip,nwt),ave(ncls,nwt),dist(ncls,nwt)
   REAL*8 first,cdrate,ltrate,ltrs,cds,dataredist(nrpw,nvip,nstrata,ntype),
  +   ltrrev,ltrpcs,ltrwt,cdrev,cdpcs,cdwt,fltrev,fltpcs,fltwt,fcdrev,
  +   fcdpcs,fcdwt

   COMMON /map/vips,ave,ratetable,dist,
  +   maxwt,vip_to_rate
   COMMON /red/
  +   ltrrev,ltrpcs,ltrwt,cdrev,cdpcs,cdwt,
  +   dataredist,
  +   fltrev,fltpcs,fltwt,fcdrev,fcdpcs,fcdwt

   vip = vips(ivip)
   cdvip = vips(ivip)
   ltrvip = vips(ivip)
   cdcls = icls
```

```
      ltrcls = icls
      cdrate = first
      ltrate = first
      icdvip = ivip
      iltrvip = ivip


      IF (vip(5:5).eq.'3') THEN      ! card
         ltrvip(5:5) = '1'
         iltrvip = searchc(vips,nvip,ltrvip)
         ltrcls = vip_to_rate(iltrvip)
         ltrate = ratetable(iltrvip,1)
      ELSE
         cdvip(5:5) = '3'
         icdvip = searchc(vips,nvip,cdvip)
         cdcls = vip_to_rate(icdvip)
         cdrate = ratetable(icdvip,1)
      END IF

      ltrs = pcs - (rev - ltrate*pcs)/(cdrate - ltrate)
      cds  = pcs - ltrs

      IF (cds.le.0.or.ltrs.le.0) THEN
         redist = 7
C         PRINT *, "Redist Problem:"
C         WRITE(6,'(A5," Rate: ",F7.3," Cards: ",F10.0," Letters: ",F10.0)')
C     +      vip,rev/pcs,cds,ltrs
         IF (vip(5:5).eq.'3') THEN      ! convert cards to letters
            cds = 0.0
            ltrs = rev/ltrate
            wt = ltrs*ave(ltrcls,1)/16.0
            fltrev = fltrev + rev
            fltpcs = fltpcs + pcs
            fltwt = fltwt + wt
         ELSE                           ! convert letters to cards
            ltrs = 0.0
            cds = rev/cdrate
            wt = cds*ave(cdcls,1)/16.0
            fcdrev = fcdrev + rev
            fcdpcs = fcdrev + pcs
            fcdwt = fcdwt + wt
         END IF
      ELSE                ! cards > 0 and letters > 0
         redist = 6
         cdrev = cdrev + cdrate*cds
         ltrrev = ltrrev + ltrate*ltrs
         ltrpcs = ltrpcs + ltrs
         cdpcs = cdpcs + cds
         ltrwt = ltrwt + ave(ltrcls,1)/16.0*ltrs
         cdwt = cdwt + ave(cdcls,1)/16.0*cds
         wt = ave(ltrcls,1)/16.0*ltrs + ave(cdcls,1)/16.0*cds
      END IF
      dataredist(1,iltrvip,istrata,itype) =
     +   dataredist(1,iltrvip,istrata,itype) + ltrate*ltrs
      dataredist(2,iltrvip,istrata,itype) =
     +   dataredist(2,iltrvip,istrata,itype) + ltrs
      dataredist(3,iltrvip,istrata,itype) =
     +   dataredist(3,iltrvip,istrata,itype) + ave(ltrcls,1)/16.0 * ltrs
      dataredist(1,icdvip,istrata,itype) =
     +   dataredist(1,icdvip,istrata,itype) + cdrate*cds
      dataredist(2,icdvip,istrata,itype) =
     +   dataredist(2,icdvip,istrata,itype) + cds
      dataredist(3,icdvip,istrata,itype) =
     +   dataredist(3,icdvip,istrata,itype) + ave(cdcls,1)/16.0 * cds


      RETURN
      END

C-----------------------------------------------------------------------------

      INTEGER*4 FUNCTION type(vip,icls)

      INTEGER*4 icls
      CHARACTER*5 vip

         IF (vip(1:1).eq.'1'.or.vip(1:1).eq.'2') THEN
            type = 2                ! All Stamped and Metered
         ELSE IF (vip(1:1).eq.'0') THEN
            type = 1                ! PI
```

```
   END IF

RETURN
END
```

```
      PROGRAM ave

C     DESCRIPTION: Compute Average SM Revenue to Fill in Missing CBCIS offices
C                  on a quarterly basis
C
C     CREATED BY:       Bill Humphries
C     DATE:             Mon Apr 24 10:42:21 CDT 1995
C
C     LAST MODIFIED BY: Tom Ayen
C     DATE:             sep 24  1996; update for 97
C                       oct 6,  1998; update for 99

      IMPLICIT NONE

      INTEGER*4 nfin,nap,currentap
      PARAMETER (nfin=10826,nap=13)    !!! update if new fins
      PARAMETER (currentap = 13)        ! one or two digits

      REAL*8    smrev(nfin)/nfin*0.0/,pirev(nfin)/nfin*0.0/,avesm(nfin)/nfin*0.0/
      REAL*8    avepi(nfin)/nfin*0.0/
      REAL*8    pi,sm,sp
      INTEGER*4 apcount(nfin)/nfin*0/
      INTEGER*4 ifin,i,ier,iap,in,found,nonzero,searchc
      CHARACTER*6 fins(nfin),fin
      CHARACTER*2 cap(nap)/'01','02','03','04','05','06','07','08','09','10',
     +   '11','12','13'/

C     Open and Read Map of Finance Numbers and Their Strata

      OPEN(1,FILE='finstrata.new.00')

2     FORMAT(a6,1x,i3)
      DO i = 1, nfin
         READ(1,2) fins(i)
      END DO

      PRINT *, "Read Strata Maps."

11    FORMAT(A6,3F10.0)

      DO iap=1,nap
         ier = 0
         in = 0
         nonzero = 0
         found = 0
        IF (iap.le.currentap) THEN    ! read current FY files in current dir
         OPEN(10,FILE='finrev.'//cap(iap))
         DO WHILE (ier.eq.0)
            READ(10,11,IOSTAT=ier,END=20) fin,pi,sm,sp
            in = in + 1
            IF (sm.gt.0.or.pi.gt.0) THEN
               nonzero = nonzero + 1
               ifin = searchc(fins,nfin,fin)
               IF (ifin.gt.0) THEN
                  found = found + 1
                  apcount(ifin) = apcount(ifin) + 1
                  smrev(ifin) = smrev(ifin) + sm
                  pirev(ifin) = pirev(ifin) + pi
               END IF
            END IF
         END DO
20       PRINT *, "ap ",iap," of current FY"
         PRINT *, "in: ",in," nonzero: ",nonzero," found: ",found
        ELSE                          ! use prior year files
         OPEN(10,FILE='../CBCIS99/finrev.'//cap(iap))
         DO WHILE (ier.eq.0)
            READ(10,11,IOSTAT=ier,END=21) fin,pi,sm,sp
            in = in + 1
            IF (sm.gt.0.or.pi.gt.0) THEN
               nonzero = nonzero + 1
               ifin = searchc(fins,nfin,fin)
               IF (ifin.gt.0) THEN
                  found = found + 1
                  apcount(ifin) = apcount(ifin) + 1
                  smrev(ifin) = smrev(ifin) + sm
                  pirev(ifin) = pirev(ifin) + pi
               END IF
            END IF
         END DO
21       PRINT *, "ap ",iap," of last FY"
```

```
         PRINT *, "in: ",in," nonzero: ",nonzero," found: ",found
        END IF
      END DO

      OPEN(30,FILE='averev.dat')
31    FORMAT(A6,2F20.5,1X,I2)
      DO ifin=1,nfin
         IF (apcount(ifin).gt.0) THEN
            avesm(ifin) = smrev(ifin)/apcount(ifin)
            avepi(ifin) = pirev(ifin)/apcount(ifin)
         ELSE
            avesm(ifin) = 0.0
            avepi(ifin) = 0.0
         END IF
         WRITE(30,31) fins(ifin),avepi(ifin),avesm(ifin),apcount(ifin)
      END DO

      STOP
      END
```

```
      PROGRAM fit_nonp

C     DESCRIPTION:
C         Estimates SM Revenues for YTD Non-CBCIS sites
C         on AP basis using regression equation
C
C
C     CREATED BY:        Bill Humphries
C     DATE:              Thu Nov 10 09:03:52 CST 1994
C
C     LAST MODIFIED BY: Tom Ayen
C     DATE:              01/19/96; use last ap smhat if meter is negative
C                        02/28/96; add new parameters, remove finrev read
C                        11/12/96; update for FY97
C                        04/04/97; if Meter acct is neg, set SMhat = 0.
C                        08/21/97; put smhat for fins not in map into 999999
C                        10/06/98; update for FY99
C
C     Needs the 2 digit ap argument
C
C     NOTE; needs manual edit for ap 1 run for opening 'previous ap' file
C        nonpbfit.13.ap since it will be in a different directory.
C

      IMPLICIT NONE

C     Parameter Estimates:

      REAL*8 a1,b1,c1
      REAL*8 a2,b2,c2
      REAL*8 a3,b3,c3
      REAL*8 a4,b4,c4
      REAL*8 a,b,c
      PARAMETER (a1=0.234259,b1=0.187064,c1=-0.004364)  ! updated 2-28-96
      PARAMETER (a2=0.244983,b2=0.248894,c2=-0.004150)
      PARAMETER (a3=0.240415,b3=0.254393,c3=-0.004106)
      PARAMETER (a4=0.261537,b4=0.161755,c4=-0.003343)

C     Constants

      INTEGER*4 nfin, nnegmeter, iap, nlast


      CHARACTER*6 fin, finex
      character*6 fins(20000)/20000*' '/
      CHARACTER*2 ap, lastap

      REAL*8    stamp,meter,piprsrt,smhat,total
      REAL*8    stamp9/0/,meter9/0/,piprsrt9/0/,smhat9/0/
      REAL*8    avesm(20000)/20000*0.0/,avepi(20000)/20000*0.0/
      REAL*8    min,max,tstamp,tmeter,tpiprsrt
      INTEGER*4 ier,i,j,searchc,out,in,nnon,ifin,lastcount,inx
      INTEGER*4 unmap/0/

      ier = 0

C     Get Argument

      CALL getarg(1,ap)
      PRINT *, "Working on AP",ap
      READ(ap,'(I2)') iap


      OPEN(15,FILE='averev.dat')
1     FORMAT(a6,2F20.5)
      ier = 0
      i = 1
      DO WHILE (ier.eq.0)
         READ (15,1,IOSTAT=ier,END=19) fins(i), avesm(i),avepi(i)
         i = i + 1
      END DO
19    PRINT *, "Read ",i-1," lines from averev.dat."
      nfin = i - 1

      ier = 0
      nnegmeter=0
      min = 99999999999.0
      max = 0
      total=0.0
```

```
      tstamp = 0.0
      tmeter = 0.0
      tpiprsrt = 0.0
      in=0
      out=0
      nnon=0


      OPEN(21,FILE='findata.ap'//ap)
      OPEN(30,FILE='nonpbfit.'//ap//'.rev')

3     FORMAT(A6,3F15.2)
4     FORMAT(A6,",",F20.10,3F12.0)
      i=0
      lastcount = 0
      DO WHILE (ier.eq.0)
         READ(21,3,IOSTAT=ier) fin,stamp,meter,piprsrt
         in = in + 1
         stamp = - stamp
         meter = - meter
         piprsrt = - piprsrt
         ifin = searchc(fins,nfin,fin)
         IF (ifin.gt.0) THEN     ! fin found in averev file
            IF ((avesm(ifin)+avepi(ifin)).eq.0) THEN ! need to fit revenue
               nnon = nnon + 1
               i=i+1
               IF (meter.lt.0) THEN ! if negative meter, set smhat = 0
                  WRITE(6,3) fin,stamp,meter,piprsrt
                  smhat=0
                  nnegmeter=nnegmeter+1
                        go to 200
               ELSE
                  IF ((iap.eq.1).or.(iap.eq.2).or.(iap.eq.3)) THEN
                     a=a1
                     b=b1
                     c=c1
                  ELSE IF ((iap.eq.4).or.(iap.eq.5).or.(iap.eq.6)) THEN
                     a=a2
                     b=b2
                     c=c2
                  ELSE IF ((iap.eq.7).or.(iap.eq.8).or.(iap.eq.9)) THEN
                     a=a3
                     b=b3
                     c=c3
                  ELSE
                     a=a4
                     b=b4
                     c=c4
                  END IF
                  smhat = a*meter + b*piprsrt + c*meter*meter/1000000
                  IF (smhat.lt.0) THEN
                     smhat=0
                  END IF
               END IF
200            continue
               tstamp = tstamp + stamp
               tmeter = tmeter + meter
               tpiprsrt = tpiprsrt + piprsrt
               IF (smhat.ge.0) THEN
                  WRITE(30,4) fin,smhat,meter,stamp,piprsrt
                  IF (smhat.gt.max) max = smhat
                  IF (smhat.lt.min) min = smhat
                  total = total + smhat
                  out=out+1
               ELSE
                  PRINT *, "Estimated Stamped and Metered Revenue for Finance Number ", fin, " negative."
                  PRINT *, meter, stamp, piprsrt
               END IF
            END IF
         ELSE          ! fin not in strata map
            IF (meter.gt.0.and.piprsrt.gt.0) THEN   ! need to fit revenue
               nnon = nnon + 1
               i=i+1
               unmap = unmap + 1
               finex = fin
               fin = '999999'   ! put all data in holder

                  IF ((iap.eq.1).or.(iap.eq.2).or.(iap.eq.3)) THEN
                     a=a1
                     b=b1
```

```
                  c=c1
               ELSE IF ((iap.eq.4).or.(iap.eq.5).or.(iap.eq.6)) THEN
                  a=a2
                  b=b2
                  c=c2
               ELSE IF ((iap.eq.7).or.(iap.eq.8).or.(iap.eq.9)) THEN
                  a=a3
                  b=b3
                  c=c3
               ELSE
                  a=a4
                  b=b4
                  c=c4
               END IF
               smhat = a*meter + b*piprsrt + c*meter*meter/1000000
               IF (smhat.lt.0) THEN
                  smhat=0
               END IF
            tstamp = tstamp + stamp
            tmeter = tmeter + meter
            tpiprsrt = tpiprsrt + piprsrt
            IF (smhat.ge.0) THEN
               smhat9 = smhat9 + smhat
               meter9 = meter9 + meter
               stamp9 = stamp9 + stamp
               piprsrt9 = piprsrt9 + piprsrt
               total = total + smhat
               out=out+1
C               print*,finex,' unmapped est    ',smhat
            ELSE
               PRINT *, "Estimated Stamped and Metered Revenue for Finance Number ", fin, " negative."
               PRINT *, meter, stamp, piprsrt
            END IF
         END IF
      END IF
   END DO

C  write out for fin 999999

14    FORMAT("999999,",F20.10,3F12.0)
      WRITE(30,14) smhat9,meter9,stamp9,piprsrt9
   PRINT *, "IER = ",ier

   PRINT *,'Done with AP ',ap
   IF ((iap.eq.1).or.(iap.eq.2).or.(iap.eq.3)) THEN
      print*,'Used quarter one parameter estimates'
   ELSE IF ((iap.eq.4).or.(iap.eq.5).or.(iap.eq.6)) THEN
      print*,'Used quarter two parameter estimates'
   ELSE IF ((iap.eq.7).or.(iap.eq.8).or.(iap.eq.9)) THEN
      print*,'Used quarter three parameter estimates'
   ELSE
      print*,'Used quarter four parameter estimates'
   END IF

   PRINT *
   PRINT *, "Read ",in-1," records"
   PRINT *, nnon," records need to be fitted."
   PRINT *, nnegmeter, " records had negative metered revenue."
   PRINT *, out," records had non-negative fitted values."
   PRINT *, "Estimated ",total," in revenue."
   PRINT *, "Min: ",min
   PRINT *, "Max: ",max
   PRINT *, "Ave: ",total/out
   PRINT *, "Stamped: ",tstamp
   PRINT *, "Metered: ",tmeter
   PRINT *, "SM Tot.: ",tstamp+tmeter
   PRINT *, "PI Psrt: ",tpiprsrt
   PRINT *, "Ratios"
   PRINT *, "Stamped & Metered to PI Psrt:"
   PRINT *, (tstamp+tmeter)/tpiprsrt
   PRINT *, "Stamped to PI Psrt:"
   PRINT *, tstamp/tpiprsrt
   PRINT *, "Metered to PI Psrt:"
   PRINT *, tmeter/tpiprsrt
   PRINT *, "non mapped fins smhat total "
   PRINT *, smhat9
   PRINT *, "non mapped fins  "
   PRINT *, unmap

   STOP
```

END

```
      PROGRAM control_shape_half

C     DESCRIPTION: Control distributions of PERMIT revenue, pieces
C        and weight to RPW revenue, aggregate to augmented RPW category list
C        (vipmap.lst). Estimate weight of bad weight pieces, redistribute
C        non-identical pieces. Write control factors to a file for use in
C        other control and distribution programs.
C
C     This progaram uses files that only have ID GOOD wt data with half oz increments

C     CREATED BY:       TA
C     DATE:             feb 3 2000
C
C     LAST MODIFIED BY:

C >>>>>>>  USE ONLY FOR Q2 1999 AND BEYOND. THE THE .OLD PROGRAM FOR Q1 99

      IMPLICIT NONE

      INTEGER*4 nwt,nstrata,nrpw,nid,ngb,ngbn
      PARAMETER (nwt=17,nstrata=20,nrpw=3,nid=2,ngb=2,ngbn=3)
      INTEGER*4 nap,nq,ncls,nind,nnewstr,ntype,size,size1,size3
      parameter (nap=13,nq=4)                !!! update - also in subroutine
      PARAMETER (ncls=34,nind=3,nnewstr=4,ntype=3)

      INTEGER*4 iwt,istrata,irpw,iid,igb,iap,iq,icls,iind,istratax
      INTEGER*4 i,j,k,ier,indicia,isize,igbn

      REAL*8    rev,pcs,wt,line(nq),rateline(nwt),rateline00(nwt)
      INTEGER*4 in


      REAL*8 permit(ncls,nnewstr,nwt,nind,nrpw,nq,ngbn)
      REAL*8 id2(ncls,nwt,nrpw,nq), ni(ncls,nwt,nrpw,nq)
      REAL*8 avewt(ncls,nwt), totid(ncls,nwt)
      REAL*8 totwt(ncls,nwt), data(ncls,nwt,nind,nrpw,nq)
      REAL*8 report(ncls,nwt,nind,nrpw,nq), total(nrpw,nind)
      REAL*8 reportx(ncls,nwt,nrpw,nq)
      REAL*8 ppriority(nnewstr,nq) ,midr(ncls,nq)
      REAL*8 midp(ncls,nq), mnir(ncls,nq)
      REAL*8 mnip(ncls,nq), distribution(ncls,nwt)
      REAL*8 indshare(ncls,nind,nq),indrev(ncls,nind,nq)
      REAL*8 ratetable(ncls,nwt), ratetbl99(ncls,nwt), ratetbl(ncls,nwt)

      COMMON /rate/ratetable,midp,mnip,midr,distribution

      CHARACTER*2 cap(nap)/'01','02','03','04','05','06','07','08','09',
     +                     '10','11','12','13'/                  ! update
      INTEGER*4 apq(nap)/1,1,1,2,2,2,3,3,3,4,4,4,4/           ! update
      INTEGER*4 stratamap(20)/1,1,1,1,1,1,1,1,1,1,1,1,1,2,2,3,3,4,4/

      REAL*8 id(ncls,nwt,nind,nrpw,4)     ! special case
      REAL*8 bad(ncls,nwt,nind,nrpw,4)

      REAL*8 rpw(nq),smrpw(nq),rrpw(nq)
      REAL*8 smrv(nnewstr,nq)
      REAL*8 psmrv(nnewstr,nq)
      REAL*8 pirv(nnewstr,nq)
      REAL*8 ppirv(nnewstr,nq)
      REAL*8 trv(nnewstr,nq)
      REAL*8 ptrv(nnewstr,nq)
      REAL*8 pitb(nnewstr,nq)
      REAL*8 ppitb(nnewstr,nq)
      REAL*8 prv(nnewstr,nq,nind)
      real*8 prvx(20,4,4)              ! temp
      real*8 prvx1/0.0/                          ! temp
      REAL*8 pirv1(nnewstr,nq),pirv2(nnewstr,nq),pirv3(nq)
      REAL*8 smrv1(nnewstr,nq),smrv2(nnewstr,nq),smrv3(nq),smrv4(nq)
      REAL*8 rv1(nq)
      REAL*8 rv2(nq)
      REAL*8 raw(nrpw)
      REAL*8 picontrol(nnewstr,nq)
      REAL*8 xpicontrol(nnewstr,nq)
      REAL*8 smcontrol(nnewstr,nq)
      REAL*8 xsmcontrol(nnewstr,nq)
      REAL*8 ppisp(nnewstr,nq)
      REAL*8 psmsp(nnewstr,nq)
      REAL*8 sp(nnewstr,nq)
      REAL*8 rcontrol(nq)
      REAL*8 uninflated(nrpw,ncls)
```

```
      REAL*8 inflated(nrpw,ncls)
      REAL*8 finalrev(ncls)
      REAL*8 finalwt(ncls)
      REAL*8 check(nq)
      real*8 grand/0.0/,grev/0.0/,infpcs/0.0/,checkall/0.0/
      real*8 totind(4)/4*0.0/
      REAL*8 prp,rprp,indtot
      REAL*8 checkpi(nnewstr,nq),checksm(nnewstr,nq),allraw,tpipri
      REAL*8 piraw(nnewstr,nq),smraw(nnewstr,nq),tin,tpin,totalrawrev
      real*8 tpitb/0.0/,tsm/0.0/
      real*8 totnctbq1/0.0/,totnctbq2/0.0/,totnctbq3/0.0/,totnctbq4/0.0/
      real*8 revadd(4,4)/16*0.0/
      real*8 tpipriq(4)/4*0.0/
      real*8 totcheckpi(4)
      real*8 tot4
      real*8 totapwt(4)    ! 4 qtrs


C     Zero-out PERMIT  matricies

      tin = 0.0
      tpin = 0.0
      totalrawrev = 0.0

      do irpw = 1,nrpw
         raw(irpw) = 0.0
         do iind = 1,nind
            total(irpw,iind) = 0.0
            do icls = 1,ncls
               do iwt = 1,nwt
                  do iq = 1,nq
                     id(icls,iwt,iind,irpw,iq) = 0.0
                     bad(icls,iwt,iind,irpw,iq) = 0.0
                     data(icls,iwt,iind,irpw,iq) = 0.0
                     report(icls,iwt,iind,irpw,iq) = 0.0
                     do igbn = 1,ngbn
                        do istrata = 1,nnewstr
                           permit(icls,istrata,iwt,iind,irpw,iq,igbn) = 0.0
                        end do
                     end do
                  end do
               end do
            end do
         end do
      end do

      do istrata = 1,nnewstr
         do iq = 1,nq
            pitb(istrata,iq) = 0.0
            ppitb(istrata,iq) = 0.0
            ppirv(istrata,iq) = 0.0
            psmrv(istrata,iq) = 0.0
            ptrv(istrata,iq) = 0.0
            smrv(istrata,iq) = 0.0
            smrv2(istrata,iq) = 0.0
            trv(istrata,iq) = 0.0
            pirv(istrata,iq) = 0.0
            picontrol(istrata,iq) = 0.0
            xpicontrol(istrata,iq) = 0.0
            smcontrol(istrata,iq) = 0.0
            xsmcontrol(istrata,iq) = 0.0
            ppisp(istrata,iq) = 0.0
            psmsp(istrata,iq) = 0.0
            sp(istrata,iq) = 0.0
            do iind = 1,nind
               prv(istrata,iq,iind) = 0.0
            end do
         end do
      end do

      do icls = 1,ncls
         do iwt = 1,nwt
            distribution(icls,iwt) = 0.0
            avewt(icls,iwt) = 0.0
            totid(icls,iwt) = 0.0
            totwt(icls,iwt) = 0.0
            do irpw = 1,nrpw
               do iq = 1,nq
                  id2(icls,iwt,irpw,iq) = 0.0
                  ni(icls,iwt,irpw,iq) = 0.0
```

```
                  reportx(icls,iwt,irpw,iq) = 0.0
               end do
            end do
         end do
      end do

      do iq = 1,nq
         smrv3(iq) = 0.0
         smrv4(iq) = 0.0
         do icls = 1,ncls
            midr(icls,iq) = 0.0
            midp(icls,iq) = 0.0
            mnir(icls,iq) = 0.0
            mnip(icls,iq) = 0.0
            do iind = 1,nind
               indshare(icls,iind,iq) = 0.0
               indrev(icls,iind,iq) = 0.0
            end do
         end do
      end do

      do irpw = 1,nrpw
         do icls = 1, ncls
            uninflated(irpw,icls) = 0.0
            inflated(irpw,icls) = 0.0
         end do
      end do


      PRINT *, "Zeroed PERMIT  Matrices."

C     Read in nctb revenue for all offices-including nonpermit

      OPEN(10,FILE='trevenue.dat',readonly) !SM incl fitted
      OPEN(20,FILE='prevenue.dat',readonly) !SM for PERMIT
C                  offices from finrev.1st with missing data added

      OPEN(40,FILE='tnctbstr.dat',readonly) ! NCTB PI presort
      OPEN(50,FILE='pnctbstr.dat',readonly) ! PERMIT offices

32    FORMAT(4F15.0)
      DO istrata=1,nstrata
         READ(10,32) line
         DO iq=1,nq
            trv(stratamap(istrata),iq) =
     +         trv(stratamap(istrata),iq) + line(iq)
         tin = tin + line(iq)
         END DO
         READ(10,32) line
         DO iq=1,nq
            pirv(stratamap(istrata),iq) =
     +         pirv(stratamap(istrata),iq) + line(iq)
         END DO
         READ(10,32) line
         DO iq=1,nq
            smrv(stratamap(istrata),iq) =
     +         smrv(stratamap(istrata),iq) + line(iq)
         tsm = tsm + line(iq)
         END DO

         READ(20,32) line
         DO iq=1,nq
            ptrv(stratamap(istrata),iq) =
     +         ptrv(stratamap(istrata),iq) + line(iq)
         tpin = tpin + line(iq)
         END DO
         READ(20,32) line
         DO iq=1,nq
            ppirv(stratamap(istrata),iq) =
     +         ppirv(stratamap(istrata),iq) + line(iq)
         END DO
         READ(20,32) line
         DO iq=1,nq
            psmrv(stratamap(istrata),iq) =
     +         psmrv(stratamap(istrata),iq) + line(iq)
         END DO

      END DO
      DO istrata=1,nstrata
         READ(40,32) line
```

```
          DO iq=1,nq
             pitb(stratamap(istrata),iq) =
     +          pitb(stratamap(istrata),iq) + line(iq)
          END DO
          READ(50,32) line
          DO iq=1,nq
             ppitb(stratamap(istrata),iq) =
     +          ppitb(stratamap(istrata),iq) + line(iq)
          END DO
       END DO


 33    FORMAT(A20,4F15.0)
 34    FORMAT(A20,4F15.10)

       WRITE (6,33) "All,     PMT   :     ", tin, tpin
       print*,' '
       print*,'                              Q1              Q2              Q3          Q4    '

       DO istrata=1,nnewstr
          WRITE(6,33) "Total Revenue:  ",(trv(istrata,iq),iq=1,nq)
       END DO
       DO istrata=1,nnewstr
          WRITE(6,33) "PI Presort    :  ",(pirv(istrata,iq),iq=1,nq)
       END DO
       DO istrata=1,nnewstr
          WRITE(6,33) "SM :             ",(smrv(istrata,iq),iq=1,nq)
       END DO
       DO istrata=1,nnewstr
          WRITE(6,33) "PERMIT Revenue: ",(ptrv(istrata,iq),iq=1,nq)
       END DO
       DO istrata=1,nnewstr
          WRITE(6,33) "PI Presort    :  ",(ppirv(istrata,iq),iq=1,nq)
       END DO
       DO istrata=1,nnewstr
          WRITE(6,33) "SM :             ",(psmrv(istrata,iq),iq=1,nq)
       END DO

       OPEN(8,FILE='newrates.half.00',readonly)

       DO i=1,ncls
          READ(8,'(8X,17(F12.3))') rateline00
          DO iwt=1, nwt
             ratetable(i,iwt) = rateline00(iwt)
          END DO
       END DO

C      READ PERMIT  Arrays by Quarter

C      Format for reading revenue, pieces and weight arrays:
 11    FORMAT(I3,I2,1x,I2,3I1,3F20.10)
 32    FORMAT(3X,3F15.0)
       DO iap=1,nap
          iq = apq(iap)
          OPEN(10,FILE='perhalf.1st.'//cap(iap),readonly)  ! <<<<<< NOTE
          PRINT *, "Opened perrolln.1st.",cap(iap)
          ier = 0
          in = 0
          DO WHILE (ier.eq.0)
             READ(10,11,IOSTAT=ier,END=12) icls,iwt,istrata,iind,
     +          iid,igb,rev,pcs,wt
C             istratax = istrata             ! temp
             istrata = stratamap(istrata)
             IF (((icls.ge.8.and.icls.le.14).or.icls.eq.18.or.icls.eq.19.or.
     +          icls.eq.20.or.icls.eq.27.or.icls.eq.28).and.iwt.gt.2.and.iap.gt.3) THEN
                PRINT *, "PERMIT Error:"
                PRINT *, "Heavy Nonstandard icls = ",icls," iwt = ",iwt,
     +             " iap = ",iap
             END IF
             IF (iid.eq.1.and.igb.eq.1) THEN
                igbn = 1
             ELSE IF (iid.eq.1.and.igb.eq.0) THEN
                igbn = 2
             ELSE
                igbn = 3
             END IF
             prv(istrata,iq,iind) = prv(istrata,iq,iind) + rev
C             prvx(istratax,iq,iind) = prvx(istratax,iq,iind) + rev   ! temp
             permit(icls,istrata,iwt,iind,1,iq,igbn) = rev +
     +          permit(icls,istrata,iwt,iind,1,iq,igbn)
```

```
            permit(icls,istrata,iwt,iind,2,iq,igbn) = pcs +
     +         permit(icls,istrata,iwt,iind,2,iq,igbn)
            permit(icls,istrata,iwt,iind,3,iq,igbn) = wt +
     +         permit(icls,istrata,iwt,iind,3,iq,igbn)
            raw(1) = raw(1) + rev
            raw(2) = raw(2) + pcs
            raw(3) = raw(3) + wt
            uninflated(1,icls) = uninflated(1,icls) + rev
            uninflated(2,icls) = uninflated(2,icls) + pcs
            IF (igb.eq.1) uninflated(3,icls) = uninflated(3,icls) + wt
            in = in + 1
         END DO
12       PRINT *, "Read ",in," records from perrolln.1st.",cap(iap),"."
         PRINT *, "ier = ",ier
         WRITE (6,31) (raw(irpw),irpw=1,3)
         DO irpw = 1,nrpw
            raw(irpw) = 0.0
         END DO
         CLOSE(10)
      END DO


C     Generate Controls ; Note case for Q2 99

C       OWN CONTROL
      DO istrata=1,nnewstr
         DO iq=1,nq
            IF (prv(istrata,iq,1).gt.0.0)  THEN
              xpicontrol(istrata,iq) = ppitb(istrata,iq)/ prv(istrata,iq,1)
            ELSE
               xpicontrol(istrata,iq) = 0.0
            END IF
            IF (prv(istrata,iq,2).gt.0.0) THEN
              xsmcontrol(istrata,iq) = psmrv(istrata,iq) / prv(istrata,iq,2)
            ELSE
               xsmcontrol(istrata,iq) = 0.0
            END IF

            pirv1(istrata,iq)  = prv(istrata,iq,1)*xpicontrol(istrata,iq)
            smrv1(istrata,iq) =  prv(istrata,iq,2)*xsmcontrol(istrata,iq)

         END DO
      END DO

      print*,' '

      DO istrata=1,nnewstr
         WRITE(6,33) "Uncontroled PPI   : ",(prv(istrata,iq,1),iq=1,nq)
      END DO
      DO istrata=1,nnewstr
         WRITE(6,33) "PPI Trial Balance: ",(ppitb(istrata,iq),iq=1,nq)
      END DO
      DO istrata=1,nnewstr
         WRITE(6,34) "PPI Control       : ",(xpicontrol(istrata,iq),iq=1,nq)
      END DO
      DO istrata=1,nnewstr
         WRITE(6,33) "Controled PPI     : ",(prv(istrata,iq,1)*
     +      xpicontrol(istrata,iq),iq=1,nq)
      END DO

      DO istrata=1,nnewstr
         WRITE(6,33) "Uncontroled PSM   : ",(prv(istrata,iq,2),iq=1,nq)
      END DO
      DO istrata=1,nnewstr
         WRITE(6,34) "PSM Control       : ",(xsmcontrol(istrata,iq),iq=1,nq)
      END DO
      DO istrata=1,nnewstr
         WRITE(6,33) "Controled PSM     : ",(prv(istrata,iq,2)*
     +      xsmcontrol(istrata,iq),iq=1,nq)
      END DO


C        SECOND STAGE CONTROL
      DO istrata=1,nnewstr
         DO iq=1,nq
            picontrol(istrata,iq) = pitb(istrata,iq)/pirv1(istrata,iq)

            pirv2(istrata,iq) = prv(istrata,iq,1)
     +         *xpicontrol(istrata,iq) * picontrol(istrata,iq)
            smcontrol(istrata,iq) = smrv(istrata,iq)/smrv1(istrata,iq)
```

```
            smrv2(istrata,iq) = prv(istrata,iq,2)*xsmcontrol(istrata,iq)
     +         *smcontrol(istrata,iq)
            smrv3(iq) = smrv3(iq) + smrv2(istrata,iq)
         END DO
      END DO

      print*,' '
      DO istrata=1,nnewstr
         WRITE(6,33) "Own Office PI      : ",(pirv1(istrata,iq),iq=1,nq)
      END DO
      DO istrata=1,nnewstr
         WRITE(6,33) "PI Trial Balance :  ",(pitb(istrata,iq),iq=1,nq)
      END DO
      DO istrata=1,nnewstr
         WRITE(6,34) "Non P Control    : ",(picontrol(istrata,iq),iq=1,nq)
      END DO
      DO istrata=1,nnewstr
         WRITE(6,33) "Controlled         : ",(pirv2(istrata,iq),iq=1,nq)
      END DO

      DO istrata=1,nnewstr
         WRITE(6,33) "Own Office SM      : ",(smrv1(istrata,iq),iq=1,nq)
      END DO
      DO istrata=1,nnewstr
         WRITE(6,33) "Total SM           : ",(smrv(istrata,iq),iq=1,nq)
      END DO
      DO istrata=1,nnewstr
         WRITE(6,34) "Non P Control    : ",(smcontrol(istrata,iq),iq=1,nq)
      END DO
      DO istrata=1,nnewstr
         WRITE(6,33) "Controlled         : ",(smrv2(istrata,iq),iq=1,nq)
      END DO

      do iq = 1,nq
         totcheckpi(iq) = 0.0
         do istrata = 1,nnewstr
            checkpi(istrata,iq) = 0.0
            checksm(istrata,iq) = 0.0
         end do
      end do

      DO istrata=1,nnewstr
         DO iq=1,nq
            checkpi(istrata,iq) = prv(istrata,iq,1)*xpicontrol(istrata,iq)
     +         *picontrol(istrata,iq)
            checksm(istrata,iq) = prv(istrata,iq,2)*xsmcontrol(istrata,iq)
     +         *smcontrol(istrata,iq)
         END DO
      END DO

      DO istrata=1,nnewstr
         DO iq=1,nq
            totcheckpi(iq) = totcheckpi(iq) + checkpi(istrata,iq)
         END DO
      END DO

      DO istrata=1,nnewstr
         WRITE(6,'("PI :",4F15.0)') (checkpi(istrata,iq),iq=1,nq)
         DO iq=1,nq
            checkall = checkpi(istrata,iq) + checkall
         END DO
      END DO
      WRITE(6,'("TOT PI :",4F15.0)') (totcheckpi(iq),iq=1,nq)
      DO istrata=1,nnewstr
         WRITE(6,'("SM :",4F15.0)') (checksm(istrata,iq),iq=1,nq)
         DO iq=1,nq
            checkall = checksm(istrata,iq) + checkall
         END DO
      END DO
      print*,' '
      WRITE(6,'("ALL:",F15.0)') checkall

C     Apply Inflation Factors to PERMIT PI Presort and SM

      do iq = 1,nq
         check(iq) = 0.0
         do istrata = 1,nnewstr
            checkpi(istrata,iq) = 0.0
            checksm(istrata,iq) = 0.0
            piraw(istrata,iq) = 0.0
```

```
            smraw(istrata,iq) = 0.0
         end do
      end do

      DO igbn=1,ngbn
         DO iq=1,nq
            DO irpw=1,nrpw
               DO iwt=1,nwt
                  DO istrata=1,nnewstr
                     DO icls=1,ncls
                        IF (irpw.eq.1) THEN
                           piraw(istrata,iq) = piraw(istrata,iq) +
     +                        permit(icls,istrata,iwt,1,irpw,iq,igbn)
                           smraw(istrata,iq) = smraw(istrata,iq) +
     +                        permit(icls,istrata,iwt,2,irpw,iq,igbn)
                        END IF


                        permit(icls,istrata,iwt,1,irpw,iq,igbn)
     +                     = permit(icls,istrata,iwt,1,irpw,iq,igbn)
     +                     * picontrol(istrata,iq)*xpicontrol(istrata,iq) !PI
                        permit(icls,istrata,iwt,2,irpw,iq,igbn)
     +                     = permit(icls,istrata,iwt,2,irpw,iq,igbn)
     +                     * smcontrol(istrata,iq)*xsmcontrol(istrata,iq) !SM


                        IF (irpw.eq.1.and.igbn.eq.3) THEN
                           indrev(icls,1,iq) = indrev(icls,1,iq) +
     +                        permit(icls,istrata,iwt,1,1,iq,igbn)
                           indrev(icls,2,iq) = indrev(icls,2,iq) +
     +                        permit(icls,istrata,iwt,2,1,iq,igbn)
                           totind(iq) = totind(iq) +
     +                        permit(icls,istrata,iwt,1,1,iq,igbn) +
     +                        permit(icls,istrata,iwt,2,1,iq,igbn)
                        END IF
                     END DO
                  END DO
               END DO
            END DO
         END DO
      END DO

      PRINT *, "Total q1 non identical Revenue = ",totind(1)
      PRINT *, "Total q2 non identical Revenue = ",totind(2)
      PRINT *, "Total q3 non identical Revenue = ",totind(3)
      PRINT *, "Total q4 non identical Revenue = ",totind(4)
      print*,' '


      DO iq=1,nq
         DO iind=1,nind
            DO icls=1,ncls
               IF (indrev(icls,1,iq)+
     +            indrev(icls,2,iq)+indrev(icls,3,iq).gt.0) THEN
                  indshare(icls,iind,iq) =
     +               indrev(icls,iind,iq)/(indrev(icls,1,iq)+
     +               indrev(icls,2,iq))
               ELSE
                  indshare(icls,iind,iq) = 0.0
               END IF
            END DO
         END DO
      END DO

C Merge Identical, Nonidentical and Bad Weight Matrices separately

      DO igbn=1,ngbn
         DO iq=1,nq
            DO irpw=1,nrpw
               DO iind=1,nind
                  DO iwt=1,nwt
                     DO istrata=1,nnewstr
                        DO icls=1,ncls
                           IF (igbn.eq.1) THEN
                              id(icls,iwt,iind,irpw,iq) =
     +                           permit(icls,istrata,iwt,
     +                           iind,irpw,iq,igbn) +
     +                           id(icls,iwt,iind,irpw,iq)
                           ELSE IF (igbn.eq.2) THEN
                              bad(icls,iwt,iind,irpw,iq) =
```

```
     +                          permit(icls,istrata,iwt,
     +                          iind,irpw,iq,igbn) +
     +                          bad(icls,iwt,iind,irpw,iq)
                           ELSE IF (igbn.eq.3) THEN
                              ni(icls,iwt,irpw,iq) =
     +                          permit(icls,istrata,iwt,
     +                          iind,irpw,iq,igbn) +
     +                          ni(icls,iwt,irpw,iq)
                           END IF
                        END DO
                     END DO
                  END DO
               END DO
            END DO
         END DO
      END DO

C use only ID GOOD data

      DO iq=1,nq
         DO irpw=1,nrpw
            DO iwt=1,nwt
               DO icls=1,ncls
                  data(icls,iwt,1,irpw,iq) =
     +               data(icls,iwt,1,irpw,iq) +
     +               id(icls,iwt,1,irpw,iq)
                  data(icls,iwt,2,irpw,iq) =
     +               data(icls,iwt,2,irpw,iq) +
     +               id(icls,iwt,2,irpw,iq)
                  data(icls,iwt,3,irpw,iq) =
     +               data(icls,iwt,3,irpw,iq) +
     +               id(icls,iwt,3,irpw,iq)
                  IF (irpw.eq.2) infpcs = infpcs +
     +               id(icls,iwt,2,irpw,iq) +
     +               id(icls,iwt,1,irpw,iq) +
     +               id(icls,iwt,3,irpw,iq)
               END DO
            END DO
         END DO
      END DO



      DO iq=1,nq
         DO iwt = 1, nwt
         DO irpw=1,nrpw
            DO iind=1,nind
               DO icls=1,ncls
                  report(icls,iwt,iind,irpw,iq) = report(icls,iwt,iind,irpw,iq) +
     +               data(icls,iwt,iind,irpw,iq)
                  total(irpw,iind) = data(icls,iwt,iind,irpw,iq) +
     +               total(irpw,iind)
                  inflated(irpw,icls) = inflated(irpw,icls) +
     +               data(icls,iwt,iind,irpw,iq)
               END DO
            END DO
         END DO
         END DO
      END DO

      PRINT *,  "Final  :   Permit Imprint|SM & Automation|Single Piece"
      WRITE(6,'("Revenue: ",3F16.0)') (total(1,iind),iind=1,nind)
      WRITE(6,'("Pieces : ",3F16.0)') (total(2,iind),iind=1,nind)
      WRITE(6,'("Weight : ",3F16.0)') (total(3,iind),iind=1,nind)
      WRITE(6,'("Total Trial Balance PI ", F16.0)') tpitb
      WRITE(6,'("Total SM:",F16.0)') tsm

C     Write Controled RPW

      PRINT *, "Writing Inflated File."

      OPEN(10,FILE='half00x1.csv',RECL=300)
19    FORMAT(I2,",",I2,",",I2,",",17(",",F15.0))

      totalrawrev = 0.0
      allraw = 0.0

C     sum across indicia
```

```
 DO icls = 1, ncls
  DO iind=1,nind
     DO iwt = 1, nwt
        DO irpw = 1, nrpw
           DO iq = 1, nq
              reportx(icls,iwt,irpw,iq) =  reportx(icls,iwt,irpw,iq) +
+                report(icls,iwt,iind,irpw,iq)
           END DO
         END DO
      END DO
   END DO
 END DO

 do irpw = 1,nrpw
 DO iq=1,nq
    DO icls=1,ncls
       WRITE(10,19) iq,irpw,icls,(reportx(icls,iwt,irpw,iq),iwt=1,nwt)
    END DO
 END DO
 end do

 DO icls=1,ncls
    WRITE(6,'(I2,1X,4F16.0)') icls,uninflated(2,icls),inflated(2,icls),
+      uninflated(1,icls),inflated(1,icls)
    totalrawrev = totalrawrev + uninflated(1,icls)
    grand = grand + inflated(2,icls)
    grev = grev + inflated(1,icls)
    allraw = allraw + uninflated(1,icls)
 END DO
 WRITE(6,'(19X,F16.0,1X,2F16.0)') grand,totalrawrev,grev
 WRITE(6,'("All Raw Revenue    ",F16.0)') allraw
 WRITE(6,'("Pcs After Infl.    ",F16.0)') infpcs

 STOP
 END
```

```
      PROGRAM control_shape

C     DESCRIPTION: Control distributions of PERMIT revenue, pieces
C        and weight to RPW revenue, aggregate to augmented RPW category list
C        (vipmap.lst). Estimate weight of bad weight pieces, redistribute
C        non-identical pieces. Write control factors to a file for use in
C        other control and distribution programs.
C
C        This version of program adds PERMIT priority presort PI
C        revenues to PERMIT First-Class presort PI revenues before
C        computing an own office control.
C
C     CREATED BY:       Bill Humphries
C     DATE:             Fri Sep 30 09:53:27 CDT 1994
C
C     LAST MODIFIED BY: ta
C     DATE:             01-08-96; revise controls to aggregate PERMIT and
C           BRAVIS data instead of seperate controls for each as were done
C           for fy95.
C                       01-may-96; initialize variables to zero that previously
C                                  were not.
C                       02-may-96; update the redist subroutine so that all
C                                  cases should be solvable.
C                       18-sep-96; update for reclass
C                       30-oct-96; update for fy97, this program originated
C                                  from newcontrol_RC.f in 96.
C                       21-feb-97; update NI dist subroutine for 'reverse'
C                                  addition.
C                       18-aug-97; revise for Shaw's revenue control
C                       14-oct-97; add PI control for q4
C                       15-jan-98; update for FY98
C                       10-jul-98; add section for auto parcel wt calc
C                       25-jan-99; update for FY99
C                       07-apr-99; set q2 PERMIT IMPRINT controls to 1.0 as per RPW.
C                       Also edit for ratetable for Q2
C                       21-apr-99; create with wt increments using newcontrol.f
C                       01-feb-00; update for FY00
C                       21-nov-00; update for new compiler


      IMPLICIT NONE

      INTEGER*4 nwt,nstrata,nrpw,nid,ngb,ngbn
      PARAMETER (nwt=13,nstrata=20,nrpw=3,nid=2,ngb=2,ngbn=3)
      INTEGER*4 nap,nq,ncls,nind,nnewstr,ntype,size,size1,size3
      parameter (nap=13,nq=4)                !!! update - also in subroutine
      PARAMETER (ncls=34,nind=2,nnewstr=4,ntype=3)

      INTEGER*4 iwt,istrata,irpw,iid,igb,iap,iq,icls,iind,istratax
      INTEGER*4 i,j,k,ier,indicia,isize,igbn

      REAL*8    rev,pcs,wt,line(nq),rateline(nwt),rateline00(nwt)
      INTEGER*4 in


      REAL*8 permit(ncls,nnewstr,nwt,nind,nrpw,nq,ngbn)
      REAL*8 id2(ncls,nwt,nrpw,nq), ni(ncls,nwt,nrpw,nq)
      REAL*8 avewt(ncls,nwt), totid(ncls,nwt)
      REAL*8 totwt(ncls,nwt), data(ncls,nwt,nind,nrpw,nq)
      REAL*8 report(ncls,nwt,nind,nrpw,nq), total(nrpw,nind)
      REAL*8 reportx(ncls,nwt,nrpw,nq)
      REAL*8 ppriority(nnewstr,nq) ,midr(ncls,nq)
      REAL*8 midp(ncls,nq), mnir(ncls,nq)
      REAL*8 mnip(ncls,nq), distribution(ncls,nwt)
      REAL*8 indshare(ncls,nind,nq),indrev(ncls,nind,nq)
      REAL*8 ratetable(ncls,nwt), ratetbl00(ncls,nwt)

      COMMON /rate/ratetable,midp,mnip,midr,distribution

      CHARACTER*2 cap(nap)/'01','02','03','04','05','06','07','08','09','10','11','12','13'/    ! update
      INTEGER*4 apq(nap)/1,1,1,2,2,2,3,3,3,4,4,4,4/     ! update
      INTEGER*4 stratamap(20)/1,1,1,1,1,1,1,1,1,1,1,1,1,2,2,3,3,4,4/

      REAL*8 id(ncls,nwt,nind,nrpw,4)      ! special case
      REAL*8 bad(ncls,nwt,nind,nrpw,4)

      REAL*8 rpw(nq),smrpw(nq),rrpw(nq)
      REAL*8 smrv(nnewstr,nq)
      REAL*8 psmrv(nnewstr,nq)
      REAL*8 pirv(nnewstr,nq)
```

```
      REAL*8 ppirv(nnewstr,nq)
      REAL*8 trv(nnewstr,nq)
      REAL*8 ptrv(nnewstr,nq)
      REAL*8 pitb(nnewstr,nq)
      real*8 totpitb/0.0/
      REAL*8 ppitb(nnewstr,nq)
      REAL*8 prv(nnewstr,nq,nind)
      real*8 prvx(20,4,4)               ! temp
      real*8 prvx1/0.0/                           ! temp
      REAL*8 pirv1(nnewstr,nq),pirv2(nnewstr,nq),pirv3(nq)
      REAL*8 smrv1(nnewstr,nq),smrv2(nnewstr,nq),smrv3(nq),smrv4(nq)
      REAL*8 rv1(nq)
      REAL*8 rv2(nq)
      REAL*8 raw(nrpw)
      REAL*8 picontrol(nnewstr,nq)
      REAL*8 xpicontrol(nnewstr,nq)
      REAL*8 smcontrol(nnewstr,nq)
      REAL*8 xsmcontrol(nnewstr,nq)
      REAL*8 ppisp(nnewstr,nq)
      REAL*8 psmsp(nnewstr,nq)
      REAL*8 sp(nnewstr,nq)
      REAL*8 rcontrol(nq)
      REAL*8 uninflated(nrpw,ncls)
      REAL*8 inflated(nrpw,ncls)
      REAL*8 finalrev(ncls)
      REAL*8 finalwt(ncls)
      REAL*8 check(nq)
      real*8 grand/0.0/,grev/0.0/,infpcs/0.0/,checkall/0.0/
      real*8 totind(4)/4*0.0/
      REAL*8 prp,rprp,indtot
      REAL*8 checkpi(nnewstr,nq),checksm(nnewstr,nq),allraw,tpipri
      REAL*8 piraw(nnewstr,nq),smraw(nnewstr,nq),tin,tpin,totalrawrev
      real*8 tpitb/0.0/,tsm/0.0/
      real*8 totnctbq1/0.0/,totnctbq2/0.0/,totnctbq3/0.0/,totnctbq4/0.0/
      real*8 revadd(4,4)
      real*8 tpipriq(4)
      real*8 totcheckpi(4)
      real*8 tot4
      real*8 totapwt(4)    ! 4 qtrs


C     Zero-out PERMIT  matricies

      tin = 0.0
      tpin = 0.0
      totalrawrev = 0.0

      do irpw = 1,nrpw
         raw(irpw) = 0.0
         do iind = 1,nind
            total(irpw,iind) = 0.0
            do icls = 1,ncls
               do iwt = 1,nwt
                  do iq = 1,nq
                     id(icls,iwt,iind,irpw,iq) = 0.0
                     bad(icls,iwt,iind,irpw,iq) = 0.0
                     data(icls,iwt,iind,irpw,iq) = 0.0
                     report(icls,iwt,iind,irpw,iq) = 0.0
                     do igbn = 1,ngbn
                        do istrata = 1,nnewstr
                           permit(icls,istrata,iwt,iind,irpw,iq,igbn) = 0.0
                        end do
                     end do
                  end do
               end do
            end do
         end do
      end do

      do istrata = 1,nnewstr
         do iq = 1,nq
            pitb(istrata,iq) = 0.0
            ppitb(istrata,iq) = 0.0
            ppirv(istrata,iq) = 0.0
            psmrv(istrata,iq) = 0.0
            ptrv(istrata,iq) = 0.0
            smrv(istrata,iq) = 0.0
            trv(istrata,iq) = 0.0
            pirv(istrata,iq) = 0.0
            picontrol(istrata,iq) = 0.0
```

```
            xpicontrol(istrata,iq) = 0.0
            smcontrol(istrata,iq) = 0.0
            xsmcontrol(istrata,iq) = 0.0
            ppisp(istrata,iq) = 0.0
            psmsp(istrata,iq) = 0.0
            sp(istrata,iq) = 0.0
            do iind = 1,nind
               prv(istrata,iq,iind) = 0.0
            end do
         end do
      end do

      do icls = 1,ncls
         do iwt = 1,nwt
            distribution(icls,iwt) = 0.0
            avewt(icls,iwt) = 0.0
            totid(icls,iwt) = 0.0
            totwt(icls,iwt) = 0.0
            do irpw = 1,nrpw
               do iq = 1,nq
                  id2(icls,iwt,irpw,iq) = 0.0
                  ni(icls,iwt,irpw,iq) = 0.0
                  reportx(icls,iwt,irpw,iq) = 0.0
               end do
            end do
         end do
      end do

      do iq = 1,nq
         smrv3(iq) = 0.0
         smrv4(iq) = 0.0
         do icls = 1,ncls
            midr(icls,iq) = 0.0
            midp(icls,iq) = 0.0
            mnir(icls,iq) = 0.0
            mnip(icls,iq) = 0.0
            do iind = 1,nind
               indshare(icls,iind,iq) = 0.0
               indrev(icls,iind,iq) = 0.0
            end do
         end do
      end do

      do irpw = 1,nrpw
         do icls = 1, ncls
            uninflated(irpw,icls) = 0.0
            inflated(irpw,icls) = 0.0
         end do
      end do


      PRINT *, "Zeroed PERMIT  Matrices."

C     Read in nctb revenue for all offices-including nonpermit

      OPEN(10,FILE='trevenue.dat',readonly) !SM incl fitted
      OPEN(20,FILE='prevenue.dat',readonly) !SM for PERMIT
C                  offices from finrev.1st with missing data added

      OPEN(40,FILE='tnctbstr.dat',readonly) ! NCTB PI presort
      OPEN(50,FILE='pnctbstr.dat',readonly) ! PERMIT offices

32    FORMAT(4F15.0)
      DO istrata=1,nstrata
         READ(10,32) line
         DO iq=1,nq
            trv(stratamap(istrata),iq) =
     +         trv(stratamap(istrata),iq) + line(iq)
         tin = tin + line(iq)
      END DO
      READ(10,32) line
      DO iq=1,nq
         pirv(stratamap(istrata),iq) =
     +         pirv(stratamap(istrata),iq) + line(iq)
      END DO
      READ(10,32) line
      DO iq=1,nq
         smrv(stratamap(istrata),iq) =
     +         smrv(stratamap(istrata),iq) + line(iq)
         tsm = tsm + line(iq)
```

```
         END DO

         READ(20,32) line
         DO iq=1,nq
            ptrv(stratamap(istrata),iq) =
     +         ptrv(stratamap(istrata),iq) + line(iq)
            tpin = tpin + line(iq)
         END DO
         READ(20,32) line
         DO iq=1,nq
            ppirv(stratamap(istrata),iq) =
     +         ppirv(stratamap(istrata),iq) + line(iq)
         END DO
         READ(20,32) line
         DO iq=1,nq
            psmrv(stratamap(istrata),iq) =
     +         psmrv(stratamap(istrata),iq) + line(iq)
         END DO

       END DO
       DO istrata=1,nstrata
         READ(40,32) line
         DO iq=1,nq
            pitb(stratamap(istrata),iq) =
     +         pitb(stratamap(istrata),iq) + line(iq)
            totpitb = totpitb + line(iq)
         END DO
         READ(50,32) line
         DO iq=1,nq
            ppitb(stratamap(istrata),iq) =
     +         ppitb(stratamap(istrata),iq) + line(iq)
         END DO
       END DO


33     FORMAT(A20,4F15.0)
34     FORMAT(A20,4F15.10)

       WRITE (6,33) "All,    PMT  :    ", tin, tpin
       print*,' '
       print*,'                          Q1              Q2              Q3          Q4    '

       DO istrata=1,nnewstr
         WRITE(6,33) "Total Revenue:  ",(trv(istrata,iq),iq=1,nq)
       END DO
       DO istrata=1,nnewstr
         WRITE(6,33) "PI Presort    :  ",(pirv(istrata,iq),iq=1,nq)
       END DO
       DO istrata=1,nnewstr
         WRITE(6,33) "SM :             ",(smrv(istrata,iq),iq=1,nq)
       END DO
       DO istrata=1,nnewstr
         WRITE(6,33) "PERMIT Revenue: ",(ptrv(istrata,iq),iq=1,nq)
       END DO
       DO istrata=1,nnewstr
         WRITE(6,33) "PI Presort    :  ",(ppirv(istrata,iq),iq=1,nq)
       END DO
       DO istrata=1,nnewstr
         WRITE(6,33) "SM :             ",(psmrv(istrata,iq),iq=1,nq)
       END DO

       OPEN(8,FILE='newrates.00',readonly)

       DO i=1,ncls
         READ(8,'(2X,13(F6.3))') rateline00
         DO iwt=1, nwt
            ratetable(i,iwt) = rateline00(iwt)
         END DO
       END DO

C      READ PERMIT  Arrays by Quarter

C      Format for reading revenue, pieces and weight arrays:
11     FORMAT(I3,I2,1x,I2,3I1,3F20.10)
31     FORMAT(3X,3F15.0)
       DO iap=1,nap
         iq = apq(iap)
         OPEN(10,FILE='perrolln.1st.'//cap(iap),readonly)
         PRINT *, "Opened perrolln.1st.",cap(iap)
         ier = 0
```

```
      in = 0
      DO WHILE (ier.eq.0)
         READ(10,11,IOSTAT=ier,END=12) icls,iwt,istrata,iind,
     +      iid,igb,rev,pcs,wt
C         istratax = istrata              ! temp
         istrata = stratamap(istrata)
         IF (((icls.ge.8.and.icls.le.14).or.icls.eq.18.or.icls.eq.19.or.
     +      icls.eq.20.or.icls.eq.27.or.icls.eq.28).and.iwt.gt.1) THEN
            PRINT *, "PERMIT Error:"
            PRINT *, "Heavy Nonstandard icls = ",icls," iwt = ",iwt,
     +         " iap = ",iap
         END IF
         IF (iid.eq.1.and.igb.eq.1) THEN
            igbn = 1
         ELSE IF (iid.eq.1.and.igb.eq.0) THEN
            igbn = 2
         ELSE
            igbn = 3
         END IF
         prv(istrata,iq,iind) = prv(istrata,iq,iind) + rev
C         prvx(istratax,iq,iind) = prvx(istratax,iq,iind) + rev   ! temp
         permit(icls,istrata,iwt,iind,1,iq,igbn) = rev +
     +      permit(icls,istrata,iwt,iind,1,iq,igbn)
         permit(icls,istrata,iwt,iind,2,iq,igbn) = pcs +
     +      permit(icls,istrata,iwt,iind,2,iq,igbn)
         permit(icls,istrata,iwt,iind,3,iq,igbn) = wt +
     +      permit(icls,istrata,iwt,iind,3,iq,igbn)
         raw(1) = raw(1) + rev
         raw(2) = raw(2) + pcs
         raw(3) = raw(3) + wt
         uninflated(1,icls) = uninflated(1,icls) + rev
         uninflated(2,icls) = uninflated(2,icls) + pcs
         IF (igb.eq.1) uninflated(3,icls) = uninflated(3,icls) + wt
         in = in + 1
      END DO
12    PRINT *, "Read ",in," records from perrolln.1st.",cap(iap),"."
      PRINT *, "ier = ",ier
      WRITE (6,31) (raw(irpw),irpw=1,3)
      DO irpw = 1,nrpw
         raw(irpw) = 0.0
      END DO
      CLOSE(10)
   END DO



C     OWN CONTROL
   DO istrata=1,nnewstr
      DO iq=1,nq
         IF (prv(istrata,iq,1).gt.0.0)  THEN
           xpicontrol(istrata,iq) = ppitb(istrata,iq)/prv(istrata,iq,1)
         ELSE
            xpicontrol(istrata,iq) = 0.0
         END IF
         IF (prv(istrata,iq,2).gt.0.0) THEN
           xsmcontrol(istrata,iq) = psmrv(istrata,iq) / prv(istrata,iq,2)
         ELSE
            xsmcontrol(istrata,iq) = 0.0
         END IF

         pirv1(istrata,iq)  = prv(istrata,iq,1)*xpicontrol(istrata,iq)
         smrv1(istrata,iq) =  prv(istrata,iq,2)*xsmcontrol(istrata,iq)

      END DO
   END DO

   print*,' '

   DO istrata=1,nnewstr
      WRITE(6,33) "Uncontroled PPI  : ",(prv(istrata,iq,1),iq=1,nq)
   END DO
   DO istrata=1,nnewstr
      WRITE(6,33) "PPI Trial Balance: ",(ppitb(istrata,iq),iq=1,nq)
   END DO
   DO istrata=1,nnewstr
      WRITE(6,34) "PPI Control      : ",(xpicontrol(istrata,iq),iq=1,nq)
   END DO
   DO istrata=1,nnewstr
      WRITE(6,33) "Controled PPI    : ",(prv(istrata,iq,1)*
     +   xpicontrol(istrata,iq),iq=1,nq)
```

```
      END DO

      DO istrata=1,nnewstr
         WRITE(6,33) "Uncontroled PSM  : ",(prv(istrata,iq,2),iq=1,nq)
      END DO
      DO istrata=1,nnewstr
         WRITE(6,34) "PSM Control      : ",(xsmcontrol(istrata,iq),iq=1,nq)
      END DO
      DO istrata=1,nnewstr
         WRITE(6,33) "Controled PSM    : ",(prv(istrata,iq,2)*
     +      xsmcontrol(istrata,iq),iq=1,nq)
      END DO


C        SECOND STAGE CONTROL
      DO istrata=1,nnewstr
         DO iq=1,nq
            picontrol(istrata,iq) = pitb(istrata,iq)/pirv1(istrata,iq)

            pirv2(istrata,iq) = prv(istrata,iq,1)
     +         *xpicontrol(istrata,iq) * picontrol(istrata,iq)
            smcontrol(istrata,iq) = smrv(istrata,iq)/smrv1(istrata,iq)
            smrv2(istrata,iq) = prv(istrata,iq,2)*xsmcontrol(istrata,iq)
     +         *smcontrol(istrata,iq)
            smrv3(iq) = smrv3(iq) + smrv2(istrata,iq)
         END DO
      END DO

      print*,' '
      DO istrata=1,nnewstr
         WRITE(6,33) "Own Office PI    : ",(pirv1(istrata,iq),iq=1,nq)
      END DO
      DO istrata=1,nnewstr
         WRITE(6,33) "PI Trial Balance :  ",(pitb(istrata,iq),iq=1,nq)
      END DO
      DO istrata=1,nnewstr
         WRITE(6,34) "Non P Control    : ",(picontrol(istrata,iq),iq=1,nq)
      END DO
      DO istrata=1,nnewstr
         WRITE(6,33) "Controlled       : ",(pirv2(istrata,iq),iq=1,nq)
      END DO

      DO istrata=1,nnewstr
         WRITE(6,33) "Own Office SM    : ",(smrv1(istrata,iq),iq=1,nq)
      END DO
      DO istrata=1,nnewstr
         WRITE(6,33) "Total SM         : ",(smrv(istrata,iq),iq=1,nq)
      END DO
      DO istrata=1,nnewstr
         WRITE(6,34) "Non P Control    : ",(smcontrol(istrata,iq),iq=1,nq)
      END DO
      DO istrata=1,nnewstr
         WRITE(6,33) "Controlled       : ",(smrv2(istrata,iq),iq=1,nq)
      END DO

C     Write Control Factors
      OPEN(10,FILE='nfactors.1st.q4')
      WRITE(10,'("PERMIT PI")')
      DO istrata=1,nnewstr
         WRITE(10,'(4F20.15)') (xpicontrol(istrata,iq),iq=1,nq)
      END DO
      WRITE(10,'("PERMIT SM")')
      DO istrata=1,nnewstr
         WRITE(10,'(4F20.15)') (xsmcontrol(istrata,iq),iq=1,nq)
      END DO
      WRITE(10,'("GLOBAL PI")')
      DO istrata=1,nnewstr
         WRITE(10,'(4F20.15)') (picontrol(istrata,iq),iq=1,nq)
      END DO
      WRITE(10,'("GLOBAL SM")')
      DO istrata=1,nnewstr
         WRITE(10,'(4F20.15)') (smcontrol(istrata,iq),iq=1,nq)
      END DO

      do iq = 1,nq
         totcheckpi(iq) = 0.0
         do istrata = 1,nnewstr
            checkpi(istrata,iq) = 0.0
            checksm(istrata,iq) = 0.0
         end do
```

```
      end do

      DO istrata=1,nnewstr
         DO iq=1,nq
            checkpi(istrata,iq) = prv(istrata,iq,1)*xpicontrol(istrata,iq)
     +         *picontrol(istrata,iq)
            checksm(istrata,iq) = prv(istrata,iq,2)*xsmcontrol(istrata,iq)
     +         *smcontrol(istrata,iq)
         END DO
      END DO

      DO istrata=1,nnewstr
         DO iq=1,nq
            totcheckpi(iq) = totcheckpi(iq) + checkpi(istrata,iq)
         END DO
      END DO

      DO istrata=1,nnewstr
         WRITE(6,'("PI :",4F15.0)') (checkpi(istrata,iq),iq=1,nq)
         DO iq=1,nq
            checkall = checkpi(istrata,iq) + checkall
         END DO
      END DO
      WRITE(6,'("TOT PI :",4F15.0)') (totcheckpi(iq),iq=1,nq)
      DO istrata=1,nnewstr
         WRITE(6,'("SM :",4F15.0)') (checksm(istrata,iq),iq=1,nq)
         DO iq=1,nq
            checkall = checksm(istrata,iq) + checkall
         END DO
      END DO
      print*,' '
      WRITE(6,'("ALL:",F15.0)') checkall

C     Apply Inflation Factors to PERMIT PI Presort and SM

      do iq = 1,nq
         check(iq) = 0.0
         do istrata = 1,nnewstr
            checkpi(istrata,iq) = 0.0
            checksm(istrata,iq) = 0.0
            piraw(istrata,iq) = 0.0
            smraw(istrata,iq) = 0.0
         end do
      end do


      DO igbn=1,ngbn
         DO iq=1,nq
            DO irpw=1,nrpw
               DO iwt=1,nwt
                  DO istrata=1,nnewstr
                     DO icls=1,ncls
                        IF (irpw.eq.1) THEN
                           piraw(istrata,iq) = piraw(istrata,iq) +
     +                        permit(icls,istrata,iwt,1,irpw,iq,igbn)
                           smraw(istrata,iq) = smraw(istrata,iq) +
     +                        permit(icls,istrata,iwt,2,irpw,iq,igbn)
                        END IF


                        permit(icls,istrata,iwt,1,irpw,iq,igbn)
     +                     = permit(icls,istrata,iwt,1,irpw,iq,igbn)
     +                     * picontrol(istrata,iq)*xpicontrol(istrata,iq) !PI
                        permit(icls,istrata,iwt,2,irpw,iq,igbn)
     +                     = permit(icls,istrata,iwt,2,irpw,iq,igbn)
     +                     * smcontrol(istrata,iq)*xsmcontrol(istrata,iq) !SM


                        IF (irpw.eq.1.and.igbn.eq.3) THEN
                           indrev(icls,1,iq) = indrev(icls,1,iq) +
     +                        permit(icls,istrata,iwt,1,1,iq,igbn)
                           indrev(icls,2,iq) = indrev(icls,2,iq) +
     +                        permit(icls,istrata,iwt,2,1,iq,igbn)
                           totind(iq) = totind(iq) +
     +                        permit(icls,istrata,iwt,1,1,iq,igbn) +
     +                        permit(icls,istrata,iwt,2,1,iq,igbn)
                        END IF
                     END DO
                  END DO
               END DO
            END DO
```

```
            END DO
         END DO
      END DO

      PRINT *, "Total q1 non identical Revenue = ",totind(1)
      PRINT *, "Total q2 non identical Revenue = ",totind(2)
      PRINT *, "Total q3 non identical Revenue = ",totind(3)
      PRINT *, "Total q4 non identical Revenue = ",totind(4)
      print*,' '


      DO iq=1,nq
         DO iind=1,nind
            DO icls=1,ncls
               IF (indrev(icls,1,iq)+indrev(icls,2,iq).gt.0) THEN
                  indshare(icls,iind,iq) =
     +               indrev(icls,iind,iq)/(indrev(icls,1,iq)+
     +               indrev(icls,2,iq))
               ELSE
                  indshare(icls,iind,iq) = 0.0
               END IF
            END DO
         END DO
      END DO

C Merge Identical, Nonidentical and Bad Weight Matrices separately

      DO igbn=1,ngbn
         DO iq=1,nq
            DO irpw=1,nrpw
               DO iind=1,nind
                  DO iwt=1,nwt
                     DO istrata=1,nnewstr
                        DO icls=1,ncls
                           IF (igbn.eq.1) THEN
                              id(icls,iwt,iind,irpw,iq) =
     +                           permit(icls,istrata,iwt,
     +                           iind,irpw,iq,igbn) +
     +                           id(icls,iwt,iind,irpw,iq)
                           ELSE IF (igbn.eq.2) THEN
                              bad(icls,iwt,iind,irpw,iq) =
     +                           permit(icls,istrata,iwt,
     +                           iind,irpw,iq,igbn) +
     +                           bad(icls,iwt,iind,irpw,iq)
                           ELSE IF (igbn.eq.3) THEN
                              ni(icls,iwt,irpw,iq) =
     +                           permit(icls,istrata,iwt,
     +                           iind,irpw,iq,igbn) +
     +                           ni(icls,iwt,irpw,iq)
                           END IF
                        END DO
                     END DO
                  END DO
               END DO
            END DO
         END DO
      END DO

C     Compute Average Weight of Identical Pieces

      DO iq=1,nq
         DO iind=1,nind
            DO iwt=1,nwt
               DO icls=1,ncls
                  totid(icls,iwt) = totid(icls,iwt) +
     +               id(icls,iwt,iind,2,iq)
                  totwt(icls,iwt) = totwt(icls,iwt) +
     +               id(icls,iwt,iind,3,iq)
               END DO
            END DO
         END DO
      END DO

      DO iwt=1,nwt
         DO icls=1,ncls
            IF (totid(icls,iwt).gt.0) THEN
               avewt(icls,iwt) =
     +            totwt(icls,iwt)/totid(icls,iwt)
            END IF
         END DO
```

```
      END DO

C     Apply Average Weight to Bad Weight Pieces

      DO iq=1,nq
         DO iind=1,nind
            DO iwt=1,nwt
               DO icls=1,ncls
                  bad(icls,iwt,iind,3,iq) =
     +               bad(icls,iwt,iind,2,iq) *
     +               avewt(icls,iwt)
               END DO
            END DO
         END DO
      END DO

C     Add identical to bad
      DO iq=1,nq
         DO irpw=1,nrpw
            DO iwt=1,nwt
               DO icls=1,ncls
                  DO iind=1,nind
                     id2(icls,iwt,irpw,iq) =
     +                  id2(icls,iwt,irpw,iq) +
     +                  id(icls,iwt,iind,irpw,iq) +
     +                  bad(icls,iwt,iind,irpw,iq)
                     IF (irpw.eq.1)
     +                  midr(icls,iq) = midr(icls,iq) +
     +                  id(icls,iwt,iind,1,iq) +
     +                  bad(icls,iwt,iind,1,iq)
                     IF (irpw.eq.2)
     +                  midp(icls,iq) = midp(icls,iq) +
     +                  id(icls,iwt,iind,2,iq) +
     +                  bad(icls,iwt,iind,2,iq)
                  END DO
               END DO
            END DO
         END DO
      END DO

      DO iwt=1,nwt
         DO icls=1,ncls
            DO iq=1,nq
               mnir(icls,iq) = mnir(icls,iq) +
     +            ni(icls,iwt,1,iq)
               mnip(icls,iq) = mnip(icls,iq) +
     +            ni(icls,iwt,2,iq)
            END DO
         END DO
      END DO

      DO iq=1,nq
C        ratetable = ratetbl00
         CALL redist(id2,ni,mnir,iq)
      END DO

      DO iq=1,nq
         DO iwt=1,nwt
            DO icls=1,ncls
               ni(icls,iwt,3,iq) =
     +            ni(icls,iwt,2,iq)*avewt(icls,iwt)
            END DO
         END DO
      END DO


      DO iq=1,nq
         DO irpw=1,nrpw
            DO iwt=1,nwt
               DO icls=1,ncls
                  data(icls,iwt,1,irpw,iq) =
     +               data(icls,iwt,1,irpw,iq) +
     +               id(icls,iwt,1,irpw,iq) +
     +               bad(icls,iwt,1,irpw,iq) +
     +               ni(icls,iwt,irpw,iq)*indshare(icls,1,iq)
                  data(icls,iwt,2,irpw,iq) =
     +               data(icls,iwt,2,irpw,iq) +
     +               id(icls,iwt,2,irpw,iq) +
     +               bad(icls,iwt,2,irpw,iq) +
     +               ni(icls,iwt,irpw,iq)*indshare(icls,2,iq)
```

```
                IF (irpw.eq.2) infpcs = infpcs +
     +             id(icls,iwt,2,irpw,iq) +
     +             bad(icls,iwt,2,irpw,iq) +
     +             ni(icls,iwt,irpw,iq)*indshare(icls,2,iq)+
     +             id(icls,iwt,1,irpw,iq) +
     +             bad(icls,iwt,1,irpw,iq) +
     +             ni(icls,iwt,irpw,iq)*indshare(icls,1,iq)
              END DO
           END DO
        END DO
      END DO

      DO iq=1,nq
         DO iwt = 1, nwt
         DO irpw=1,nrpw
            DO iind=1,nind
               DO icls=1,ncls
               report(icls,iwt,iind,irpw,iq) = report(icls,iwt,iind,irpw,iq) +
     +            data(icls,iwt,iind,irpw,iq)
               total(irpw,iind) = data(icls,iwt,iind,irpw,iq) +
     +            total(irpw,iind)
               inflated(irpw,icls) = inflated(irpw,icls) +
     +            data(icls,iwt,iind,irpw,iq)
              END DO
           END DO
        END DO
        END DO
      END DO

      PRINT *,  "Final   :   Permit Imprint|SM & Automation|Single Piece"
      WRITE(6,'("Revenue: ",3F16.0)') (total(1,iind),iind=1,nind)
      WRITE(6,'("Pieces : ",3F16.0)') (total(2,iind),iind=1,nind)
      WRITE(6,'("Weight : ",3F16.0)') (total(3,iind),iind=1,nind)
      WRITE(6,'("Total Trial Balance PI ", F16.0)') totpitb
      WRITE(6,'("Total SM:",F16.0)') tsm

C     Write Controled RPW

      PRINT *, "Writing Inflated File."

      OPEN(10,FILE='shape00.csv.q4',RECL=300)
19    FORMAT(I2,",",I2,",",I2,",",13(",",F15.0))

      totalrawrev = 0.0
      allraw = 0.0

C     sum across indicia

      DO icls = 1, ncls
       DO iind=1,nind
          DO iwt = 1, nwt
             DO irpw = 1, nrpw
                DO iq = 1, nq
            reportx(icls,iwt,irpw,iq) =  reportx(icls,iwt,irpw,iq) +
     +         report(icls,iwt,iind,irpw,iq)
                END DO
              END DO
           END DO
        END DO
      END DO

      do irpw = 1,nrpw
      DO iq=1,nq
         DO icls=1,ncls
            WRITE(10,19) iq,irpw,icls,(reportx(icls,iwt,irpw,iq),iwt=1,nwt)
         END DO
      END DO
      end do

      DO icls=1,ncls
         WRITE(6,'(I2,1X,4F16.0)') icls,uninflated(2,icls),inflated(2,icls),
     +      uninflated(1,icls),inflated(1,icls)
         totalrawrev = totalrawrev + uninflated(1,icls)
         grand = grand + inflated(2,icls)
         grev = grev + inflated(1,icls)
         allraw = allraw + uninflated(1,icls)
      END DO
      WRITE(6,'(19X,F16.0,1X,2F16.0)') grand,totalrawrev,grev
      WRITE(6,'("All Raw Revenue    ",F16.0)') allraw
      WRITE(6,'("Pcs After Infl.    ",F16.0)') infpcs
```

```
      STOP
      END

C-----------------------------------------------------------------------
      SUBROUTINE redist(id2,ni,mnir,iq)

      INTEGER*4 ncls,nwt,nrpw,nind,nq
      PARAMETER (ncls=34,nwt=13,nrpw=3,nind=3,nq=4)    ! update nq
      INTEGER*4 icls,iwt,irpw,imax,imaxlast,iq,iind

      REAL*8 id2(ncls,nwt,nrpw,nq),ni(ncls,nwt,nrpw,nq)
      REAL*8 pmax,idp,ckr,nip,nir,x,share(nwt),ara,arb,pa,pb,rip,ckp,idr
      REAL*8 ratetable(ncls,nwt),midp(ncls,nq)
      REAL*8 mnip(ncls,nq),mnir(ncls,nq)
      REAL*8 midr(ncls,nq),brr(ncls)
      REAL*8 anip(ncls),anir(ncls)
      REAL*8 distribution(ncls,nwt)
      real*8 nirevpc

      logical flagx, reverse

      COMMON /rate/ratetable,midp,mnip,midr,distribution

      PRINT *, "Entering Redistribution Subroutine."

      reverse = .false.
      do icls = 1,ncls
         anip(icls) = 0.0
         anir(icls) = 0.0
      end do

      DO icls=1, ncls
         flagx = .TRUE.
         pmax = 0.0
         idp = midp(icls,iq)
         idr = midr(icls,iq)
         nip = mnip(icls,iq)
         nir = mnir(icls,iq)
         imax = 1
         imaxlast = 1
         nirevpc = nir/nip

         DO iwt=1,nwt
            IF (id2(icls,iwt,2,iq).gt.pmax) THEN
               pmax = id2(icls,iwt,2,iq)
               imax = iwt
               imaxlast = iwt
            END IF
         END DO
         IF (imax.gt.0) rip = idp - pmax
         PRINT *,"iq = ",iq
         PRINT *,"icls = ",icls
         PRINT *,"nip = ",nip," idp = ",idp," pmax = ",pmax
         PRINT *,"imax = ",imax
         PRINT *,"nirevpc = ",nirevpc

         IF ((nip.gt.0.0).and.(rip.gt.0.0)) THEN

            DO WHILE (flagx)
               ara = 0.0
               arb = 0.0
               pa = 0.0
               pb = 0.0

               print *,"new imax = ", imax
               DO iwt=1,nwt
                  IF (iwt.le.imax) THEN
                     pa = pa + id2(icls,iwt,2,iq)
                  ELSE
                     pb = pb + id2(icls,iwt,2,iq)
                  END IF
               END DO
               print *,"pa = ",pa,"   pb = ",pb
               DO iwt=1,nwt
                  IF (iwt.le.imax) THEN
                     ara = ara + (id2(icls,iwt,2,iq)/pa)*
     +                  ratetable(icls,iwt)
                  ELSE
                     IF (pb.gt.0) THEN
```

```
                arb = arb +
+                  (id2(icls,iwt,2,iq)/pb)*
+                  ratetable(icls,iwt)
             ELSE
                arb = 0.0
             END IF
          END IF
       END DO
       print *,"ara = ",ara,"   arb = ",arb

       if ((nirevpc.ge.ara).and.(nirevpc.le.arb).and.(imax.lt.12)) then
          flagx = .FALSE.
       else if ((nirevpc.lt.ara).and.(imax.lt.12)) then
          imax = imax - 1
       else if ((nirevpc.gt.arb).and.(imax.lt.12)) then
          imax = imax + 1
       else
          print*,'redist condition error ', icls
          flagx = .FALSE.
          reverse = .TRUE.
       end if

    END DO


    IF (reverse) then
       flagx = .TRUE.
       imax = imaxlast


       DO WHILE (flagx)
          ara = 0.0
          arb = 0.0
          pa = 0.0
          pb = 0.0

          print *,"new imax = ", imax
          DO iwt=1,nwt
             IF (iwt.le.imax) THEN
                pa = pa + id2(icls,iwt,2,iq)
             ELSE
                pb = pb + id2(icls,iwt,2,iq)
             END IF
          END DO
          print *,"pa = ",pa,"   pb = ",pb
          DO iwt=1,nwt
             IF (iwt.le.imax) THEN
                ara = ara + (id2(icls,iwt,2,iq)/pa)*
+                  ratetable(icls,iwt)
             ELSE
                IF (pb.gt.0) THEN
                   arb = arb +
+                     (id2(icls,iwt,2,iq)/pb)*
+                     ratetable(icls,iwt)
                ELSE
                   arb = 0.0
                END IF
             END IF
          END DO
          print *,"ara = ",ara,"   arb = ",arb

          if ((nirevpc.ge.ara).and.(nirevpc.le.arb).and.(imax.lt.12)) then
             flagx = .FALSE.
          else if ((nirevpc.gt.arb).and.(imax.lt.12)) then ! reverse
             imax = imax + 1
          else if ((nirevpc.lt.ara).and.(imax.lt.12)) then
             imax = imax - 1
          else
             print*,'redist condition error on reverse', icls
             flagx = .FALSE.
          end if

       END DO


    END IF


    x = (nir - nip*arb)/(ara - arb) ! side 'a' pieces
```

```
      IF (x.gt.0.and.x.lt.nip) THEN
         PRINT *, "Solution Found."
         PRINT *, "x       = ",x
         PRINT *, "imax    = ",imax
         PRINT *, "icls    = ",icls
         PRINT *, "nip     = ",nip
         PRINT *, "nir     = ",nir
         PRINT *, "ara     = ",ara
         PRINT *, "arb     = ",arb
         PRINT *, "pa      = ",pa
         PRINT *, "pb      = ",pb
         ckr = 0.0
         ckp = 0.0
         DO iwt=1,nwt
            IF (iwt.le.imax) THEN
               ni(icls,iwt,2,iq) =
     +            x*id2(icls,iwt,2,iq)/pa
               ni(icls,iwt,1,iq) =
     +            ni(icls,iwt,2,iq) *
     +            ratetable(icls,iwt)
               ckr = ckr + ni(icls,iwt,1,iq)
               ckp = ckp + ni(icls,iwt,2,iq)
            ELSE IF (pb.gt.0.0) THEN
               ni(icls,iwt,2,iq) =
     +            (nip-x)*id2(icls,iwt,2,iq)/pb
               ni(icls,iwt,1,iq) =
     +            ni(icls,iwt,2,iq) *
     +            ratetable(icls,iwt)
               ckr = ckr + ni(icls,iwt,1,iq)
               ckp = ckp + ni(icls,iwt,2,iq)
            END IF
         END DO
         PRINT *, "Check Revenue = ",ckr
         PRINT *, "Check Pieces  = ",ckp
         PRINT *, "Check Revenue = ",idr + nir
         PRINT *, "After Redist. = ",idr + ckr
         WRITE(6,'(11F6.3)') (ratetable(icls,iwt),iwt=1,nwt)
      ELSE
         PRINT *, "Can't Solve."
         PRINT *, "rip     = ",rip
         PRINT *, "x       = ",x
         PRINT *, "imax    = ",imax
         PRINT *, "icls    = ",icls
         PRINT *, "nip     = ",nip
         PRINT *, "idr     = ",idr
         PRINT *, "nir     = ",nir
         PRINT *, "ara     = ",ara
         PRINT *, "arb     = ",arb
C              PRINT *, "Distributing on Identical Weight"
C              ckr = 0.0
C              DO iwt=1,nwt
C                 ni(icls,iwt,ishp,2,iq) =
C     +              distribution(icls,iwt,ishp) * nip
C                 ni(icls,iwt,ishp,1,iq) =
C     +              distribution(icls,iwt,ishp) *
C     +              nip * ratetable(icls,iwt)
C                 ckr = ckr + distribution(icls,iwt,ishp) *
C     +              nip * ratetable(icls,iwt)
C              END DO
C              PRINT *, "Check Revenue = ",ckr
      END IF
   ELSE  IF (nip.gt.0.0.and.rip.le.0.0) THEN
      PRINT *, "id good with no distr; idp = pmax"
      PRINT *, "icls = ",icls
      PRINT *, "nip  = ",nip
      PRINT *, "nir  = ",nir
   END IF
END DO

RETURN
END
```

```
//H38139XX JOB HSQ02,'CUTTING',CLASS=A,NOTIFY=H38139,MSGCLASS=H
//*
//S01       EXEC SAS
//WORK      DD SPACE=(CYL,(1000,100))
//LET1      DD DSN=HSI.RPW.HQ044D01.FY00PQ1,DISP=SHR
//LET2      DD DSN=HSI.RPW.HQ044D01.FY00PQ2,DISP=SHR
//LET3      DD DSN=HSI.RPW.HQ044D01.FY00PQ3,DISP=SHR
//LET4      DD DSN=HSI.RPW.HQ044D01.FY00PQ4,DISP=SHR
//RESULTS   DD DSN=H38139.RPW.FY00.FCM.DATA,DISP=SHR
//SYSIN     DD  *

*************************************************************;
* FILE:     H38139.SAS.CNTL(JOB08AX8)                       ;
* PURPOSE:  ROLL UP FCM SP INFLATED RPW BY                  ;
*           SHAPE,INDICIA,AND OUNCE INCREMENT FOR FY00      ;
*************************************************************;

%MACRO BLD(CORP1);
DATA LET&CORP1;
        INFILE LET&CORP1;
        INPUT
                FIN_NO          6.0
                TEST_ID         6.0
                STRATA          3.0
                MEPZIP          2.0
                TEST_DT         8.0
                VERSION         2.0
                END_TIME        4.0
                SKIP            4.0
                MAILCAT4       $4.0
                VOLUME          4.0
                REV            12.3
                WEIGHT         12.5
                AG_CD           3.0
                CAG            $1.
                SS_CD           4.0
                OD_ZIP          3.0
                ZONE            1.0
                SRPD_REV        8.3
                OVPD_REV        8.3
                INDICIA        $1.
                AUTO_COM        1.0
                SHAPE          $1.0
                LENGTH          3.0
                WIDTH           2.0
                HEIGHT          2.0
                GIRTH           2.0
                SMPLMETH        1.0
                SS_RECNO        5.2
                SES_NO          2.0
                REC_NO          7.2
                USER_ID        $3.
                LAPE_FL        $1.
                CSD             3.0
                AP              2.0
                REC_LOC        11.2
                COMBO          15.5
```

```
                ADJUST          10.5;

OZ = WEIGHT*16;  * CONVERT POUNDS TO OUNCES;
VOL2 = VOLUME*COMBO*ADJUST;
REV2 = REV*COMBO*ADJUST;
WEIGHT2 = WEIGHT*COMBO*ADJUST;


IF SHAPE EQ "." THEN SHAPE = "6";
IF SHAPE EQ " " THEN SHAPE = "6";

IF (SHAPE NE "0" AND SHAPE NE "1" AND SHAPE NE "2"
    AND SHAPE NE "3" AND SHAPE NE "4"
    AND SHAPE NE "O" AND SHAPE NE "B" AND SHAPE NE "6")
    THEN SHAPE = "6";

IF (MAILCAT4 EQ "ZERO" OR MAILCAT4 EQ "XNAX" OR
    MAILCAT4 EQ "XADM") THEN MAILCAT4 = "0000";
MAILCAT = MAILCAT4 * 1.0; *IF MAILCAT EQ . THEN MAILCAT = 0;

DATA LET&CORP1; SET LET&CORP1;

  * FCM CODES;
      IF ( MAILCAT = 1100 OR MAILCAT = 1105 OR MAILCAT = 1106 OR
        MAILCAT = 1120 OR MAILCAT = 1125  OR MAILCAT = 1500 OR
        MAILCAT = 6200

        OR MAILCAT = 1210 OR MAILCAT = 1220 OR
        MAILCAT = 1235

        OR MAILCAT = 1111 OR MAILCAT = 1131

        OR MAILCAT = 1181 OR MAILCAT = 1301
        OR MAILCAT = 1311 OR MAILCAT = 1441
        OR MAILCAT = 1451

        OR MAILCAT = 1141

        OR MAILCAT = 1231

        OR MAILCAT = 1411

        OR MAILCAT = 1241
        );

STEP = MAILCAT4 * 1.0;
IF OZ GT 0 THEN FLAG1 = 1;
IF VOLUME GT 0 THEN FLAG2 = 1;
FLAG3 = FLAG1 * FLAG2;
IF FLAG3 NE 1 THEN STEP = 99;

* STEPS DEFINED BY OUNCE INCREMENT;
IF FLAG3 EQ 1 THEN DO;
  IF OZ/VOLUME GT 0.0 AND OZ/VOLUME LE 0.5 THEN STEP = 1.0;
  IF OZ/VOLUME GT 0.5 AND OZ/VOLUME LE 1.0 THEN STEP = 2.0;
  IF OZ/VOLUME GT 1.0 AND OZ/VOLUME LE 1.5 THEN STEP = 3.0;
  IF OZ/VOLUME GT 1.5 AND OZ/VOLUME LE 2.0 THEN STEP = 4.0;
```

```
  IF OZ/VOLUME GT 2.0 AND OZ/VOLUME LE 2.5 THEN STEP = 5.0;
  IF OZ/VOLUME GT 2.5 AND OZ/VOLUME LE 3.0 THEN STEP = 6.0;
  IF OZ/VOLUME GT 3.0 AND OZ/VOLUME LE 3.5 THEN STEP = 7.0;
  IF OZ/VOLUME GT 3.5 AND OZ/VOLUME LE 4.0 THEN STEP = 8.0;
  IF OZ/VOLUME GT 4.0 AND OZ/VOLUME LE 5.0 THEN STEP = 9.0;
  IF OZ/VOLUME GT 5.0 AND OZ/VOLUME LE 6.0 THEN STEP = 10.0;
  IF OZ/VOLUME GT 6.0 AND OZ/VOLUME LE 7.0 THEN STEP = 11.0;
  IF OZ/VOLUME GT 7.0 AND OZ/VOLUME LE 8.0 THEN STEP = 12.0;
  IF OZ/VOLUME GT 8.0 AND OZ/VOLUME LE 9.0 THEN STEP = 13.0;
  IF OZ/VOLUME GT 9.0 AND OZ/VOLUME LE 10.0 THEN STEP = 14.0;
  IF OZ/VOLUME GT 10.0 AND OZ/VOLUME LE 11.0 THEN STEP = 15.0;
  IF OZ/VOLUME GT 11.0 AND OZ/VOLUME LE 12.0 THEN STEP = 16.0;
  IF OZ/VOLUME GT 12.0 AND OZ/VOLUME LE 13.0 THEN STEP = 17.0;
  IF OZ/VOLUME GT 13.0 AND OZ/VOLUME LE 14.0 THEN STEP = 18.0;
  IF OZ/VOLUME GT 14.0 AND OZ/VOLUME LE 15.0 THEN STEP = 19.0;
  IF OZ/VOLUME GT 15.0 AND OZ/VOLUME LE 16.0 THEN STEP = 20.0;
  IF OZ/VOLUME GT 16.0 AND OZ/VOLUME LE 17.0 THEN STEP = 21.0;
  IF OZ/VOLUME GT 17.0 AND OZ/VOLUME LE 18.0 THEN STEP = 22.0;
  IF OZ/VOLUME GT 18.0 AND OZ/VOLUME LE 19.0 THEN STEP = 23.0;
  IF OZ/VOLUME GT 19.0 AND OZ/VOLUME LE 20.0 THEN STEP = 24.0;
  IF OZ/VOLUME GT 20.0 THEN STEP = 25.0;
  IF OZ/VOLUME EQ 0.0 THEN STEP = 26.0;
  IF OZ/VOLUME LT 0.0 THEN STEP = 27.0;

END;

PROC SUMMARY DATA=LET&CORP1 NWAY;
   CLASS MAILCAT SHAPE INDICIA STEP;
   VAR VOL2 REV2 WEIGHT2;
   OUTPUT OUT=LET&CORP1 SUM=;

DATA LET&CORP1; SET LET&CORP1;
     FILE RESULTS(PQB&CORP1);
     PUT (MAILCAT SHAPE INDICIA STEP VOL2 REV2 WEIGHT2)
         (6.0 +2 $1. +2 $2. +2 2.0 +2 18.0 19.3 22.5);

RUN;
%MEND;

%BLD(1);
%BLD(2);
%BLD(3);
%BLD(4);
```

```
      PROGRAM priority

C     Date: 18 NOV 93
C     Programmer: mcb.mhm
C
C     Last Modified: 10-30-96 ta; fy 97 update
C                    02-04-97 ta; new record layout for input file
C                    02-20-97 ta; transaction date function
C                    08-15-97 ta; revise back to processing date format
C                    08-28-97 ta; skip GOV mail records
C                    01-15-98 ta; FY98 update
C                    01-25-99 ta; update for 99
C                    07-08-99 ta; keep all Priority records, including SM
C
C     Purpose: Extract Priority records from raw PI data
C

      IMPLICIT NONE

C     VARIABLES
      integer*4 jul_AP00, ier2           ! note function
      INTEGER*4 len,lines,year,date, tdate
      CHARACTER*6 finno
      CHARACTER*2 digit, cls, class
      CHARACTER*1 type, presort
      CHARACTER*5 pmtnum

      REAL*8 rev

C     BOOKKEEPPING
      INTEGER*4 i,ier,ap,in,j,k,num , badout, totin
      INTEGER*4 index, out3, foo
      character*5008  record

C     open output files

      do ap=1, 13           !!!!!!!!!!!! EDIT if needed !!!!!!!!!!!!!
         write(digit,'(i2.2)') ap
         num=40+ap
         open(num,file='priority.'//digit,
     +    recl=5008)
      end do


      open(20,file='permit.00.other',readonly)
 19   FORMAT(A)
 21   FORMAT(A6,a2,2x,i2,i3,10x,a5,i8,1x,f12.2,24x,i1,1x,a1,56x,i3,6x,a2)

      i = 0
      ier = 0
      out3=0
      totin=0
      ier2 = 0

      DO WHILE (ier.EQ.0)
         READ(20,19,iostat=ier,end=22) record
         read(record,21,iostat=ier2) finno,type,year,date,pmtnum,tdate,rev,foo,presort,
     +        lines,class
         totin= totin+1

         IF (pmtnum(1:1).eq.'G') GO TO 200    ! skip gov mail

         len = 138+lines*48
         if (ier2.gt.0) then
            print*,'bad rec at rec number ',totin,' record: ',record(1:138),
     +         ' ier = ',ier2
            ier2 = 0
            ier = 0
         else
            ap = jul_AP00(year,date)
            if (ap.ge.1.and.ap.le.13) then
               if (class(2:2).eq.'7') then  ! note all Priority
                  num=40+ap
                  write(num,19) record(1:len)
                  out3 = out3+1
               end if
            end if
         end if
200    continue
      END DO
```

```
22    PRINT*, 'exit code from permit data is: ',ier
      print*, 'total number of records in is: ',totin
      print*, 'total priority records is:    ',out3

      END
```

```
      PROGRAM rollpriority

C     DESCRIPTION: verifies and rolls up first class PRI PERMIT system rev and pcs
C                  by processing category and VIP
C
C     CREATED BY:        TS
C     DATE:              12/13/00
C
      IMPLICIT NONE

C     Program Parameters

      integer*4    nvip,i,j,k,l,npcat,nap,nq,iq,ipcat,maxvip,iap,ier
      integer*4    lines,searchc,ivip,iproc
      parameter    (nvip=39,npcat=5,nap=13,nq=4,maxvip=20)

      character*6   finno
      character*2   rectype
      character*5   permitnum,vip
      character*8   unique
      character*960 vipfields
      character*5   mapvip(nvip)/nvip*' '/
      character*2   cap(nap)/'01','02','03','04','05','06','07','08','09','10','11','12','13'/

      integer*4    ap_to_q(nap)/1,1,1,2,2,2,3,3,3,4,4,4,4/
      integer*4    recin/0/,nvipvalid/0/
      integer*4    recvip(maxvip)/maxvip*0/

      real*8       rev,fees,pcwt,pct,wt,revin/0.0/,viprev,vippcs,vipwt,viprate,pcs
      real*8       trev/0.0/,tpcs/0.0/,pcsout/0.0/
      real*8       revenue(nq,nvip,npcat)
      real*8       pieces(nq,nvip,npcat)
      real*8       weight(nq,nvip,npcat)
      real*8       recrev(maxvip)/maxvip*0/
      real*8       recpcs(maxvip)/maxvip*0/
      real*8       recwt(maxvip)/maxvip*0/

C     Initialize Variables

      do iproc = 1,npcat
         do ivip = 1,nvip
            do iq = 1,nq
               revenue(iq,ivip,iproc) = 0.0
               pieces(iq,ivip,iproc) = 0.0
               weight(iq,ivip,iproc) = 0.0
            end do
         end do
      end do


C     Read list of Priority VIP codes

      open(3,file='vipmap.pri.00',readonly)

4     format(a5)
      do i = 1, nvip
         read(3,4) mapvip(i)
      end do
      close(3)

C     MAIN LOOP
C     Read a complete PERMIT record. Assign the VIP, revenue, pieces & weight
C     from each VIP field to the record array.


11    FORMAT(A6,a2,17x,a5,a8,1X,F12.2,2X,F12.2,13X,I1,1X,F8.4,F12.0,F14.4,20X,I3,
     +   A960)
12    FORMAT(6X,A5,1X,F6.3,F12.0,F18.7)
13    format(i6,1x,i2,1x,i4,1x,i1,1x,i3,1x,i2,1x,a13,1x,f15.2,1x,2f15.0)
19    format(1x,i4)

C Read a PERMIT system record. Add the total revenue, pieces and weight
C to the running totals.

      do iap = 1,nap
         iq = ap_to_q(iap)
         ier = 0
         OPEN(10,FILE='priority.'//cap(iap),RECL=1090,readonly)

         DO WHILE (ier.eq.0)
```

```
            READ(10,11,IOSTAT=ier,END=99) finno,rectype,permitnum,unique,
     &         rev,fees,ipcat,pcwt,pcs,wt,lines,vipfields
            trev = 0.0
            tpcs = 0.0
            pcsout = 0.0
            recin = recin + 1
            revin = revin + rev
            IF (lines.gt.maxvip) THEN
               PRINT *, "Warning: maxvip exceeded with value: ",lines,
     +            " at record ",recin,"."
               STOP
            END IF

C Read each VIP field of the transaction. Check to see if it is a valid VIP

            DO i=1,lines
               read(vipfields((i-1)*48+1:(i-1)*48+48),12) vip,viprate,vippcs,
     +            viprev
C               read(vip,19) pvip
               IF (vip(1:1).eq.' '.and.vip.ne.'    0') THEN
                  vip(1:1) = '0'
               END IF
               ivip = searchc(mapvip,nvip,vip)

               IF (ivip.gt.0) THEN
                  revenue(iq,ivip,ipcat) = revenue(iq,ivip,ipcat) + viprev
                  pieces(iq,ivip,ipcat) = pieces(iq,ivip,ipcat) + vippcs
                  weight(iq,ivip,ipcat) = weight(iq,ivip,ipcat) + (wt/pcs*vippcs)
                  trev = trev + viprev
                  tpcs = tpcs + vippcs
                  nvipvalid = nvipvalid + 1
                  recvip(i) = ivip
                  recrev(i) = viprev
                  recpcs(i) = vippcs
                  recwt(i) = wt/pcs * vippcs
               ELSE IF (ivip.le.0.and.vip.eq.'    0') THEN
                  viprev = 0.0
                  vippcs = 0.0
                  viprate = 0.0
                  nvipvalid = nvipvalid + 1
                  recvip(i) = 0
                  recrev(i) = 0
                  recpcs(i) = 0
                  print *,'vip error ', vip,' fin ',finno,' lines ',lines,' rev ',
     +               viprev
               ELSE             !  non-PI vip
                  viprev = 0.0
                  vippcs = 0.0
                  viprate = 0.0
                  nvipvalid = nvipvalid + 1
                  recvip(i) = 0
                  recrev(i) = 0
                  recpcs(i) = 0

                  print *,'vip error ', vip,' fin ',finno,' lines ',lines
               END IF
            END DO

         END DO                 ! Main Loop

99       print*,'Finished Reading AP ',cap(iap),' File with ier = ',ier
      end do

      open(15,file='priroll.00')
25    format(a5,1x,i1,5f12.2,10f12.0)

      do iq = 1, nq
         do ivip = 1, nvip
            write(15,25) mapvip(ivip), iq, (revenue(iq,ivip,iproc),iproc=1,5),
     &         (pieces(iq,ivip,iproc),iproc=1,5), (weight(iq,ivip,iproc),iproc=1,5)
         end do
      end do

      END
```

```
#!/usr/bin/ksh
for file in `more nodelist`
do
    typeset -i skip=0
    typeset -i outrec=0
    gzcat /u/mcb/second/FY00b/tape/data/S.*.${file}.data.gz > unpacked.s
    dgsort -c sortpub.sm -O -sl 1.sort
    rm unpacked.s
    grep 'Total number of output records' 1.sort >> temp
    cat temp | read junk junk junk junk junk outrec
    grep 'Number of input records skipped' 1.sort >> temp
    cat temp | read junk junk junk junk junk skip

    chkforzero
    reverse_v2 > list/Rev.${file}.out


    mv goodtrans   data/S.reversed.${file}
    mv badreverses data/S.bad.${file}
    mv matched     data/S.matched.${file}
    gzip data/S.reversed.${file}
    gzip data/S.matched.${file}
    gzip data/S.bad.${file}
    rm unpacked
    rm 1.sort
    rm temp
    echo "Node ${file}                  " >> list/rec_check.${file}
    echo "Output  Records:     $outrec " >> list/rec_check.${file}
    echo "Skipped Records:     $skip   " >> list/rec_check.${file}
done
```

```
:nput file is 'unpacked.s', recs are data sensitive  upto 4250 chars.
:utput file is 'unpacked', recs are  data sensitive  upto 4250 chars.
:ey 7/7.
:ey 1/6.
:ey 24/30.
:ey 23/23.
:ey 37/44.
:ey 104/115.
:ort.
:nd.
```

```
      PROGRAM chkforzero

C     PURPOSE: remove headers without any vip info
C     By Kaz
C     Input : Concatenated files for each node..(both reversals and non reversals)
C
C

      IMPLICIT NONE

      integer*4       maxrec, maxtran,maxvips

      parameter       (maxrec  =  31000) !MAX RECORDS IN PUB ARRAY
      parameter       (maxvips = 84)

      real*8          pcs,tpcs,vpcs,vrpcs,trev,lrev ! # of pieces in header & pcs in reverse

      integer*4       ier,i,k,j,ier2,rempc
      integer*4       nvips       ! total number of vips
      integer*4       numvips(maxrec) ! # of vip array
      integer*4       reversed(maxrec) ! Reversed array
      integer*4       reversal(maxrec) ! Indicate reversal
      integer*4       nmatch,nrec,nunmatch,nrev,npub,ngood

      character*130   header(maxrec),theader ! Header array for publications
      character*49    recbuf(maxrec,maxvips),tbuf(maxvips) ! Record Buffer array for publications
      character*7     pubno,lpubno ! Publication No & Last Publication
      character*6     file,fin, lastfin

      logical found , ndiff

      open (14,file = 'baddata',access='append',recl = 4250)
      open (15,file='Revenue.zero.fin',access='append')
      open (16,file='datafile',recl = 4250)
      open (17,file='unpacked',recl = 4250)
      open (20,file='TakeZero.out', access='append')
C     Read data one publication at a time

      ier  =  0
      ier2 = 0

      lastfin='      '
      trev=0.0
      do while(ier.eq.0)
            read (17,'(a130,84a49)',iostat = ier,end = 100) theader,tbuf
            read(theader(128:130),'(i3)',iostat=ier2) nvips
            read (theader(1:6),'(a6)') fin
            if (ier2.ne.0) then
               write(20,'(a30,i6)') 'ier2 at first read was ',ier2
               ier2=0
            end if
            if (nvips.gt.0) then
                  write (16,'(a130,84a49)') theader,(tbuf(k),k = 1,nvips)
            else
               read (theader(104:115),'(f12.2)') lrev
               if (fin.ne.lastfin) then
                  write(15,'(a6,f16.2)') lastfin, trev
                  lastfin=fin
                  trev=lrev
               else
                  trev=trev+lrev
               end if
               write (14,'(a130,84a49)') theader,(tbuf(k),k = 1,nvips)
            end if
      end do


100   write(20,'(a30,i6)') 'Doing the last loop',ier
      write(15,'(a6,f16.2)') lastfin, trev

      end
```

```
      PROGRAM reverse_v2
C
C     PURPOSE: remove reversals from PERMIT second-class data
C     By Kaz
C     Input : Concatenated files for each node..(both reversals and non reversals)
C
C
      IMPLICIT NONE

      integer*4       maxrec, maxtran,maxvips

      parameter       (maxrec  =  31000) !MAX RECORDS IN PUB ARRAY
      parameter       (maxvips = 84)

      real*8          trev,rev,pcs,tpcs,vpcs,vrpcs ! # of pieces in header & pcs in reverse

      integer*4       ier,i,k,j,ier2,tempc
      integer*4       nvips       ! total number of vips
      integer*4       numvips(maxrec) ! # of vip array
      integer*4       reversed(maxrec) ! Reversed array
      integer*4       reversal(maxrec) ! Indicate reversal
      integer*4       nmatch,nrec,nunmatch,nrev,npub,ngood
      integer*4       cnt

      character*130   header(maxrec),theader ! Header array for publications
      character*49    recbuf(maxrec,maxvips),tbuf(maxvips) ! Record Buffer array for publications
      character*7     pubno,lpubno ! Publication No & Last Publication
      character*6     finno,lfinno ! Finance # and Last Fin. #
      character*6     file

      logical found , ndiff

C      call  getarg(1,file)
      open (13,file = 'goodtrans',recl = 4250)
      open (14,file = 'badreverses',recl = 4250)
      open (15,file = 'matched',recl = 4250)
      open (16,file = 'datafile',recl = 4250)
      print*,'opened files'

C   Read data one publication at a time

      ier  =  0
      nrec = 0
      npub = 0
      ngood = 0
      nunmatch = 0
      nmatch = 0
      nrev = 0
      ier2 = 0
      tempc=0
      cnt=0


      do while(ier.eq.0)
         if (nrec.eq.0) then    ! If this is the first record
            read (16,'(a130,84a49)',iostat = ier,end = 100) theader,tbuf
            tempc=tempc+1
            nrec = nrec+1
            npub = npub+1
            header(npub) = theader

            read(theader(128:130),'(i3)',iostat=ier2) nvips
            if (ier2.ne.0) then
               print*,'ier2 at first read was ',ier2
               ier2=0
            end if

            numvips(npub) = nvips
            do i = 1,nvips
               recbuf(npub,i) = tbuf(i)
            end do
            read (theader(24:30),'(a7)',iostat=ier2)  lpubno
            if (ier2.ne.0) then
               print *,'ier2 at second read is ',ier2
               ier2=0
            end if

            read (theader(1:6),'(a6)',iostat=ier2)  lfinno
            if (ier2.ne.0) then
               print*,'ier2 at third read is ',ier2
```

```
            ier2=0
         end if
      end if

C  Start reading till new publication is reached
      read (16,'(a130,84a49)',iostat = ier,end = 100) theader,tbuf
      tempc=tempc+1
      read (theader(24:30),'(a7)',iostat=ier2)  pubno
      if (ier2.ne.0) then
         print*,'ier2 at fourth read is ',ier2
         ier2=0
      end if

      read (theader(1:6),'(a6)',iostat=ier2)  finno
      if (ier2.ne.0) then
         print*,'ier2 at fifth read is ',ier2
         ier2=0
      end if

      nrec = nrec+1
      if ((pubno.eq.lpubno).and. (finno.eq.lfinno)) then
         npub = npub+1
         header(npub) = theader
         read(theader(128:130),'(i3)',iostat=ier2) nvips
         if (ier2.ne.0) then
            print*,'ier2 at sixth read is ',ier2
            ier2=0
         end if

         numvips(npub) = nvips
         do i = 1,nvips
            recbuf(npub,i) = tbuf(i)
         end do
      else                    !  all the publication data is loaded

         do i = 1, npub       ! Initialize arrays
            reversed(i) = 0
            reversal(i) = 0
         end do

         do i = 1,npub        ! Go through & find the reversals
            if (header(i)(8:8).eq.'R') then ! Bingo we have a reversal
               reversal(i) = 1
               read (header(i)(104:115),'(f12.2)',iostat=ier2) trev
               if (ier2.ne.0) then
                  print*,'ier2 at seventh read is ',ier2
                  ier2=0
               end if

               found  =  .false.
               k  =  i - 1
               do while (.not.found.and.k.gt.0)
                  if ((header(k)(8:8).ne.'R').and.
     &                reversal(k).eq.0.and.reversed(k).eq.0) then ! check for reversals
                     read (header(k)(104:115),'(f12.2)',iostat=ier2) rev
                     if (ier2.ne.0) then
                        print*,'ier2 at eighth read is ',ier2
                        ier2=0
                     end if

                     if ((rev+trev .le. 0.01).and.(rev+trev.ge.-0.01)) then ! we might have  a winner
                        read (recbuf(i,1)(30:39),'(f10.2)',iostat=ier2)vrpcs
                        if (ier2.ne.0) then
                           print*,'ier2 at ninth read is ',ier2,recbuf(i,1)(30:39),i,'-',tempc
                           ier2=0
                        end if

                        read (recbuf(k,1)(30:39),'(f10.0)',iostat=ier2)vpcs
                        if (ier2.ne.0) then
                           print*,'ier2 at eleventh read is ',ier2,recbuf(k,1)(30:39),k,'- ',tempc
                           ier2=0
                        end if

                        if ((header(k)(128:130).eq.header(i)(128:130))
     &                     .and.(ndiff(vrpcs,vpcs))) then !YES
                           reversed(k) = 1
                           reversal(i) = 3
                           k = npub+1
                           found = .true.
```

```
                        else
                        end if
                     end if
                  end if
                  k = k - 1  ! check backwards in time for reversed transaction
               end do
               if (.not.found) then ! reversed transaction not found before reversal
                  k = i + 1
                  do while (.not.found.and.k.le.npub)
                     if ((header(k)(8:8).ne.'R').and.
     &                    reversal(k).eq.0.and.reversed(k).eq.0) then ! check for reversals
                        read (header(k)(104:115),'(f12.2)',iostat=ier2) rev
                        if (ier2.ne.0) then
                           print*,'ier2 at twelvth read is ',ier2
                           ier2=0
                        end if

                        if ((rev+trev .le. 0.01).and.(rev+trev.ge.-0.01)) then ! we might have  a winner
                           read (recbuf(i,1)(30:39),'(f10.2)',iostat=ier2) vrpcs
                           if (ier2.ne.0) then
                              print*,'ier2 at thirteenth read is ',ier2
                              ier2=0
                           end if

                           read (recbuf(k,1)(30:39),'(f10.0)',iostat=ier2) vpcs
                           if (ier2.ne.0) then
                              print*,'ier2 at fourteenth read is ',ier2
                              ier2=0
                           end if

                           if ((header(k)(128:130).eq.header(i)(128:130))
     &                          .and.(ndiff(vrpcs,vpcs))) then !YES
                              reversed(k) = 1
                              reversal(i) = 3
                              k = npub+1
                              found  =  .true.
                           else
                           end if
                        end if
                     end if
                     k = k + 1 ! check forwards in time for reversed transaction
                  end do
               end if
            end if
         end do
* Reporting and writing to files
         do i = 1,npub
            if ((reversed(i).eq.0).and.(reversal(i).eq.0)) then
               write (13,'(a130,84a49)') header(i),(recbuf(i,k),k = 1,numvips(i))
               ngood = ngood+1
            else if (reversed(i).eq.1) then
               nrev = nrev+1
            else if (reversal(i).eq.1) then
               nunmatch = nunmatch+1
               write (14,'(a130,84a49)') header(i),(recbuf(i,k),k = 1,numvips(i))
            else if (reversal(i).eq.3) then
               nmatch = nmatch+1
               write (15,'(a130,84a49)') header(i),(recbuf(i,k),k = 1,numvips(i))
            end if
         end do

         Set up arrays for next publication

         npub = 1
         header(npub) = theader
         read(theader(128:130),'(i3)',iostat=ier2) nvips
         if (ier2.ne.0) then
            print*,'ier2 at fifteenth read is ',ier2
            ier2=0
         end if

         numvips(npub) = nvips
         do i = 1,nvips
            recbuf(npub,i) = tbuf(i)
         end do
         lpubno = pubno
         lfinno = finno

      end if
   end do
```

```
100   print *,'Doing the last loop'
      do i = 1,maxrec
         reversed(i) = 0
         reversal(i) = 0
      end do
      do i = 1,npub
         if (header(i)(8:8).eq.'R') then ! we have a reversal
            reversal(i) = 1
            read (header(i)(104:115),'(f12.2)') trev
            k = i - 1
            found = .false.
            do while (.not.found.and.k.gt.0)
               if ((header(k)(8:8).ne.'R').and.
     &             reversal(k).eq.0.and.reversed(k).eq.0) then ! check for reversals
                  read (header(k)(104:115),'(f12.2)') rev

                  if ((rev+trev .le. 0.01).and.(rev+trev.ge.-0.01)) then ! we might have  a winner
                     read (recbuf(i,1)(30:39),'(f10.2)')vrpcs
                     read (recbuf(k,1)(30:39),'(f10.0)')vpcs
                     if ((header(k)(128:130).eq.header(i)(128:130))
     &                    .and.(ndiff(vrpcs,vpcs))) then
                        reversed(k) = 1
                        reversal(i) = 3
                        k = npub+1
                     else
                     end if
                  end if
               end if
               k = k - 1
            end do
            if (.not.found) then
               k = i + 1
               do while (.not.found.and.k.le.npub)
                  if ((header(k)(8:8).ne.'R').and.
     &                reversal(k).eq.0.and.reversed(k).eq.0) then ! check for reversals
                     read (header(k)(104:115),'(f12.2)') rev
                     if ((rev+trev .le. 0.01).and.(rev+trev.ge.-0.01)) then ! we might have  a winner
                        read (recbuf(i,1)(30:39),'(f10.2)') vrpcs
                        read (recbuf(k,1)(30:39),'(f10.0)') vpcs
c                             vrpcs = vrpcs*(-1)
                        if ((header(k)(128:130).eq.header(i)(128:130))
     &                      .and.(ndiff(vrpcs,vpcs))) then !YES
                           reversed(k) = 1
                           reversal(i) = 3
                           k = npub+1
                           found = .true.
                        else
c                                print *,' loop 2: pcs ',pcs,' tpcs ',tpcs,' vpcs ',vpcs,' vrpcs ',vrpcs
                        end if
                     end if
                  end if
                  k = k + 1      ! check forwards in time for reversed transaction
               end do
            end if
         end if
      end do
c Reporting and writing to files
      do i = 1,npub
         if ((reversed(i).eq.0).and.(reversal(i).eq.0)) then
            write (13,'(a130,84a49)') header(i),(recbuf(i,k),k = 1,numvips(i))
            ngood = ngood+1
         else if (reversed(i).eq.1) then
            nrev = nrev+1
         else if (reversal(i).eq.1) then
            nunmatch = nunmatch+1
            write (14,'(a130,84a49)') header(i),(recbuf(i,k),k = 1,numvips(i))
         else if (reversal(i).eq.3) then
            nmatch = nmatch+1
            write (15,'(a130,84a49)') header(i),(recbuf(i,k),k = 1,numvips(i))
         end if
      end do

c      print *,'Node Name is: ',file
      print *,'Number of records              : ',nrec
      print *,'Number of reversed transactions: ',nrev
      print *,'Number of matched reversals    : ',nmatch
```

```
      print *,'Number of unmatched reversals  : ',nunmatch

      end
C ------------------ ------------------ ----------------------------------------
      logical function ndiff(r1, r2)

      real*8 r1, r2

      if (nint(r1).eq.nint(r2)) then
         ndiff = .true.
      else
         ndiff = .false.
      end if

      return
      end
```

```
      program  sepdata

c     written by Karina - 11/99
c     This program checks rates for different VIP codes to
c     separate data into two data sets.  One is the data pre rate
c     change (1/99), and the other is post rate change.  The date
c     alone does not do a satisfactory job of testing.  There are a
c     number of transactions where the date is post rate change, however
c     the VIP code structure is pre rate change.
c


      implicit none
      integer*4    nchk, nq
      parameter    (nchk = 142)
      parameter    (nq = 4)

      real*8       rev(nchk)
      real*8       pcs(nchk)
      real*8       wght(nchk)
      real*8       cps(nchk)
      real*8       rate1(nchk,nq)
      real*8       rate2(nchk,nq)
      real*8       rate3(nchk,nq)

      integer*4    chkcon(nchk,5)
      character*1  ratetype
      character*4  vipc(nchk)
      character*5  vips(nchk),vip
      character*49 viprec(nchk)
      character*4246 recs
      character*130 recl
      real*8       chkr1
      real*8       r,p,w,c

      integer*4    i,j,k,icls, searchc
      integer*4    ivip,classind
      integer*4    ier, ier2, type
      integer*4    nvips,count
      character*7  date

      logical      vipused(nchk)
      logical      found
      logical      diff,diffa

c COMMON BLOCK FOR CONSISTENCY CHECK


      open(15,file='seccon.950101.950931')
150   format(6x,a4,3i3,i2,i3)
      open(16,file='seccon.981004.990109')
160   format(27x,4f9.5)
      open(17,file='seccon.990110.present')
      open(18,file='seccon.971005.981003')

      do i = 1, nchk
         read(15,150)  vipc(i), (chkcon(i,j),j=1,5)
         read(18,160) (rate1(i,j),j=1,4)
         read(16,160) (rate2(i,j),j=1,4)
         read(17,160) (rate3(i,j),j=1,4)
      end do

      print*, 'seccon.dat has been read '
      close(15)
      close(16)
      close(17)
      close(18)

      open(20,file='prechange.data',recl=4500)
      open(21,file='postchange.data',recl=4500)
      open(22,file='S.reversed.all',recl=4500)
95    format(a,100(a))

      ier=0
      count=0
      do while (ier.eq.0)
c        count=count+1
         read(22,'(a)',iostat=ier,end=100) recs
         type=0
         chkr1=0
         ier2=0
```

```
      nvips=0
      do i=1,nchk
         vipused(i) = .false.
         rev(i)=0
         pcs(i)=0
         wght(i)=0
         cps(i)=0
      end do
      read(recs(10:16),'(a7)') date
      read(recs(53:53),'(a1)') ratetype
      ier2=0
      read(recs(128:130),'(i3)',iostat=ier2) nvips
      if (nvips.gt.0) then
         if (ier2.eq.0) then
            rec1 = recs(1:130)
            do i = 1, nvips
               read(recs((131+(i-1)*49):(179+(i-1)*49)),'(a49)') viprec(i)
            end do
            do j = 1, nvips
               read (viprec(j),'(a5,f12.2,f12.0,2f10.0)') vip,r,p,w,c
               ivip = searchc(vipc,nchk,vip(2:5))
               if (vip.ne.'     ') then
                  if (ivip.gt.0) then
                     vipused(ivip) = .true.
                     vips(ivip) = vip
                     rev(ivip) = r
                     pcs(ivip) = p
                     wght(ivip) = w
                     cps(ivip) = c
                  else
                     print *,ivip, 'my j', j
                     print *,' vip ',vip,' not found in check list '
                     print *,vip(2:5),r,'-',p,'-',w,'-',c
                  end if
               end if
            end do
            icls = classind(ratetype)
            found=.false.
            do i = 1, nchk
               if (vipused(i).eq..true.) then
                  if (((vipc(i).eq.'1360').or.(vipc(i).eq.'1370')).and.
&                    (found.eq..false.)) then
                     if (rev(i).gt.0) then
                        chkr1 = wght(i) * rate3(i,icls)
                        if (diffa(rev(i),chkr1)) then
                           chkr1 = wght(i) * rate2(i,icls)
                           if (diffa(rev(i),chkr1)) then
                              chkr1 = wght(i) * rate1(i,icls)
                              if (diffa(rev(i),chkr1)) then ! rate doesn't match any, problem!
                                 type=0
                              else
                                 type=1
                                 found=.true.
                              end if
                           else
                              type=1
                              found=.true.
                           end if
                        else
                           type=2
                           found=.true.
                        end if
                     end if
                  end if
               end if
            end do
            do i = 1, nchk
               if (vipused(i).eq..true.) then
                  if (((vipc(i).eq.'2160').or.(vipc(i).eq.'2190').or.
&                    (vipc(i).eq.'2210').or.(vipc(i).eq.'2220').or.
&                    (vipc(i).eq.'2214').or.(vipc(i).eq.'2215').or.
&                    (vipc(i).eq.'2230').or.(vipc(i).eq.'2240').or.
&                    (vipc(i).eq.'2244').or.(vipc(i).eq.'2245').or.
&                    (vipc(i).eq.'2220')).and.(found.eq..false.)) then
                     if (rev(i).gt.0) then
                        chkr1 = pcs(i) * rate3(i,icls)
                        if (diff(rev(i),chkr1)) then
                           chkr1 = pcs(i) * rate2(i,icls)
                           if (diff(rev(i),chkr1)) then
                              chkr1 = pcs(i) * rate1(i,icls)
```

```
                              if (diff(rev(i),chkr1)) then ! rate doesn't match any, problem!
                                 type=0
                              else
                                 type=1
                                 found=.true.
                              end if
                           else
                              type=1
                              found=.true.
                           end if
                        else
                           type=2
                           found=.true.
                        end if
                     end if
                  end if
               end do
               if (found.eq..false.) then ! didn't find type from piece vips
                  do i=1,nchk
                     if (vipused(i).eq..true.) then
                        if (((vipc(i).eq.'2060').or.(vipc(i).eq.'2070').or.
     &                     (vipc(i).eq.'2080').or.(vipc(i).eq.'2090'))
     &                     .and.(found.eq..false.)) then
                           if (rev(i).gt.0) then
                              chkr1 = wght(i) * rate3(i,icls)
                              if (diff(rev(i),chkr1)) then
                                 chkr1 = wght(i) * rate2(i,icls)
                                 if (diff(rev(i),chkr1)) then
                                    chkr1 = wght(i) * rate1(i,icls)
                                    if (diff(rev(i),chkr1)) then ! rate doesn't match any, problem!
                                       type=0
                                    else
                                       type=1
                                       found=.true.
                                    end if
                                 else
                                    type=1
                                    found=.true.
                                 end if
                              else
                                 type=2
                                 found=.true.
                              end if
                           end if
                        end if
                     end do
                  end if
                  if (type.eq.0) then ! didn't find type from weight vips
                     if (date.ge.'1999010') then
                        type=2
                     else
                        type=1
                     end if
                  end if

                  if (type.ne.0) then
                     if (type.eq.1) then
                        write(20,25) rec1(1:130),(viprec(i),i=1,nvips)
                     else if (type.eq.2) then
                        write (21,25) rec1(1:130),(viprec(i),i=1,nvips)
                     end if
                  end if
               else
                  count=count+1
               end if
            end if

      end do

100   print *,'end of file read, ier = ',ier,' count=',count

      end

C     -------  -------  ---------  ----------------  ----------------
      function diff(v1,v2)

      logical diff
      real*8    v1, v2
```

```
      real*8    tol
      parameter (tol = 0.005)

      if (dabs(v1-v2).gt.tol) then
         diff = .true.
      else
         diff = .false.
      end if

      return
      end
C-----------  ----------------------  ------------------------------------
       function diffa(v1,v2)

      logical diffa
      real*8    v1, v2
      real*8    tol
      parameter (tol = 0.001)

      if (dabs(v1 v2).gt.tol) then
         diffa = .true.
      else
         diffa   .false.
      end if

      return
      end
C-----------  ----------  ------------------------------------------
        function    classind(code)

        character*1 code
        integer*4   classind

        if (code.eq.'9') then
           classind   4
        else if (code.eq.'6') then
           classind = 1
        else if (code.eq.'7') then
           classind = 2
        else if (code.eq.'8') then
           classind = 3
        else
           classind = 4
        end if

        return
        end
```

```
      program bin2nd2

C     Mike McGrane 11-15-96

C     This program reads second-class PERMIT transactions
C     and checks them for internal consistency
C     It writes three output files in binary number format:
C     good transactions, bad transactions, and good trans.
C     revenue by finance number and quarter.
C
C     last Modified to handle 2000 data
C

      implicit none

      integer*4 nvip, nfin, nq, nchk, nerror, ngcnt, nrec, fixed, intlflag

      parameter (nvip = 528)     ! number of vips
      parameter (nfin = 1877)    ! number of PERMIT finance number
      parameter (nq = 4)         ! number of quarters
      parameter (nchk = 142)     ! number of 4-digit VIP costs
      parameter (nerror = 25)    ! number of error codes
      parameter (ngcnt = 7)      ! number of transaction repair codes
      parameter (nrec = 20000)   ! number of records in recs array


      real*8  rev(nchk), cps(nchk), wght(nchk), pcs(nchk), twght(nchk)
      real*8  brev(nerror), bpcs(nerror), bwght(nerror)
      real*8  r, p, w, c, edper, zwo, nw, wpc, totr, totp, totw
      real*8  rate1(nchk,4), rate2(nchk,4), rate3(nchk,4)
      real*8  grevenue(nq,nfin), brevenue(nq,nfin)
      real*8  goodrev, goodpcs,tviprev,viprevchk,tvipcps,tpvipcps
      real*8  tvipcpsi,tvipcpso,tpvipcpsi,tpvipcpso

      integer*4 ichk, ivip, ierror, iq, ifin, i, j, k, ier, irec, nvips, igood, icls
      integer*4 chkcon(nchk,5), gcnt(ngcnt,nchk)
      integer*4 nbad(nerror), ngood,cnt
      integer*4 qind, searchc, il
      integer*4 ncommerror, rlength, classind, ier2,vipcount

      character*4    vipc(nchk)
      character*5    vips(nchk), vip
      character*4246 recs(nrec)
      character*4246 outrec
      character*130  header
      character*49   tail(nchk)
      character*6    fins(nfin), fin, finno
      character*6    pub, lastpub
      character*8    tdate, date, jul_to_txt
      character*1    ratetype, cratetype
      character*13   tr_num
      character*15   transno
      character*4    clength
      character*1    clind

      logical vipused(nchk)

C     common block for passing data to checking subroutine

      common /check/ rev, pcs, wght, cps, totp, totr, chkcon, vipc, gcnt, rate1,
     &   rate2, rate3, vipused,vips,tr_num,finno

      logical first, itype2, mflag, good, badvips

C     Initialize values and arrays
      ncommerror = 0
      ngood = 0
      do i = 1, nerror
         brev(i) = 0.
         bpcs(i) = 0.
         bwght(i) = 0.
         nbad(i) = 0
      end do
      do ifin = 1, nfin
         do iq = 1, nq
            brevenue(iq,ifin) = 0.
            grevenue(iq,ifin) = 0.
         end do
      end do
```

```
      print *,'starting program'

C READ CONTROL FILE

      open(15,file='seccon.971005.981003')
150   format(6x,a4,3i3,i2,i3,3x,4f9.5)
      open(16,file='seccon.981004.990109')
160   format(27x,4f9.5)
      open(17,file='seccon.950101.950931')
      open(18,file='seccon.990110.present')

      do i = 1, nchk
         read(17,150)  vipc(i), (chkcon(i,j),j=1,5)
         read(15,160) (rate1(i,j),j=1,4)
         read(16,160) (rate2(i,j),j=1,4)
         read(18,160) (rate3(i,j),j=1,4)
      end do

      print*, 'seccon.dat has been read '
      close(15)
      close(16)

C     Read PERMIT finance numbers

      open(15,file='fin.map')
      do i=1,nfin
         read(15,'(a6)') fins(i)
      end do


C      read PERMIT records, one publication at a time, check for internal
C      consistency and write to output file

      open(20,file='prechange.data',recl=4500)

C      open(30,file='bin2nd.data',form='unformatted',access='sequential',
C     &   organization='dynamic',carriagecontrol='none',iointent='output')

      open(30,file='olddata.data',recl=4500)
      open(40,file='bin2nd.data.bad',recl=4500)
      open(50,file='bin2nd.data.badvips',recl=4500)
      open(48,file='oldrevdiff.txt',recl=4500)
      open(49,file='oldrevheaddiff.txt',recl=4500)
      open(46,file='oldfixeddiff.txt',recl=4500)
      open(47,file='oldfixedhead.txt',recl=4500)



      ier = 0
      first = .true.
      irec = 1
      cnt=0

      do while (ier.eq.0)
         read(20,'(a)',iostat=ier,end=100) recs(irec)

         if (first) then
            lastpub = recs(irec)(25:30)
            first = .false.
         end if
         pub = recs(irec)(25:30)
         do while (pub.eq.lastpub) ! read all the records for a single publication
            irec = irec + 1
            read(20,'(a)',iostat=ier,end=100) recs(irec)
            pub = recs(irec)(25:30)
         end do
100      if (ier.ne.0) then
            print *,' read exit of input file = ',ier
         end if

C        process records for the publication

         do i = 1, irec - 1
            cnt=cnt+1
            outrec(1:130) = recs(i)(1:130)
            ratetype = recs(i)(53:53)
            date = jul_to_txt(recs(i)(10:16))
            iq = qind(date)
            ifin = searchc(fins,nfin,recs(i)(1:6))
```

```
      finno = fins(ifin)
      cratetype = ' '
      do ivip = 1, nchk
         rev(ivip) = 0.
         pcs(ivip) = 0.
         wght(ivip) = 0.
         cps(ivip) = 0.
         vips(ivip) = '     '
         vipused(ivip) = .false.
         tviprev=0.0
      end do
      tr_num = recs(i)(10:22)
      if (recs(i)(83:83).eq.'Y') then ! Search for commingled transaction
         transno = recs(i)(84:98)
         transno = recs(i)(97:98)//recs(i)(84:96)
         do j = 1, irec - 1
            if (recs(j)(8:22).eq.transno) then
               cratetype = recs(j)(53:53)
               print *,'found this match'
            end if
         end do
         if (cratetype.eq.' ') then
            print *,' nonsubscriber commingled trans match failed ',lastpub,',',recs(i)(1:6),'-',recs(j)(8:22),'-',
&              transno
            ncommerror = ncommerror + 1
         end if
      end if
      read(recs(i)(128:130), '(i3)', iostat=ier2) nvips
      vipcount=0
      if (ier2.eq.0) then
         badvips = .false.
         do j = 1, nvips
            read (recs(i)((131+(j-1)*49):(179+(j-1)*49)),'(a5,f12.2,f12.0,2f10.0)')
&              vip, r,  p, w, c
            if (vip(1:1).eq.'X') then
               if (cratetype.ne.' ') then
                  vip(1:1) = clind(cratetype) ! get rate type of parent transaction
                  ratetype = cratetype
               else
                  vip(1:1) = '9' ! requestor
               end if
            end if
            ivip = searchc(vipc,nchk,vip(2:5))
            if (ivip.gt.0) then
               vipused(ivip) = .true.
               vips(ivip) = vip
               rev(ivip) = r
               pcs(ivip) = p
               wght(ivip) = w
               cps(ivip) = c
               tviprev=rev(ivip)*chkcon(ivip,2)+tviprev

               if (ivip.le.28) then !IC
                  tvipcpsi=cps(ivip)*chkcon(ivip,5)+tvipcpsi ! Total Copies

                  if (((ivip.ge.64).and.(ivip.le.83)).or.((ivip.ge.8).and.(ivip.le.25))) then
                     tpvipcpsi = tpvipcpsi + cps(ivip)

                  end if
               else       !OC
                  tvipcpso=cps(ivip)*chkcon(ivip,5)+tvipcpso ! Total Copies
                  if (((ivip.ge.64).and.(ivip.le.83)).or.((ivip.ge.8).and.(ivip.le.25))) then
                     tpvipcpso = tpvipcpso + cps(ivip)

                  end if
               end if

               if ((ivip.ge.90).and.(ivip.le.99)) then
                  intlflag = 1
               end if



            else
               print *,' vip ',vip,' not found in check list '
               badvips = .true.
            end if
         end do
         read(recs(i)(104:127),'(f12.2,f12.0)') totr, totp
      else
```

```
      read(recs(i)(104:127),'(f12.2,f12.0)') totr, totp
      if (totp.gt.0) then
         print *,' error - no vip detail with totp > 0 ',totp,',',lastpub,',',recs(1)(1:6)
      end if
   end if
   icls = classind(ratetype)
   call checktrn(itype2,good,mflag,wpc,icls,ierror,igood,edper,tdate)
   if (good.and.iq.gt.0.and..not.badvips.and.ifin.gt.0) then

      rlength = 130 + (37*(nvips-vipcount)) + 4 ! length of variable format record
      j=0
      do ivip=1,nchk
         if (vipused(ivip)) then
            j=j+1
         end if
      end do
      write (recs(i)(128:130),'(i3)') j
      if (((tviprev-totr).gt..1).or.((tviprev-totr).lt.-.1)) then !Rev diff
         fixed = 0
         tvipcps = tvipcpsi + tvipcpso
         tpvipcps = tpvipcpsi + tpvipcpso
         if (tvipcps.gt.0) then
            if (((totr-tviprev)/tvipcps.ge.0.095).and.((totr-tviprev)/tvipcps.le.0.105)) then
               fixed = 1
               if (tvipcpsi.gt.0) then
                  nvips = nvips + 1
                  vipused(nvips) = .true.
                  vips(nvips) = '99998'
                  rev(nvips) = tvipcpsi*0.10
                  pcs(nvips) = 0
                  wght(nvips) = 0
                  cps(nvips) = 0
                  write(recs(i)((131+(nvips-1)*49):(179+(nvips-1)*49)),'(a5,f12.2,f12.0,2f10.0)')
&                    vips(nvips),rev(nvips),pcs(nvips),wght(nvips),cps(nvips)
               end if
               if (tvipcpso.gt.0) then
                  nvips = nvips + 1
                  vipused(nvips) = .true.
                  vips(nvips) = '99999'
                  rev(nvips) = tvipcpso*0.10
                  pcs(nvips) = 0
                  wght(nvips) = 0
                  cps(nvips) = 0
                  write(recs(i)((131+(nvips-1)*49):(179+(nvips-1)*49)),'(a5,f12.2,f12.0,2f10.0)')
&                    vips(nvips),rev(nvips),pcs(nvips),wght(nvips),cps(nvips)
               end if
            end if
         end if
         if (fixed.eq.0) then
            if (tpvipcps.gt.0) then

               if (((totr-tviprev)/tpvipcps.ge.0.095).and.((totr-tviprev)/tpvipcps.le.0.105)) then
                  fixed = 1
                  if (tpvipcpsi.gt.0) then
                     nvips = nvips + 1
                     vipused(nvips) = .true.
                     vips(nvips) = '99998'
                     rev(nvips) = tpvipcpsi*0.10
                     pcs(nvips) = 0
                     wght(nvips) = 0
                     cps(nvips) = 0
                     write(recs(i)((131+(nvips-1)*49):(179+(nvips-1)*49)),'(a5,f12.2,f12.0,2f10.0)')
&                       vips(nvips),rev(nvips),pcs(nvips),wght(nvips),cps(nvips)
                  end if
                  if (tpvipcpso.gt.0) then
                     nvips = nvips + 1
                     vipused(nvips) = .true.
                     vips(nvips) = '99999'
                     rev(nvips) = tpvipcpso*0.10
                     pcs(nvips) = 0
                     wght(nvips) = 0
                     cps(nvips) = 0
                     write(recs(i)((131+(nvips-1)*49):(179+(nvips-1)*49)),'(a5,f12.2,f12.0,2f10.0)')
&                       vips(nvips),rev(nvips),pcs(nvips),wght(nvips),cps(nvips)
                  end if
               end if
            end if
         end if

      write (recs(i)(128:130),'(i3)') nvips
```

```
               if ((fixed.eq.0).and.(intlflag.eq.0)) then
                  write(48,'(a)') recs(i)(1:130+(nvips*49))
                  write(49,'(a36,10f18.2)') recs(i)(1:36),totr,tviprev,totr-tviprev,totp,tvipcps,tpvipcps
     &               ,tvipcpsi,tvipcpso,tpvipcpsi,tpvipcpso
               end if
               if ((fixed.eq.1)) then
                  write(46,'(a)') recs(i)(1:130+(nvips*49))
                  write(47,'(a36,10f18.2)') recs(i)(1:36),totr,tviprev,totr-tviprev,totp,tvipcps,tpvipcps
     &               ,tvipcpsi,tvipcpso,tpvipcpsi,tpvipcpso
               end if
            end if
            write (recs(i)(128:130),'(i3)') nvips
            write (30,'(a)') recs(i)(1:130+(nvips*49))
            j = 0
            do ivip = 1, nchk
               if (vipused(ivip)) then
                  j = j + 1
               end if
            end do
            if (j.ne.(nvips-vipcount)) then
               print *, ' fatal error nvips neq count of vipused in good data '
            end if
            ngood = ngood + 1
            goodrev = goodrev + totr
            goodpcs = goodpcs + totp
            grevenue(iq,ifin) = grevenue(iq,ifin) + totr
         else                    ! write to bad file
            if (good.and.iq.eq.0) ierror = 23
            if (badvips) ierror = 24
            if (good.and.ifin.eq.0) then
               ierror = 25
               print *,' *** Bad Finance Number: ',recs(i)(1:6),' *** '
            end if
            if (.not.badvips) then
               write(40,'(a)') recs(i)(1:130+(nvips*49))
            else
               write(50,'(a)') recs(i)(1:130+(nvips*49))
            end if
            nbad(ierror) = nbad(ierror) + 1
            brev(ierror) = brev(ierror) + totr
            bpcs(ierror) = bpcs(ierror) + totp
            if (iq.gt.0.and.ifin.gt.0)
     &         brevenue(iq,ifin) = brevenue(iq,ifin) + totr
         end if
      end do

c     Move first record of next publication to beginning of array

      recs(1) = recs(irec)
      irec = 2
      lastpub = pub
   end do
   close (30)
   close (40)

c  Write out array of total revenue
   open(50,file='bin2nd.data.revenue')
51 format(a6,4f14.2)
52 format('Bad Transaction Revenue:')

   do ifin = 1, nfin
      write (50,51) fins(ifin), (grevenue(iq,ifin),iq=1,nq)
   end do
   write (50,52)
   do ifin = 1, nfin
      write (50,51) fins(ifin), (brevenue(iq,ifin),iq=1,nq)
   end do



c  Write out report statisitics

   print *,' read exit code = ',ier
300 format(' Good transactions  :  ',i10)
301 format(' Good trans. revenue: ',f12.0)
302 format(' Good trans. pieces : ',f12.0)
303 format(' Error number ',i2,', trans: ',i6,', rev: ',f12.0,', pcs: ',f12.0)
   write (*,300) ngood
```

```
      write (*,301) goodrev
      write (*,302) goodpcs
      print *,'number of commingled err is :',ncommerror
      do i = 1, nerror
         if (nbad(i).gt.0) then
            write (*,303) i, nbad(i), brev(i), bpcs(i)
         end if
      end do

      end

C -----------------------------------------------------------------------
        function    clind(code)

        character*1 code
        character*1 clind

        if (code.eq.'R') then
           clind = '9'
        else if (code.eq.'S') then
           clind = '7'
        else if (code.eq.'A') then
           clind = '6'
        else if (code.eq.'C') then
           clind = '8'
        else if (code.eq.'Q') then
           clind = '9'
        else if (code.eq.'N') then
           clind = 'X'
        else
           clind = 'Z'
        end if

        return
        end

C -----------------------------------------------------------------------
        function    classind(code)

        character*1 code
        integer*4   classind

        if (code.eq.'9') then
           classind = 4
        else if (code.eq.'6') then
           classind = 1
        else if (code.eq.'7') then
           classind = 2
        else if (code.eq.'8') then
           classind = 3
        else
           classind = 4
        end if

        return
        end

C ---------------------------------------------------
      function qind(date)

      integer*4 qind
      character*8 date

      if (date.ge.'19990911'.and.date.le.'19991203') then
         qind = 1
      else if (date.ge.'19991204'.and.date.le.'20000225') then
         qind = 2
      else if (date.ge.'20000226'.and.date.le.'20000519') then
         qind = 3
      else if (date.ge.'20000520'.and.date.le.'20000908') then
         qind = 4
      else
         qind = 0
         print *,'bad date',date
      end if

      return
      end
C -----------------------------------------------------
```

```
      subroutine  checktrn(itype2,good,mflag,wpc,icls,ierror,igood,edper,date)

C     Mike McGrane   1/5/94
C        modified: 5/94 - 10/94
C           - 9/94 added total checks
C           - 11/96 - modified for fy 1996 and reclass
C
C     Subroutine checks transaction for internal consistency and
C     also checks for monthly statement

      implicit none
      integer*4    nchk, ngcnt, nq, nfin
      real*8       revtol, tottol
      parameter    (nchk = 142)
      parameter    (ngcnt = 7)
      parameter    (nq = 4)
      parameter    (nfin = 1819)
      parameter    (revtol = 0.01) ! tolerance for diff in vip rev and calc. rev
      parameter    (tottol = 0.05) ! tol. for diff between sum of vips and total

      real*8       rev(nchk)
      real*8       pcs(nchk)
      real*8       wght(nchk)
      real*8       cps(nchk)
      real*8       chkrate(nchk,nq)
      real*8       rate1(nchk,nq)
      real*8       rate2(nchk,nq)
      real*8       rate3(nchk,nq)
      real*8       edper
      real*8       rtot, rate, totr, totp

      integer*4    chkcon(nchk,5)
      integer*4    gcnt(ngcnt,nchk)
      integer*4    errnum(nchk)
      character*4  vipc(nchk)
      character*5  vips(nchk)
      real*8       wpc, mratio, lmratio, chkrev, chkr1, chkr2
      real*8       zco, sco, po, zwo, nw, pdiff, newzco
      real*8       zci, sci, pi, zwi, adwtcpy
      real*8       chkwght, szci, szco, upper, lower, newcps
      real*8       testdiff

      integer*4    i,j,k,icls, searchc
      integer*4    imratio, igood
      integer*4    ierror, iv1, iv2, nbad, ibad
      character*8  date
      character*6  fins(nfin),finno
      character*13 tr_num
      character*15 transno

      logical      good, mflag, diff, regular, mdiff, itype2, debug
      logical      vipused(nchk)

C COMMON BLOCK FOR CONSISTENCY CHECK

      common  /check/ rev, pcs, wght, cps, totp, totr, chkcon, vipc, gcnt,
     &   rate1, rate2, rate3, vipused,vips,tr_num,finno

      debug =     .true.
      lmratio =  0.
      good        .true.
      mflag =     .false.
      itype2 =    .false.
      regular =   .false.
      ierror =    0
      igood =     0
      zci =       0.
      sci =       0.
      pi =        0.
      zwi =       0.
      nw =        0.
      zco =       0.
      sco =       0.
      po =        0.
      zwo =       0.
      szci =      0.
      szco =      0.
      edper =     0.
      rtot =      0.
      testdiff = 0.
```

```
      Zero error number array

      do i = 1, nchk
         errnum(i) = 0
      end do

      ************************************************************
      Set chkrate array to appropriate time period
      ************************************************************

      if (date.le.'19981004') then ! pre-rate change
         do j = 1, nq
            do i = 1, nchk
               chkrate(i,j) = rate1(i,j)
            end do
         end do
      else
         do j = 1, nq
            do i = 1, nchk
               chkrate(i,j) = rate2(i,j)
            end do
         end do
      end if


      ************************************************************
      Separate pre-reclass barcoded vips into letters and flats
      We are unable to separate x141x (in-county basic) because
      the rates are the same for letters and flats

      This edit is not necessary post-reclass because letter and
      flat automation have separate vip codes in regular rate
      ************************************************************

      do i = 1, nchk
         if ((vipc(i).eq.'1440'.or.vipc(i).eq.'1470'.or.vipc(i).eq.'2180'.or.
     &       vipc(i).eq.'2210'.or.vipc(i).eq.'2240').and.vipused(i)) then
            chkr1 = pcs(i) * chkrate(i+1,icls) ! check letter rate  xxxx4
            chkr2 = pcs(i) * chkrate(i+2,icls) ! and flat rate      xxxx5
            if (diff(rev(i),chkr1).and.diff(rev(i),chkr2)) then
               chkr1 = pcs(i) * rate3(i+1,icls) ! check letter rate  xxxx4
               chkr2 = pcs(i) * rate3(i+2,icls) ! and flat rate      xxxx5
               if (diff(rev(i),chkr1).and.diff(rev(i),chkr2)) then
                  chkr1 = pcs(i) * rate2(i+1,icls) ! check letter rate  xxxx4
                  chkr2 = pcs(i) * rate2(i+2,icls) ! and flat rate      xxxx5
                  if (diff(rev(i),chkr1).and.diff(rev(i),chkr2)) then
                     chkr1 = pcs(i) * rate1(i+1,icls) ! check letter rate  xxxx4
                     chkr2 = pcs(i) * rate1(i+2,icls) ! and flat rate      xxxx5
                  else
                     errnum(i) = 22
                     ierror = 22
                     good = .false.
                  end if
               end if
            end if
            if (.not.diff(rev(i),chkr1)) then
               rev(i+1) = rev(i) ! move data to barcoded letter vip
               pcs(i+1) = pcs(i)
               cps(i+1) = cps(i)
               vipused(i+1) = .true.
               vips(i+1)=vips(i)(1:1)//vipc(i+1)
               rev(i) = 0.
               pcs(i) = 0.
               cps(i) = 0.
               vipused(i) = .false.
               vips(i)='     '
            else if (.not.diff(rev(i),chkr2)) then
               rev(i+2) = rev(i) ! move data to barcoded flat vip
               pcs(i+2) = pcs(i)
               cps(i+2) = cps(i)
               vipused(i+2) = .true.
               vips(i+2)=vips(i)(1:1)//vipc(i+2)
               rev(i) = 0.
               pcs(i) = 0.
               cps(i) = 0.
               vips(i)='     '
               vipused(i) = .false.
            end if
         end if
```

```
      end do

C     ************************************************************
C         Loop through all vip codes in transactions - check
C         rates at each vip code and sum check totals for
C         comparison to check totals
C     ************************************************************

      i = 1
      do while((i.le.nchk).and.good)
         if(vipc(i)(4:4).ne.'9') then

C     ************************************************************
C      Sum various information for comparison to subtotals
C     ************************************************************

            rtot = rtot + chkcon(i,2)*rev(i)
            if(vipc(i)(1:1).ne.'1') then
               if (vipc(i).ne.'2120') then
                  zwo = zwo + chkcon(i,4)*wght(i) ! OUTSIDE COUNTY ADVERTISING WEIGHT
               else
                  nw = wght(i)  ! NON-ADVERTISING WEIGHT
               end if
               zco = zco + chkcon(i,4)*cps(i) ! OUTSIDE COUNTY ZONE COPIES
               sco = sco + chkcon(i,3)*cps(i) ! OUTSIDE COUNTY SORTATION COPIES
               po  = po  + chkcon(i,1)*pcs(i) ! OUTSIDE COUNTY SORTATION PIECES
            else
               zci = zci + chkcon(i,4)*cps(i) ! IN-COUNTY COPIES
               zwi = zwi + chkcon(i,4)*wght(i) ! IN-COUNTY WEIGHT
               sci = sci + chkcon(i,3)*cps(i) ! IN-COUNTY SORTATION COPIES
               pi =  pi +  chkcon(i,1)*pcs(i) ! IN-COUNTY SORTATION PIECES
            end if

C     ************************************************************
C       Check that revenue matches pieces * piece rate for
C       piece rate vips, if not check for type 2 monthlies
C     ************************************************************

            if ((chkcon(i,1).eq.1).and.(pcs(i).gt.0.)) then
               chkrev = pcs(i) * chkrate(i,icls)
               if(dabs(rev(i)-chkrev).gt.revtol) then ! rate does not match revenue
                  if (date.le.'19981003') then ! pre-reclass, check post reclass rates
                     chkr1 = pcs(i) * rate2(i,icls)
                     if (dabs(rev(i)-chkr1).le.revtol) then
                        chkrev = chkr1
                     else
                        chkr1 = pcs(i) * rate3(i,icls)
                        if (dabs(rev(i) chkr1).le.revtol) chkrev = chkr1
                     end if
                  else          ! post-reclass, check pre reclass rates
                     chkr1 = pcs(i) * rate1(i,icls)
                     if (dabs(rev(i)-chkr1).le.revtol) then
                        chkrev = chkr1
                     else
                        chkr1 = pcs(i) * rate3(i,icls)
                        if (dabs(rev(i)-chkr1).le.revtol) chkrev = chkr1
                     end if
                  end if
               end if
               if(dabs(chkrev-rev(i)).gt.revtol) then ! CHECK FOR TYPE 2 MONTHLY
                  if (rev(i).gt.chkrev) then ! A type 2 monthly has the
                     if (chkrev.eq.0.0) then
                        mratio=rev(i)
                     else
                        mratio =  rev(i) / chkrev ! pieces for a single issue recorded
                     end if
                     imratio = nint(mratio) ! but revenue for all issues in the month
                     if (.not.diff(mratio,dble(imratio)).and.mratio.le.31.0) then
                        mflag = .true.
                        if (lmratio.eq.0) then
                           if (regular) then
                              good = .false. ! NON MONTHLY RECORD IN SAME TRANSACTION
                              errnum(i) = 1
                              ierror = 1
                           else
                              lmratio = mratio
                           end if
                        else if (diff(lmratio,mratio)) then
                           good = .false. ! DIFFERENT MONTHLY RATIOS IN SAME TRANSACTION
```

```
                        errnum(i) = 2
                        ierror = 2
                     end if
                  else
                     good = .false. ! RATE CHECK FAILS, NOT A MONTHLY
                     errnum(i) = 3
                     ierror = 3
                  end if
               else
                  good = .false. ! RATE CHECK FAILS, NOT A MONTHLY
                  errnum(i) = 4
                  ierror = 4
               end if
            else
               if (mflag) then
                  good = .false. ! MIXED MONTHLY AND NON-MONTHLY IN SAME TRANSACTION
                  errnum(i) = 5
                  ierror = 5
               end if
               regular = .true.
            end if

C     ************************************************************
C       Check that revenue matches weight * pound rate for
C       pound rate vips
C     ************************************************************

         else if((chkcon(i,4).eq.1).and.(wght(i).gt.0.)) then ! CHECK WEIGHT RATES
            if((vipc(i).ne.'2110').and.(vipc(i).ne.'4111')) then
               chkrev = wght(i) * chkrate(i,icls)
               if(dabs(rev(i)-chkrev).gt.revtol) then ! rate does not match revenue
                  if (date.le.'19971004') then ! pre-reclass, check post reclass rates
                     chkr1 = wght(i) * rate2(i,icls)
                     if (dabs(rev(i)-chkr1).le.revtol) then
                        chkrev = chkr1
                     else
                        chkr1 = wght(i) * rate3(i,icls)
                        if (dabs(rev(i)-chkr1).le.revtol) chkrev = chkr1
                     end if
                  else       ! post-reclass, check pre reclass rates
                     chkr1 = wght(i) * rate1(i,icls)
                     if (dabs(rev(i)-chkr1).le.revtol) then
                        chkrev = chkr1
                     else
                        chkr1 = wght(i) * rate3(i,icls)
                        if (dabs(rev(i)-chkr1).le.revtol) chkrev = chkr1
                     end if
                  end if
               end if
               if(dabs(chkrev-rev(i)).gt.revtol) then
                  good = .false.
                  errnum(i) = 6
                  ierror = 6 ! WEIGHT RATE CHECK FAILED
               endif
            end if
         end if

C     ************************************************************
C        Check that sum of parts equals subtotal line revenue
C     ************************************************************

         else if(vipc(i)(4:4).eq.'9') then
            if(vipc(i).eq.'1389'.or.vipc(i).eq.'1539'.or.vipc(i).eq.'2159'
     &         .or.vipc(i).eq.'2339') then
               testdiff = rev(i) - rtot
               if(dabs(testdiff).gt.tottol) then
                  good = .false.
                  errnum(i) = 7
                  ierror = 7    ! FAILED TOTAL CHECK
               end if
            end if
            rtot = 0.0
         end if
         i = i + 1
      end do

C     ************************************************************
C      Rate comparison of individual vips completed
C     ************************************************************
```

```
C     ************************************************************
C       Check that zone weight lines also have copies -
C       estimate number of copies if necessary
C     ************************************************************

      if(good) then
         do i = 1, nchk
            if((vipc(i).eq.'2010').or.(vipc(i).eq.'2020').or.
     &         (vipc(i).eq.'2030').or.(vipc(i).eq.'2040').or.
     &         (vipc(i).eq.'2050').or.(vipc(i).eq.'2060').or.
     &         (vipc(i).eq.'2070').or.(vipc(i).eq.'2080').or.
     &         (vipc(i).eq.'2090')) then
               if((wght(i).gt.0.0).and.(cps(i).eq.0.0)) then ! VIP IS MISSING COPIES
                  igood = 5
                  gcnt(igood,i) = gcnt(igood,i) + 1
                  if (sco.eq.0) then
                     cps(i) = dnint(wght(i) / zwo )
                  else
                     cps(i) = dnint(wght(i) / (zwo / sco))
                  end if
                  zco = zco + cps(i)
               end if
            else if((vipc(i).eq.'4011').or.(vipc(i).eq.'4021').or.
     &              (vipc(i).eq.'4031').or.(vipc(i).eq.'4041').or.
     &              (vipc(i).eq.'4051').or.(vipc(i).eq.'4061').or.
     &              (vipc(i).eq.'4071').or.(vipc(i).eq.'4081').or.
     &              (vipc(i).eq.'4091')) then
               if((wght(i).gt.0.0).and.(cps(i).eq.0.0)) then ! COMMINGLED VIP IS MISSING COPIES
                  igood = 6
                  gcnt(igood,i) = gcnt(igood,i) + 1
                  if (sco.eq.0.0) then
                     cps(i) =  dnint(wght(i) / zwo )
                  else
                     cps(i) =  dnint(wght(i) / (zwo / sco))
                  end if
                  zco = zco + cps(i)
               end if
            end if
         end do
      end if

C     ********************************************************************
C       Check that discounts were not given for more than
C       the total number of pieces in the mailing
C     ********************************************************************

      if(good) then
         j = searchc(vipc,nchk,'2300')
         if(rev(j).gt.0.0) then
            if(pcs(j).gt.po) then
               errnum(j) = 8
               ierror = 8        ! THE DELIVERY UNIT DISCOUNT WAS GIVEN ON AN AMOUNT
               good = .false.    ! OF PIECES THAT EXCEEDS TOTAL OUTSIDE COUNTY
            end if
         end if
         j = searchc(vipc,nchk,'2310')
         if(rev(j).gt.0.0) then
            if(pcs(j).gt.po) then
               if (pcs(j).le.sco) then ! A common error is SCF discount based on copies
                  pcs(j) = po   ! reset to number of outside county pieces
                  rev(j) = po * chkrate( j, icls) ! set revenue (credit) to pieces time rate
               else
                  errnum(j) = 9
                  ierror = 9    !THE SCF DISCOUNT WAS GIVEN ON AN AMOUNT OF PIECES
                  good = .false. !THAT EXCEEDS TOTAL OUTSIDE COUNTY SORTATION copies
               end if
            end if
         end if
      end if

C     ************************************************************
C       Adjust pieces and copies for type 2 monthly transaction
C     ************************************************************


      if (good) then
         if (mflag) then        ! TYPE 2 MONTHLY STATEMENT TRANSACTION
            if (zwo.gt.0.) then
               if (diff(zci+zco),(sci+sco)) then
```

```
            if (dabs((zci+zco)-(sci+sco))/(zci+zco)).gt..10) then ! ALLOW SOME DIFFERENCE
               good = .false. ! ZONE COPIES SHOULD EQUAL SORT COPIES
               ierror = 10
            end if
         end if
         if ((pi+po).gt.(sci+sco)) then
            good = .false. ! PIECES SHOULD BE LESS THAN OR EQUAL TO COPIES
            ierror = 11
         end if
         if (good) then
            itype2 = .true.
            sci = sci * mratio
            sco = sco * mratio
            if ((sci+sco).eq.0.0) then
               wpc = (zwi + zwo + nw)
            else
               wpc = (zwi + zwo + nw) / (sci + sco)
            end if
            do i = 1, nchk ! APPLY MRATIO TO PIECES AND COPIES
               if (chkcon(i,1).eq.1) then
                  pcs(i) = pcs(i) * mratio
                  cps(i) = cps(i) * mratio
               else if (chkcon(i,4).eq.1) then
                  cps(i) = cps(i) * mratio
               else if(chkcon(i,5).eq.1) then
                  cps(i) = cps(i) * mratio
               end if
            end do
         end if
      else
         if ((pi+po).gt.(sci+sco)) then
            good = .false.
            ierror = 12   ! SORTATION PIECES ARE GREATER THAN SORTATION COPIES
         end if
         if (zco.gt.0.) then
            good = .false.
            ierror = 13   ! ZONE WEIGHT IS ZERO But Zone COPIES ARE GREATER THAN ZERO
         end if
         if (good) then
            sci = sci * mratio
            sco = sco * mratio
            if ((sci+sco).eq.0.0) then
               wpc = (zwi + nw)
            else
               wpc = (zwi + nw) / (sci + sco)
            end if
            do i = 1, nchk ! APPLY MRATIO
               if (chkcon(i,1).eq.1) then
                  pcs(i) = pcs(i) * mratio
                  cps(i) = cps(i) * mratio
               end if
            end do
         end if
      end if
   else

C     ***************************************************************
C       Check whether transaction is normal or type 1 or 3 monthly
C          Type 3 monthly has pieces = issues pieces * number issues
C          but sort copies = number of copies for 1 issue
C     ***************************************************************

         if (zwo.gt.0) then
            if (.not.diff((zci+zco),(sci+sco))) then ! sort copies = zone copies, either normal or type 3
               if ((pi+po).gt.(sci+sco)) then ! possible type 3 monthly pieces gt sort copies
                  if ((sci+sco).eq.0.0) then
                     mratio = (pi+po)
                  else
                     mratio = (pi+po) / (sci+sco)
                  end if
                  imratio = nint(mratio)
                  if (mratio.le.31.0) then
                     if ((mratio-imratio) .le. 0.05) then
                        mflag = .true. ! TYPE 3 MONTHLY
                        sci = sci * mratio
                        sco = sco * mratio
                        if ((sci+sco).eq.0.0) then
                           wpc = (zwi + zwo + nw)
                        else
                           wpc = (zwi + zwo + nw) / (sci + sco)
```

```
                        end if
                        do i = 1, nchk ! APPLY MRATIO TO COPIES
                           if (chkcon(i,4).eq.1) then
                              cps(i) = cps(i) * mratio
                           else if (chkcon(i,1).eq.1) then
                              cps(i) = cps(i) * mratio
                           else if(chkcon(i,5).eq.1) then
                              cps(i) = cps(i) * mratio
                           end if
                        end do
                     else
                        good = .false.
                        ierror = 14 ! BAD TYPE 3 MRATIO
                     end if
                  end if
               end if
            else

      ***************************************************************
         Check for type 1 monthly
             Type 1 monthly has sort copies = copies * number issues
             but zone copies = copies for single issue
      ***************************************************************
               if ((zci+zco).eq.0.0) then
                  mratio = (sci + sco)
               else
                  mratio = (sci + sco) / (zci + zco)
               end if

               imratio = nint(mratio)

               if ((mratio.gt.1.1).and.(mratio.le.31.0)) then
                  if (dabs(mratio-imratio) .le. 0.05) then
                     mflag = .true.
                     if ((pi + po).gt.(sci + sco)) then
                        good = .false.
                        ierror = 15 ! SORTATION PIECES ARE GREATER THAN SORTATION COPIES
                     end if
                     if ((sci+sco).eq.0.0) then
                        wpc = (zwi + zwo + nw)
                     else
                        wpc = (zwi + zwo + nw) / (sci + sco)
                     end if
                     do i = 1, nchk ! APPLY MRATIO TO COPIES
                        if (chkcon(i,4).eq.1) then
                           cps(i) = cps(i) * mratio
                        else if(chkcon(i,5).eq.1) then
                           cps(i) = cps(i) * mratio
                        end if
                     end do
                  else
                      print *,'martio: ',mratio,'and imratio: ',imratio
                      print *,'sci',sci,'sco',sco,'zci',zci,'zco',zco,'mytype',itype2
                     ! TRY RECASTING COPIES
      ***************************************************************
        mrm - 11/96  I deleted a section here which recomputed
                     zone copies and zone weight by setting any
                     zone that had a weight of 1.0 pound to 0.5 pounds
                     and re-estimating copies for the monthly check
                     If a lot of error 16 is encountered we may want
                     to re-insert this.

        mrm - 12/96  The problem seems to be ?????
      ***************************************************************
                     if ((sco.gt.0.).and.(zwo.gt.0.).and.(sci.eq.zci)) then
                        adwtcpy = zwo / sco
                        iv1 = searchc(vipc,nchk,'2010')
                        iv2 = searchc(vipc,nchk,'2090')
                        nbad = 0
                        ibad = 0
                        do i = iv1, iv2
                           if (wght(i).gt.0) then
                              cps(i) = wght(i) / adwtcpy
                           end if
                        end do
                     else
                        good = .false.
                        ierror = 16 ! problem is in both inside and outside
                     end if
                  end if
```

```
            end if
          end if
        else

      ************************************************************
        Outside county weight is zero - pub has no (<10%) advertising

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

          if((nw.gt.0).and.(zwi.eq.0)) then ! no in-county mail
            if(po.gt.sco) then
              good = .false.
              ierror = 18 ! SORTATION PIECES ARE GREATER THAN SORTATION COPIES
            else
              if (sco.ne.0.0) then
                wpc = nw / sco
              else
                wpc=nw
              end if
            end if
          else              ! in-county mail - check copies
            if(.not.diff(sci,zci)) then ! REGULAR
              if (pi.gt.sci) then
                good = .false.
                ierror = 19 ! IN-COUNTY SORTATION PIECES ARE GREATER THAN COPIES
              end if
            else            ! possible type 1 monthly
              if (zci.eq.0.0) then
                mratio=sci
              else
                mratio = sci / zci
              end if
              imratio = nint(mratio)
              if(.not.mdiff(mratio,dble(imratio)).and.mratio.le.31) then
                mflag = .true.
                if(pi.gt.sci) then
                  good = .false.
                  ierror = 20 ! IN-COUNTY SORTATION PIECES ARE GREATER THAN COPIES
                end if
                if (sci.eq.0.0) then
                  wpc=zwi
                else
                  wpc = zwi / sci
                end if
                do i = 1, nchk ! APPLY MRATIO
                  if(chkcon(i,4).eq.1) then
                    cps(i) = cps(i) * mratio
                  else if(chkcon(i,5).eq.1) then
                    cps(i) = cps(i) * mratio
                  end if
                end do
              else
                good = .false.
                ierror = 21 ! BAD IN-COUNTY MRATIO
              end if
            end if
          end if
        end if
      end if

      if(good) then             ! CALCULATE EDITORIAL PERCENTAGE FOR QUALIFIER/NONQUALIFIER
        if ((zwo+nw).gt.0) then
          edper = (nw / (zwo + nw)) *100
        else
        end if
      end if

      if(.not.mflag.and.good) then ! CALCULATE WEIGHT PER COPY
        if((sci+sco).gt.0.0) then
          wpc = (zwi+zwo+nw) / (sci+sco)
        else
          wpc = 0.0
        end if
      end if

900   format(' Bad transaction, last error = ',i3,' Date = ',a8)
910   format(' Vip     Rev     Pcs     Wght    Cps     Rate     Chkrate   Error # ')
920   format(i1,a4,f10.2,3f8.0,2f10.4,i6)
930   format('finance - ',a6,' trans number = ',a13)
```

```
      if (debug.and..not.good) then ! print out diagnostics for transaction
         write (*,801) ierror, date
         write (*,804) finno,tr_num
         write (*,802)
         do i   1, nchk
            if (rev(i).gt.0.or.pcs(i).gt.0.or.wght(i).gt.0.or.cps(i).gt.0) then
               if (chkcon(i,1).eq.1) then
                  if (pcs(i).gt.0) then
                     chkr1 = rev(i) / pcs(i)
                  else
                     chkr1 = 0.
                  end if
               else if (chkcon(i,4).eq.1) then
                  if (wght(i).gt.0) then
                     chkr1 = rev(i) / wght(i)
                  else
                     chkr1 = 0.
                  end if
               else if (chkcon(i,2).eq.-1) then
                  if (pcs(i).gt.0) then
                     chkr1 = rev(i) / pcs(i)
                  else
                     chkr1 = 0.
                  end if
               else
                  chkr1 = 0.
               end if
               write (*,803) icls+5, vipc(i), rev(i), pcs(i), wght(i),
     &            cps(i), chkr1, chkrate(i,icls), errnum(i)
            end if
         end do
      end if

      return
      end
C     ----- --- ------ -------        ---------------------------------------
      function diff(v1,v2)

      logical diff
      real*8    v1, v2
      real*8    tol
      parameter (tol = 0.01)

      if (dabs(v1-v2).gt.tol) then
        diff = .true.
      else
        diff = .false.
      end if

      return
      end
C     ---------------------------------------------------------------------
      function mdiff(v1,v2)

      logical mdiff
      real*8    v1, v2
      real*8    mtol
      parameter (mtol = 0.1)

      if (dabs(v1-v2).gt.mtol) then
        mdiff = .true.
      else
        mdiff = .false.
      end if

      return
      end
C     ------------ - - --  -------- ----------------------------------------
```

```
      function jul_to_txt(juldate)

      AUTHOR: Mary E. Larson
      DATE: May 6, 1985
      PROJECT: System Library
      PURPOSE: Given date - returns index for date. Jan 1, 1900 index = 1

      Input parameters :
          INTEGER*4  YEAR  - 4 digit year
          INTEGER*4  MONTH - 2 digit month
          INTEGER*4  DAY   - 2 digit day
      Value returned : INTEGER*4 INDEX for date.
      Algorithm :

      IMPLICIT NONE

      INTEGER*4       YEAR
      INTEGER*4       MONTH
      INTEGER*4       DAY
      INTEGER*4       days(12,2)/31,28,31,30,31,30,31,31,30,31,30,31,
     +                          31,29,31,30,31,30,31,31,30,31,30,31/
      INTEGER*4       days_in_year(0:1)/366,365/
      INTEGER*4       BASE_YEAR
      INTEGER*4       NUM_YEARS
      INTEGER*4       MO, YR          ! Loop control
      INTEGER*4       SUM_INDEX
      integer*4       julday

      character*5     juldate
      character*6     jul_to_txt

      logical         DEBUG/.true./
      logical         DEBUG/.false./
      logical         leap, leapyear

      PARAMETER (BASE_YEAR = 1900)

      if (DEBUG) print *,' Enter Function get-index '
      if (DEBUG) print *,' Date = ',month,'-',day,'-',year

      read(juldate,'(i2,i3)') year, julday


      Get month and day of year

      sum_index   0
      yr = year + base_year

      leapyear = leap(yr)

      mo = 0
      do while (sum_index.lt.julday)
         mo = mo + 1
         if (leapyear) then
            sum_index = sum_index + days(mo,2)
         else
            sum_index = sum_index + days(mo,1)
         end if
      end do

      day = sum_index - julday
      if (leapyear) then
         day   days(mo,2) - day
      else
         day = days(mo,1) - day
      end if

      write (jul_to_txt,'(i2.2,i2.2,i2.2)') year, mo, day

      return
      end
```

```
      LOGICAL FUNCTION LEAP(YEAR)

      AUTHOR: Mary E. Larson
      DATE: August 15, 1985
      PROJECT: System Library
      PURPOSE: Determine whether a 4 digit year is a leap year or not
      Input parameters :
           INTEGER*4  YEAR (4 DIGIT)
      Value Returned : TRUE - If year is a leap year
                       FALSE - If year in not a leap year
      Algorithm : Use mod to determine if year is a leap year
                  Take Leap Centuries into account also.

      IMPLICIT NONE

      integer*4       year, four, oneh, fourh
      PARAMETER (FOUR = 4)
      PARAMETER (ONEH = 100)
      PARAMETER (FOURH = 400)

      logical         DEBUG/.true./
      logical         DEBUG/.false./

      if (DEBUG) print *,' Enter Function Leap'

      leap = ((MOD(year,FOUR).eq.0).and.(MOD(year,ONEH).ne.0)) .or.
     +            (MOD(year,FOURH).eq.0))

      if (DEBUG) print *,' Exit from Function Leap '

      return
      end
```

```
      program bin2nd2_new

C     Mike McGrane 11-15-96

C     This program reads second-class PERMIT transactions
C     and checks them for internal consistency
C     It writes three output files in binary number format:
C     good transactions, bad transactions, and good trans.
C     revenue by finance number and quarter.
C
C     last Modified to handle 2000 data for vips after 1999 rate change
C

      implicit none

      integer*4 nvip, nfin, nq, nchk, nerror, ngcnt, nrec, fixed,intlflag

      parameter (nvip = 528)     ! number of vips
      parameter (nfin = 1904)    ! number of PERMIT finance number
      parameter (nq = 4)         ! number of quarters
      parameter (nchk = 142)     ! number of 4-digit VIP costs
      parameter (nerror = 25)    ! number of error codes
      parameter (ngcnt = 7)      ! number of transaction repair codes
      parameter (nrec = 20000)   ! number of records in recs array


      real*8  rev(nchk), cps(nchk), wght(nchk), pcs(nchk), twght(nchk)
      real*8  brev(nerror), bpcs(nerror), bwght(nerror)
      real*8  r, p, w, c, edper, zwo, nw, wpc, totr, totp, totw
      real*8  rate1(nchk,4), rate2(nchk,4), rate3(nchk,4)
      real*8  grevenue(nq,nfin), brevenue(nq,nfin)
      real*8  goodrev, goodpcs,tviprev,viprevchk,tvipcps,tpvipcps
      real*8  tvipcpsi,tvipcpso,tpvipcpsi,tpvipcpso

      integer*4 ichk, ivip, ierror, iq, ifin, i, j, k, ier, irec, nvips, igood, icls
      integer*4 chkcon(nchk,5), gcnt(ngcnt,nchk)
      integer*4 nbad(nerror), ngood
      integer*4 qind, searchc
      integer*4 ncommerror, rlength, classind, ier2,vipcount

      character*4    vipc(nchk)
      character*5    vips(nchk), vip
      character*4246 recs(nrec)
      character*6    fins(nfin), fin, finno
      character*6    pub, lastpub
      character*8    tdate, date, jul_to_txt
      character*1    ratetype, cratetype
      character*13   tr_num
      character*15   transno
      character*130  header
      character*49   tail(nchk)
      character*4    clength
      character*1    clind

      logical vipused(nchk)

C     common block for passing data to checking subroutine

      common /check/ rev, pcs, wght, cps, totp, totr, chkcon, vipc, gcnt, rate1,
     &   rate2, rate3, vipused,vips,tr_num,finno

      logical first, itype2, mflag, good, badvips

C     Initialize values and arrays
      ncommerror = 0
      ngood = 0
      do i = 1, nerror
         brev(i) = 0.
         bpcs(i) = 0.
         bwght(i) = 0.
         nbad(i) = 0
      end do
      do ifin = 1, nfin
         do iq = 1, nq
            brevenue(iq,ifin) = 0.
            grevenue(iq,ifin) = 0.
         end do
      end do
```

```
      print *,'starting program'

C READ CONTROL FILE

      open(15,file='seccon.971005.981003')
150   format(6x,a4,3i3,i2,i3,3x,4f9.5)
      open(16,file='seccon.981004.990109')
160   format(27x,4f9.5)
      open(17,file='seccon.950101.950931')
      open(18,file='seccon.990110.present')

      do i = 1, nchk
         read(17,150)  vipc(i), (chkcon(i,j),j=1,5)
         read(15,160) (rate1(i,j),j=1,4)
         read(16,160) (rate2(i,j),j=1,4)
         read(18,160) (rate3(i,j),j=1,4)
      end do

      print*, 'seccon.dat has been read '
      close(15)
      close(16)

C     Read PERMIT finance numbers

      open(15,file='fin.map')
      do i=1,nfin
         read(15,'(a6)') fins(i)
      end do


C      read PERMIT records, one publication at a time, check for internal
C      consistency and write to output file

      open(20,file='postchange.data',recl=4500)

      open(30,file='newdata.data',recl=4500)
      open(48,file='revdiff.txt',recl=4500)
      open(49,file='revheaddiff.txt',recl=4500)
      open(46,file='fixeddiff.txt',recl=4500)
      open(47,file='fixedhead.txt',recl=4500)

      open(40,file='bin2nd.data.bad',recl=4500)
      open(50,file='bin2nd.data.badvips',recl=4500)



      ier = 0
      first = .true.
      irec = 1

      do while (ier.eq.0)
         read(20,'(a)',iostat=ier,end=100) recs(irec)
         if (first) then
            lastpub = recs(irec)(25:30)
            first = .false.
         end if
         pub = recs(irec)(25:30)
         do while (pub.eq.lastpub) ! read all the records for a single publication
            irec = irec + 1
            read(20,'(a)',iostat=ier,end=100) recs(irec)
            pub = recs(irec)(25:30)
         end do
100      if (ier.ne.0) then
            print *,' read exit of input file = ',ier
         end if

C        process records for the publication

         do i = 1, irec - 1
            header = recs(i)(1:130)
            ratetype = recs(i)(53:53)
            date = jul_to_txt(recs(i)(10:16))
            tdate=date
            iq = qind(tdate)
            ifin = searchc(fins,nfin,recs(i)(1:6))
            finno = fins(ifin)
            cratetype = ' '
            do ivip = 1, nchk
               rev(ivip) = 0.
```

```
            pcs(ivip) = 0.
            wght(ivip) = 0.
            cps(ivip) = 0.
            vips(ivip) = '      '
            vipused(ivip) = .false.
            tviprev=0.0
            tvipcpso=0.0
            tvipcpsi=0.0
            tpvipcpso=0.0
            tpvipcpsi=0.0
            tvipcps=0.0
            tpvipcps=0.0
            intlflag=0
         end do
         tr_num = recs(i)(10:22)
         if (recs(i)(83:83).eq.'Y') then ! Search for commingled transaction
            transno = recs(i)(84:98)
            transno = recs(i)(97:98)//recs(i)(84:96)
            do j = 1, irec - 1
               if (recs(j)(8:22).eq.transno) then
                  cratetype   recs(j)(53:53)
                  print *,'found this match'
               end if
            end do
            if (cratetype.eq.' ') then
               print *,' nonsubscriber commingled trans match failed ',lastpub,',',recs(i)(1:6),'-',recs(j)(8:22),'-',
&                 transno
               ncommerror = ncommerror + 1
            end if
         end if
         read(recs(i)(128:130), '(i3)', iostat=ier2) nvips
         vipcount=0
         if (ier2.eq.0) then
            badvips = .false.
            do ichk= 1, nvips
               read(recs(i)(131+(ichk-1)*49):(179+(ichk-1)*49)),'(a49)') tail(ichk)
            end do
            do j = 1, nvips
               read (recs(i)((131+(j-1)*49):(179+(j-1)*49)),'(a5,f12.2,f12.0,2f10.0)')
&                 vip, r,  p, w, c
               if (vip(1:1).eq.'X') then
                  if (cratetype.ne.' ') then
                     vip(1:1) = clind(cratetype) ! get rate type of parent transaction
                     ratetype = cratetype
                  else
                     vip(1:1) = '9' ! requestor
                  end if
               end if
               ivip = searchc(vipc,nchk,vip(2:5))
               if (ivip.gt.0) then
                  vipused(ivip) = .true.
                  vips(ivip)   vip
                  rev(ivip) = r
                  pcs(ivip) = p
                  wght(ivip) = w
                  cps(ivip) = c
                  tviprev=rev(ivip)*chkcon(ivip,2)+tviprev

                  if (ivip.le.28) then !IC
                     tvipcpsi=cps(ivip)*chkcon(ivip,5)+tvipcpsi ! Total Copies

                     if (((ivip.ge.64).and.(ivip.le.83)).or.((ivip.ge.8).and.(ivip.le.25))) then
                        tpvipcpsi = tpvipcpsi + cps(ivip)

                     end if
                  else       !OC
                     tvipcpso=cps(ivip)*chkcon(ivip,5)+tvipcpso ! Total Copies
                     if (((ivip.ge.64).and.(ivip.le.83)).or.((ivip.ge.8).and.(ivip.le.25))) then
                        tpvipcpso = tpvipcpso + cps(ivip)

                     end if
                  end if

                  if ((ivip.ge.90).and.(ivip.le.99)) then
                     intlflag = 1
                  end if

               else
                  print *,' vip ',vip,' not found in check list '
                  badvips = .true.
```

```
            end if
         end do
         read(recs(i)(104:127),'(f12.2,f12.0)') totr, totp

      else
         read(recs(i)(104:127),'(f12.2,f12.0)') totr, totp
         if (totp.gt.0) then
            print *,' error - no vip detail with totp > 0 ',totp,',',lastpub,',',recs(1)(1:6)
         end if
      end if
      icls = classind(ratetype)
      call checktrn(itype2,good,mflag,wpc,icls,ierror,igood,edper,tdate)
      if (good.and.iq.gt.0.and..not.badvips.and.ifin.gt.0) then



         rlength = 130 + (37*(nvips-vipcount)) + 4 ! length of variable format record
         j=0
         do ivip=1,nchk
            if (vipused(ivip)) then
               j=j+1
            end if
         end do
         write (recs(i)(128:130),'(i3)') j
         if (((tviprev-totr).gt..1).or.((tviprev-totr).lt.-.1)) then !Rev diff
            fixed = 0
            tvipcps = tvipcpsi + tvipcpso
            tpvipcps = tpvipcpsi + tpvipcpso
            if (tvipcps.gt.0) then
               if (((totr-tviprev)/tvipcps.ge.0.095).and.((totr-tviprev)/tvipcps.le.0.105)) then
                  fixed = 1
                  if (tvipcpsi.gt.0) then
                     nvips = nvips + 1
                     vipused(nvips) = .true.
                     vips(nvips) = '99998'
                     rev(nvips) = tvipcpsi*0.10
                     pcs(nvips) = 0
                     wght(nvips) = 0
                     cps(nvips) = 0
                     write(recs(i)((131+(nvips-1)*49):(179+(nvips-1)*49)),'(a5,f12.2,f12.0,2f10.0)')
&                       vips(nvips),rev(nvips),pcs(nvips),wght(nvips),cps(nvips)
                  end if
                  if (tvipcpso.gt.0) then
                     nvips = nvips + 1
                     vipused(nvips) = .true.
                     vips(nvips) = '99999'
                     rev(nvips) = tvipcpso*0.10
                     pcs(nvips) = 0
                     wght(nvips) = 0
                     cps(nvips) = 0
                     write(recs(i)((131+(nvips-1)*49):(179+(nvips-1)*49)),'(a5,f12.2,f12.0,2f10.0)')
&                       vips(nvips),rev(nvips),pcs(nvips),wght(nvips),cps(nvips)
                  end if
               end if
            end if
            if (fixed.eq.0) then
               if (tpvipcps.gt.0) then

                  if (((totr-tviprev)/tpvipcps.ge.0.095).and.((totr-tviprev)/tpvipcps.le.0.105)) then
                     fixed = 1
                     if (tpvipcpsi.gt.0) then
                        nvips = nvips + 1
                        vipused(nvips) = .true.
                        vips(nvips) = '99998'
                        rev(nvips) = tpvipcpsi*0.10
                        pcs(nvips) = 0
                        wght(nvips) = 0
                        cps(nvips) = 0
                        write(recs(i)((131+(nvips-1)*49):(179+(nvips-1)*49)),'(a5,f12.2,f12.0,2f10.0)')
&                          vips(nvips),rev(nvips),pcs(nvips),wght(nvips),cps(nvips)
                     end if
                     if (tpvipcpso.gt.0) then
                        nvips = nvips + 1
                        vipused(nvips) = .true.
                        vips(nvips) = '99999'
                        rev(nvips) = tpvipcpso*0.10
                        pcs(nvips) = 0
                        wght(nvips) = 0
                        cps(nvips) = 0
                        write(recs(i)((131+(nvips-1)*49):(179+(nvips-1)*49)),'(a5,f12.2,f12.0,2f10.0)')
```

```
&                          vips(nvips),rev(nvips),pcs(nvips),wght(nvips),cps(nvips)
                        end if
                     end if
                  end if
               end if

            write (recs(i)(128:130),'(i3)') nvips

               if ((fixed.eq.0).and.(intlflag.eq.0)) then
                  write(48,'(a)') recs(i)(1:130+(nvips*49))
                  write(49,'(a36,10f18.2)') recs(i)(1:36),totr,tviprev,totr-tviprev,totp,tvipcps,tpvipcps
&                    ,tvipcpsi,tvipcpso,tpvipcpsi,tpvipcpso
               end if
               if ((fixed.eq.1)) then
                  write(46,'(a)') recs(i)(1:130+(nvips*49))
                  write(47,'(a36,10f18.2)') recs(i)(1:36),totr,tviprev,totr-tviprev,totp,tvipcps,tpvipcps
&                    ,tvipcpsi,tvipcpso,tpvipcpsi,tpvipcpso
               end if
            end if

            write (recs(i)(128:130),'(i3)') nvips

            write (30,'(a)') recs(i)(1:130+(nvips*49))

            j = 0
            do ivip = 1, nchk
               if (vipused(ivip)) then
                  j = j + 1
               end if
            end do
            if (j.ne.(nvips-vipcount)) then
               print *, ' fatal error nvips neq count of vipused in good data '
            end if
            ngood = ngood + 1
            goodrev = goodrev + totr
            goodpcs = goodpcs + totp
            grevenue(iq,ifin) = grevenue(iq,ifin) + totr
         else                  ! write to bad file
            if (good.and.iq.eq.0) ierror = 23
            if (badvips) ierror = 24
            if (good.and.ifin.eq.0) then
               ierror = 25
               print *,' *** Bad Finance Number: ',recs(i)(1:6),' *** '
            end if
            if (.not.badvips) then
               write (40,'(a)') recs(i)(1:130+(nvips*49))
            else
               write (50,'(a)') recs(i)(1:130+(nvips*49))
            end if
            nbad(ierror) = nbad(ierror) + 1
            brev(ierror) = brev(ierror) + totr
            bpcs(ierror) = bpcs(ierror) + totp
            if ((iq.gt.0).and.(ifin.gt.0))
&              brevenue(iq,ifin) = brevenue(iq,ifin) + totr
         end if
      end do

      Move first record of next publication to beginning of array

      recs(1) = recs(irec)
      irec = 2
      lastpub = pub
   end do
   close (30)
   close (40)

   Write out array of total revenue
   open(50,file='bin2nd.data.revenue')
51 format(a6,4f14.2)
52 format('Bad Transaction Revenue:')

   do ifin = 1, nfin
      write (50,51) fins(ifin), (grevenue(iq,ifin),iq=1,nq)
   end do
   write (50,52)
   do ifin = 1, nfin
      write (50,51) fins(ifin), (brevenue(iq,ifin),iq=1,nq)
   end do
```

```
C     Write out report statisitics

      print *,' read exit code = ',ier
300   format(' Good transactions  :  ',i10)
301   format(' Good trans. revenue: ',f12.0)
302   format(' Good trans. pieces : ',f12.0)
303   format(' Error number ',i2,', trans: ',i6,', rev: ',f10.0,', pcs: ',f10.0)
      write (*,300) ngood
      write (*,301) goodrev
      write (*,302) goodpcs
      print *,'number of commingled err is :',ncommerror
      do i = 1, nerror
         if (nbad(i).gt.0) then
            write (*,303) i, nbad(i), brev(i), bpcs(i)
         end if
      end do

      end

C ------------------------------------------------------------------------------
        function    clind(code)

        character*1 code
        character*1 clind

        if (code.eq.'R') then
           clind = '9'
        else if (code.eq.'S') then
           clind = '7'
        else if (code.eq.'A') then
           clind = '6'
        else if (code.eq.'C') then
           clind = '8'
        else if (code.eq.'Q') then
           clind = '9'
        else if (code.eq.'N') then
           clind = 'X'
        else
           clind = 'Z'
        end if

        return
        end

C ------------------------------------------------------------------------------
        function    classind(code)

        character*1 code
        integer*4   classind

        if (code.eq.'9') then
           classind = 4
        else if (code.eq.'6') then
           classind = 1
        else if (code.eq.'7') then
           classind = 2
        else if (code.eq.'8') then
           classind = 3
        else
           classind = 4
        end if

        return
        end

C ------------------------------------------------------------------------------
      function qind(date)

      integer*4 qind
      character*8 date

      if (date.ge.'19990911'.and.date.le.'19991203') then
         qind = 1
      else if (date.ge.'19991204'.and.date.le.'20000225') then
         qind = 2
      else if (date.ge.'20000226'.and.date.le.'20000519') then
         qind = 3
      else if (date.ge.'20000520'.and.date.le.'20000908') then
         qind = 4
```

```
      else
         qind = 0
         print *,'bad date',date
      end if

      return
      end
C ------------ ----------------------------------------------
```

```
#!/usr/bin/csh

mt -f /dev/rmt/1 rewind
foreach ap (01 02 03 04 05 06 07 08 09 10 11 12 13)
dd if=/dev/rmt/1n ibs=32741 of=infile
dqsort -c extract.sm -sl l.ext${ap} -O
mv outfile ../data/nctbext.00${ap}
rm infile
end
```

```
Input file is "infile", recs are 29 chars.
output file is "outfile",  recs are 29 chars.
if 7/11 <> "41310" and 7/11 <> "41320" and 7/11 <> "41411" and 7/11 <>
 "41412" and 7/11 <> "41413" and 7/11 <> "41414" and 7/11 <> "41416" and 7/11
<> "41418" and 7/11 <> "41316" and 7/11 <> "41410" and 7/11 <> "41419" then skip.
copy.
end.
```

```
%   MAPFIN.SM
%
%   CREATE FILE FOR MAP OF FINANCE NUMBERS.
%
INPUT FILE IS "nctbext.0001", recs are 29 chars.
INPUT FILE IS "nctbext.0002", recs are 29 chars.
INPUT FILE IS "nctbext.0003", recs are 29 chars.
INPUT FILE IS "nctbext.0004", recs are 29 chars.
INPUT FILE IS "nctbext.0005", recs are 29 chars.
INPUT FILE IS "nctbext.0006", recs are 29 chars.
INPUT FILE IS "nctbext.0007", recs are 29 chars.
INPUT FILE IS "nctbext.0008", recs are 29 chars.
INPUT FILE IS "nctbext.0009", recs are 29 chars.
INPUT FILE IS "nctbext.0010", recs are 29 chars.
INPUT FILE IS "nctbext.0011", recs are 29 chars.
INPUT FILE IS "nctbext.0012", recs are 29 chars.
INPUT FILE IS "nctbext.0013", recs are 29 chars.
OUTPUT FILE IS "mapfin", recs are data sensitive upto 7 chars.
KEY 1/6.
REFORMAT 1/6.
INSERT "<012>" AFTER LAST.
SORT DELETING DUPLICATES.
END.
```

```
      PROGRAM REVACCTS_BYAP.Q4
C
C       PURPOSE: CREATE FILES FOR EACH PI REV ACCT OF THE AMOUNT BY AP.
C
C       AUTHOR: MM
C       DATE  : 15 JUL 93
C       MODIFIED : amr 10/3/94 for 94 processing
C
C       PROJECT: MCB:TASK9
C

      IMPLICIT NONE

      integer*4   maxfin,naccts,naps
      parameter   (maxfin=30000)
      parameter   (naccts=13)
      parameter   (naps=13)     ! <<<<<<< Update >>>>>>>

      integer*4   ind, inx, fin(maxfin), ier, i, totfin, ap, loc
      integer*4   finx, acctx, finpos, acct(naccts), totact
      integer*4   searchi, act, inz, a, f,z

      real*8     rev (naps,maxfin,naccts)
      real*8       revcur (naps,maxfin,naccts)
      real*8       revamt

      character*2 aps(naps), apx

      logical DEBUG/.true./
C     logical DEBUG/.false./

      data   acct/41310,41316,41320,41410,41411,41412,41413,41414,41416,41418,41419,41440,41441/

C  initialize arrays to 0.

      do ap = 1,naps
         do i = 1,maxfin
            do act = 1,naccts
               rev(ap,i,act) = 0.0
               revcur(ap,i,act) = 0.0
            end do
         end do
      end do


C  READ MAP OF ALL FINANCE NUMBERS

      open(16,file='mapfin')
17    format(i6)

      ind = 1
      ier = 0
      do while (ier.eq.0)
         read(16,17,iostat=ier,end=101) fin(ind)
         ind = ind + 1
      end do

101   totfin = ind-1
      print*, 'mapfin read with',totfin,'lines'
      print*, 'exit code read as  ', ier

c  WRITE OUT DATA FOR EACH AP

C  set up arrays <<<<<<<< update >>>>>>>>>>

      aps(1) = '01'
      aps(2) = '02'
      aps(3) = '03'
      aps(4) = '04'
      aps(5) = '05'
      aps(6) = '06'
      aps(7) = '07'
      aps(8) = '08'
      aps(9) = '09'
      aps(10) = '10'
      aps(11) = '11'
      aps(12) = '12'
      aps(13) = '13'

C      open NCTBEXT files by ap
```

```
      ap = 0
      inx = 0

      do i = 1,naps

         apx = aps(i)
         ap = ap + 1
         ier = 0
         open(30,file='nctbext.00'//apx,access='direct',form='formatted',recl=29)
31       format(i6,i5,3x,f15.2,1x)
         z=1
         do while (ier.eq.0)
            read (30,31,iostat=ier,rec=z,err=130) finx,acctx,revamt
            z=z+1
            inx = searchi(fin,totfin,finx)
            inz = searchi(acct,naccts,acctx)

            if ((inx.gt.0).and.(inz.gt.0)) then
               a=inz
               f=inx
               rev(ap,f,a) =revamt
            else if (inx.lt.1) then
               print*,'finance number ',finx,'  for acct',acctx,' for $ ',
     +            revamt,' is not found'
            end if
         end do
130      print*,' read exit code =',ier
         close (30)
      end do

C     rename rev(1, , ) - revcur(1, , ) for write purposes

      do finpos = 1,totfin
         do act=1,naccts
            revcur(1,finpos,act)=rev(1,finpos,act)
         end do
      end do

C     Correct the amount of revenue for each AP.
C     Subtract rev in AP1 from rev in AP2, etc.

      do finpos = 1,totfin
         do ap=1,naps-1              ! <<<<<<<<<< Update >>>>>>>>>>>
            do act=1,naccts
               revcur(ap+1,finpos,act)=rev(ap+1,finpos,act)-rev(ap,finpos,act)
            end do
         end do
      end do



c  Now write the data to files - (A[acct#].TXT)

      open(40,file='A41310.TXT')
41    format(i6,13(f15.2))      ! <<<<<<<<< Update >>>>>>>>>

      do loc = 1,totfin
         write(40,41) fin(loc), (revcur(ap,loc,1), ap = 1,naps) ! <<<< Update>>>>
      end do

      close(40)

      open(42,file='A41316.TXT')

      do loc = 1,totfin
         write(42,41) fin(loc), (revcur(ap,loc,2), ap = 1,naps) ! <<<< Update>>>>
      end do

      close(42)

      open(43,file='A41320.TXT')

      do loc = 1,totfin
         write(43,41) fin(loc), (revcur(ap,loc,3), ap = 1,naps) ! <<<< Update>>>>
      end do

      close(43)

      open(44,file='A41410.TXT')
```

```
      do loc = 1,totfin
         write(44,41) fin(loc), (revcur(ap,loc,4), ap = 1,naps) ! <<<< Update>>>>
      end do

      close(44)

      open(45,file='A41411.TXT')

      do loc = 1,totfin
C Correction for sites that have zero data AP1 to 5 then a huge amount in AP6
C We are zeroing out AP 6.
C         if ((fin(loc).eq.116918).OR.(fin(loc).eq.324803).OR.
C     &      (fin(loc).eq.335673)) then
C            revcur(6,loc,5) = 0
C         end if
         write(45,41) fin(loc), (revcur(ap,loc,5), ap = 1,naps) ! <<<< Update>>>>
      end do
      close(45)
      open(46,file='A41412.TXT')

      do loc = 1,totfin
         write(46,41) fin(loc), (revcur(ap,loc,6), ap = 1,naps) ! <<<< Update>>>>
      end do

      close(46)

      open(47,file='A41413.TXT')

      do loc = 1,totfin
         write(47,41) fin(loc), (revcur(ap,loc,7), ap = 1,naps) ! <<<< Update>>>>
      end do

      close(47)

      open(48,file='A41414.TXT')

      do loc = 1,totfin
         write(48,41) fin(loc), (revcur(ap,loc,8), ap = 1,naps) ! <<<< Update>>>>
      end do
      close(48)

      open(49,file='A41416.TXT')

      do loc = 1,totfin
         write(49,41) fin(loc), (revcur(ap,loc,9), ap = 1,naps) ! <<<< Update>>>>
      end do
      close(49)

      open(50,file='A41418.TXT')

      do loc = 1,totfin
         write(50,41) fin(loc), (revcur(ap,loc,10), ap = 1,naps) ! <<<< Update>>>>
      end do
      close(50)

      open(51,file='A41419.TXT')

      do loc = 1,totfin
         write(51,41) fin(loc), (revcur(ap,loc,11), ap = 1,naps) ! <<<< Update>>>>
      end do
      close(51)

      open(51,file='A41440.TXT')

      do loc = 1,totfin
         write(51,41) fin(loc), (revcur(ap,loc,12), ap = 1,naps) ! <<<< Update>>>>
      end do
      close(51)
      open(51,file='A41441.TXT')

      do loc = 1,totfin
         write(51,41) fin(loc), (revcur(ap,loc,13), ap = 1,naps) ! <<<< Update>>>>
      end do

      close(51)


      end
```

```
#!/usr/bin/ksh

for file in 41310 41316 41320 41410 41411 41412 41413 41414 41416 41418 41419
do

   rm revfile
   echo "$file"
   cp ../data/A$file.TXT revfile
   strata_00 > l.strata_00.$file
   mv strata.hy ../strata/strata.$file

done
```

```
      PROGRAM strata_00
C
C     PURPOSE: Stratify permit and bravis offices by revenue decades.
C
C     AUTHOR: mrm
C     DATE  : 12-Feb-92
C     LAST MODIFIED: 1-27-94 weh
C     Using new method to assign offices to strata.
C                    10/3/94 amr for 94 nctb processing
C     PROJECT: mcb
C
C     SUBROUTINES:
C     LINKING:
C

      IMPLICIT NONE

      integer*4   numcag, maxfin, numper, numbra, size, numdec,nap

      parameter   (numdec = 20)
      parameter   (maxfin = 31391)
      parameter   (nap=13)      ! change number of aps, check formats in prog

      real*8      revenue(maxfin), rev(nap), decrev, rt
      real*8      revbyap(maxfin,nap)
      real*8      revdec(numdec)
      real*8      totrev , accrev, limit
      real*8      cutoff(numdec)

      integer*4   offdec(numdec)
      integer*4   i, j, k, l, jlast, numfin
      integer*4   ier,ifin,iap,idec
      integer*4   searchi, numpos, firstpos, declen
      integer*4   finnos(maxfin), finno
      integer*4   finsrt(maxfin)
      integer*4   cpos

      logical found(maxfin)

      logical  switch

      do ifin=1,maxfin
         found(ifin)=.false.
         revenue(ifin) = 0
         do iap = 1,nap
            revbyap(ifin,iap) = 0
         end do
      end do
      totrev = 0
      do idec = 1,numdec
         revdec(idec) = 0
         cutoff(idec) = 0
         offdec(idec) = 0
      end do
      ier = 0

      open(20,file='revfile')
21    format(i6,13f15.2)
C change format each quarter for correct number of APs
      ier=0
      numfin = 1
      do while (ier.eq.0)
         cpos = 0
         read (20,21,iostat=ier,end=200) finno, rev
         rt = 0.
         do j = 1, nap
c            print*, 'j = ', j, ' rev = ', -rev(j)
            rt = rt - rev(j)
         end do
C Section added because of data problems with quarter 2 replacing negative data with average

         if (rt.lt.0) then
            rt = 0
            do j = 1,nap
               if (rev(j).lt.0) then
                  rt = rt - rev(j)
                  cpos = cpos + 1
               end if
            end do
            do j = 1,nap
```

```
               if (rev(j).gt.0) then
                  rev(j)=rt/cpos
               end if
            end do
            rt = 0
            do j = 1,nap
               rt = rt - rev(j)
            end do
         end if

         if (rt.ne.0.0) then
            finnos(numfin) = finno
            finsrt(numfin) = finno
            revenue(numfin) = rt
            totrev = totrev + rt
            do j = 1, nap
               revbyap(numfin,j) = -rev(j)
            end do
            numfin = numfin + 1
         end if
      end do
200   print *,' read exit of revenue file = ',ier
      numfin = numfin - 1
      print *, numfin,' finance numbers with positive (or negative) revenue '

      call  sort(numfin,revenue,finnos) ! sort offices by increasing rev.
      ! equal revenue in each stratum

      open(40,file='strata.hy')
41    format(i6.6,i3,f15.2,13f12.2)
C change format each quarter for correct number of APs

      decrev = (totrev + 1.0) / dble(numdec)
      print *,' total revenue   = ',totrev
      print *,' stratum revenue = ',decrev

      accrev = 0.0
      j = 1
      limit = decrev
      do i = numfin, 1, -1
         if ((j.lt.numdec).and.
     +      (accrev+revenue(i).gt.limit).and.
     +      (abs(limit-accrev).lt.
     +      abs(limit-(accrev+revenue(i))))) then
            j = j + 1
            cutoff(j) = revenue(i)
            limit = decrev * j
         end if
         accrev = accrev + revenue(i)
         k = searchi(finsrt,numfin,finnos(i))
C         if (k.gt.0) then
         write (40,41) finnos(i), j, revenue(i), (revbyap(k,l),l=1,nap)
         offdec(j) = offdec(j) + 1
         revdec(j) = revdec(j) + revenue(i)
C         else
C            print*,'fin not found: ',finnos(i),' i: ',i
C         end if
      end do


61    format(5x,i2,i8,2f16.0)
      do i = 1, numdec
         write (*,61) i, offdec(i), cutoff(i), revdec(i)
      end do


      end

C ---------------------------------------------------------------------------
        subroutine sort(n,array1,array2)

C
C       PURPOSE: Sort costalloc output for fungroup processing
C
C
C       AUTHOR: mrm      - Algorithim is heapsort from numerical recipes.
C       DATE  : 13-Feb-92
C       LAST MODIFIED:
C
C       PROJECT:
```

```
C
C       SUBROUTINES:
C
C       LINKING:
C

        integer*4   i, j, l, n, ir
        real*8      rra
        integer*4   rra2
        real*8      array1(n)
        integer*4   array2(n)


        l=n/2+1
        ir=n
10      continue

        if(l.gt.1)then
           l=l-1
           rra=array1(l)
           rra2=array2(l)
        else
           rra=array1(ir)
           rra2=array2(ir)
           array1(ir)=array1(1)
           array2(ir)=array2(1)
           ir=ir-1
           if(ir.eq.1)then
              array1(1)=rra
              array2(1)=rra2
              return
           end if
        end if
        i=l
        j=l+l
20      if (j.le.ir) then
           if (j.lt.ir) then
              if (array1(j).lt.array1(j+1)) j=j+1
           end if
           if (rra.lt.array1(j)) then
              array1(i)=array1(j)
              array2(i)=array2(j)
              i=j
              j=j+j
           else
              j=ir+1
           end if
           goto 20
        end if
        array1(i)=rra
        array2(i)=rra2
        goto 10
        end
```

```
      PROGRAM strata_99_22
C
C     PURPOSE: Stratify permit and bravis offices by revenue decades.
C
C     AUTHOR: mrm
C     DATE  : 12-Feb-92
C     LAST MODIFIED: 1-27-94 weh
C     Using new method to assign offices to strata.
C                    10/3/94 amr for 94 nctb processing
C     PROJECT: mcb
C
C     SUBROUTINES:
C     LINKING:
C

      IMPLICIT NONE

      integer*4   numcag, maxfin, numper, numbra, size, numdec,nap

      parameter   (numdec = 22)
      parameter   (maxfin = 7242)
      parameter   (nap=13)       ! change number of aps, check formats in prog

      real*8      revenue(maxfin), rev(nap), decrev, rt
      real*8       revbyap(maxfin,nap)
      real*8      revdec(numdec)
      real*8      totrev , accrev, limit
      real*8      cutoff(numdec)

      character*6   fin,finnos(maxfin),finno
      character*36  rnew
      character*156 rec(maxfin)

      integer*4   offdec(numdec)
      integer*4   i, j, k, l, jlast, numfin
      integer*4   ier,ifin,iap,idec
      integer*4   searchi, numpos, firstpos, declen
      integer*4   finsrt(maxfin)
      integer*4   cpos

      logical found(maxfin)

      logical  switch

      do ifin=1,maxfin
         found(ifin)=.false.
         revenue(ifin) = 0
         do iap = 1,nap
            revbyap(ifin,iap) = 0
         end do
      end do
      totrev = 0
      do idec = 1,numdec
         revdec(idec) = 0
         cutoff(idec) = 0
         offdec(idec) = 0
      end do
      ier = 0

      open(20,file='strata.41310')
21    format(a6,3x,f15.2,a156)
C change format each quarter for correct number of APs
      ier=0
      numfin = 1
      do ifin=1,maxfin
         cpos = 0
         read (20,21,iostat=ier,end=200) finnos(ifin),revenue(ifin),rec(ifin)
         read(rec(ifin),'(13f12.2)') (revbyap(ifin,iap),iap=1,13)
         revenue(ifin)=0
         do iap=1,13
            revenue(ifin)=revenue(ifin)+revbyap(ifin,iap)
         end do
         totrev = totrev + revenue(ifin)
         numfin = numfin + 1
      end do
200   print *,' read exit of revenue file = ',ier
      numfin = numfin - 1
      print *, numfin,' finance numbers with positive (or negative) revenue '

      call  sort(numfin,revenue,finnos,rec) ! sort offices by increasing rev.
```

```
      ! equal revenue in each stratum

      open(40,file='strata.41310.n22')
11    format(a6,i3,f15.2,a156)
c change format each quarter for correct number of APs

      decrev = (totrev * 1.0) / dble(numdec-2)
      print *,' total revenue   = ',totrev
      print *,' stratum revenue = ',decrev

      accrev = 0.0
      j = 1
      limit = decrev
      do i = numfin, 1, -1
         if ((j.lt.numdec+2).and.
     +       (accrev+revenue(i).gt.limit).and.
     +       (abs(limit-accrev).lt.
     +       abs(limit-(accrev+revenue(i))))) then
            j = j + 1
            cutoff(j) = revenue(i)
            if (j.le.18) then
               limit = decrev * j
            else
               limit = (decrev * 18) + ((decrev/2)*(j-18))
            end if

         end if
         accrev = accrev + revenue(i)
         write (40,41) finnos(i), j, revenue(i), rec(i)
         offdec(j) = offdec(j) + 1
         revdec(j) = revdec(j) + revenue(i)
      end do


31    format(5x,i2,i8,2f16.0)
      do i = 1, numdec
         write (*,61) i, offdec(i), cutoff(i), revdec(i)
      end do


      end

c     ---------------- ---------------------------------------------------
        subroutine sort(n,array1,array2,array3)

c
c       PURPOSE: Sort costalloc output for fungroup processing
c       AUTHOR: mrm       - Algorithim is heapsort from numerical recipes.
c       DATE  : 13-Feb-92

        integer*4     i, j, l, n, ir
        real*8        rra
        character*6   rra2
        character*156 rra3
        real*8        array1(n)
        character*6   array2(n)
        character*156 array3(n)


        l=n/2+1
        ir=n
10      continue

        if(l.gt.1)then
           l=l-1
           rra=array1(l)
           rra2=array2(l)
           rra3=array3(l)
        else
           rra=array1(ir)
           rra2=array2(ir)
           rra3=array3(ir)
           array1(ir)=array1(1)
           array2(ir)=array2(1)
           array3(ir)=array3(1)
           ir=ir-1
           if(ir.eq.1)then
              array1(1)=rra
              array2(1)=rra2
              array3(1)=rra3
```

```
            return
         end if
      end if
      i=l
      j=l+l
20    if (j.le.ir) then
         if (j.lt.ir) then
            if (array1(j).lt.array1(j+1)) j=j+1
         end if
         if (rra.lt.array1(j)) then
            array1(i)=array1(j)
            array2(i)=array2(j)
            array3(i)=array3(j)
            i=j
            j=j+j
         else
            j=ir+1
         end if
         goto 20
      end if
      array1(i)=rra
      array2(i)=rra2
      array3(i)=rra3
      goto 10
      end
```

```
      PROGRAM strata_99_22_41316
C
C     PURPOSE: Stratify permit and bravis offices by revenue decades.
C
C     AUTHOR: mrm
C     DATE  : 12-Feb-92
C     LAST MODIFIED: 1-27-94 weh
C     Using new method to assign offices to strata.
C                    10/3/94 amr for 94 nctb processing
C     PROJECT: mcb
C
C     SUBROUTINES:
C     LINKING:
C

      IMPLICIT NONE

      integer*4   numcag, maxfin, numper, numbra, size, numdec,nap

      parameter   (numdec = 22)
      parameter   (maxfin = 5481)
      parameter   (nap=13)       ! change number of aps, check formats in prog

      real*8      revenue(maxfin), rev(nap), decrev, rt
      real*8       revbyap(maxfin,nap)
      real*8      revdec(numdec)
      real*8      totrev , accrev, limit
      real*8      cutoff(numdec)

      character*6   fin,finnos(maxfin),finno
      character*36  rnew
      character*156 rec(maxfin)

      integer*4   offdec(numdec)
      integer*4   i, j, k, l, jlast, numfin
      integer*4   ier,ifin,iap,idec
      integer*4   searchi, numpos, firstpos, declen
      integer*4   finsrt(maxfin)
      integer*4   cpos

      logical found(maxfin)

      logical  switch

      do ifin=1,maxfin
         found(ifin)=.false.
         revenue(ifin) = 0
         do iap = 1,nap
            revbyap(ifin,iap) = 0
         end do
      end do
      totrev = 0
      do idec = 1,numdec
         revdec(idec) = 0
         cutoff(idec) = 0
         offdec(idec) = 0
      end do
      ier = 0

      open(20,file='strata.41316')
21    format(a6,3x,f15.2,a156)
C change format each quarter for correct number of APs
      ier=0
      numfin = 1
      do ifin=1,maxfin
         cpos = 0
         read (20,21,iostat=ier,end=200) finnos(ifin),revenue(ifin),rec(ifin)
         read(rec(ifin),'(13f12.2)') (revbyap(ifin,iap),iap=1,13)
         revenue(ifin)=0
         do iap=1,13
            revenue(ifin)=revenue(ifin)+revbyap(ifin,iap)
         end do
         totrev = totrev + revenue(ifin)
         numfin = numfin + 1
      end do
200   print *,' read exit of revenue file = ',ier
      numfin = numfin - 1
      print *, numfin,' finance numbers with positive (or negative) revenue '

      call  sort(numfin,revenue,finnos,rec) ! sort offices by increasing rev.
```

```
      ! equal revenue in each stratum

      open(40,file='strata.41316.n22')
10    format(a6,i3,f15.2,a156)
c change format each quarter for correct number of APs

      decrev = (totrev + 1.0) / dble(numdec-2)
      print *,' total revenue   = ',totrev
      print *,' stratum revenue = ',decrev

      accrev = 0.0
      j = 1
      limit = decrev
      do i = numfin, 1, -1
         if ((j.lt.numdec+2).and.
     +       (accrev+revenue(i).gt.limit).and.
     +       (abs(limit-accrev).lt.
     +       abs(limit-(accrev+revenue(i))))) then
            j = j + 1
            cutoff(j) = revenue(i)
            if (j.le.18) then
               limit = decrev * j
            else
               limit = (decrev * 18) + ((decrev/2)*(j-18))
            end if

         end if
         accrev = accrev + revenue(i)
         write (40,41) finnos(i), j, revenue(i), rec(i)
         offdec(j) = offdec(j) + 1
         revdec(j) = revdec(j) + revenue(i)
      end do


51    format(5x,i2,i8,2f16.0)
      do i = 1, numdec
         write (*,61) i, offdec(i), cutoff(i), revdec(i)
      end do


      end

c     ------------------------------------------------------------------------
      subroutine sort(n,array1,array2,array3)

c
c     PURPOSE: Sort costalloc output for fungroup processing
c     AUTHOR: mrm      - Algorithim is heapsort from numerical recipes.
c     DATE  : 13-Feb-92

      integer*4     i, j, l, n, ir
      real*8        rra
      character*6   rra2
      character*156 rra3
      real*8        array1(n)
      character*6   array2(n)
      character*156 array3(n)


      l=n/2+1
      ir=n
10    continue

      if(l.gt.1)then
         l=l-1
         rra=array1(l)
         rra2=array2(l)
         rra3=array3(l)
      else
         rra=array1(ir)
         rra2=array2(ir)
         rra3=array3(ir)
         array1(ir)=array1(1)
         array2(ir)=array2(1)
         array3(ir)=array3(1)
         ir=ir-1
         if(ir.eq.1)then
            array1(1)=rra
            array2(1)=rra2
            array3(1)=rra3
```

```
            return
         end if
      end if
      i=l
      j=l+l
20    if (j.le.ir) then
         if (j.lt.ir) then
            if (array1(j).lt.array1(j+1)) j=j+1
         end if
         if (rra.lt.array1(j)) then
            array1(i)=array1(j)
            array2(i)=array2(j)
            array3(i)=array3(j)
            i=j
            j=j+j
         else
            j=ir+1
         end if
         goto 20
      end if
      array1(i)=rra
      array2(i)=rra2
      array3(i)=rra3
      goto 10
      end
```

```
*
*
*
input file is 'strata.41310.n22', recs are data sens upto 300 chars.
output file is 'strata.41310.n22.srt', recs are data sens upto 300 chars.
*
key 1/6.
sort deleting duplicates.
insert '<012>' after last.
end.
```

```
;
;
;
input file is 'strata.41316.n22', recs are data sens upto 300 chars.
output file is 'strata.41316.n22.srt', recs are data sens upto 300 chars.
;
key 1/6.
sort deleting duplicates.
insert '<012>' after last.
end.
```

```
      program mkstr
C     purpose: sum trial balance revenues by quarter.  This file is needed for input
C     into the control program that calcualates inflation factors.  This program
C     needs to be run twice, once for the in-county account (41316) and
C     once for the outside county account (41310).
C
C     author: mcb.pms
C     date: 1/2/96
C     project: 2c stratification
C     modified 12/99 by KS for FY99 processing
C
      IMPLICIT NONE

      integer*4    nfin, npmt
      parameter    (nfin = 7242)
      parameter    (npmt = 1877)

      integer*4    cnt, ier, i, j
      integer*4    searchc, ifin
      integer*4    oldx(nfin)
      character*6  fin, finx(nfin), finp(npmt)
      real*8       revx(nfin)
      real*8        aprev(nfin,13),qrev(nfin,4)

      open(14,file='fin.map')
      do i=1,npmt
         read(14,'(a6)') finp(i)
      end do

C OPEN OUTPUT FILE
      open(20,file='key00_oc.2nd')

C READ IN STRATA FILE
      open(10,file='strata.41310.n22.srt')
      ier = 0
      cnt = 0
      do while (ier.eq.0)
         do i = 1, nfin
            read(10,'(a6,i3,f15.2,13f12.2)',iostat=ier,end=99) finx(i), oldx(i), revx(i),(aprev(i,j),j=1,13)
            qrev(i,1)=aprev(i,1)+aprev(i,2)+aprev(i,3)
            qrev(i,2)=aprev(i,4)+aprev(i,5)+aprev(i,6)
            qrev(i,3)=aprev(i,7)+aprev(i,8)+aprev(i,9)
            qrev(i,4)=aprev(i,10)+aprev(i,11)+aprev(i,12)+aprev(i,13)

            cnt = cnt + 1
         end do
      end do
99    print *, 'strata.41316 read with ',cnt, ' records and ier = ',ier
      close(10)

      do ifin = 1, nfin
         write(20,'(a6,i3,f15.2,4f15.2)') finx(ifin), oldx(ifin),
     &      revx(ifin),(qrev(ifin,j),j=1,4)
         j = searchc(finp,npmt,finx(ifin))
         if (j.gt.0.and.oldx(ifin).eq.0)
     &      print *,' PERMIT ',finx(ifin),',',oldx(ifin),',',revx(ifin)
      end do

      end
```

```
      program mkstr41316
C     purpose: sum trial balance revenues by quarter.  This file is needed for input
C     into the control program that calcualates inflation factors.  This program
C     needs to be run twice, once for the in-county account (41316) and
C     once for the outside county account (41310).
C
C     author: mcb.pms
C     date: 1/2/96
C     project: 2c stratification
C     modified 12/99 by KS for FY99 processing
C
      IMPLICIT NONE

      integer*4    nfin, npmt
      parameter    (nfin = 5481)
      parameter    (npmt = 1877)

      integer*4    cnt, ier, i, j
      integer*4    searchc, ifin
      integer*4    oldx(nfin)
      character*6  fin, finx(nfin), finp(npmt)
      real*8       revx(nfin)
      real*8        aprev(nfin,13),qrev(nfin,4)

      open(14,file='fin.map')
      do i=1,npmt
         read(14,'(a6)') finp(i)
      end do

C OPEN OUTPUT FILE
      open(20,file='key99_ic.2nd')

C READ IN STRATA FILE
      open(10,file='strata.41316.n22.srt')
      ier = 0
      cnt = 0
      do while (ier.eq.0)
         do i = 1, nfin
            read(10,'(a6,i3,f15.2,13f12.2)',iostat=ier,end=99) finx(i), oldx(i), revx(i),(aprev(i,j),j=1,13)
            qrev(i,1)=aprev(i,1)+aprev(i,2)+aprev(i,3)
            qrev(i,2)=aprev(i,4)+aprev(i,5)+aprev(i,6)
            qrev(i,3)=aprev(i,7)+aprev(i,8)+aprev(i,9)
            qrev(i,4)=aprev(i,10)+aprev(i,11)+aprev(i,12)+aprev(i,13)

            cnt = cnt + 1
         end do
      end do
99    print *, 'strata.41316 read with ',cnt, ' records and ier = ',ier
      close(10)

      do ifin = 1, nfin
         write(20,'(a6,i3,f15.2,4f15.2)') finx(ifin), oldx(ifin),
     &      revx(ifin),(qrev(ifin,j),j=1,4)
         j = searchc(finp,npmt,finx(ifin))
         if (j.gt.0.and.oldx(ifin).eq.0)
     &      print *,' PERMIT ',finx(ifin),',',oldx(ifin),',',revx(ifin)
      end do

      end
```

```
      program mkmap

c     Purpose: To make a map of each finance number with it's inside and outside county strata

      implicit none

      integer*4    nifin,nofin
      parameter    (nifin = 5481)
      parameter    (nofin = 7242)
      character*6  finso(nofin),finsi(nifin)
      character*6  fin
      integer*4    stratai(nifin)
      integer*4    stratao(nofin)
      integer*4    searchc
      integer*4    ifin,ifin2
      integer*4    istrata,istrata2
      integer*4    ier,i,j

      open(10,file='strata.41316.n22.srt')
      do ifin=1,nifin
         read(10,'(a6,i3)') finsi(ifin),stratai(ifin)
      end do
      close(10)

      open(10,file='strata.41310.n22.srt')
      do ifin=1,nofin
         read(10,'(a6,i3)') finso(ifin),stratao(ifin)
      end do
      close(10)

      open(20,file='fin_strata.map')
25    format(a6,i3,i3)

      open(10,file='fin.map')
      ier=0
      do while (ier.eq.0)
         read(10,'(a6)',iostat=ier,end=100) fin
         istrata=0
         istrata2=0
         ifin=searchc(finsi,nifin,fin)
         ifin2=searchc(finso,nofin,fin)
         if (ifin.gt.0) then
            istrata=stratai(ifin)
         else
            print *,'fin not in ic, fin = ',fin
         end if
         if (ifin2.gt.0) then
            istrata2=stratao(ifin2)
         else
            print *,'fin not in oc, fin = ',fin
         end if
         write(20,25) fin,istrata,istrata2
      end do

100   print *,'end of file read, ier = ',ier

      end
```

```
      PROGRAM sepgov

C     PURPOSE: check the bin2nd.data to get separate government and non-government
C     revenue files by quarter.  The output file gov.rev will be used for
C     determining the inflation factors in control.f.
C     The reason that we have to separate
C     government revenues out is that government revenues are included in
C     the RPW, but not in the NCTB data.  This affects the inflation factors when
C     we first roll up to NCTB, and then to RPW.
C
C
C     Input : bin2nd.data (unzipped file), finance number map of fins used
C     during the year.
C
C     12/99 notes - Karina
C     I am changing this program to separate in-county rate mail
C     from outside county mail.  This is because we are going to be calculating
C     separate inflation factors.
C
      IMPLICIT NONE

      integer*4      maxvips,nfin,nchk,nq

      parameter      (nchk=144)
      parameter      (nq=4)
      parameter      (nfin=7389)

C     Character strorage

      character*13 tno          !  transaction number (1:7) julian date and year
      character*6  tin          !  finance number
      character*5  vip          !  VIP code
      character*1  uspsst       !  USPS status - M=master, P=pending
      character*7  uspsno       !  USPS number
      character*1  uspsst_re    !  Related USPS status
      character*7  uspsno_re    !  Related USPS number
      character*1  Prc_cat      !  Process Category

C     Parameters
      integer*4  nvips          !  maximum number of vips
      parameter  (nvips = 84)

      character*5 vipd(nvips)   ! vip codes

      integer*4  numvip         ! integer value of clines,number of vips to follow
      integer*4  il             ! index for vip detail
      integer*4  ier            ! read error iostat
      integer*4  cnt            ! counter for number of transaction read
      real*8     vippc(nvips)   ! VIP pieces
      real*8     vipwt(nvips)   ! VIP weight
      real*8     viprev(nvips)  ! VIP revenue
      real*8     vipcop(nvips)  ! VIP copies

      character*1   rty
      character*4   vipmap(nchk)
      character*6   tusps
      character*6   fins(nfin)
      character*6   tdate,jul_to_txt
      character*5   vipname
      character*4244 rec
      character*8   tdate2

      integer*4     searchc
      integer*4     vipty,ivip
      integer*4     ipub
      integer*4     vipmap_in(nchk,5)
      integer*4     icat,procescat
      integer*4     iq,qind
      integer*4     ifin,iloc

      logical       debug

      real*8         grev(nfin,2,4),ngrev(nfin,2,4)
      real*8         rtot,lrev,chk

      integer*4      i,j

C     initialize values
```

```
do i=1, nfin
   do il=1,2
      do j=1,4
         grev(i,il,j)=0.0
         ngrev(i,il,j)=0.0
      end do
   end do
end do

open(10,file='../../stratab/maps/fin.map')
do i=1, nfin
   read(10,'(a6)') fins(i)
end do
print *,'finished reading finance number map'
close(10)

open (15,file='../../sepdata/seccon.950101.950931')
do i=1,nchk-2
   read(15,'(6x,a4,3i3,i2,i3)',iostat=ier) vipmap(i),(vipmap_in(i,j),j=1,5)
end do
vipmap(nchk-1) = '9998'
do j = 1,5
   vipmap_in(nchk-1,j) = 1
end do
vipmap(nchk) = '9999'
do j = 1,5
   vipmap_in(nchk,j) = 1
end do

print *,'Finished reading the vip code map',ier
close(15)

open (20,file='gov_ic.rev')
open (25,file='gov_oc.rev')
open (30,file='nongov_ic.rev')
open (35,file='nongov_oc.rev')

ier = 0
open(80,file='/u/mcb/data/second/FY00/newdata.data')
do while (ier.eq.0)

   read(80,'(a4244)',end=100,iostat=ier) rec
   read (rec(1:6),'(a6)') fin
   read (rec(7:7),'(a1)') rty
   read (rec(10:22),'(a13)') tno
   read (rec(23:23),'(a1)') uspsst
   read (rec(24:30),'(a7)') uspsno
   read (rec(37:37),'(a1)') uspsst_re
   read (rec(38:44),'(a7)') uspsno_re
   read (rec(103:103),'(a1)') prc_cat
   read (rec(104:115),'(f12.2)') rtot
   read (rec(128:130),'(i3)') numvip

   cnt = cnt + 1
   do il = 1, numvip
      read (rec((131+(il-1)*49):(179+(il-1)*49)),'(a5,f12.2,f12.0,2f10.0)')
&        vipd(il),viprev(il),vippc(il),vipwt(il),vipcop(il)
   end do

   if (fin.eq.'115605') fin='118929'
   if (fin.eq.'118675') fin='115854'
   if (fin.eq.'161274') fin='161540'
   if (fin.eq.'162865') fin='161544'
   if (fin.eq.'166027') fin='161540'
   if (fin.eq.'414582') fin='410636'
   if (fin.eq.'484143') fin='484149'
   if (fin.eq.'512704') fin='516540'
   if (fin.eq.'411626') fin='416088'

   tdate=jul_to_txt(tno(3:7))
   if (tdate(1:2).eq.'00') then
      tdate2(1:2) = '20'
      tdate2(3:8) = tdate(1:6)
   else
      tdate2(1:2) = '19'
      tdate2(3:8) = tdate(1:6)
   end if

   iq = qind(tdate2)
   if (iq.eq.0) print*,tno(3:7)
```

```
ifin=searchc(fins,nfin,fin)
if (ifin.gt.0) then
   do i=1,numvip
      ivip=searchc(vipmap,nchk,vipd(i)(2:5))
      if (ivip.gt.0) then
         if (ivip.eq.14) then ! 1430
            ivip=15
         else if (ivip.eq.15) then ! 1440
            ivip=14
         else if (ivip.eq.16) then ! 1444
            ivip=14
         else if (ivip.eq.17) then ! 1445
            ivip=14
         else if (ivip.eq.18) then ! 1450
            ivip=13
         else if (ivip.eq.19) then ! 1460
            ivip=18
         else if (ivip.eq.20) then ! 1470
            ivip=14
         else if (ivip.eq.21) then ! 1474
            ivip=14
         else if (ivip.eq.22) then ! 1475
            ivip=14
         else if (ivip.eq.23) then ! 1480
            ivip=19
         else if (ivip.eq.24) then ! 1490
            ivip=20
         else if (ivip.eq.25) then ! 1500
            ivip=23
         else if (ivip.eq.66) then ! 2174
            ivip=65
         else if (ivip.eq.68) then ! 2184
            ivip=67
         else if (ivip.eq.69) then ! 2185
            ivip=67
         else if (ivip.eq.72) then ! 2204
            ivip=71
         else if (ivip.eq.74) then ! 2214
            ivip=73
         else if (ivip.eq.75) then ! 2215
            ivip=73
         else if (ivip.eq.76) then ! 2220
            ivip=70
         else if (ivip.eq.77) then ! 2230
            ivip=71
         else if (ivip.eq.78) then ! 2240
            ivip=73
         else if (ivip.eq.79) then ! 2244
            ivip=73
         else if (ivip.eq.80) then ! 2245
            ivip=73
         else if (ivip.eq.81) then ! 2250
            ivip=76
         else if (ivip.eq.82) then ! 2260
            ivip=77
         else if (ivip.eq.83) then ! 2270
            ivip=78
         else if (ivip.eq.128) then ! 4221
            ivip=122
         else if (ivip.eq.129) then ! 4231
            ivip= 123
         else if (ivip.eq.130) then ! 4241
            ivip=125
         else if (ivip.eq.131) then ! 4244
            ivip=125
         else if (ivip.eq.132) then ! 4245
            ivip=125
         else if (ivip.eq.133) then ! 4251
            ivip=128
         else if (ivip.eq.134) then ! 4261
            ivip=129
         else if (ivip.eq.135) then ! 4271
            ivip=130
         end if
      end if
      if (((ivip.ge.1).and.(ivip.le.28)).or.(ivip.eq.143)) then
         iloc=1
      else
         iloc=2
      end if
```

```
            if (ivip.ne.0) then
               if (iq.ne.0) then
                  if ((vipmap_in(ivip,2).eq.1).or.(vipmap_in(ivip,2).eq.-1)) then
                     if ((rty.eq.'G').or.(rty.eq.'D')) then
                        grev(ifin,iloc,iq)=grev(ifin,iloc,iq)+(viprev(i)*vipmap_in(ivip,2))
                     else if ((rty.eq.'P').or.(rty.eq.'C')) then
                        ngrev(ifin,iloc,iq)=ngrev(ifin,iloc,iq)+(viprev(i)*vipmap_in(ivip,2))
                     end if
                  end if
               end if
            end if
         end do
      else
         print *,'finance number not found, fin =',fin
      end if
   end do
   print *,'Doing the last loop',ier
   close(80)

   ier=0
   open(80,file='/u/mcb/data/second/FY00/olddata.data')
   do while (ier.eq.0)

      read(80,'(a4244)',end=200,iostat=ier) rec
      read (rec(1:6),'(a6)') fin
      read (rec(7:7),'(a1)') rty
      read (rec(10:22),'(a13)') tno
      read (rec(23:23),'(a1)') uspsst
      read (rec(24:30),'(a7)') uspsno
      read (rec(37:37),'(a1)') uspsst_re
      read (rec(38:44),'(a7)') uspsno_re
      read (rec(103:103),'(a1)') prc_cat
      read (rec(104:115),'(f12.2)') rtot
      read (rec(128:130),'(i3)') numvip

      cnt = cnt + 1
      do il = 1, numvip
         read (rec((131+(il-1)*49):(179+(il-1)*49)),'(a5,f12.2,f12.0,2f10.0)')
     &      vipd(il),viprev(il),vippc(il),vipwt(il),vipcop(il)
      end do

      if (fin.eq.'115605') fin='118929'
      if (fin.eq.'118675') fin='115854'
      if (fin.eq.'161274') fin='161540'
      if (fin.eq.'162865') fin='161544'
      if (fin.eq.'166027') fin='161540'
      if (fin.eq.'414582') fin='410636'
      if (fin.eq.'484143') fin='484149'
      if (fin.eq.'512704') fin='516540'
      if (fin.eq.'411626') fin='416088'

      tdate=jul_to_txt(tno(3:7))
      if (tdate(1:2).eq.'00') then
         tdate2(1:2) = '20'
         tdate2(3:8) = tdate(1:6)
      else
         tdate2(1:2) = '19'
         tdate2(3:8) = tdate(1:6)
      end if
      iq = qind(tdate2)
      ifin=searchc(fins,nfin,fin)
      if (ifin.gt.0) then
         do i=1,numvip
            ivip=searchc(vipmap,nchk,vipd(i)(2:5))
            if (((ivip.ge.1).and.(ivip.le.28)).or.(ivip.eq.143)) then
               iloc=1
            else
               iloc=2
            end if
            if (ivip.ne.0) then
               if (iq.ne.0) then
                  if ((vipmap_in(ivip,2).eq.1).or.(vipmap_in(ivip,2).eq.-1)) then
                     if ((rty.eq.'G').or.(rty.eq.'D')) then
                        grev(ifin,iloc,iq)=grev(ifin,iloc,iq)+(viprev(i)*vipmap_in(ivip,2))
                     else if ((rty.eq.'P').or.(rty.eq.'C')) then
                        ngrev(ifin,iloc,iq)=ngrev(ifin,iloc,iq)+(viprev(i)*vipmap_in(ivip,2))
                     end if
                  end if
               end if
            end if
```

```
            end do
         else
            print *,'finance number not found, fin = ',fin
         end if
      end do
200   print *,'Doing the last loop',ier
      close(80)

      do i=1,nfin
         chk=0
         do il=1,4
            chk=chk+grev(i,1,il)
         end do
         if (chk.gt.0) then
            write(20,'(a6,4(f16.2))') fins(i),(grev(i,1,il),il=1,4)
         end if
         chk=0
         do il=1,4
            chk=chk+grev(i,2,il)
         end do
         if (chk.gt.0) then
            write(25,'(a6,4(f16.2))') fins(i),(grev(i,2,il),il=1,4)
         end if
         chk=0
         do il=1,4
            chk=chk+ngrev(i,1,il)
         end do
         if (chk.gt.0) then
            write(30,'(a6,4(f16.2))') fins(i),(ngrev(i,1,il),il=1,4)
         end if
         chk=0
         do il=1,4
            chk=chk+ngrev(i,2,il)
         end do
         if (chk.gt.0) then
            write(35,'(a6,4(f16.2))') fins(i),(ngrev(i,2,il),il=1,4)
         end if
      end do

      end
C     ------------------------- ------- -----------------
      function qind(date)

      integer*4 qind
      character*8 date

      if (date.ge.'19990911'.and.date.le.'19991203') then
         qind = 1
      else if (date.ge.'19991204'.and.date.le.'20000225') then
         qind = 2
      else if (date.ge.'20000226'.and.date.le.'20000519') then
         qind = 3
      else if (date.ge.'20000520'.and.date.le.'20000908') then
         qind = 4
      else
         qind = 0
         print *,'bad date',date
      end if

      return
      end
C     ------------------------------------------------------
```

```
      program control_10

C  This program calculates the control factors for the second class strata
C  Input:    finance numbers by 22 strata and 2 trial balance accounts
C            Revenue files from permit /bravis systems
C            Revenue files from NCTB
C Updated for 99 RPW numbers
C     This program has been changed from earlier years to inflate in-county
C     revenue separately from outside county revenue.  This is possible because
C     starting in AP5 of FY99, trial balance created a new account number
C     for in-county mail.  After checking characteristics of the different
C     strata (sortation and zone), we decided to combine some of the similar
C     strata.  This is done in the stratafind function.

      implicit  none

      integer*4       i,j,k,ier,nstrata,nq
      integer*4       nfin_ic,nfin_oc
      integer*4       ngov_oc,ngov_ic
      integer*4       npmt
      parameter       (nstrata=10)
      parameter       (nq=4)
      parameter       (npmt=7389)
      parameter       (nfin_ic=4986)
      parameter       (nfin_oc=7557)
      parameter       (ngov_ic=0)
      parameter       (ngov_oc=12)

      integer*4       istr,strata,iq,ifin
      integer*4       searchc
      integer*4       istrata,stratafind,strataic
      integer*4       ic_strata(npmt),oc_strata(npmt)

      real*8          revenue(4),cpprev(13),govtot(nq)
      real*8          strrev_nc(nstrata,nq,2)
      real*8          strrev_pb(nstrata,nq,2)
      real*8          str_cpp(nq)
      real*8          str_ncpp(nq)
      real*8          str_fac_n(nstrata,nq,2)
      real*8          str_fac_c(nq)
      real*8          rpw_fac(nq)
      real*8          rpwrev(nq)
      real*8          nrpwrev(nq),govrev(nq),nrpwrev2(nq)
      real*8          fstr_ncpp(nstrata,nq,2)
      real*8          fstr_cpp(nq),q2rev
      character*6     fin,fins_ic(nfin_ic),fins_oc(nfin_oc)
      character*6     govfin
      character*6     fins(npmt)

C Write the RPW weight manually here
      do i=1,4
         rpwrev(i)=0
         nrpwrev(i)=0
      end do
      rpwrev(1)=503983556
      rpwrev(2)=492787413
      rpwrev(3)=524454408
      rpwrev(4)=640327002
      print *,'initialized RPW values'

      do i=1,nstrata
         do j=1,nq
            do k=1,2
               nrpwrev2(j) = 0.0
               fstr_ncpp(i,j,k)=0.0
               strrev_nc(i,j,k)=0.0
               strrev_pb(i,j,k)=0.0
               str_fac_n(i,j,k)=0.0
            end do
         end do
      end do
      print *,'initialized 2-array values'
      do j=1,nq
         fstr_cpp(j)=0.0
         str_cpp(j)=0.0
         str_ncpp(j)=0.0
         str_fac_c(j)=0.0
         rpw_fac(j)=0.0
         govtot(j)=0.0
```

```
      end do
      print *,'initialized 1-array values'

      open(10,file='fin_strata.map')
15    format(a6,i3,i3)
      do i=1,npmt
         read(10,15) fins(i),ic_strata(i),oc_strata(i)
         ic_strata(i) = strataic(ic_strata(i))
         oc_strata(i) = stratafind(oc_strata(i))
      end do
      close(10)

      open (10,file='key00_ic.2nd')
      print *,'opened key99_ic.2nd'
      ier=0
      do while (ier.eq.0)
         read(10,'(a6,i3,15x,4f15.2)',iostat=ier,end=112) fin,istr,(revenue(j),j=1,nq)
         istrata=strataic(istr)
         do j=1,nq
            strrev_nc(istrata,j,1)=revenue(j)+strrev_nc(istrata,j,1)
         end do
      end do
112   print *,'end of key ic file read, ier = ',ier
      close(10)
      do i=1,nstrata
         write (*,'(i3,4f15.2)') i,(strrev_nc(i,j,1),j=1,nq)
      end do

      open (10,file='key99_oc.2nd')
      print *,'opened key99_oc.2nd'
      ier=0
      do while (ier.eq.0)
         read(10,'(a6,i3,15x,4f15.2)',iostat=ier,end=115) fin,istr,(revenue(j),j=1,nq)
         istrata=stratafind(istr)
         do j=1,nq
            strrev_nc(istrata,j,2)=revenue(j)+strrev_nc(istrata,j,2)
         end do
      end do
115   print *,'end of key oc file read, ier = ',ier
      close(10)
      do i=1,nstrata
         write (*,'(i3,4f15.2)') i,(strrev_nc(i,j,2),j=1,nq)
      end do

      open(10,file='gov_oc.rev')
      print *,'opened gov.rev'
      do i=1,ngov_oc
         read(10,'(a6,4f16.2)') govfin,(govrev(j),j=1,4)
         do j=1,4
            govtot(j) = govtot(j) + govrev(j)
         end do
      end do
      print *,'read oc gov.rev'
      close(10)
      write(*,'(4f15.2)') (govtot(j),j=1,4)



      open(47,file='nongov_oc.rev')
      ier=0
      do while (ier.eq.0)
         read (47,'(a6,4f16.2)',iostat=ier,end=500) fin,(revenue(j),j=1,nq)
         if ((fin.ne.'355826').and.(fin.ne.'000000')) then
            i=searchc(fins,npmt,fin)
            if (i.gt.0) then
               istr=oc_strata(i)
               if (istr.eq.0) then
                  istr=ic_strata(i)
               end if
               if (istr.eq.0) then
                  istr=22
                  print *,'no strata for fin ',fin
               end if
               do iq=1,nq
                  strrev_pb(istr,iq,2)=strrev_pb(istr,iq,2)+revenue(iq)
               end do
            else
               print *, 'Could not locate the finance number that exists in Permit / Bravis',fin
            end if
         end if
```

```
      if (fin.eq.'355826') then
         do iq=1,nq
            str_cpp(iq)=revenue(iq)+str_cpp(iq)
         end do
      end if
   end do
500 print *,ier,'Strata Name, quarterly revenues'
   close(47)

   do i=1,nstrata
      write (*,'(i3,4f15.2)') i,(strrev_pb(i,j,2),j=1,nq)
   end do

   open(47,file='nongov_ic.rev')
   ier=0
   do while (ier.eq.0)
      read (47,'(a6,4f16.2)',iostat=ier,end=600) fin,(revenue(j),j=1,nq)
      if ((fin.ne.'355826').and.(fin.ne.'000000')) then
         i=searchc(fins,npmt,fin)
         if (i.gt.0) then
            istr=ic_strata(i)
            if (istr.eq.0) then
               istr=oc_strata(i)
            end if
            if (istr.eq.0) then
               istr=22
               print *,'no strata for fin ',fin
            end if
            do iq=1,nq
               strrev_pb(istr,iq,1)=strrev_pb(istr,iq,1)+revenue(iq)
            end do
         else
            print *, 'Could not locate the finance number that exists in Permit / Bravis',fin
         end if
      end if
      if (fin.eq.'355826') then
         do iq=1,nq
            str_cpp(iq)=revenue(iq)+str_cpp(iq)
         end do
      end if
   end do
500 print *,ier,'Strata Name, quarterly revenues'
   close(47)

   do i=1,nstrata
      write (*,'(i3,4f15.2)') i,(strrev_pb(i,j,1),j=1,nq)
   end do

   ier=0
   open (25,file='strata.41320')
   read(25,'(24x,13f12.2)') (cpprev(i),i=1,13)
   str_ncpp(1)=cpprev(1)+cpprev(2)+cpprev(3)
   str_ncpp(2)=cpprev(4)+cpprev(5)+cpprev(6)
   str_ncpp(3)=cpprev(7)+cpprev(8)+cpprev(9)
   str_ncpp(4)=cpprev(10)+cpprev(11)+cpprev(12)+cpprev(13)
   close(25)

   print *,'Cpp revenue from NCTB'
   write(*,'(f12.2)') (str_ncpp(i),i=1,nq)
   print *,'Cpp revenue from Permit / Bravis'
   write(*,'(f12.2)') (str_cpp(i),i=1,nq)

   print *,'Starting Non-Cpp NCTB Factor'
   do istr=1,nstrata
      do iq=1,nq
         do i=1,2
            if (strrev_pb(istr,iq,i).ne.0) then
               str_fac_n(istr,iq,i)=strrev_nc(istr,iq,i)/strrev_pb(istr,iq,i)
            else
               str_fac_n(istr,iq,i) = 0.0
            end if
         end do
      end do
   end do


   do i=1,2
      do istr=1,nstrata
         write (*,'(i3,1x,i2,4f8.4,4f15.2)') istr,i,(str_fac_n(istr,j,i),j=1,nq),(strrev_pb(istr,iq,i),iq=1,nq)
      end do
```

```
end do

print *,'Printing Cpp Factor'
do iq=1,nq
   str_fac_c(iq)=str_ncpp(iq)/str_cpp(iq)
end do
write (*,'(4f8.4,4f15.2)') (str_fac_c(j),j=1,nq),(str_cpp(iq),iq=1,nq)

do iq=1,nq
   do istr=1,nstrata
      do i=1,2
         nrpwrev(iq)=nrpwrev(iq)+strrev_nc(istr,iq,i)
      end do
   end do
   nrpwrev(iq)=nrpwrev(iq)+str_ncpp(iq)
end do

do iq=1,nq
   do istr=1,nstrata
      do i=1,2
         print*,'nrpwrev2 is ',nrpwrev2(iq),istr,iq,i
         print*,'strrev_pb is ',strrev_pb(istr,iq,i),istr,iq,i
         print*,'str_fac_n is ',str_fac_n(istr,iq,i),istr,iq,i
         nrpwrev2(iq)=nrpwrev2(iq)+(strrev_pb(istr,iq,i)*str_fac_n(istr,iq,i))
      end do
   end do
   nrpwrev2(iq)=nrpwrev2(iq)+(str_cpp(iq)*str_fac_c(iq))
end do



print *,'ToTal Revenue from NCTB'
write (*,'(4f15.2)') (nrpwrev(j),j=1,nq)
print *,'ToTal Revenue from NCTB'
write (*,'(4f15.2)') (nrpwrev2(j),j=1,nq)

print *, 'Total Revenue from RPW'
write (*,'(4f15.2)')( rpwrev(j),j=1,nq)

do iq=1,nq
   nrpwrev(iq)=nrpwrev(iq)+govtot(iq)
end do

print *,'ToTal Revenue from NCTB with goverment'
write (*,'(4f15.2)') (nrpwrev(j),j=1,nq)

print *,'Control Factors are'
do iq=1,nq
   rpw_fac(iq)=rpwrev(iq)/nrpwrev(iq)
end do

write (*,'(4f8.6)') (rpw_fac(j),j=1,nq)

print *,'Starting to build the matrix'

do iq=1,nq
   do istr=1,nstrata
      do i=1,2
         fstr_ncpp(istr,iq,i)=rpw_fac(iq)*str_fac_n(istr,iq,i)
      end do
   end do
   fstr_cpp(iq)=rpw_fac(iq)*str_fac_c(iq)
end do

print *,'The matrix for control factors Non-CPP'
do istr=1,nstrata
   do i=1,2
      write (*,'(i3,i2,4f12.6)') istr,i,(fstr_ncpp(istr,j,i),j=1,nq)
   end do
end do

print *,'The matrix for control factors for CPP'
write (*,'(4f12.6)') (fstr_cpp(j),j=1,nq)

open(30,file='Control.In.NCTB.NonCpp')
do i=1,nstrata
   write (30,'(i3,4f12.8)') i,(str_fac_n(i,j,1),j=1,nq)
end do
close (30)
open(30,file='Control.Out.NCTB.NonCpp')
```

```
      do i=1,nstrata
         write (30,'(i3,4f12.8)') i,(str_fac_n(i,j,2),j=1,nq)
      end do
      close (30)
      open(30,file='Control.Matrix.NCTB.Cpp')
      write (30,'(4f12.8)') (str_fac_c(j),j=1,nq)
      close (30)
      open(30,file='Control.Matrix.RPW')
      write (30,'(4f12.8)') (rpw_fac(j),j=1,nq)
      close (30)

      end


c ---------------- ------------------------------------------------------------

      function strataic(istr)

      integer*4   istr
      integer*4   strataic

c      print*,'inside sub ',istr
      if (istr.le.6) then
         strataic=1
      else if (istr.le.9) then
         strataic=2
      else if (istr.le.13) then
         strataic=3
      else if (istr.le.16) then
         strataic=4
      else if (istr.le.18) then
         strataic=5
      else if (istr.eq.19) then
         strataic=6
      else if (istr.eq.20) then
         strataic=7
      else if (istr.eq.21) then
         strataic=8
      else if (istr.eq.22) then
         strataic=9
      else
         print *,'strata not found, istr = ',istr
         strataic=9
      end if

      return
      end
c ------ ------------------ ---------- ------------------------------------

      function stratafind(istr)

      integer*4   istr
      integer*4   stratafind

      if (istr.le.14) then
         stratafind=1
      else if (istr.le.15) then
         stratafind=2
      else if (istr.eq.16) then
         stratafind=3
      else if (istr.eq.17) then
         stratafind=4
      else if (istr.eq.18) then
         stratafind=5
      else if (istr.eq.19) then
         stratafind=6
      else if (istr.eq.20) then
         stratafind=7
      else if (istr.eq.21) then
         stratafind=8
      else if (istr.eq.22) then
         stratafind=9
      else
         print *,'strata not found, istr = ',istr
         stratafind=9
      end if

      return
      end
```

```
      program roll2nd_99_inf_weight

      implicit none

      integer*4     h,i,j,k,nfin,nchk,l
      integer*4     ncls,nq,nwe,s
      integer*4     nvips
      integer*4     nfin_ic,nfin_oc
      integer*4     nstrata,npub,iwgt,nwgt,wc0cnt,wivip


      parameter     (nchk=149)
      parameter     (ncls=4)
      parameter     (nq=4)
      parameter     (nwe=4)
      parameter     (nwgt=21)
      parameter     (nfin=7389)
      parameter     (nvips=84)
      parameter     (npub=25759)
      parameter     (nstrata=22)

      Character strorage

      character*13 tno          !  transaction number (1:7) julian date and year
      character*6  fin          !  finance number
      character*5  vip          !  VIP code
      character*1  uspsst       !  USPS status - M=master, P=pending
      character*7  uspsno       !  USPS number
      character*1  uspsst_re    !  Related USPS status
      character*7  uspsno_re    !  Related USPS number
      character*1  Prc_cat      !  Process Category
      character*1   rty
      character*4   vipmap(nchk)
      character*6   tusps
      character*6   fins(nfin)
      character*6   pubs(npub)
      character*6   tdate,jul_to_txt
      character*5   vipname
      character*4244 rec
      character*8   tdate2

      character*5 vipd(nvips)   ! vip codes

      integer*4     numvip      ! integer value of clines,number of vips to follow
      integer*4     il          ! index for vip detail
      integer*4     ier         ! read error iostat
      integer*4     cnt         ! counter for number of transaction read
      integer*4     searchc
      integer*4     vipty,ivip
      integer*4     ipub
      integer*4     vipmap_in(nchk,5)
      integer*4     icat,procescat
      integer*4     iq,qind
      integer*4     ic_strata(nfin),oc_strata(nfin)
      integer*4     ifin
      integer*4     a_pub(npub),itype
      integer*4     stratafind,strataic,wgtfun


      real*8        vippc(nvips) ! VIP pieces
      real*8        vipwt(nvips) ! VIP weight
      real*8        viprev(nvips) ! VIP revenue
      real*8        vipcop(nvips) ! VIP copies
      real*8        chk
      real*8        inf_ic(nstrata,nq)
      real*8        inf_oc(nstrata,nq)
      real*8        inf_cpp(nq)
      real*8        inf_rpw(8)
      real*8        ifact(2)
      real*8        revchk(nfin,nq)
      real*8        vipinfo(nwe,nchk,nq,ncls,4,nwgt) ! process catg, VIP, quarter, subclass, rpwc, weight increment
      real*8        ictr(nstrata,nq),octr(nstrata,nq),cppictr(nq),cppoctr(nq)
      real*8        wc,wcvipwt,wcvipcop

      logical       recls
      logical       debug

      do i=1,nwe
         do j=1,nchk
            do il=1,nq
```

```
            do k=1,ncls
               do l=1,4
                  do iwgt = 1,nwgt
                     vipinfo(i,j,il,k,l,iwgt)=0
                  end do
               end do
            end do
         end do
      end do
   end do
   do i = 1,nstrata
      do iq = 1,nq
         ictr(i,iq) = 0.0
         octr(i,iq) = 0.0
         cppictr(iq) = 0.0
         cppoctr(iq) = 0.0
      end do
   end do

   do i=1,nfin
      do j=1,nq
         revchk(i,j)=0
      end do
   end do

   wc0cnt = 0

   debug=.true.
   ier=0
   open (20,file='seccon.950101.950931')
   do i=1,nchk-2
      read(20,'(6x,a4,3i3,i2,i3)',iostat=ier) vipmap(i),(vipmap_in(i,j),j=1,5)
   end do
   vipmap(nchk-1) = '9998'
   do j = 1,5
      vipmap_in(nchk-1,j) = 1
   end do
   vipmap(nchk) = '9999'
   do j = 1,5
      vipmap_in(nchk,j) = 1
   end do

   print *,'Finished reading the vip code map',ier
   close(20)

   open(10,file='fin_strata.map')
15 format(a6,i3,i3)
   do i=1,nfin
      read(10,15) fins(i),ic_strata(i),oc_strata(i)
      ic_strata(i) = strataic(ic_strata(i))
      oc_strata(i) = stratafind(oc_strata(i))

   end do
   close(10)

   open(30,file='pubs.map')
   do ipub=1,npub
      read(30,'(a6,3x,i1)') pubs(ipub),a_pub(ipub)
   end do
   close(30)

   ier=0
   open (20,file='Control.In.NCTB.NonCpp')
   do i=1,nstrata
      read(20,'(3x,4f12.8)',iostat=ier) (inf_ic(i,j),j=1,4)
   end do
   close (20)
   print *,'Finished reading the IC Inflation Factors',ier

   if (debug) then
      do i=1,nstrata
         write (*,'(4f12.8)') (inf_ic(i,j),j=1,4)
      end do
   end if

   ier=0
   open (20,file='Control.Out.NCTB.NonCpp')
   do i=1,nstrata
      read(20,'(3x,4f12.8)',iostat=ier) (inf_oc(i,j),j=1,4)
   end do
```

```
close (20)
print *,'Finished reading the OC Inflation Factors',ier
if (debug) then
   do i=1,nstrata
      write (*,'(4f12.8)') (inf_oc(i,j),j=1,4)
   end do
end if

ier=0
open(20,file='Control.Matrix.RPW.')
read(20,'(8f12.8)',iostat=ier) (inf_rpw(j),j=1,8)
close(20)
print *,'Finished reading the RPW Inflation Factors',ier
if (debug) write (*,'(8f12.8)') (inf_rpw(j),j=1,8)

ier=0
open(20,file='Control.Matrix.NCTB.Cpp')
read(20,'(4f12.8)',iostat=ier) (inf_cpp(j),j=1,4)
close(20)
print *,'Finished reading the CPP Inflation Factors',ier
if (debug) write (*,'(4f12.8)') (inf_cpp(j),j=1,4)

recls = .true.
ier=0
open(80,file='newdata.data')
do while (ier.eq.0)

   read(80,'(a4244)',end=100,iostat=ier) rec
   read (rec(1:6),'(a6)') fin
   read (rec(7:7),'(a1)') rty
   read (rec(10:22),'(a13)') tno
   read (rec(23:23),'(a1)') uspsst
   read (rec(24:30),'(a7)') uspsno
   read (rec(37:37),'(a1)') uspsst_re
   read (rec(38:44),'(a7)') uspsno_re
   read (rec(69:77),'(f9.6)') wc
   read (rec(103:103),'(a1)') prc_cat
   read (rec(128:130),'(i3)') numvip

   if ((rty.ne.'G').and.(rty.ne.'D')) then

      cnt = cnt + 1
      do il = 1, numvip
         read (rec((131+(il-1)*49):(179+(il-1)*49)),'(a5,f12.2,f12.0,2f10.0)')
&           vipd(il),viprev(il),vippc(il),vipwt(il),vipcop(il)
      end do

      if (uspsst.eq.'M') then
         tusps=uspsno_re(2:7)
      else if ((uspsno(7:7).eq.' ').and.(uspsno(1:1).ne.' ')) then
         tusps=uspsno(1:6)
      else
         tusps = uspsno(2:7)
      end if

      ipub = searchc(pubs,npub,tusps)
      if (ipub.eq.0) then
         itype = 7
      else
         itype = a_pub(ipub)
      end if

      if (fin.eq.'115605') fin='118929'
      if (fin.eq.'118675') fin='115854'
      if (fin.eq.'161274') fin='161540'
      if (fin.eq.'162865') fin='161544'
      if (fin.eq.'166027') fin='161540'
      if (fin.eq.'414582') fin='410636'
      if (fin.eq.'484143') fin='484149'
      if (fin.eq.'512704') fin='516540'
      if (fin.eq.'411626') fin='416088'

      do i=1,2
         ifact(i)=0
      end do
      tdate = jul_to_txt(tno(3:7))
      if (tdate(1:2).eq.'00') then
         tdate2(1:2) = '20'
         tdate2(3:8) = tdate(1:6)
      else
```

```
            tdate2(1:2) = '19'
            tdate2(3:8) = tdate(1:6)
         end if

         iq=qind(tdate2)
         ifin=searchc(fins,nfin,fin)
         if (iq.gt.0) then
            if (ifin.gt.0) then
               if (fin.ne.'355826') then
                  if ((rty.eq.'G').or.(rty.eq.'D')) then
                     ifact(1) = inf_rpw(2*iq-1)
                     ifact(2) = inf_rpw(2*iq)
                  else if ((rty.eq.'C').or.(rty.eq.'P')) then
                     if (oc_strata(ifin).ne.0) then
                        ifact(1) = inf_oc(oc_strata(ifin),iq)*inf_rpw(2*iq-1)
                     else if (ic_strata(ifin).ne.0) then
                        ifact(1) = inf_oc(ic_strata(ifin),iq)*inf_rpw(2*iq-1)
                     else
                        ifact(1) = inf_oc(22,iq)*inf_rpw(2*iq-1)
                     end if
                     if (ic_strata(ifin).ne.0) then
                        ifact(2) = inf_ic(ic_strata(ifin),iq)*inf_rpw(2*iq)
                     else if (oc_strata(ifin).ne.0) then
                        ifact(2) = inf_ic(oc_strata(ifin),iq)*inf_rpw(2*iq)
                     else

                        ifact(2) = inf_ic(22,iq)*inf_rpw(2*iq)
                     end if
                  else
                     print *,'type not found, rty = ',rty
                  end if
               else
                  if ((rty.eq.'G').or.(rty.eq.'D')) then
                     ifact(1) = inf_rpw(2*iq-1)
                     ifact(2) = inf_rpw(2*iq)
                  else if ((rty.eq.'C').or.(rty.eq.'P')) then
                     ifact(1) = inf_cpp(iq)*inf_rpw(2*iq-1)
                     ifact(2) = inf_cpp(iq)*inf_rpw(2*iq)
                  end if
               end if
            else
               print *,'fin not found, fin = ',fin
            end if
         end if

         if (numvip.ge.1) then
            if (vipd(1)(1:1).ne.'Z') then
               read(vipd(1)(1:1),'(i1)') vipty
               if ((vipty.eq.5).and.(numvip.ge.2)) then
                  read(vipd(2)(1:1),'(i1)') vipty
               else if ((vipty.eq.5).and.(numvip.eq.1)) then
                  vipty=9
               else if (vipty.lt.5) then
                  print *,'Read vip type is wrong - ',vipd(1)(1:1),' -Type is: ',vipty
                  vipty=9
               end if
            else
               vipty=9
               print *,'Read vip type is wrong - ',vipd(1)(1:1),' -Type is: ',vipty
            end if
         else
            Print *,'No Vip detail'
         end if

         if (itype.eq.2) then
            vipty=7
         end if

         if((tusps.eq.'590110').or.(tusps.eq.'399680').or.
     &      (tusps.eq.'167360').or.(tusps.eq.'311190').or.
     &      (tusps.eq.'006524').or.(tusps.eq.'710740').or.
     &      (tusps.eq.'265820').or.(tusps.eq.'074560').or.
     &      (tusps.eq.'696610').or.(tusps.eq.'000061').or.
     &      (tusps.eq.'757310').or.(tusps.eq.'007753').or.
     &      (tusps.eq.'374580').or.(tusps.eq.'455380').or.
     &      (tusps.eq.'853680')) then
            if ((vipty.eq.8).and.(debug)) then
               print *,'This USPS number is mailed under classroom but authorized np',tusps
               vipty=vipty-1
            end if
```

```
      else if((tusps.eq.'121870').or.(tusps.eq.'777240')) then

         if((vipty.eq.7).and.debug) then
            print *,'This USPS number is mailed under non-pr but classroom',tusps
            vipty=vipty+1
         endif
      endif


      do i=1,numvip
         ivip=searchc(vipmap,nchk,vipd(i)(2:5))
         if (ivip.ne.0) then
            If ((ifin.gt.0).and.(iq.gt.0).and.((vipmap_in(ivip,2).eq.1).or.
     &          (vipmap_in(ivip,2).eq.-1))) then
               revchk(ifin,iq)=revchk(ifin,iq)+(viprev(i)*vipmap_in(ivip,2))
            end if
         else
            print *,'Unidentified Vip...Fatal..',vipd(i)(2:5)
         end if
      end do
      icat = procescat(prc_cat)

      vipty=vipty-5

      do i=1,numvip
         ivip=searchc(vipmap,nchk,vipd(i)(2:5))
         if (recls) then
            if (ivip.gt.0) then
               if (ivip.eq.14) then ! 1430
                  ivip=15
               else if (ivip.eq.15) then ! 1440
                  ivip=14
               else if (ivip.eq.16) then ! 1444
                  ivip=14
               else if (ivip.eq.17) then ! 1445
                  ivip=14
               else if (ivip.eq.18) then ! 1450 IC NA 5 dig - Needs to be split
                  ivip=143
               else if (ivip.eq.19) then ! 1460 IC Auto 5 dig L - In right spot
                  ivip=18
               else if (ivip.eq.20) then ! 1470 IC Auto 5 dig F - Needs to be split
                  ivip=144
               else if (ivip.eq.21) then ! 1474 IC Auto 5 dig F - Needs to be split
                  ivip=144
               else if (ivip.eq.22) then ! 1475 IC Auto 5 dig F - Needs to be split
                  ivip=144
               else if (ivip.eq.23) then ! 1480
                  ivip=19
               else if (ivip.eq.24) then ! 1490
                  ivip=20
               else if (ivip.eq.25) then ! 1500
                  ivip=23
               else if (ivip.eq.66) then ! 2174
                  ivip=65
               else if (ivip.eq.68) then ! 2184
                  ivip=67
               else if (ivip.eq.69) then ! 2185
                  ivip=67
               else if (ivip.eq.72) then ! 2204
                  ivip=71
               else if (ivip.eq.74) then ! 2214
                  ivip=73
               else if (ivip.eq.75) then ! 2215
                  ivip=73
               else if (ivip.eq.76) then ! 2220 5 dig NA - Needs to be split
                  ivip=145
               else if (ivip.eq.77) then ! 2230 5 dig Auto L - Needs to be split
                  ivip=146
               else if (ivip.eq.78) then ! 2240 5 dig Auto F - Needs to be split
                  ivip=147
               else if (ivip.eq.79) then ! 2244 5 dig Auto F - Needs to be split
                  ivip=147
               else if (ivip.eq.80) then ! 2245 5 dig Auto F - Needs to be split
                  ivip=147
               else if (ivip.eq.81) then ! 2250
                  ivip=76
               else if (ivip.eq.82) then ! 2260
                  ivip=77
               else if (ivip.eq.83) then ! 2270
```

```
                  ivip=78
               else if (ivip.eq.128) then ! 4221 5 dig NA - Needs to be split
                  ivip=122
               else if (ivip.eq.129) then ! 4231 5 dig Auto L - Needs to be split
                  ivip= 123
               else if (ivip.eq.130) then ! 4241 5 dig Auto F - Needs to be split
                  ivip=125
               else if (ivip.eq.131) then ! 4244 5 dig Auto F - Needs to be split
                  ivip=125
               else if (ivip.eq.132) then ! 4245 5 dig Auto F - Needs to be split
                  ivip=125
               else if (ivip.eq.133) then ! 4251
                  ivip=128
               else if (ivip.eq.134) then ! 4261
                  ivip=129
               else if (ivip.eq.135) then ! 4271
                  ivip=130
               end if
            end if
         end if
C  Start rolling up the vip data
         if (rec(69:77).eq.'000000000') then

            wcvipwt = 0
            wcvipcop = 0
            do wivip = 1,numvip
               wcvipwt = wcvipwt + vipwt(wivip)
               wcvipcop = wcvipcop + vipcop(wivip)
            end do
            wc = wcvipwt/wcvipcop
            if ((wc.eq.0.0).or.(wc.gt.10)) then
               print*,'wt out of bounds ',wc,' ',rec(1:100)
            end if
         end if

         if ((wc.lt.0.00001)) then
            print*,'wt out of bounds2 ',wc,' ',rec(1:100)
         end if

         iwgt = wgtfun(wc)

         if ((ivip.gt.0).and.(icat.gt.0).and.(iq.gt.0).and.
     &        (ivipty.gt.0).and.(ivipty.le.4))) then
            if ((ivip.ne.0).and.(ivip.ne.93).and.(ivip.ne.96)) then !remove foreign
               if ((ivip.le.28).or.(ivip.eq.148).or.(ivip.eq.143).or.(ivip.eq.144))  then


                  vipinfo(icat,ivip,iq,vipty,1,iwgt) = vipinfo(icat,ivip,iq,vipty,1,iwgt) + viprev(i)*ifact(2)
                  vipinfo(icat,ivip,iq,vipty,2,iwgt) = vipinfo(icat,ivip,iq,vipty,2,iwgt) + vippc(i)*ifact(2)
                  vipinfo(icat,ivip,iq,vipty,3,iwgt) = vipinfo(icat,ivip,iq,vipty,3,iwgt) + vipcop(i)*ifact(2)
                  vipinfo(icat,ivip,iq,vipty,4,iwgt) = vipinfo(icat,ivip,iq,vipty,4,iwgt) + vipwt(i)*ifact(2)
                  if (fin.ne.'355826') then
                     ictr(ic_strata(ifin),iq) = ictr(ic_strata(ifin),iq) + (viprev(i)*vipmap_in(ivip,2))
                  else
                     cppictr(iq) = cppictr(iq) + (viprev(i)*vipmap_in(ivip,2))
                  end if
               else

                  vipinfo(icat,ivip,iq,vipty,1,iwgt) = vipinfo(icat,ivip,iq,vipty,1,iwgt) + viprev(i)*ifact(1)
                  vipinfo(icat,ivip,iq,vipty,2,iwgt) = vipinfo(icat,ivip,iq,vipty,2,iwgt) + vippc(i)*ifact(1)
                  vipinfo(icat,ivip,iq,vipty,3,iwgt) = vipinfo(icat,ivip,iq,vipty,3,iwgt) + vipcop(i)*ifact(1)
                  vipinfo(icat,ivip,iq,vipty,4,iwgt) = vipinfo(icat,ivip,iq,vipty,4,iwgt) + vipwt(i)*ifact(1)
                  if (fin.ne.'355826') then
                     octr(oc_strata(ifin),iq) = octr(oc_strata(ifin),iq) + (viprev(i)*vipmap_in(ivip,2))
                  else
                     cppoctr(iq) = cppoctr(iq) + (viprev(i)*vipmap_in(ivip,2))
                  end if
               end if
            end if
         else
            If (debug) then
               if (iq.gt.0) then
                  print *,' Error: fatal.., '
                  print *,'icat number: ',icat
                  print *,'quarter: ',iq
                  print *,'Vip index: ',ivip
                  print *,'Viptype: ',vipty
               end if
            end if
```

```
            end if
         end do
      end if
   end do
100   print *,ier
   close (80)

   recls = .false.
   ier=0
   open(80,file='olddata.data')
   do while (ier.eq.0)

      read(80,'(a4244)',end=200,iostat=ier) rec
      read (rec(1:6),'(a6)') fin
      read (rec(7:7),'(a1)') rty
      read (rec(10:22),'(a13)') tno
      read (rec(23:23),'(a1)') uspsst
      read (rec(24:30),'(a7)') uspsno
      read (rec(37:37),'(a1)') uspsst_re
      read (rec(38:44),'(a7)') uspsno_re
      read (rec(69:77),'(f9.6)') wc
      read (rec(103:103),'(a1)') prc_cat
      read (rec(128:130),'(i3)') numvip

      if ((rty.ne.'G').and.(rty.ne.'D')) then

         cnt = cnt + 1
         do il = 1, numvip
            read (rec((131+(il-1)*49):(179+(il-1)*49)),'(a5,f12.2,f12.0,2f10.0)')
     &         vipd(il),viprev(il),vippc(il),vipwt(il),vipcop(il)
         end do

         if (uspsst.eq.'M') then
            tusps=uspsno_re(2:7)
         else if ((uspsno(7:7).eq.' ').and.(uspsno(1:1).ne.' ')) then
            tusps=uspsno(1:6)
         else
            tusps = uspsno(2:7)
         end if

         ipub = searchc(pubs,npub,tusps)
         if (ipub.eq.0) then
            itype = 7
         else
            itype = a_pub(ipub)
         end if

         if (fin.eq.'115605') fin='118929'
         if (fin.eq.'118675') fin='115854'
         if (fin.eq.'161274') fin='161540'
         if (fin.eq.'162865') fin='161544'
         if (fin.eq.'166027') fin='161540'
         if (fin.eq.'414582') fin='410636'
         if (fin.eq.'484143') fin='484149'
         if (fin.eq.'512704') fin='516540'
         if (fin.eq.'411626') fin='416088'

         do i=1,2
            ifact(i)=0
         end do
         tdate = jul_to_txt(tno(3:7))
         if (tdate(1:2).eq.'00') then
            tdate2(1:2) = '20'
            tdate2(3:8) = tdate(1:6)
         else
            tdate2(1:2) = '19'
            tdate2(3:8) = tdate(1:6)
         end if

         iq=qind(tdate2)
         ifin=searchc(fins,nfin,fin)
         if (iq.gt.0) then
            if (ifin.gt.0) then
               if (fin.ne.'355826') then
                  if ((rty.eq.'G').or.(rty.eq.'D')) then
                     ifact(1) = inf_rpw(2*iq-1)
                     ifact(2) = inf_rpw(2*iq)
                  else if ((rty.eq.'C').or.(rty.eq.'P')) then
                     if (oc_strata(ifin).ne.0) then
                        ifact(1) = inf_oc(oc_strata(ifin),iq)*inf_rpw(2*iq-1)
```

```
                else if (ic_strata(ifin).ne.0) then
                   ifact(1) = inf_oc(ic_strata(ifin),iq)*inf_rpw(2*iq-1)
                else
                   ifact(1) = inf_oc(22,iq)*inf_rpw(2*iq-1)
                end if
                if (ic_strata(ifin).ne.0) then
                   ifact(2) = inf_ic(ic_strata(ifin),iq)*inf_rpw(2*iq)
                else if (oc_strata(ifin).ne.0) then
                   ifact(2) = inf_ic(oc_strata(ifin),iq)*inf_rpw(2*iq)
                else
                   ifact(2) = inf_ic(22,iq)*inf_rpw(2*iq)
                end if
             else
                print *,'type not found, rty = ',rty
             end if
          else
             if ((rty.eq.'G').or.(rty.eq.'D')) then
                ifact(1) = inf_rpw(2*iq-1)
                ifact(2) = inf_rpw(2*iq)
             else if ((rty.eq.'C').or.(rty.eq.'P')) then
                ifact(1) = inf_cpp(iq)*inf_rpw(2*iq-1)
                ifact(2) = inf_cpp(iq)*inf_rpw(2*iq)
             end if
          end if
       else
          print *,'fin not found, fin = ',fin
       end if
    end if

    if (numvip.ge.1) then
       if (vipd(1)(1:1).ne.'Z') then
          read(vipd(1)(1:1),'(i1)') vipty
          if ((vipty.eq.5).and.(numvip.ge.2)) then
             read(vipd(2)(1:1),'(i1)') vipty
          else if ((vipty.eq.5).and.(numvip.eq.1)) then
             vipty=9
          else if (vipty.lt.5) then
             print *,'Read vip type is wrong - ',vipd(1)(1:1),' -Type is: ',vipty
             vipty=9
          end if
       else
          vipty=9
          print *,'Read vip type is wrong - ',vipd(1)(1:1),' -Type is: ',vipty
       end if
    else
       Print *,'No Vip detail'
    end if

    if (itype.eq.2) then
       vipty=7
    end if

    if((tusps.eq.'590110').or.(tusps.eq.'399680').or.
&      (tusps.eq.'167360').or.(tusps.eq.'311190').or.
&      (tusps.eq.'006524').or.(tusps.eq.'710740').or.
&      (tusps.eq.'265820').or.(tusps.eq.'074560').or.
&      (tusps.eq.'696610').or.(tusps.eq.'000061').or.
&      (tusps.eq.'757310').or.(tusps.eq.'007753').or.
&      (tusps.eq.'374580').or.(tusps.eq.'455380').or.
&      (tusps.eq.'853680')) then
       if ((vipty.eq.8).and.(debug)) then
          print *,'This USPS number is mailed under classroom but authorized np',tusps
          vipty=vipty-1
       end if

    else if((tusps.eq.'121870').or.(tusps.eq.'777240')) then
       !CHANGE PUBS THAT WERE MAILING NONPROFIT BUT WERE
       !AUTHORIZED CLASSROOM

       if((vipty.eq.7).and.debug) then
          print *,'This USPS number is mailed under non-pr but classroom',tusps
          vipty=vipty+1
       endif
    endif

    do i=1,numvip
       ivip=searchc(vipmap,nchk,vipd(i)(2:5))
       if (ivip.ne.0) then
          If ((ifin.gt.0).and.(iq.gt.0).and.((vipmap_in(ivip,2).eq.1).or.
&            (vipmap_in(ivip,2).eq.-1))) then
```

```
                  revchk(ifin,iq)=revchk(ifin,iq)+(viprev(i)*vipmap_in(ivip,2))
               end if
            else
               print *,'Unidentified Vip...Fatal..',vipd(i)(2:5)
            end if
         end do
         icat = procescat(prc_cat)

         vipty=vipty-5

         do i=1,numvip
            ivip=searchc(vipmap,nchk,vipd(i)(2:5))
            if (rec(69:77).eq.'000000000') then
               wcvipwt = 0
               wcvipcop = 0
               do wivip = 1,numvip
                  wcvipwt = wcvipwt + vipwt(wivip)
                  wcvipcop = wcvipcop + vipcop(wivip)
               end do
               wc = wcvipwt/wcvipcop
               if ((wc.eq.0.0).or.(wc.gt.10)) then
                  print*,'wt out of bounds ',wc,' ',rec(1:100)
               end if
            end if

            if ((wc.le.0.00001)) then
               print*,'wt out of bounds2 ',wc,' ',rec(1:100)
            end if
            iwgt = wgtfun(wc)

C  Start rolling up the vip data
            if ((ivip.gt.0).and.(icat.gt.0).and.(iq.gt.0).and.
     &          ((vipty.gt.0).and.(vipty.le.4))) then
               if ((ivip.ne.0).and.(ivip.ne.93).and.(ivip.ne.96)) then !remove foreign
                  if ((ivip.le.28).or.(ivip.eq.148).or.(ivip.eq.143).or.(ivip.eq.144))  then


                     vipinfo(icat,ivip,iq,vipty,1,iwgt) = vipinfo(icat,ivip,iq,vipty,1,iwgt) + viprev(i)*ifact(2)
                     vipinfo(icat,ivip,iq,vipty,2,iwgt) = vipinfo(icat,ivip,iq,vipty,2,iwgt) + vippc(i)*ifact(2)
                     vipinfo(icat,ivip,iq,vipty,3,iwgt) = vipinfo(icat,ivip,iq,vipty,3,iwgt) + vipcop(i)*ifact(2)
                     vipinfo(icat,ivip,iq,vipty,4,iwgt) = vipinfo(icat,ivip,iq,vipty,4,iwgt) + vipwt(i)*ifact(2)
                     if (fin.ne.'355826') then
                        ictr(ic_strata(ifin),iq) = ictr(ic_strata(ifin),iq) + (viprev(i)*vipmap_in(ivip,2))
                     else
                        cppictr(iq) = cppictr(iq) + (viprev(i)*vipmap_in(ivip,2))
                     end if
                  else

                     vipinfo(icat,ivip,iq,vipty,1,iwgt) = vipinfo(icat,ivip,iq,vipty,1,iwgt) + viprev(i)*ifact(1)
                     vipinfo(icat,ivip,iq,vipty,2,iwgt) = vipinfo(icat,ivip,iq,vipty,2,iwgt) + vippc(i)*ifact(1)
                     vipinfo(icat,ivip,iq,vipty,3,iwgt) = vipinfo(icat,ivip,iq,vipty,3,iwgt) + vipcop(i)*ifact(1)
                     vipinfo(icat,ivip,iq,vipty,4,iwgt) = vipinfo(icat,ivip,iq,vipty,4,iwgt) + vipwt(i)*ifact(1)
                     if (fin.ne.'355826') then
                        octr(oc_strata(ifin),iq) = octr(oc_strata(ifin),iq) + (viprev(i)*vipmap_in(ivip,2))
                     else
                        cppoctr(iq) = cppoctr(iq) + (viprev(i)*vipmap_in(ivip,2))
                     end if
                  end if
               end if
            else
               If (debug) then
                  if (iq.gt.0) then
                     print *,' Error: fatal.., '
                     print *,'icat number: ',icat
                     print *,'quarter: ',iq
                     print *,'Vip index: ',ivip
                     print *,'Viptype: ',vipty
                  end if
               end if
            end if
         end do
      end if
   end do
200   print *,ier
      close (80)

      print*,'wc 0 count is ',wc0cnt

      do i = 1,10
```

```
      write(*,'(4f15.2)') ictr(i,1),ictr(i,2),ictr(i,3),ictr(i,4)
   end do
   do i = 1,10
      write(*,'(4f15.2)') octr(i,1),octr(i,2),octr(i,3),octr(i,4)
   end do
   write(*,'(4f15.2)') cppictr(1),cppictr(2),cppictr(3),cppictr(4)
   write(*,'(4f15.2)') cppoctr(1),cppoctr(2),cppoctr(3),cppoctr(4)

   open (66,file='Rolled3.inf.00.weight')
   do i=1,nchk
      vipname='6'//vipmap(i)
      do j=1,nwe
         do iwgt = 1,nwgt
            write(66,'(a5,i2,i3,16f15.2)') vipname,j,iwgt,((vipinfo(j,i,s,1,k,iwgt),k=1,4),s=1,nq)
         end do
      end do
   end do
   do i=1,nchk
      vipname='7'//vipmap(i)
      do j=1,nwe
         do iwgt = 1,nwgt
            write(66,'(a5,i2,i3,16f15.2)') vipname,j,iwgt,((vipinfo(j,i,s,2,k,iwgt),k=1,4),s=1,nq)
         end do
      end do
   end do
   do i=1,nchk
      vipname='8'//vipmap(i)
      do j=1,nwe
         do iwgt = 1,nwgt
            write(66,'(a5,i2,i3,16f15.2)') vipname,j,iwgt,((vipinfo(j,i,s,3,k,iwgt),k=1,4),s=1,nq)
         end do
      end do
   end do
   do i=1,nchk
      vipname='9'//vipmap(i)
      do j=1,nwe
         do iwgt = 1,nwgt
            write(66,'(a5,i2,i3,16f15.2)') vipname,j,iwgt,((vipinfo(j,i,s,4,k,iwgt),k=1,4),s=1,nq)
         end do
      end do
   end do

   open(35,file='Rev3.Check.wei.weight')
   do i=1,nfin
      write(35,'(a6,4f18.2)') fins(i),(revchk(i,j),j=1,nq)
   end do

   end

!-------------- ---------------------------------------------
   function qind(date)

   integer*4 qind
   character*8 date

   if (date.ge.'19990911'.and.date.le.'19991203') then
      qind = 1
   else if (date.ge.'19991204'.and.date.le.'20000225') then
      qind = 2
   else if (date.ge.'20000226'.and.date.le.'20000519') then
      qind = 3
   else if (date.ge.'20000520'.and.date.le.'20000908') then
      qind = 4
   else
      qind = 0
      print *,'bad date',date
   end if

   return
   end
!-----  -------- ------------------------------------------

   function procescat(Prc_cat)

   character*1 Prc_cat
   integer*2   procescat

   if (prc_cat.eq.'0') then
      procescat=1
   else if (prc_cat.eq.'1') then
```

```
         procescat=2
      else if (prc_cat.eq.'2') then
         procescat=3
      else
         procescat=4
      end if

      return
      end

C     --- ... ----------------------------------------------------------------

      function strataic(istr)

      integer*4   istr
      integer*4   strataic

C      print*,'inside sub ',istr
      if (istr.le.6) then
         strataic=1
      else if (istr.le.9) then
         strataic=2
      else if (istr.le.13) then
         strataic=3
      else if (istr.le.16) then
         strataic=4
      else if (istr.le.18) then
         strataic=5
      else if (istr.eq.19) then
         strataic=6
      else if (istr.eq.20) then
         strataic=7
      else if (istr.eq.21) then
         strataic=8
      else if (istr.eq.22) then
         strataic=9
      else
         print *,'strata not found, istr = ',istr
         strataic=9
      end if

      return
      end
C     ---- -------- ----------------------------------------------------------

      function stratafind(istr)

      integer*4   istr
      integer*4   stratafind

      if (istr.le.14) then
         stratafind=1
      else if (istr.le.15) then
         stratafind=2
      else if (istr.eq.16) then
         stratafind=3
      else if (istr.eq.17) then
         stratafind=4
      else if (istr.eq.18) then
         stratafind=5
      else if (istr.eq.19) then
         stratafind=6
      else if (istr.eq.20) then
         stratafind=7
      else if (istr.eq.21) then
         stratafind=8
      else if (istr.eq.22) then
         stratafind=9
      else
         print *,'strata not found, istr = ',istr
         stratafind=9
      end if

      return
      end


C     --- ... ----------------------------------------------------------------
      function wgtfun(wc)
```

```
      real*8      wc
      integer*4   wgtfun

      wgtfun = 0

      if (wc.le.(0.5/16)) then
         wgtfun = 1
      else if (wc.le.(1.0/16)) then
         wgtfun = 2
      else if (wc.le.(1.5/16)) then
         wgtfun = 3
      else if (wc.le.(2.0/16)) then
         wgtfun = 4
      else if (wc.le.(2.5/16)) then
         wgtfun = 5
      else if (wc.le.(3.0/16)) then
         wgtfun = 6
      else if (wc.le.(3.5/16)) then
         wgtfun = 7
      else if (wc.le.(4.0/16)) then
         wgtfun = 8
      else if (wc.le.(5.0/16)) then
         wgtfun = 9
      else if (wc.le.(6.0/16)) then
         wgtfun = 10
      else if (wc.le.(7.0/16)) then
         wgtfun = 11
      else if (wc.le.(8.0/16)) then
         wgtfun = 12
      else if (wc.le.(9.0/16)) then
         wgtfun = 13
      else if (wc.le.(10.0/16)) then
         wgtfun = 14
      else if (wc.le.(11.0/16)) then
         wgtfun = 15
      else if (wc.le.(12.0/16)) then
         wgtfun = 16
      else if (wc.le.(13.0/16)) then
         wgtfun = 17
      else if (wc.le.(14.0/16)) then
         wgtfun = 18
      else if (wc.le.(15.0/16)) then
         wgtfun = 19
      else if (wc.le.1) then
         wgtfun = 20
      else
         wgtfun = 21
c        print*,'large wgtfun ',wc
      end if

      return
      end
```

```
      program roll_bill_wgtinc.f

C     This Program Rolls Up the 2nd Class data by pub number, and billing determinants
C     (here we only use regular and non-profit).  Only outside county info and flat
C     shaped pieces are counted.
C     The totals are either by revenue, pieces, or weight, and the presort level.
C     This program uses the 2000 data in non-binary format
C     Make sure to look at the begining of the program to set certain variables

      implicit none

      integer*4    ier,i,nchk,ipcline,npcline,searchc,nwtline
      integer*4    ivip,iwtline,getwtline,getpcline,iq,nq,isub,nsub
      integer*4    ishape,nshape,irpw,nrpw,nwgt,iwgt
      parameter    (nsub=5)
      parameter    (npcline=10)
      parameter    (nwtline=10)
      parameter    (nchk=149)
      parameter    (nq=4)
      parameter    (nrpw=4)
      parameter    (nshape=4)
      parameter    (nwgt=22)
      character*4  vipmap(nchk),vipx
      real*8       pinfo(nq,nsub,nshape,npcline,nrpw,nwgt)
      real*8       rev1,rev2,rev3,rev4
      real*8       pcs1,pcs2,pcs3,pcs4
      real*8       wgt1,wgt2,wgt3,wgt4
      real*8       cps1,cps2,cps3,cps4,pdis(nq,nsub,nshape,nwgt)
      real*8       winfo(nq,nsub,nshape,nwtline,nrpw,nwgt)
      integer*4    vipc

      integer*4    vipmult(nchk)

      do iq = 1,nq
         do isub = 1,nsub
            do ishape = 1,nshape
               do ipcline = 1,npcline
                  do irpw = 1,nrpw
                     do iwtline = 1,nwtline
                        do iwgt = 1,nwgt
                           pinfo(iq,isub,ishape,ipcline,irpw,iwgt) = 0.0
                           winfo(iq,isub,ishape,iwtline,irpw,iwgt) = 0.0
                           pdis(iq,isub,ishape,iwgt) = 0.0
                        end do
                     end do
                  end do
               end do
            end do
         end do

      end do


      ier=0
      open (20,file='seccon.950101.950931')
      do i=1,nchk-2
         read(20,'(6x,a4,3x,i3)',iostat=ier) vipmap(i),vipmult(i)
      end do
      vipmap(nchk-1) = '9998'
      vipmult(nchk-1) = 1
      vipmap(nchk) = '9999'
      vipmult(nchk) = 1



      open (40,file='Rolled3.inf.00.weight')
10    format(i1,a4,i2,i3,16f15.2)

      ier = 0
      do while (ier.eq.0)
         read(40,40,iostat=ier,end=100) vipc,vipx,ishape,iwgt,rev1,pcs1,cps1,wgt1,rev2,pcs2,cps2,wgt2,
     &      rev3,pcs3,cps3,wgt3,rev4,pcs4,cps4,wgt4
         ivip=searchc(vipmap,nchk,vipx)
         ipcline=0
         iwtline=0
         iwtline=getwtline(ivip)
         ipcline=getpcline(ivip)

         isub = vipc - 5
```

```
      if ((iwtline.eq.99).and.(ipcline.eq.99).and.(vipmult(ivip).eq.1)) then
         if (rev1+rev2+rev3+rev4.ne.0) then
            print*,'problem with rec'
            print*,'iq is ',iq
            print*,'isub is ',isub
            print*,'ishape is ',ishape
            print*,'ipcline is ',ipcline,' ',vipx
            print*,'iwtline is ',iwtline,' ',vipx
         end if
      end if



      if ((vipx(1:1).ne.'9').or.(vipx(2:3).eq.'99')) then

         if (((vipx(2:3).ge.'36').and.(vipx(2:3).le.'53')).or.(vipx(2:4).eq.'998')) then ! In county
            isub=1
         else if ((vipx(2:3).ge.'55').and.(vipx(2:3).le.'57')) then
            isub=5
         else
            if ((isub.eq.4).or.(isub.eq.1))  then ! regular
               isub=2
            else if (isub.eq.3) then ! classroom
               isub=3
            else
               isub=4
            end if

         end if
      else
         if ((vipx(2:2).eq.'1').or.(vipx(2:2).eq.'2')) then
            isub = 1
         else
            if ((isub.eq.4).or.(isub.eq.1))  then ! regular
               isub=2
            else if (isub.eq.3) then ! classroom
               isub=3
            else
               isub=4
            end if
         end if
      end if

      if (vipmult(ivip).eq.-1) then
         pdis(1,isub,ishape,iwgt) = pdis(1,isub,ishape,iwgt) - rev1
         pdis(2,isub,ishape,iwgt) = pdis(2,isub,ishape,iwgt) - rev2
         pdis(3,isub,ishape,iwgt) = pdis(3,isub,ishape,iwgt) - rev3
         pdis(4,isub,ishape,iwgt) = pdis(4,isub,ishape,iwgt) - rev4
      end if


      if (((ipcline.gt.0).and.(ipcline.le.10)).and.(iq.gt.0).and.(isub.gt.0)
     &    .and.(ishape.gt.0))then

         if (vipmult(ivip).eq.1) then
            pinfo(1,isub,ishape,ipcline,1,iwgt) = pinfo(1,isub,ishape,ipcline,1,iwgt) + rev1
            pinfo(2,isub,ishape,ipcline,1,iwgt) = pinfo(2,isub,ishape,ipcline,1,iwgt) + rev2
            pinfo(3,isub,ishape,ipcline,1,iwgt) = pinfo(3,isub,ishape,ipcline,1,iwgt) + rev3
            pinfo(4,isub,ishape,ipcline,1,iwgt) = pinfo(4,isub,ishape,ipcline,1,iwgt) + rev4
            pinfo(1,isub,ishape,ipcline,2,iwgt) = pinfo(1,isub,ishape,ipcline,2,iwgt) + pcs1
            pinfo(2,isub,ishape,ipcline,2,iwgt) = pinfo(2,isub,ishape,ipcline,2,iwgt) + pcs2
            pinfo(3,isub,ishape,ipcline,2,iwgt) = pinfo(3,isub,ishape,ipcline,2,iwgt) + pcs3
            pinfo(4,isub,ishape,ipcline,2,iwgt) = pinfo(4,isub,ishape,ipcline,2,iwgt) + pcs4
            pinfo(1,isub,ishape,ipcline,3,iwgt) = pinfo(1,isub,ishape,ipcline,3,iwgt) + cps1
            pinfo(2,isub,ishape,ipcline,3,iwgt) = pinfo(2,isub,ishape,ipcline,3,iwgt) + cps2
            pinfo(3,isub,ishape,ipcline,3,iwgt) = pinfo(3,isub,ishape,ipcline,3,iwgt) + cps3
            pinfo(4,isub,ishape,ipcline,3,iwgt) = pinfo(4,isub,ishape,ipcline,3,iwgt) + cps4
            pinfo(1,isub,ishape,ipcline,4,iwgt) = pinfo(1,isub,ishape,ipcline,4,iwgt) + wgt1
            pinfo(2,isub,ishape,ipcline,4,iwgt) = pinfo(2,isub,ishape,ipcline,4,iwgt) + wgt2
            pinfo(3,isub,ishape,ipcline,4,iwgt) = pinfo(3,isub,ishape,ipcline,4,iwgt) + wgt3
            pinfo(4,isub,ishape,ipcline,4,iwgt) = pinfo(4,isub,ishape,ipcline,4,iwgt) + wgt4
         end if
      else
          print*,'problem with rec'
          print*,'iq is ',iq
          print*,'isub is ',isub
          print*,'ishape is ',ishape
          print*,'ipcline is ',ipcline,' ',vipx
      end if
```

```
         if (((iwtline.gt.0).and.(iwtline.le.10)).and.(iq.gt.0).and.(isub.gt.0)
     &       .and.(ishape.gt.0))then

            if (vipmult(ivip).eq.1) then
               winfo(1,isub,ishape,iwtline,1,iwgt) = winfo(1,isub,ishape,iwtline,1,iwgt) + rev1
               winfo(2,isub,ishape,iwtline,1,iwgt) = winfo(2,isub,ishape,iwtline,1,iwgt) + rev2
               winfo(3,isub,ishape,iwtline,1,iwgt) = winfo(3,isub,ishape,iwtline,1,iwgt) + rev3
               winfo(4,isub,ishape,iwtline,1,iwgt) = winfo(4,isub,ishape,iwtline,1,iwgt) + rev4
               winfo(1,isub,ishape,iwtline,2,iwgt) = winfo(1,isub,ishape,iwtline,2,iwgt) + pcs1
               winfo(2,isub,ishape,iwtline,2,iwgt) = winfo(2,isub,ishape,iwtline,2,iwgt) + pcs2
               winfo(3,isub,ishape,iwtline,2,iwgt) = winfo(3,isub,ishape,iwtline,2,iwgt) + pcs3
               winfo(4,isub,ishape,iwtline,2,iwgt) = winfo(4,isub,ishape,iwtline,2,iwgt) + pcs4
               winfo(1,isub,ishape,iwtline,3,iwgt) = winfo(1,isub,ishape,iwtline,3,iwgt) + cps1
               winfo(2,isub,ishape,iwtline,3,iwgt) = winfo(2,isub,ishape,iwtline,3,iwgt) + cps2
               winfo(3,isub,ishape,iwtline,3,iwgt) = winfo(3,isub,ishape,iwtline,3,iwgt) + cps3
               winfo(4,isub,ishape,iwtline,3,iwgt) = winfo(4,isub,ishape,iwtline,3,iwgt) + cps4
               winfo(1,isub,ishape,iwtline,4,iwgt) = winfo(1,isub,ishape,iwtline,4,iwgt) + wgt1
               winfo(2,isub,ishape,iwtline,4,iwgt) = winfo(2,isub,ishape,iwtline,4,iwgt) + wgt2
               winfo(3,isub,ishape,iwtline,4,iwgt) = winfo(3,isub,ishape,iwtline,4,iwgt) + wgt3
               winfo(4,isub,ishape,iwtline,4,iwgt) = winfo(4,isub,ishape,iwtline,4,iwgt) + wgt4
            end if
         end if

      end do
100   print*,'loop exited with ier of ',ier

80    format(i2,i2,i2,i2,i3,4f14.2)
      open(80,file='pcinfo_we')
      do iq = 1,nq
         do isub = 1,nsub
            do ishape = 1,nshape
               do ipcline = 1,npcline
                  do iwgt = 1,nwgt
                     write(80,80) iq,isub,ishape,ipcline,iwgt,(pinfo(iq,isub,ishape,ipcline,irpw,iwgt),irpw=1,nrpw)
                  end do
               end do
            end do
         end do
      end do

      close(80)

      open(80,file='wtinfo_we')
      do iq = 1,nq
         do isub = 1,nsub
            do ishape = 1,nshape
               do iwtline = 1,nwtline
                  do iwgt = 1,nwgt
                     write(80,80) iq,isub,ishape,iwtline,iwgt,(winfo(iq,isub,ishape,iwtline,irpw,iwgt),irpw=1,nrpw)
                  end do
               end do
            end do
         end do
      end do

      close(80)
      open(80,file='disinfo_we')
90    format(i2,i2,i2,i3,f14.2)
      do iq = 1,nq
         do isub = 1,nsub
            do ishape = 1,nshape
               do iwgt = 1,nwgt
c                 print*,'pdis is ',pdis(iq,isub,ishape)
                  write(80,90) iq,isub,ishape,iwgt,pdis(iq,isub,ishape,iwgt)
               end do
            end do
         end do
      end do




      end
```

```
      function getpcline(ivip)

      integer*4  ivip,getpcline


      if (((ivip.ge.9).and.(ivip.le.12)).or.((ivip.ge.65).and.(ivip.le.69)).or.
     &   ((ivip.ge.117).and.(ivip.le.121))) then
         getpcline=1            ! basic auto
      else if ((ivip.eq.8).or.(ivip.eq.64).or.(ivip.eq.116)) then
         getpcline=2            ! basic non-auto
      else if (((ivip.ge.14).and.(ivip.le.17)).or.((ivip.ge.71).and.(ivip.le.75)).or.
     &      ((ivip.ge.123).and.(ivip.le.127))) then
         getpcline=3            ! 3 auto
      else if ((ivip.eq.13).or.(ivip.eq.70).or.(ivip.eq.122)) then
         getpcline=4            ! 3 non-auto
      else if ((ivip.eq.18).or.(ivip.eq.144).or.(ivip.eq.146).or.(ivip.eq.147)) then
         getpcline=5            ! 5 auto
      else if ((ivip.eq.143).or.(ivip.eq.145)) then
         getpcline=6            ! 5 Non-auto
      else if ((ivip.eq.19).or.(ivip.eq.76).or.(ivip.eq.128)) then
         getpcline=7            ! CR
      else if (((ivip.ge.20).and.(ivip.le.22)).or.(ivip.eq.77).or.
     &      (ivip.eq.129)) then
         getpcline=8            ! high density
      else if ((ivip.eq.23).or.((ivip.ge.78).and.(ivip.le.80)).or.
     &      ((ivip.ge.130).and.(ivip.le.132))) then
         getpcline=9            ! saturation
      else if ((ivip.eq.148).or.(ivip.eq.149)) then
         getpcline=10
      else
         getpcline=99
      end if


      return
      end


C---  -----  ---  ----------------------------  ----------------

      function getwtline(ivip)

      integer*4 ivip,getwtline

      if ((ivip.eq.1).or.(ivip.eq.29)
     &   .or.(ivip.eq.102)) then
         getwtline=1          !  zone = DDU
      else if ((ivip.eq.4).or.(ivip.eq.32)
     &      .or.(ivip.eq.103)) then
         getwtline=2          ! in county, zone = all other; other domestic, zone = DSCF
      else if ((ivip.eq.35).or.(ivip.eq.104)) then
         getwtline=3          ! zone = 1&2
      else if ((ivip.eq.38).or.(ivip.eq.105)) then
         getwtline=4          ! zone = 3
      else if ((ivip.eq.41).or.(ivip.eq.106)) then
         getwtline=5          ! zone = 4
      else if ((ivip.eq.44).or.(ivip.eq.107)) then
         getwtline=6          ! zone = 5
      else if ((ivip.eq.47).or.(ivip.eq.108)) then
         getwtline=7          ! zone = 6
      else if ((ivip.eq.50).or.(ivip.eq.109)) then
         getwtline=8          ! zone = 7
      else if ((ivip.eq.53).or.(ivip.eq.110)) then
         getwtline=9          ! zone = 8
      else if ((ivip.eq.60).or.(ivip.eq.113)) then
         getwtline=10
      else
         getwtline=99
      end if


      return
      end
```

```
      program pmtstrata

C     Date: DATE
C     Programmer: mcb.weh
C     Project: mcb
C
C     Last Modified: 10/11/94 amr for FY1994
C                    10/27/95 lpl for FY1995
c                    07/28/98     for q1 to q3, 1998
C
C     Purpose: create permit strata maps for use in rollup programs

      IMPLICIT NONE

      INTEGER*4 nstrata,nap,nelm,npmt,nbad,nclass

      PARAMETER (nstrata=20,nap=13,npmt=2274,nbad=20)
      parameter (nclass=3)

      REAL*8 tabrev(3,nstrata,nap)

      CHARACTER*6 key1st(npmt),key3rd(npmt),key3np(npmt),sites(npmt)
      INTEGER*4 str1st(npmt),str3rd(npmt),str3np(npmt), searchc

      INTEGER*4 noffice(3,nstrata)

      REAL*8 rev(nap)
      INTEGER*4 strata

      INTEGER*4 n1st,n3rd,n3np,nsites,nmaps
      integer*4 test

      INTEGER*4 i,j,k,ier,idx,icl,iap,iclass,ifin,istr

      CHARACTER*165 record
      CHARACTER*6 finno,bad(nbad),good(nbad)
      do ifin = 1,npmt
         str1st(ifin) = 99
         str3rd(ifin) = 99
         str3np(ifin) = 99
      end do


      do icl = 1,nclass
         do istr = 1,nstrata
            noffice(icl,istr) = 0
            do iap = 1,nap
               tabrev(icl,istr,iap)=0
            end do
         end do
      end do

      do iap = 1,nap
         rev(iap)=0
      end do

C     Read the list of permit sites.

      open(10,file='finno.pmt')
C     Sorted file of PERMIT finance numbers.  A list of PERMIT finance numbers is
C     created in the processing of the San Mateo PERMIT tapes.

      ier = 0
      nsites = 1
      DO WHILE (ier.EQ.0)
         READ(10,'(A6)',IOSTAT=ier,END=19) sites(nsites)

         nsites = nsites + 1
      END DO
19    nsites = nsites - 1
      PRINT *, 'Read ',nsites,' permit sites  ier = ', ier

C     Read Strata lists

      OPEN(20,FILE='strata.41416',recl=1000)
      ier = 0
      n1st=1
      DO WHILE (ier.EQ.0)
         READ(20,'(A6,I3)',IOSTAT=ier,END=27) finno, strata
         idx = searchc(sites,nsites,finno)
```

```
         IF (idx.GT.0) THEN
            str1st(idx) = strata
            n1st=n1st+1
         END IF
      END DO
27    n1st = n1st - 1
      PRINT *, 'Read ',n1st,' valid 1st Class PI sites. ier = ', ier
      CLOSE(20)

      OPEN(20,FILE='strata.41411',recl=1000)
      ier = 0
      n3rd = 1
      DO WHILE (ier.EQ.0)
         READ(20,'(A6,I3)',IOSTAT=ier,END=28) finno, strata
         idx = searchc(sites,nsites,finno)
         IF (idx.GT.0) THEN
            str3rd(idx) = strata
            n3rd = n3rd + 1
         END IF
      END DO
28    n3rd = n3rd - 1
      PRINT *, 'Read ',n3rd,' valid PERMIT 3rd Class PI sites. ier = ', ier
      CLOSE(20)

      OPEN(20,FILE='strata.41414',recl=1000)
      ier = 0
      n3np = 1
      DO WHILE (ier.EQ.0)
         READ(20,'(A6,I3)',IOSTAT=ier,END=29) finno, strata
         idx = searchc(sites,nsites,finno)
         IF (idx.GT.0) THEN
            str3np(idx) = strata
            n3np = n3np + 1
         END IF
      END DO
29    n3np = n3np - 1
      PRINT *, 'Read ',n3np,' valid PERMIT 3np Class PI sites. ier = ', ier
      CLOSE(20)

C     Write report

      OPEN(40,FILE='finstrata.00')
44    FORMAT(a6,1x,3(i3))
      DO strata = 1, nsites
         WRITE(40,44) sites(strata), str1st(strata),str3rd(strata),str3np(strata)
         test = 0
         test = str1st(strata) + str3rd(strata) + str3np(strata)
         if (test.gt.220) then
            print*,'PMT finno does not match NCTB  ',sites(strata)
         end if
      END DO

      END
C-------------------------------------------------------------------------------
```

```
      program rev_cov

C     Date: 07/29/98
C     Programmer: lpl
C     Project: lpl
C
C     Last Modified:
C
C
C     Purpose:  to compare PI revenue reported in PERMIT with NCTB Revenue
C               office by office

      IMPLICIT NONE

      INTEGER*4 nap,npmt,nstrata,ntype

      PARAMETER (nap=13,npmt=2257)
      PARAMETER (ntype=2,nstrata=20)

      CHARACTER*6 pfins(npmt)
      INTEGER*4  searchc, ip,str


      REAL*8  aprev(nap), revpmt(nap)
      real*8  offaprev(npmt,nap), nctbrev(npmt,nap)
      real*8  test(npmt,nap)
      real*8  revenue(ntype,nstrata),data_rev(ntype,nstrata)
      real*8  rev_nctb(nstrata)

      real*8  check_pmt(npmt), check_nctb(npmt)

      INTEGER*4 npfins

      INTEGER*4 i,j,k,ier,idx,itype,ia

      CHARACTER*175 record
      CHARACTER*6 finno


      integer*4    noffices(ntype,nstrata)
      integer*4    nctbap(ntype,nstrata)
      integer*4    off_nctb(nstrata),off_stat(2,nstrata)
      integer*4    fit_flag(nap),byap_flag(nap)

      character*3  class
      character*5  account
      integer*4    cid

      integer*4    ido

      real*8       check_ap(nap)

      logical   done

      class = 'xxx'
      call getarg(1,class)

      if (class.eq.'1st') account='41416'
      if (class.eq.'3rd') account='41411'
      if (class.eq.'3np') account='41414'
      if (class.eq.'132') account='41413'
      if (class.eq.'bpm') account='41412'
      if (class.eq.'4sp') account='41418'
      if (class.eq.'ppt') account='41419'

      if (class.eq.'1st') cid=1
      if (class.eq.'3rd') cid=2
      if (class.eq.'3np') cid=3
      if (class.eq.'bpm') cid=4
      if (class.eq.'132') cid=7
      if (class.eq.'4sp') cid=5
      if (class.eq.'ppt') cid=8

      if (class.eq.'xxx') then
         print*,'Invalid argument not, 1st,3rd,3np'
         stop
      end if
      do i = 1,2
         do j = 1,nstrata
            off_stat(i,j) = 0
```

```
            noffices(i,j) = 0
            nctbap(i,j) = 0
            data_rev(i,j)= 0.0
         end do
      end do
      do i= 1,npmt
         check_pmt(i)= 0
         check_nctb(i) = 0
         do j = 1,nap
            check_ap(j) = 0.0
            offaprev(i,j) = 0
            nctbrev(i,j) = 0
            test(i,j) = 0.0
         end do
      end do


      ido = 1

      do str=1,nstrata
         off_nctb(str) = 0
         rev_nctb(str) = 0
         do itype = 1,ntype
            data_rev(itype,str) = 0
            revenue(itype,str) = 0
         end do
      end do
      itype=0
C     Read the list of permit pfins.

      OPEN(10,FILE='finstrata.00')
      ier = 0
      npfins = 1
      DO WHILE (ier.EQ.0)
         READ(10,'(A6)',IOSTAT=ier,END=14) pfins(npfins)
         npfins = npfins + 1
      END DO
14    npfins = npfins - 1
      PRINT *, 'Read ',npfins,' PERMIT pfins. ier =', ier



C     Read PERMIT revenues by AP
      OPEN(30,FILE='PIrevbyacct.00.ye')
31    FORMAT(1x,A6,1X,13F12.2)

      ier = 0
      i = 0
      j=0
      DO WHILE (ier.EQ.0)
         READ(30,'(A)',IOSTAT=ier,END=39) record
c         print*, 'rec = ', record
         READ(record(2:7),'(A6)') finno

         IF (record(1:1).EQ.'5') THEN
            j = 1
         ELSE IF (record(1:1).EQ.'1') THEN
            j = 2
         ELSE IF (record(1:1).EQ.'4') THEN
            j = 3
         ELSE IF (record(1:1).EQ.'2') THEN
            j = 4
         ELSE IF (record(1:1).eq.'6') then
            j= 5
         ELSE IF (record(1:1).eq.'3') then
            j= 7
         ELSE IF (record(1:1).eq.'7') then
            j= 8
         ELSE IF (record(1:1).eq.'9') then
            j = 99
         ELSE IF (record(1:1).eq.'8') then
            j = 99
         end if
         i = i + 1
         idx = searchc(pfins,npfins,finno)
         IF (idx.GT.0) THEN
            READ(record,31) finno,(revpmt(k), k = 1, nap)
            DO k = 1, nap
               test(idx,k) = test(idx,k) + revpmt(k)
            END DO
```

```
            IF (j.EQ.cid) THEN

               DO k = 1, nap
                  offaprev(idx,k) = offaprev(idx,k) + revpmt(k)
               END DO
            end if
         ELSE
            PRINT *, 'j = ',j,' Did not find finno: you pmt ',finno
         END IF

      END DO
39    PRINT *,'Read ',i,' records from q1. ier = ',ier
      close(30)


      open(70,file='strata.'//account)
      open(73,file='finsbyap.pmt.00')
      open(72,file='to_be_fit.'//account)
73    format(a6,13i2)

      do ip = 1,npmt
         do k = 1,nap
            byap_flag(k) = 0
         end do
         do k = 1,nap
            if (test(ip,k).gt.0) then
               byap_flag(k) = 1
            end if
         end do
         write(73,73) pfins(ip),(byap_flag(k),k=1,nap)
      end do

72    format(a6,i3,13i2)

71    format(a6,i3,15x,13f12.2)
      ier =0
      do while (ier.eq.0)
         read(70,71,iostat=ier,end=170) finno, str, aprev
         done=.false.
         do ip = 1,nap
            fit_flag(ip) = 1
         end do
         off_nctb(str) = off_nctb(str) + 1
         do k = 1,nap
            check_ap(k) = check_ap(k) + aprev(k)
            rev_nctb(str) = rev_nctb(str) + aprev(k)
         end do
         ip = searchc(pfins,npmt,finno)
         if(ip.gt.0) then
            do k = 1,nap
               if (test(ip,k).gt.0) then
                  fit_flag(k) = 0
                  byap_flag(k) = 1
               end if
            end do
            itype=1
            off_stat(1,str) = off_stat(1,str) + 1
            noffices(itype,str) = noffices(itype,str) + 1
            do k = 1,nap
               if(test(ip,k).gt.0) then
                  itype=1
                  nctbrev(ip,k) = nctbrev(ip,k) + aprev(k)
                  if  (ido.eq.1) then
                     data_rev(itype,str) = data_rev(itype,str) + offaprev(ip,k)
                  end if
                  revenue(itype,str) = revenue(itype,str) + aprev(k)
                  nctbap(itype,str) = nctbap(itype,str) + 1
               else
                  itype=2
                  if(.not.done) then
                     off_stat(2,str) = off_stat(2,str) + 1
                     off_stat(1,str) = off_stat(1,str) - 1
                     noffices(itype,str) = noffices(itype,str) + 1
                     done=.true.
                  end if
                  revenue(itype,str) = revenue(itype,str) + aprev(k)
                  nctbap(itype,str) = nctbap(itype,str) + 1
               end if
            end do
         else
```

```
            itype = 2
            noffices(itype,str) = noffices(itype,str) + 1
            do k = 1,nap
               revenue(itype,str) = revenue(itype,str) + aprev(k)
               nctbap(itype,str) = nctbap(itype,str) + 1
            end do
         end if
         write(72,72) finno,str,(fit_flag(k),k=1,nap)
         ip=0
      end do
170   print*,'read exit of strata.',account,'  ier = ',ier
      ier = 0
      close(70)


      do ip = 1,npfins
         do ia = 1,nap
            check_pmt(ip) = check_pmt(ip) + offaprev(ip,ia)
            check_nctb(ip) = check_nctb(ip) + nctbrev(ip,ia)
         end do
      end do


      do k=1,nap
         print*,'AP, ',k,'  ',check_ap(k)
      end do

      END
C-------------------------------------------------------------------------
```

```
#!/usr/bin/csh
# Shell Script stdadisk
foreach ap (01 02 03 04 05 06 07 08 09 10 11 12 13)

cat  /u/mcb/permit00/files/permit.00.stda.${ap} > tempfile
stda_roll ${ap} > stda_roll.out.${ap}

rm tempfile

end
```

```
      program stda_roll

C     Mike McGrane   9-21-94
C
C     Check PERMIT system transactions for internal consistency
C


      implicit none

C     Parameters

      integer*4  nvip, nshp, nwt, nstr, nms, ntype, nfin,nrpw, nerr, nind
      integer*4  maxln

      parameter  (nvip = 670)   ! number of 3rd class regular rate vips
      parameter  (nshp = 5)     ! number of shapes
      parameter  (nwt  = 16)    ! number of weight increments
      parameter  (nstr = 20)    ! number of revenue strata
      parameter  (ntype = 4)    ! number of transaction types
      parameter  (nfin = 2257)  ! number of permit finance numbers
      parameter  (nrpw = 3)     ! r, p are held as a dim in array
      parameter  (nerr = 29)    ! number of error codes (incl good trans)
      parameter  (nind = 2)     ! number of indicia (for ms trans counts)
      parameter  (maxln = 50)

C     Real storage

      real*8   rerr(nfin,nerr,nshp,nind) ! r, p, w for errors and good trans
      real*8   perr(nfin,nerr,nshp,nind)
      real*8   werr(nfin,nerr,nshp,nind)
      real*8   rerr_x(nerr,nshp) ! r, p, w for errors and good trans
      real*8   perr_x(nerr,nshp)
      real*8   werr_x(nerr,nshp)
      real*8   fintran(2,nfin,nind,4)
      real*8   rv(nvip)          ! r, p, w for a transaction
      real*8   pv(nvip)
      real*8   wv(nvip)
      real*8   rtv(nvip)
      real*8   rts(2,nvip), fxrts(2,nvip) ! rates for each vip code
      real*8   rts96(2,nvip),rts95(2,nvip) ! rates for each vip code
      real*8   rtot, ptot, wtot, rspc
      real*8   r, p, w, rt, t
      real*8   wpp

C     Integer storage

      integer*4 rpttran(nstr)   ! number of trans by strata

      integer*4 iv, ip, iw, it, is, im, il, if, inf, ie, ir, ii, ctold

      integer*4 ier, i, j, k, lines, ierror, n, nt, ctshp, ctn
      integer*4 searchc, msind, day
      integer*4 itype(nvip)     ! array to indicate type of each vip in trans
      integer*4 iwt(nvip)       ! array to indicate weight increment of each vip
      integer*4 strmap(nfin)    ! map from finance number to strata
      integer*4 cnffins(maxln)
      integer*4 cnterr(nfin,nerr,nshp,nind)
      integer*4 cnterr_x(nerr,nshp)
      integer*4 ntran, iind

C     Character strorage

      character*137  rechead, temphead ! header portion of record
      character*48   recarray(maxln) ! array to be equived with rest of record

      character*5008 record, temprec,out_rec
      character*50   foot
      character*48   errorcodes(nerr)
      character*13   tno, newtran,foo
      character*8    trans,tdate
      character*6    fins(nfin), fin, lfin
      character*6    nffins(maxln)
      character*5    vips(nvip), vip
      character*5    vipprn(maxln)
      character*5    perno
      character*4    len,clines
      character*2    viptype(nvip), tcode, seqno, newseq, ap
      character*2    indic
      character*2    nprof
```

```
      character*1    viptyp(maxln) ! identical/non-identical/bad-weight
      character*1    mixid,surflag,cprc

      integer*4      length,badlen,pplen
      integer*4      year
      integer*4      iob,eob

      real*8         rtprn(maxln),pprn(maxln),rprn(maxln),wtprn(maxln)

c      character*20   rec_chk


C     Logicals

      logical   tflag, fflag, first,ppflag

C     Common block for checking subroutine
      common  /check/ fins,rv, pv, wv, rtv, rts, fxrts, rtot, ptot, wtot, rspc,
     &   wpp, itype, iwt, ip,if, viptype, vips, tcode, ppflag,lines

      CALL GETARG(1,ap)

c      open(1, file='bugfile', iointent='output')
c2     format(a20)

      print*, 'Starting bin prog for ap ', ap

      open(40,file='permit.new.stda.'//ap,recl=5008)
      open(59,file='tempfile',recl=5008)

      print*, 'Opened good data file'

      open(45,file='badrec.stda.'//ap,recl=5008)

      print*, 'Opened bad data file'

C     Read list of vip codes with rates and flags

      open(15,file='vip_stda99.new')
16    format(a5,4f8.3)

      do iv = 1, nvip
         read (15,16) vip, (rts96(i,iv),i=1,2),(rts95(i,iv),i=1,2)
         if (vip(1:1).eq.' ')  vip(1:1)='0'
         vips(iv) = vip
      end do

      print *,' vipmap96.dat read '
      close (15)

C     Read list of finance numbers and strata they map to

      open(17,file='finstrata.00')
18    format(a6)
      do if = 1, nfin
         read (17,18) fins(if)
         strmap(if) = 20
         if (strmap(if).eq.99) then
            strmap(if) = 20
            print *,' zero strata for ',fins(if)
         end if
      end do

      print *,' read finance numbers and strata '
      close (17)

C     Read file of error codes

      open(19,file='error.codes')
      do ier = 1, nerr
         read (19,'(a48)') errorcodes(ier)
      end do
      print *,"error codes read"

C     Initialize storage arrays

      do if = 1, nfin
         do ie = 1, nerr
            do ip = 1, nshp
               do ii = 1, nind
```

```
                  rerr(if,ie,ip,ii) = 0.
                  perr(if,ie,ip,ii) = 0.
                  werr(if,ie,ip,ii) = 0.
                  cnterr(if,ie,ip,ii) = 0.
               end do
            end do
         end do
      end do

      do ie = 1, nerr
         do ip = 1,nshp
            rerr_x(ie,ip) = 0.
            perr_x(ie,ip) = 0.
            werr_x(ie,ip) = 0.
            cnterr_x(ie,ip) = 0
         end do
      end do

      do ir = 1,2
         do iv = 1,nvip
            rts(ir,iv) = 0
            fxrts(ir,iv) = 0
            iwt(iv) = 0
         end do
      end do

      do ie = 1,2
         do if = 1,nfin
            do ii = 1,nind
               do ip = 1,4
                  fintran(ie,if,ii,ip) = 0
               end do
            end do
         end do
      end do
C     Read through permit file and pigeon hole records

34    format(a137)
31    format(a6,a2,a13,a2,a2,a5,a8,a1,f12.2,a2,f12.2,a13,i1,a1,f8.4,f12.0,f14.4,20x,i3)
32    format(6x,a5,1x,f6.3,f12.0,f18.7)
33    format(8x,a13,a2,16x,f12.2)
      ier = 0
      inf = 0
      ctn = 0
      ctshp = 0
      ntran = 0
      lfin = ' '
      first = .true.


98    do while (ier.eq.0)
         ppflag=.false.
         tflag=.true.
         do il = 1,maxln
            viptyp(il)='X'
            wtprn(il) = 0
            pprn(il) = 0
            rtprn(il) = 0
            rprn(il) = 0
         end do
         do iv = 1, nvip
            rv(iv) = 0.
            pv(iv) = 0.
            wv(iv) = 0.
            rtv(iv) = 0.
         end do
         read (59,'(a)',iostat=ier,end=100) record
         read(record,34)  rechead

c*************************************************************
c         read (record, 2) rec_chk
c         write(1,2) rec_chk
c*************************************************************

        ntran = ntran + 1
c         if (ntran.gt.282530) then
c            print*, record
c         end if
99       read (rechead,31) fin,indic,tno,seqno,tcode,perno,tdate,mixid,rtot,
     &      nprof,rspc,foo,ip,surflag,wpp, ptot, wtot, lines
```

```
badlen = 138 + lines*48


day = 0
read(tno(5:7),'(i3)') day
read(tno,'(i4)') year
if ((year.eq.1999).and.(day.ge.010)) then
   do iv = 1,nvip
      rts(1,iv) = rts96(1,iv)
      rts(2,iv) = rts96(2,iv)
      fxrts(1,iv) = rts96(1,iv)
      fxrts(2,iv) = rts96(2,iv)
   end do
else
   do iv = 1,nvip
      rts(1,iv) = rts96(1,iv)
      rts(2,iv) = rts96(2,iv)
      fxrts(1,iv) = rts96(1,iv)
      fxrts(2,iv) = rts96(2,iv)
   end do
end if
if (ip.gt.4) then
   print *, "Shape category = ", ip
   print *, "Transaction number ", ntran
   ctshp = ctshp + 1
   ip=4
end if

if (tcode(1:1).eq.'N') then
   ctn = ctn + 1
end if
if (indic.eq.'PI') then
   iind=1
else
   iind=2
end if
if (fin.ne.lfin) then  ! look up finance number
   lfin = fin

   if = searchc(fins,nfin,fin)
   if (if.eq.0) then   ! if finance number not found
      print *, " "
      print *, "Finance Number not found ", fin
      print *, " "
      if (inf.eq.0) then ! first one not found ?
         inf = 1

         nffins(inf) = fin
      else
         fflag = .false.
         do i = 1, inf ! check if this one has been recorded
            if (fin.eq.nffins(i)) then
               fflag = .true.
            end if
         end do
         if (.not.fflag) then ! if a new one add to arrays
            inf = inf + 1
            if (inf.le.-1) then
               nffins(inf) = fin
            else
               print*,'finno mapping error ',fin
               stop  ! ***** ERROR: too many unmapped finance numbers **** !
            end if
         end if
      end if
      is = 0
   else
      is = if          ! if a good finance number, map to strata
   end if
end if
nt = 1

if (if.gt.0) then      ! if a good finance number
   do il=1,lines
      n  = 138+(il-1)*48
      recarray(il) = record(n:n+47)
   end do

   if (lines.gt.0) then ! if there is vip detail
      do il = 1, lines
```

```
                read (recarray(il),32)  vip, rt, p, r
                if (recarray(il)(2:5).eq.'6000') then ! Parcel Surcharge
                   vip = '99999'
                   if (abs(p-(r*10.0)).gt.1.0) then
                      print*,'vip6000 ',p,' ',r
                   end if
                end if
C               Check processing category
                if (vip(5:5).eq.'1') then
                   if (ip.ne.1) then
                      print*, 'Pcat error ',ip,' ',vip
C                      print*,record
                      ip = 1
                   end if
                else if (vip(5:5).eq.'2') then
                   if (ip.ne.2) then
                      print*,'Pcat error ',ip,' ',vip
C                      print*,record
                      ip = 2
                   end if
                else if (vip(5:5).eq.'4') then
                   if (ip.lt.3) then
                      print*,'Pcat error ',ip,' ',vip
C                      print*,record
                      ip = 3
                   end if
                else if (vip(5:5).eq.'6') then
                   if (ip.ne.1) then
                      print*,'Pcat error ',ip,' ',vip
C                      print*,record
                      ip = 1
                   end if
                else if (vip(5:5).eq.'7') then
                   if (ip.lt.2) then
                      print*,'Pcat error ',ip,' ',vip
C                      print*,record
                      ip = 2
                   end if
                else if (vip(5:5).eq.'9') then
                   if (ip.lt.3) then
                      print*,'Pcat error ',ip,' ',vip
C                      print*,record
                      ip = 3
                   end if
                else
                   print*,'No processing category ',vip
                   ierror = 29
                   ip = 1
                end if

                if (vip(1:1).eq.' ')  vip(1:1)= '0'
                if (vip.eq.'03490') vip ='03401'
                vipprn(il) = vip
                rtprn(il) = rt
                if ((vip(1:1).eq.'3').and.(vip(2:4).ne.'340')) then
                   if (vip(5:5).gt.'5') then ! Wt portion
                      w=p/10000
                      p=0.
                   else
                      w=0.
                   end if
                end if
                iv = searchc(vips,nvip,vip)
                if (iv.gt.0) then ! look up vip


                   if (vip(1:1).eq.'3') then
                      rv(iv) = rv(iv) + r
                      pv(iv) = pv(iv) + p
                      wv(iv) = wv(iv) + w
                   else
                      rv(iv) = r
                      rtv(iv) = rt
                      if (vip(5:5).lt.'5') then ! piece vip
                         pv(iv) = p

                      else if (vip.ne.'99999') then ! pound vip
                         wv(iv) = p / 10000.0
                      else
                         rv(iv) = r
```

```
                  pv(iv) = p
               end if
            end if
         else          ! bad vip code
            ! once maps have been updated for 94 this branch
            ! should not be reached
            print *,' vip not in vip_stda96.dat ',vip
            tflag=.false.
         end if
      end do

      if(tflag) then
         call check_stda(tflag,ierror) !perform consistency and error chk
         il = 0
         lines = 0
         do iv = 1,nvip
            if((rv(iv).gt.0).or.(pv(iv).gt.0)) then
               il = il + 1
               lines = lines + 1
               vipprn(il) = vips(iv)
               write(viptyp(il),'(i1.1)') itype(iv)
               wtprn(il) = wv(iv)
               pprn(il) = pv(iv)
               if (vips(iv)(5:5).lt.'5') then
                  rtprn(il) = rv(iv)/pv(iv)
               else if (vips(iv).ne.'99999') then
                  rtprn(il) = rv(iv)/wv(iv)
               else
                  rtprn(il) = rv(iv)/pv(iv)
               end if
               rprn(il) = rv(iv)
            end if
         end do
      end if

      if (tflag) then  ! good transaction
         length = 137
         write(rechead(135:137),'(i3.3)') lines
         write(rechead(79:79),'(i1.1)') ip
         write(rechead(101:114),'(f14.4)') wtot

         rechead(1:6) = fin
         write(out_rec(1:137),34) rechead
         eob = 137

         do il = 1,lines
            iob = eob + 1
            eob = iob + 49
            write(foot,'(1x,a5,a2,f6.3,f12.0,2f12.4)') vipprn(il),viptyp(il),rtprn(il),pprn(il),wtprn(il),rprn(il)
            write(out_rec(iob:eob),'(a49)') foot
            length = length + 50
         end do

         write(40,'(a)') out_rec(1:length)

         i = 0
         i = ierror
         rerr(if,i,ip,iind) = rerr(if,i,ip,iind) + rtot ! good trans are counted under
         perr(if,i,ip,iind) = perr(if,i,ip,iind) + ptot ! error types 1, 2, 3, and 24
         werr(if,i,ip,iind) = werr(if,i,ip,iind) + wtot
         cnterr(if,i,ip,iind) = cnterr(if,i,ip,iind) + 1 ! transaction count
         ! count transactions for mailing size tables
         if (ii.eq.0) then
            ii = 1     ! permit imprint
         else
            ii = 2     ! precan or meter
         end if
      else             ! bad transaction flag
         write(45,'(a)')  record(1:badlen)
C         write(45,'(a)') record(1:badlen)
         if (ppflag) write(45,'(a)') record(1:badlen)
C         write(45,'(a)') temprec(1:pplen)
         rerr(if,ierror,ip,iind) = rerr(if,ierror,ip,iind) + rtot ! store totals under
         perr(if,ierror,ip,iind) = perr(if,ierror,ip,iind) + ptot ! error number
         werr(if,ierror,ip,iind) = werr(if,ierror,ip,iind) + wtot
         cnterr(if,ierror,ip,iind) = cnterr(if,ierror,ip,iind) + 1
      end if
   else                ! lines = 0
      ierror = 26
      rerr(if,ierror,ip,iind) = rerr(if,ierror,ip,iind) + rtot ! store totals under
```

```
            perr(if,ierror,ip,iind) = perr(if,ierror,ip,iind) + ptot ! error number
            werr(if,ierror,ip,iind) = werr(if,ierror,ip,iind) + wtot
            cnterr(if,ierror,ip,iind) = cnterr(if,ierror,ip,iind) + 1
         end if
      else                    ! finance number not in strata map
         ierror = 27
         rerr(if,ierror,ip,iind) = rerr(if,ierror,ip,iind) + rtot ! store totals under
         perr(if,ierror,ip,iind) = perr(if,ierror,ip,iind) + ptot ! error number
         werr(if,ierror,ip,iind) = werr(if,ierror,ip,iind) + wtot
         cnterr(if,ierror,ip,iind) = cnterr(if,ierror,ip,iind) + 1
      end if

      if (nt.eq.2) then       ! process next tran
         record = temprec
         rechead = temphead
         go to 99
      end if


   end do

100   print*,' read exit of ier =  ',ier
   print*,' Number of transactions read = ',ntran
c     print*,' Number of transaction dates before July 1, 1996 ', ctold
   print*,' Number of shape category 5 transactions ', ctshp
   print*,' Number of non-identical records ', ctn

c     write (98, '(a)') 'do we hit this part?'  ---- not bug



C     write out data arrays

C     write out processing summary and error information to file

   do if =1,nfin
      do ie = 1,nerr
         do ip = 1,nshp
            do ii=1,nind
               cnterr_x(ie,ip)=cnterr_x(ie,ip) + cnterr(if,ie,ip,ii)
               rerr_x(ie,ip)=rerr_x(ie,ip) + rerr(if,ie,ip,ii)
               perr_x(ie,ip)=perr_x(ie,ip) + perr(if,ie,ip,ii)
               werr_x(ie,ip)=werr_x(ie,ip) + werr(if,ie,ip,ii)
            end do
         end do
      end do
   end do

   open(50,file='stda.summary.'//ap)
51    format(i2,2x,i8,2x,f14.2,2f12.0,2x,a48)
52    format(4x,i8,2x,f14.2,2f12.0,2x,' Total All Transactions ')

   do ip = 1, nshp
      write (50,*) ' Third-Class Regular Rate Processing Summary '
      write (50,*) ' AP ',ap,', PFY 1996 '
      write (50,*) ' Processing Category ', ip
      write (50,*) '   # Trans     Revenue         Pieces        Weight      Description '
      write (50,*) '   -------     -------         ------        ------      ----------- '
      rtot = 0.
      ptot = 0.
      wtot = 0.
      ntran = 0.
      do ie = 1, nerr
         write(50,51) ie, cnterr_x(ie,ip), rerr_x(ie,ip), perr_x(ie,ip),
     &      werr_x(ie,ip), errorcodes(ie)
         ntran = ntran + cnterr_x(ie,ip)
         rtot = rtot + rerr_x(ie,ip)
         ptot = ptot + perr_x(ie,ip)
         wtot = wtot + werr_x(ie,ip)
      end do
      write (50,*)
      write (50,52) ntran, rtot, ptot, wtot
      write (50,*)
   end do

   do if =1,nfin
      do ie = 1,nerr
         do ip = 1,nshp
            do ii = 1,nind
               if (ie.le.3) then
```

```
                  fintran(1,if,ii,1) = fintran(1,if,ii,1) + cnterr(if,ie,ip,ii)
                  fintran(1,if,ii,2) = fintran(1,if,ii,2) + rerr(if,ie,ip,ii)
                  fintran(1,if,ii,3) = fintran(1,if,ii,3) + perr(if,ie,ip,ii)
                  fintran(1,if,ii,4) = fintran(1,if,ii,4) + werr(if,ie,ip,ii)
               else
                  fintran(2,if,ii,1) = fintran(2,if,ii,1) + cnterr(if,ie,ip,ii)
                  fintran(2,if,ii,2) = fintran(2,if,ii,2) + rerr(if,ie,ip,ii)
                  fintran(2,if,ii,3) = fintran(2,if,ii,3) + perr(if,ie,ip,ii)
                  fintran(2,if,ii,4) = fintran(2,if,ii,4) + werr(if,ie,ip,ii)
               end if
            end do
         end do
      end do
   end do

   open(70,file='finrev.dat.'//ap,recl=1000)
55 format(a6,4(f8.0,f16.4,f16.0,f16.4))
   do if = 1,nfin
      write(70,55) fins(if),((fintran(1,if,ii,ip),ip=1,4),(fintran(2,if,ii,ie),ie=1,4),ii=1,nind)
   end do

c     close(1)

   end



C ---------------------------------------------------------------------------
      function msind(ptot)

      integer*4   msind
      real*8      ptot

      if (ptot.lt.200) then
         msind = 1
      else if (ptot.lt.250) then
         msind = 2
      else if (ptot.lt.300) then
         msind = 3
      else if (ptot.lt.350) then
         msind = 4
      else if (ptot.lt.400) then
         msind = 5
      else if (ptot.lt.450) then
         msind = 6
      else if (ptot.lt.500) then
         msind = 7
      else if (ptot.lt.750) then
         msind = 8
      else if (ptot.lt.1000) then
         msind = 9
      else if (ptot.lt.2000) then
         msind = 10
      else if (ptot.lt.5000) then
         msind = 11
      else if (ptot.lt.10000) then
         msind = 12
      else if (ptot.lt.25000) then
         msind = 13
      else if (ptot.lt.50000) then
         msind = 14
      else if (ptot.lt.100000) then
         msind = 15
      else if (ptot.lt.250000) then
         msind = 16
      else if (ptot.lt.500000) then
         msind = 17
      else if (ptot.ge.500000) then
         msind = 18
      end if

      return
      end
C ---------------------------------------------------------------------------
```

```
      program wgt_std_roll2.f

C     Paul Loetscher
C     Roll up raw PERMIT system transactions for third class
C
C     1)  RPW x vip x shape x wt incr x strata x ntype
C
C       where ntype = 1 - identical
C                     2 - non-identical
C                     3 - bad weight (identical or non-identical)
C     Check PERMIT system transactions for internal consistency
C

      implicit none

      include 'permit_stda.h'

C     Storage for rollup

      integer*4  nvip, nfin, nrpw, ntype,nwt,nshp,nstrata

      parameter  (nvip = 670)   ! number of 3rd class regular rate vips
      parameter  (nshp = 6)     ! number of shapes
      parameter  (nrpw = 3)     ! r, p w are held as a dim in array
      parameter  (ntype = 3)    ! number of transaction types
      parameter  (nwt  = 20)    ! number of weight increments
      parameter  (nfin = 2257)  ! number of permit finance numbers
      parameter  (nstrata  = 20) ! number of NCTB strata


C     Real storage

      real*8         wi(nrpw,nvip,nshp,nwt,ntype,nstrata) ! r,p,w by weight increment

      real*8         vipwpp,viprtx(nvip,2)

      integer*4      ir,iv, ip, iw, if, it,is,il2,iv2
      integer*4      searchc
      integer*4      wind       ! to indicate weight increment of each vip
      integer*4      strata(nfin)
      character*5    vips(nvip),vipx
      character*6    fins(nfin),lastfin
      character*2    ap
      character*1    pc,lb
      integer*4      pci,lbi
      integer*4      pcat_fun


      logical        fndpc, fndwgt,debug

      call getarg(1,ap)

      debug = .false.

      open(40,file='pclb2.stda98',recl=5008)
31    format(a4,a6,a2,a13,a2,a2,a5,a1,f12.2,a2,f12.2,a5,a2,a1,a1,a1,a1,a1,a1,a1,a1,f8.4,f12.0,f14.4,a4,
     &   20(a2,a5,f7.3,f12.0,f18.7,f14.4))


      open(15,file='vip_stda99.new')
16    format(a5,2f8.3)
      do iv = 1, nvip
         read (15,16) vip,viprtx(iv,1),viprtx(iv,2)
         if (vip(1:1).eq.' ')  vip(1:1)='0'
         vips(iv) = vip
      end do

      print *,' vip_stda.dat read '
      close (15)

      open(17,file='finstrata.00')
18    format(a6,4x,i3)
      do if = 1, nfin
         read (17,18) fins(if),strata(if)
         if (strata(if).eq.99) strata(if)=20
         if (strata(if).eq.0) strata(if)=20
      end do

      print *,' finstrata.00 read '
      close (17)
```

```
C      Initialize arrays

       lastfin='XXXXXX'
       do ir = 1, nrpw
          do iv = 1, nvip
             do ip = 1,nshp
                do iw = 1, nwt
                   do it = 1,ntype
                      do is = 1,nstrata
                         wi(ir,iv,ip,iw,it,is) = 0.
                      end do
                   end do
                end do
             end do
          end do
       end do

       print*, 'Initialized'

C*************READ IN PERMIT TRANSTACTION*********************************
C*************************************************************************

       open(20,file='datafile.dat',recl=5008)
c       open(25, file='datafile.dat', iointent='input')
15     format(a6,a2,a13,a2,a2,a5,a8,a1,f12.2,a2,f12.2,a5,a2,8a1,f8.4,f12.0,
     &    f14.4,20x,a3,a4871)
25     format(100(1x,a5,1x,a1,f6.3,f12.0,f12.4,f11.3,1x))
       ier = 0
       cnt = 0


       do while (ier.eq.0)

C      Read permit file

          include 'permit_read_stda.h'


C ******************END PERMIT READ ************************************

          if (mod(cnt,25000).eq.0) print*, 'Working on transaction ',cnt


C          if (debug) print*, 'Transaction ',cnt
C          if (debug) print*,'ip ',ip

          if (fin.ne.lastfin) then
             if = searchc(fins,nfin,fin)
             if (if.gt.0) then
                is = strata(if)
             else
                print*,'Finance number not found ',fin
                stop
             end if
          end if

C          if (debug) print*,'lines,',lines
          if (perno(1:1).ne.'G') then ! Insert 'G' if Government transactions are to be shipped
             do il = 1, lines
                iv = searchc(vips,nvip,vipd(il))
                if (iv.gt.0) then
                   read(viptyp(il),'(i1)') it
                   if (it.eq.0) then
C                      print*,'type error ' ,it , ' ' ,vipd(il)

                      if (vipd(il).eq.'99999') then
                         it = 1
                      end if
                   end if
                   ip = pcat_fun(vips(iv)(5:5))
C          Check for parcel surcharge
                   if (vips(iv)(5:5).eq.'4') then
                      if ((vips(iv)(1:1).eq.'0').or.(vips(iv)(1:1).eq.'3')) then
                         if (abs(1-viprt(il)/viprtx(iv,1)).le.0.05) then
                            ip = 6
                         end if
                      else if (vippc(il).gt.0) then
                         vipwpp = vipwt(il)/vippc(il)
                         if (abs(1.0 - (viprt(il)/(viprtx(iv,1)+(vipwpp*viprtx(iv,2))))).le.0.10) then
```

```
                  ip = 6
               end if
            end if
         end if

         if (vippc(il).gt.0) then
            vipwpp = vipwt(il)/vippc(il)
         end if
      else
         print*,'vip not found', vipd(il)
         print*, rec
      end if
      fndwgt = .false.
      fndpc = .false.

      if ((vipd(il)(1:1).eq.'0').or.(vipd(il)(1:1).eq.'3')) then
         if ((vipd(il)(3:3).ge.'5').and.(vipd(il)(5:5).le.'4')) then
            if (debug) print*, 'entering first remap'
            vipwpp = 0
            read(vipd(il),'(4x,i1)') pci
            if (pci.eq.3) pci = 1
            lbi = pci + 5
            write(lb,'(i1)') lbi
            vipx = vipd(il)(1:4)//lb
            iv2 = searchc(vips,nvip,vipd(il))
            do il2 = 1, lines
               if ((vipx.eq.vipd(il2))) then
                  if ((vippc(il).gt.0)) then
                     vipwpp = vipwt(il2)/vippc(il)
                     fndwgt = .true.
                  end if
               end if
            end do

            if (fndwgt.eq..false.) then
               write(40,31) len,fin,indic,tno,seqno,tcode,perno,
&                 mix_indic,rtot,non_prof,sp_fees,perno_for,gst_typ,
&                 co_used,cpp_used,op_proc,class,by_for,pre_srt,shape,
&                 surflag,wpp,ptot,wtot,clines,'  ',vipd(il2),
&                 viprt(il2),vippc(il2),vipwt(il2),viprev(il2),
&                 il2 = 1,lines)

            end if

         else if ((vipd(il)(3:3).ge.'5').and.(vipd(il)(5:5).ge.'5')) then
            if (debug) print*,'entering second remap'
            vipwpp = 0
            read(vipd(il),'(4x,i1)') lbi
            pci= lbi - 5
            write(pc,'(i1)') pci
            vipx = vipd(il)(1:4)//pc



            iv2 = searchc(vips,nvip,vipd(il))
            do il2 = 1, lines
               if ((vipx.eq.vipd(il2))) then
                  if ((vippc(il2).gt.0)) then
                     vipwpp = vipwt(il)/vippc(il2)
                     fndpc = .true.
                  end if
               end if
            end do
            if (debug) then
               print*,'vipd(il) ',vipd(il),' vipx ', vipx , 'vipwpp', vipwpp
            end if
            if (fndpc.eq..false.) then
               write(40,31) len,fin,indic,tno,seqno,tcode,perno,
&                 mix_indic,rtot,non_prof,sp_fees,perno_for,gst_typ,
&                 co_used,cpp_used,op_proc,class,by_for,pre_srt,shape,
&                 surflag,wpp,ptot,wtot,clines,'  ',vipd(il2),
&                 viprt(il2),vippc(il2),vipwt(il2),viprev(il2),
&                 il2 = 1,lines)
            end if
         end if
      end if
      iw = wind(vipwpp)
      if ((iw.le.0)) then
         if ((vipd(il)(2:4).ne.'340')) then
```

```
                  if (ptot.gt.0) then
                     vipwpp=wtot/ptot
                  end if
                  iw = wind(vipwpp)
                  if (iw.lt.0) then
                     print*,'Bad weight 2 ',vipd(il),' ',wtot,' ',ptot
                     it = 3
                     iw = 1
                     print*, rec
                  end if
                  vipwt(il)=vippc(il)*vipwpp
                  wi(1,iv,ip,iw,it,is) = wi(1,iv,ip,iw,it,is) + viprev(il)
                  wi(2,iv,ip,iw,it,is) = wi(2,iv,ip,iw,it,is) + vippc(il)
                  wi(3,iv,ip,iw,it,is) = wi(3,iv,ip,iw,it,is) + vipwt(il)
               else if ((vipd(il)(2:4).eq.'340')) then
                  it=1
                  if (ptot.gt.0) then
                     vipwpp=wtot/ptot
                  end if
                  iw = wind(vipwpp)
                  vipwt(il)=vippc(il)*vipwpp
                  if (iw.lt.0) then
                     print*,'BAD weight 1 ',wtot, 'ptot',ptot,' ',lines
                  else
                     wi(1,iv,ip,iw,it,is) = wi(1,iv,ip,iw,it,is) + viprev(il)
                     wi(2,iv,ip,iw,it,is) = wi(2,iv,ip,iw,it,is) + vippc(il)
                     wi(3,iv,ip,iw,it,is) = wi(3,iv,ip,iw,it,is) + vipwt(il)
                  end if
               end if

            else
               wi(1,iv,ip,iw,it,is) = wi(1,iv,ip,iw,it,is) + viprev(il)
               wi(2,iv,ip,iw,it,is) = wi(2,iv,ip,iw,it,is) + vippc(il)
               wi(3,iv,ip,iw,it,is) = wi(3,iv,ip,iw,it,is) + vipwt(il)
            end if
         end do
      else
         print*,'Skipping Government ',perno
      end if

   end do

100   print*,'read ',cnt
      print*, 'ier = ', ier

      open(50,file='pmt_stda.wi.'//ap)
51    format(i2,1x,i1,1x,i3,1x,i1,1x,i3,f14.4,f12.0,f14.4)

      do is = 1,nstrata
         do it = 1, ntype
            do iw = 1, nwt
               do ip = 1, nshp
                  do iv = 1, nvip
                     if ((wi(1,iv,ip,iw,it,is).gt.0.0).or.(wi(2,iv,ip,iw,it,is).gt.0.0)) then
                        write(50,51) is, it, iw, ip, iv,
     &                     (wi(ir,iv,ip,iw,it,is), ir = 1, nrpw)
                     end if
                  end do
               end do
            end do
         end do
      end do
      close(50)

      close(20)


      end

C ---------------------------------------------------------------------------
C       wind  - return index of weight graduation
C
C       mrm  8-14-92

      function wind(wpp)

      integer*4   wind, ioz
      real*8      wpp, ozs, oz2

      ozs = wpp * 16.0
```

```
      ioz = int(ozs)
      if ((ozs.gt.4).and.(ozs.le.16)) then        ! wgt greater than 4 oz, increment by 1
         if ((ozs-dble(ioz)).gt.0.0) ioz = ioz + 1
         if ((ioz.ge.5).and.(ioz.le.16)) then
            wind = ioz + 4      ! Subsript
         end if
      else if ((ozs.gt.0).and.(ozs.le.4)) then
         oz2 = ozs - ioz
         if (oz2.eq.0) then
            wind = ioz*2
         else if (oz2.gt.0.5) then
            wind = (ioz+1)*2
         else
            wind = (ioz+1)*2-1
         end if

      else
         wind = -9              ! piece is too heavy or is negative.
      end if

      if (wpp.eq.0.0) wind = 0  ! mixed weight transaction

      return
      end

C-------------------------------------------------------------------------------

      function pcat_fun(vip)

      integer*4    pcat_fun
      character*1  vip

      pcat_fun = -9

      if (vip.eq.'1') then
         pcat_fun  = 1
      else if (vip.eq.'2') then
         pcat_fun = 2
      else if (vip.eq.'4') then
         pcat_fun = 3
      else if (vip.eq.'6') then
         pcat_fun = 1
      else if (vip.eq.'7') then
         pcat_fun = 2
      else if (vip.eq.'9') then
         pcat_fun = 3
      end if

      return
      end
```

```
      program fitdat

c     Author:  tcg, lpl  (11/21/97)

c     Purpose:  Write out stamped/metered data for nctb data by AP
c               Uses the corrected NCTB data

      implicit none

      integer*4      maxfin, nap
      parameter      (maxfin=30000)
      parameter      (nap=13)

      character*6   finno(maxfin),finx
      character*2   apx,ape
      integer*4     iap

      integer*4     ier,cnt,nfin,if,searchc

      real*8        rev(maxfin,nap)
      real*8        end411(maxfin,nap),end412(maxfin,nap)
      real*8        beg411(maxfin,nap),beg412(maxfin,nap)
      real*8        a411,a412

      do if = 1,maxfin
         do  iap = 1, nap
            end411(if,iap)=0.
            end412(if,iap)=0.
            beg411(if,iap)=0.
            beg412(if,iap)=0.
            finno(if) ='xxxxxx'
            rev(if,iap)=0.
         end do
      end do

      print*, 'Matrices initialized'

      open (10,file='revfile')
C     Sorted strata.41411

10    format(a6,18x,13f12.2)

      ier=0

      cnt = 0
      do while(ier.eq.0)
         cnt = cnt + 1
         read(10,10,iostat=ier,end = 100) finno(cnt), (rev(cnt,iap), iap = 1, nap)
      end do

100   print*,'read revfile.ye with ier ',ier, ' and cnt ',cnt

      nfin=cnt

      do iap = 1, nap

         write(apx,'(i2.2)') iap
         write(ape,'(i2.2)') iap-1

         open(20,file='tb00'//apx//'.dat')
20       format(a6,14x,2f14.2)
         ier = 0
         cnt = 0

         do while(ier.eq.0)
            read(20,20,iostat = ier,end = 200) finx,a411,a412
            cnt = cnt + 1
            if = searchc(finno,nfin,finx)
            if (if.gt.0) then
               end411(if,iap) = -a411
               end412(if,iap) = -a412
            end if
         end do
200      print*, 'tb00',apx,' read with ier = ',ier,' and cnt ',cnt

         if (iap.ge.2) then
            open(30,file='tb00'//ape//'.dat')
            ier = 0
            cnt = 0
```

```
            do while(ier.eq.0)
               read(30,20,iostat = ier,end = 300) finx,a411,a412
               cnt = cnt + 1
               if = searchc(finno,nfin,finx)
               if (if.gt.0) then
                  beg411(if,iap) = -a411
                  beg412(if,iap) = -a412
               end if
            end do
300         print*, 'tb00',ape,' read with ier = ',ier,' and cnt ',cnt
         end if

      end do

      open(40,file='nctb_fit_ye.00',recl=560)
40    format(a6,39f14.2)


      do if = 1,nfin
         write(40,40) finno(if),(end411(if,iap)-beg411(if,iap),end412(if,iap)-beg412(if,iap),
     &      rev(if,iap), iap = 1, nap)
      end do


      end
```

```
      PROGRAM fit_stda


C     DESCRIPTION:
C         Estimates SM Revenues for YTD Non-CBCIS sites
C         on AP basis using regression equation

      IMPLICIT NONE

C     Parameter Estimates:

      REAL*8 a1_lg,b1_lg,c1_lg,a1_sm,b1_sm,c1_sm
      REAL*8 a2_lg,b2_lg,c2_lg,a2_sm,b2_sm,c2_sm
      REAL*8 a3_lg,b3_lg,c3_lg,a3_sm,b3_sm,c3_sm
      REAL*8 a4_lg,b4_lg,c4_lg,a4_sm,b4_sm,c4_sm
      REAL*8 a,b,c
      PARAMETER (a1_lg=0.023253,b1_lg=0.097314,c1_lg=-0.000104)
      PARAMETER (a2_lg=0.019722,b2_lg=0.103178,c2_lg=-0.000110)
      PARAMETER (a3_lg=0.016672,b3_lg=0.130030,c3_lg=-0.000048)
      PARAMETER (a4_lg=0.015445,b4_lg=0.115697,c4_lg=-0.000045)
      PARAMETER (a1_sm=0.009477,b1_sm=0.086807,c1_sm= 0.000336)
      PARAMETER (a2_sm=0.011302,b2_sm=0.100748,c2_sm= 0.000123)
      PARAMETER (a3_sm=0.006696,b3_sm=0.103331,c3_sm= 0.000523)
      PARAMETER (a4_sm=0.005279,b4_sm=0.121616,c4_sm= 0.000422)

C     Constants

      INTEGER*4 nfin, iap, nstrata,nq,nap

      parameter (nfin=15682)
      parameter (nstrata=20)
      parameter (nq=4)
      parameter (nap = 13)

      CHARACTER*6 fin
      character*6 fins(nfin)
      CHARACTER*2 ap

      REAL*8    stamp(nap),meter(nap),piprsrt(nap),smhat,total,fit3rd(nstrata,nq)
      REAL*8    min,max,tstamp,tmeter,tpiprsrt,pmtrev(nstrata)

      INTEGER*4 ier,i,j,searchc,out,in,nnon,ifin,lastcount,inx,iq
      integer*4 strata(nfin), strata1, fit(nfin,nap)

C     Get Arguments

C      CALL getarg(1,ap)
C      PRINT *, "Working on AP",ap


      OPEN(10,FILE='to_be_fit.41411')
10    FORMAT(a6,i3,13i2)

      do i = 1, nfin
         READ (10,10) fins(i), strata(i), (fit(i,iap),iap = 1,nap)
      END DO

      PRINT *, 'NCTB finance numbers read'

      do i = 1, nstrata
         do iq = 1,nq
            fit3rd(i,iq) = 0.0
            pmtrev(i) = 0.0
         end do
      end do

      ier = 0
      nnon=0
      in = 0

      OPEN(21,FILE='nctb_fit_ye.00',recl=2000)

21    FORMAT(A6,39F14.2)

      DO WHILE (ier.eq.0)
         READ(21,21,iostat=ier,end=100) fin,(stamp(iap),meter(iap),piprsrt(iap),iap=1,nap)
         in = in + 1
         ifin = searchc(fins,nfin,fin)
         do iap = 1,nap
            IF (ifin.gt.0) THEN ! fin found
```

```
            IF (fit(ifin,iap).eq.1) THEN ! need to fit revenue
               nnon = nnon + 1
               strata1 = strata(ifin)
               pmtrev(strata1) = pmtrev(strata1) + piprsrt(iap)
               IF (iap.le.3) THEN
                  iq = 1
                  if (strata1.lt.19) then
                     a=a1_lg
                     b=b1_lg
                     c=c1_lg
                  else
                     a=a1_sm
                     b=b1_sm
                     c=c1_sm
                  end if
               ELSE IF (iap.le.6) THEN
                  iq = 2
                  if (strata1.lt.19) then
                     a=a2_lg
                     b=b2_lg
                     c=c2_lg
                  else
                     a=a2_sm
                     b=b2_sm
                     c=c2_sm
                  end if
               ELSE IF (iap.le.9) THEN
                  iq = 3
                  if (strata1.lt.19) then
                     a=a3_lg
                     b=b3_lg
                     c=c3_lg
                  else
                     a=a3_sm
                     b=b3_sm
                     c=c3_sm
                  end if
               ELSE IF (iap.le.13) THEN
                  iq = 4
                  if (strata1.lt.19) then
                     a=a4_lg
                     b=b4_lg
                     c=c4_lg
                  else
                     a=a4_sm
                     b=b4_sm
                     c=c4_sm
                  end if
               END IF

               smhat = a*meter(iap) + b*piprsrt(iap) + c*meter(iap)*meter(iap)/1000000
c               if (iq.eq.4) print*, 'smhat1: ', smhat

               if ((meter(iap).gt.0).or.(piprsrt(iap).gt.0)) then

                  if ((meter(iap).le.0).or.(piprsrt(iap).le.0)) then
                     print*,iap,' ',fin,' ',meter(iap),' ',piprsrt(iap)
                  end if
               end if
               if(strata1.eq.13) print*,'stata13 ', fin

               IF (smhat.lt.0) THEN
                  smhat=0
               END IF
               fit3rd(strata1,iq) = fit3rd(strata1,iq) + smhat
c               if (iq.eq.4) print*, 'fit3rd(stratal,4) ', fit3rd(strata1,4)
            end if
         else
            print*, 'Fin not found ', fin
            strata1 = 20
            nnon = nnon + 1
            pmtrev(strata1) = pmtrev(strata1) + piprsrt(iap)
            IF (iap.le.3) THEN
               iq = 1
               if (strata1.lt.19) then
                  a=a1_lg
                  b=b1_lg
                  c=c1_lg
               else
                  a=a1_sm
```

```
                  b=b1_sm
                  c=c1_sm
               end if
            ELSE IF (iap.le.6) THEN
               iq = 2
               if (strata1.lt.19) then
                  a=a2_lg
                  b=b2_lg
                  c=c2_lg
               else
                  a=a2_sm
                  b=b2_sm
                  c=c2_sm
               end if
            ELSE IF (iap.le.9) THEN
               iq = 3
               if (strata1.lt.19) then
                  a=a3_lg
                  b=b3_lg
                  c=c3_lg
               else
                  a=a3_sm
                  b=b3_sm
                  c=c3_sm
               end if
            ELSE IF (iap.le.13) THEN
               iq = 4
               if (strata1.lt.19) then
                  a=a4_lg
                  b=b4_lg
                  c=c4_lg
               else
                  a=a4_sm
                  b=b4_sm
                  c=c4_sm
               end if
            END IF
            smhat = a*meter(iap) + b*piprsrt(iap) + c*meter(iap)*meter(iap)/1000000
c            if (iq.eq.4)  print*, 'smhat4: ', smhat
            IF (smhat.lt.0) THEN
               smhat=0
            END IF
            fit3rd(strata1,iq) = fit3rd(strata1,iq) + smhat
c            if (iq.eq.4)  print*, 'fit3rd(stratal, 4) ', fit3rd(strata1,4)
         end if
      end do
   END DO

100   print*, "ier = ", ier
      PRINT *, "Read ",in-1," records"
      PRINT *, nnon," records need to be fitted."

      open(40,file='fit_stda_ye.00')
41    format(i3,4f14.2)

      do i = 1, nstrata
         write(40,41) i, (fit3rd(i,iq),iq=1,nq)
         print*, "permit imprint rev for strata ", i , " is ", pmtrev(i)
      end do


      end
```

```
      program eststda_20

C     Paul Loetscher 1-06-96
C
C
C
C
C     Estimate third-class volumes by weight increment and class
C     using PERMIT and BRAVIS data

C     Input files:
C     permit rollup matrices: ../pmt/pmt3rd.wi.*
C     bravis rollup matrices: ../brv/brv3rd.wi.*
C     trial balance revenues: /u/mcb/vol96/nctb/strata.41411
C     lists of permit and bravis finance numbers
C
C     Output files:
C     est3rd.csv  -  to be imported into excel
C     est3rd.factors - control factors


      implicit none

C     Parameters

      integer*4  nvip, nshp, nwt, nstr, ntype, nrpw, nind, ncls
      integer*4  npmt, nq, nap

      parameter  (nvip = 669)   ! number of 3rd class regular rate vips
      parameter  (nshp = 6)     ! number of shapes
      parameter  (nwt  = 20)    ! number of weight increments
      parameter  (nstr = 4)     ! number of revenue strata
      parameter  (ntype = 3)    ! number of transaction types
      parameter  (nrpw = 3)     ! r, p, w are held as a dim in array
      parameter  (nind = 2)     ! number of indicia (for ms trans counts)
      parameter  (ncls = 85)    ! number of subclasses
      parameter  (npmt = 2257)  ! number of permit finance numbers
      parameter  (nq = 4)
      parameter  (nap = 13)

C     Real Storage

      real*8   wipr(ncls,nind,nshp,nwt,ntype,nstr,nq) ! r, p, w  - permit
      real*8   wipp(ncls,nind,nshp,nwt,ntype,nstr,nq) ! r, p, w  - permit
      real*8   wipw(ncls,nind,nshp,nwt,ntype,nstr,nq) ! r, p, w  - permit

      real*8   result(nrpw,nwt,nshp,ncls) ! array to write out

      real*8   shrwt(nwt), pishrwt(nwt,ncls,nshp) ! arrays used for redist
      real*8   badwght(nrpw,nshp,ncls)
      real*8   sm_fitted(nstr,nq)

      real*8   pmtrev(nstr,nq)  ! permit revenue by strata and indicia
      real*8   smrev(nstr,nq)   ! permit revenue by strata and indicia
      real*8   nctbpmt(nstr,nq) ! trial balance permit office rev.
      real*8   nctball(nstr,nq) ! trial balance total revenue by strata
      real*8   factpmt(nstr,nq) ! control factors for permit office revenue
      real*8   factall(nstr,nq) ! control factors for total revenue (nctb)
      real*8   factall_sm(nstr,nq) ! control factors for total revenue (nctb)
      real*8   rpwfact(nq)      ! control factor to total rpw revenue
      real*8   revin(nap)
      real*8   r, p, w, wpp,x,xc
      real*8   pieces_by_shp(nshp),rvtst(nq)
      real*8   q1,q2,q3,q4
      real*8   rev_str(20,2)

C     Integer Storage

      integer*4 ir, ic, ii, ip, iw, it, is, iq, if, ia, iv,id
      integer*4 ier, i
      integer*4 searchc, qind, iind
      integer*4 strind          ! function to assign condensed strata
      integer*4 clsmap(nvip)/nvip*0/ ! map from vip to class index
      integer*4 pflags(nap,npmt) ! flag for finance number in permit for ap
      integer*4 minrate(ncls)/ncls*0./ ! flag for minimum rate pieces
      integer*4 shpflag(ncls)/ncls*0./ ! flag for allowable shapes

C     Character Storage

      character*6   pfins(npmt), fin
```

```
      character*2   ap
      character*5   vips(nvip), vip
      real*8        fuzz(20)
C     Logical Storage

C     Intialize arrays

C     RPW revenue = total regular rate revenue from penalty and franked
C     as classes report


      do ip = 1,nshp
         pieces_by_shp(ip) = 0
      end do
      do iq = 1, nq
         do is = 1, nstr
            sm_fitted(iq,is) = 0
            do it = 1, ntype
               do iw = 1, nwt
                  do ip = 1, nshp
                     do ii = 1, nind
                        do ic = 1, ncls
                           wipr(ic,ii,ip,iw,it,is,iq) = 0.
                           wipp(ic,ii,ip,iw,it,is,iq) = 0.
                           wipw(ic,ii,ip,iw,it,is,iq) = 0.
                        end do
                     end do
                  end do
               end do
            end do
         end do
      end do
      do ic = 1, ncls
         do ip = 1, nshp
            do iw = 1, nwt
               pishrwt(iw,ic,ip) = 0.
               do ir = 1, nrpw
                  result(ir,iw,ip,ic) = 0.

               end do
            end do
         end do
      end do
      do ic = 1, ncls
         do ip = 1, nshp
            do ir = 1, nrpw
               badwght(ir,ip,ic) = 0.
            end do
         end do
      end do
      do iq = 1, nq
         rvtst(iq) = 0
         do is = 1, nstr
            pmtrev(is,iq)     = 0.
            smrev(is,iq)      = 0.
            nctbpmt(is,iq)    = 0.
            nctball(is,iq)    = 0.
            factpmt(is,iq)    = 0.
            factall(is,iq)    = 0.
            factall_sm(is,iq)    = 0.
         end do
      end do

      do is = 1,20
         rev_str(is,1) = 0
         rev_str(is,2) = 0
         fuzz(is) = 0
      end do

      do if = 1,npmt
         do ia = 1,nap
            pflags(ia,if) = 0
         end do
      end do

      open(90,file='factors_20.txt')
90    format(i3,i3,f14.2,f14.2,f10.6)

C     Read finance numbers and reported ap's
```

```
      open(15,file='finsbyap.pmt.00')
15    format(a6,13i2)
      do if = 1, npmt
         read (15,15) pfins(if), (pflags(ia,if),ia=1,nap)
      end do
      close (15)
      print *,' PERMIT finance numbers read '



C     Read Trial balance revenue and assign to Permit, Bravis and
C     Total revenue arrays

      open(18,file='strata.41411')
18    format(a6,i3,f15.2,13f12.2)

      if = 0
      ier = 0
      do iq = 1,nq
         rvtst(iq) = 0
      end do
      do while (ier.eq.0)
         read (18,18,iostat=ier,end=180) fin, is, r, revin
         if = if + 1
         id = is
         do ia = 1,nap
            rev_str(id,1) = rev_str(id,1) + revin(ia)
            fuzz(id) = fuzz(id) + revin(ia)
         end do
         ip = 0
         is = strind(is)
         ip = searchc(pfins,npmt,fin)
         do ia = 1, nap
            iq = qind(ia)
            nctball(is,iq) = nctball(is,iq) + revin(ia)
            if (ip.gt.0) then
               rev_str(id,2) = rev_str(id,2) + revin(ia)
               if(pflags(ia,ip).eq.0) then
                  if (is.eq.2) then
                     print*, 'no ap ', pfins(ip),' ia ', ia ,' ', pflags(ia,ip)
                     rvtst(iq) = rvtst(iq) + revin(ia)
                  end if
               else
                  nctbpmt(is,iq) =  nctbpmt(is,iq) + revin(ia)
               end if
            end if
         end do
      end do

180   print *,' Strata.41411 read with ',if,' records, ier = ',ier



      open(19,file='fit_stda_ye.00')
19    format(i3,4f14.2)

      do i = 1, 20
         read(19,19) is, q1, q2, q3,q4
         rev_str(is,1) = rev_str(is,1) + q1 + q2 + q3 + q4
         rev_str(is,2) = rev_str(is,2) + q1 + q2 + q3  + q4

         is = strind(is)
         sm_fitted(is,1) = sm_fitted(is,1) + q1
         sm_fitted(is,2) = sm_fitted(is,2) + q2
         sm_fitted(is,3) = sm_fitted(is,3) + q3
         sm_fitted(is,4) = sm_fitted(is,4) + q4
      end do

      print*, "Stamped/Metered fitted values read"



C     Build class map, shape and min rate flags from vip3rd.94


      open(20,file='vip_stda99.new')
20    format(a5,34x,i2)

      do iv = 1, nvip
         read(20,20) vip, ic
```

```
         vips(iv) = vip
         clsmap(iv) = ic
         if (ic.gt.0) then
            if (ic.lt.62) minrate(ic) = 1
         end if
      end do

      print*,'vips read '


C     Read Permit Transactions

31    format(i2,1x,i1,1x,i3,1x,i1,1x,i3,f14.4,f12.0,f14.4)
32    format(i2,i1,i3,i1,i3,f14.4,2f12.0)

      do ia = 10,13
         iq = qind(ia)
         write (ap,'(i2.2)') ia
         open(30,file='pmt_stda.wi.'//ap)
         i = 0
         ier = 0
         do while (ier.eq.0)
            read (30,31,iostat=ier,end=300) is, it, iw, ip, iv, r, p, w
C            write(*,31) is, it, iw, ip, iv, r, p, w
            id = is
            if (iv.ne.670) then
               if (ip.gt.3) ip = 3
               pieces_by_shp(ip) = pieces_by_shp(ip) + p
               if(iv.gt.0) then
                  ic = clsmap(iv)
C                 print*,'ic is ',ic , ' vip is ',vips(iv), ' iv is ',iv
                  is = strind(is)
                  ii = iind(vips(iv))

                  if (ic.ge.62.and.iw.lt.7) then
                     if ((it.ne.3).and.(ic.ne.0)) then
                        print*,'Bad weight in PERMIT is too small: ',is,
     +                     ' it: ',it,' iw: ',iw,' ip: ',ip,' iv: ',vips(iv),' ic: ',ic,' rpw: ',r,p,w
                     end if
                     it = 3
                  else if (ic.lt.62.and.iw.gt.7) then
                     if ((it.ne.3).and.(ic.ne.0)) then
                        print*,'Bad weight in PERMIT is too large: ',is,
     +                     ' it: ',it,' iw: ',iw,' ip: ',ip,' iv: ',vips(iv),' ic: ',ic,' rpw: ',r,p,w
                     end if
                     it = 3
                  end if

                  if (ic.gt.0) then
                     if (ip.gt.3) ip = 3
                     if (ii.ge.2) then
                        rev_str(id,1) = rev_str(id,1) + r
                        rev_str(id,2) = rev_str(id,2) + r
                     end if
                     wipr(ic,ii,ip,iw,it,is,iq) = wipr(ic,ii,ip,iw,it,is,iq) + r
                     wipp(ic,ii,ip,iw,it,is,iq) = wipp(ic,ii,ip,iw,it,is,iq) + p
                     wipw(ic,ii,ip,iw,it,is,iq) = wipw(ic,ii,ip,iw,it,is,iq) + w
                     if (ii.eq.1) pmtrev(is,iq) = pmtrev(is,iq) + r
                  else
                     if (vips(iv)(2:4).ne.'340') then
                        print*,'No Class ',vips(iv)
                     end if
                  end if

               else
                  print*,'Bad vip remapping ',(iv)
               end if
            end if
         end do
300      print *,' read exit of pmt_stda.wi.',ap,' = ',ier
         close(30)
      end do


C     Calculate Trial Balance Permit and Bravis Own Factors
      write(90,'(a40)') 'Permit/Bravis Own Factors'
      write(90,'(a40)') '   '
      do iq = 1, nq
         do is = 1, nstr
```

```
            if (pmtrev(is,iq).gt.0.) then
               factpmt(is,iq) = nctbpmt(is,iq) / pmtrev(is,iq)
            else
               print *, ' ?!  No PERMIT revenue in strata ',is,' quarter ',iq
            end if
            write(90,90) is,iq,nctbpmt(is,iq),pmtrev(is,iq),factpmt(is,iq)
            pmtrev(is,iq) = 0.
         end do
      end do

400   format(i3,4f8.4,4f14.0)
410   format(i3,8f14.0)
      print *,'  '
      print *,' PERMIT Own Factors and NCTB Revenue: '
      do is = 1, nstr
         write (*,400) is, (factpmt(is,iq),iq=1,nq), (nctbpmt(is,iq), iq=1,nq)
      end do

C     Apply Own Factors and Check

      do iq = 1, nq
         do is = 1, nstr
            do it = 1, ntype
               do iw = 1, nwt
                  do ip = 1, nshp
                     do ii = 1, nind
                        do ic = 1, ncls
                           wipr(ic,ii,ip,iw,it,is,iq) =
     &                        wipr(ic,ii,ip,iw,it,is,iq) * factpmt(is,iq)
                           wipp(ic,ii,ip,iw,it,is,iq) =
     &                        wipp(ic,ii,ip,iw,it,is,iq) * factpmt(is,iq)
                           wipw(ic,ii,ip,iw,it,is,iq) =
     &                        wipw(ic,ii,ip,iw,it,is,iq) * factpmt(is,iq)
                           if (ii.eq.1) then
                              pmtrev(is,iq) = pmtrev(is,iq) +
     &                           wipr(ic,ii,ip,iw,it,is,iq)
                           else
C                               print*,'is ',is
c                               print*,'iq ',iq
c                               print*,'ic ',ic
c                               print*,'ii ',ii
c                               print*,'ip ',ip
c                               print*,'iw ',iw
c                               print*,'it ',it

                              smrev(is,iq) = smrev(is,iq) + wipr(ic,ii,ip,iw,it,is,iq)
                           end if
                        end do
                     end do
                  end do
               end do
            end do
         end do
      end do

      print *,'  '
      print *,' PERMIT Permit Imprint Controlled Revenue and NCTB Revenue: '
      do is = 1, nstr
         write (*,410) is, (pmtrev(is,iq),iq=1,nq), (nctbpmt(is,iq), iq=1,nq)
      end do

C     Calculate Trial Balance Total Factors
      write(90,'(a40)') 'Trial Balance Total Factors'
      write(90,'(a40)') '   '
      do iq = 1, nq
         do is = 1, nstr
            if (pmtrev(is,iq).gt.0.) then
               factall(is,iq) = nctball(is,iq) /
     &            (pmtrev(is,iq))
               write(90,90) is,iq,nctball(is,iq),pmtrev(is,iq),factall(is,iq)
               factall_sm(is,iq) = (sm_fitted(is,iq)+smrev(is,iq)) /
     &            (smrev(is,iq))
               write(90,90) is,iq,(sm_fitted(is,iq) + smrev(is,iq)),smrev(is,iq),factall_sm(is,iq)
            else
               print *, ' ?! No PERMIT or BRAVIS revenue strata ',is,' q ',iq
            end if
            pmtrev(is,iq) = 0.
            smrev(is,iq) = 0.
         end do
      end do
```

```
      print *,'  '
      print *,' All Office Factors and NCTB Revenue: Permit Imprint'
      do is = 1, nstr
         write (*,400) is, (factall(is,iq),iq=1,nq), (nctball(is,iq), iq=1,nq)
      end do
      print *,'  '
      print *,' All Office Factors and NCTB Revenue: Stamped/Metered'
      do is = 1, nstr
         write (*,400) is, (factall_sm(is,iq),iq=1,nq), (sm_fitted(is,iq), iq=1,nq)
      end do


C     Apply Total Factor and Check

      do iq = 1, nq
         do is = 1, nstr
            do it = 1, ntype
               do iw = 1, nwt
                  do ip = 1, nshp
                     do ii = 1, nind
                        do ic = 1, ncls
                           if (ii.eq.1) then
                              wipr(ic,ii,ip,iw,it,is,iq) =
     &                           wipr(ic,ii,ip,iw,it,is,iq) * factall(is,iq)
                              wipp(ic,ii,ip,iw,it,is,iq) =
     &                           wipp(ic,ii,ip,iw,it,is,iq) * factall(is,iq)
                              wipw(ic,ii,ip,iw,it,is,iq) =
     &                           wipw(ic,ii,ip,iw,it,is,iq) * factall(is,iq)
                           else
                              wipr(ic,ii,ip,iw,it,is,iq) =
     &                           wipr(ic,ii,ip,iw,it,is,iq) * factall_sm(is,iq)
                              wipp(ic,ii,ip,iw,it,is,iq) =
     &                           wipp(ic,ii,ip,iw,it,is,iq) * factall_sm(is,iq)
                              wipw(ic,ii,ip,iw,it,is,iq) =
     &                           wipw(ic,ii,ip,iw,it,is,iq) * factall_sm(is,iq)
                           end if
                           if (ii.eq.1) then
                              pmtrev(is,iq) = pmtrev(is,iq) +
     &                           wipr(ic,ii,ip,iw,it,is,iq)
                           else
                              smrev(is,iq) = smrev(is,iq) +
     &                           wipr(ic,ii,ip,iw,it,is,iq)

                           end if
                        end do

                     end do
                  end do
               end do
            end do
         end do
      end do

      print *,'  '
      print *,' PERMIT/BRAVIS Controlled Revenue and NCTB Revenue: Permit Imprint'
      do is = 1, nstr
         write (*,410) is, (pmtrev(is,iq),iq=1,nq), (nctball(is,iq), iq=1,nq)
      end do

      print *,'  '
      print *,' PERMIT/BRAVIS Controlled Revenue and NCTB Revenue: Stamped'
      do is = 1, nstr
         write (*,410) is, (smrev(is,iq),iq=1,nq), (sm_fitted(is,iq), iq=1,nq)
      end do

C     Distribute Bad Weight Data to Identical Weight

      open(50,file='stda.csv',recl=500)
51    format(i2,',',i1,',',20(f16.3,','),f16.3,',',f16.3,',',f16.3)

C     Sum identical and non-identical into result array,
C      bad weight into badwght array

      x=0
      xc=0
      do ii = 1, nind
         do iq = 1, nq
            do is = 1, nstr
               do it = 1, 2     ! skip bad weight
```

```
                  do iw = 1, nwt
                     do ip = 1, nshp
                        do ic = 1, ncls
                           result(1,iw,ip,ic) =
     &                        result(1,iw,ip,ic) +
     &                        wipr(ic,ii,ip,iw,it,is,iq)
                           result(2,iw,ip,ic) =
     &                        result(2,iw,ip,ic) +
     &                        wipp(ic,ii,ip,iw,it,is,iq)
                           result(3,iw,ip,ic) =
     &                        result(3,iw,ip,ic) +
     &                        wipw(ic,ii,ip,iw,it,is,iq)

                           xc = xc + wipr(ic,ii,ip,iw,it,is,iq)
                        end do
                     end do
                  end do
                  do iw = 1, nwt
                     do ip = 1, nshp
                        do ic = 1, ncls
                           badwght(1,ip,ic) = badwght(1,ip,ic) +
     &                        wipr(ic,ii,ip,iw,3,is,iq)
                           badwght(2,ip,ic) = badwght(2,ip,ic) +
     &                        wipp(ic,ii,ip,iw,3,is,iq)
                           badwght(3,ip,ic) = badwght(3,ip,ic) +
     &                        wipw(ic,ii,ip,iw,3,is,iq)
                           xc = xc + wipr(ic,ii,ip,iw,3,is,iq)
                           x= x + wipr(ic,ii,ip,iw,3,is,iq)
                        end do
                     end do
                  end do
               end do
            end do
            print*,'sum of bad weight revenues  ', x
            print*,'sum of total revenues ',xc

            x=0

C     Now distribute bad weight pieces

            do ic = 1, ncls
               do ip = 1, nshp
                  do iw = 1, nwt
                     shrwt(iw) = 0.
                  end do
                  p = 0.
                  do iw = 1, nwt
                     p = p + result(2,iw,ip,ic)
                     shrwt(iw) = shrwt(iw) + result(2,iw,ip,ic)
                  end do
                  if (p.gt.0) then
                     do iw = 1, nwt
                        shrwt(iw) = shrwt(iw)/p
                        if (ii.eq.1) then
                           pishrwt(iw,ic,ip) = shrwt(iw)
                        end if
                        if (badwght(2,ip,ic).gt.0) then
C                        write(*,'(3i3,f6.3)') ic,ip,iw,shrwt(iw)
                        end if
                     end do
                  end if
                  if (badwght(2,ip,ic).gt.0) then
                     if (minrate(ic).eq.1) then
                        if (p.gt.0.) then
                           do iw = 1, nwt
                              if (result(2,iw,ip,ic).gt.0) then
                                 wpp = result(3,iw,ip,ic) / result(2,iw,ip,ic)
                              else
                                 wpp = 0.
                              end if
                              do ir = 1, 2
                                 result(ir,iw,ip,ic) =
     &                              result(ir,iw,ip,ic) +
     &                              badwght(ir,ip,ic)*shrwt(iw)
                              end do
                              x=x+badwght(1,ip,ic)*shrwt(iw)
                              result(3,iw,ip,ic) =
     &                           badwght(2,ip,ic)*(shrwt(iw)) * wpp +
     &                           result(3,iw,ip,ic)
```

```
                  end do
               else
                  do iw = 1, nwt
                     print *, "share at wgt inc ", iw, " is ", pishrwt(iw,ic,ip), " with subclass ", ic
                     if (result(2,iw,ip,ic).gt.0) then
                        wpp = result(3,iw,ip,ic) / result(2,iw,ip,ic)
                     else
                        wpp = 0.
                     end if
                     do ir = 1, 2
                        result(ir,iw,ip,ic) =
     &                     result(ir,iw,ip,ic) +
     &                     badwght(ir,ip,ic)*pishrwt(iw,ic,ip)
                     end do
                     x=x+badwght(1,ip,ic)*pishrwt(iw,ic,ip)
                     result(3,iw,ip,ic) =
     &                  badwght(2,ip,ic)*(pishrwt(iw,ic,ip)) * wpp +
     &                  result(3,iw,ip,ic)
                  end do

c                    print *,' No i/ni weight to dist bad ',ic,',',ip,',',ii
c                    print *,' r = ',badwght(1,ip,ic),' p = ',badwght(2,ip,ic)
                  end if
               else
                  print *,' Bad wght to dist non-minrate ',ic,',',ip,',',ii
               end if
            end if
         end do
      end do
      print*,'sum of badweight revenues after distribution ',x

C     Write Out Data for Importation into Excel


      x=0
      do ic = 1, ncls
         do ip = 1,3
            r = 0.
            p = 0.
            w = 0.
            do iw = 1, nwt
               r = r + result(1,iw,ip,ic)
               x = x + result(1,iw,ip,ic)
               p = p + result(2,iw,ip,ic)
               w = w + result(3,iw,ip,ic)
            end do   ! 1- revenue, 2 pieces 3 weight
            write (50,51) ic, ip, (result(2,iw,ip,ic),iw=1,nwt),
     +           p,r,w
         end do
      end do
      write(50,'(a2)') 'SM'
      do ir = 1,nrpw
         do iw = 1,nwt
            do ip = 1,nshp
               do ic = 1,ncls
                  result(ir,iw,ip,ic) = 0
                  badwght(ir,ip,ic) = 0
               end do
            end do
         end do
      end do
   end do

   print*, 'after writing revenue ',x

C     Write Out Control Factors To File

   open(60,file='eststda_20.control')
61 format(4f10.7)
   do is = 1, nstr
      write (60,61) (factpmt(is,iq),iq=1,nq)
   end do
   do is = 1, nstr
      write (60,61) (factall(is,iq),iq=1,nq)
   end do

   close(60)

   open(60,file='strrev.txt')
```

```
      do is = 1,20
         write(60,'(i3,4f15.2)') is,rev_str(is,1),(rev_str(is,1)-rev_str(is,2)),fuzz(is),rev_str(is,1)-fuzz(is)
      end do

      do ip = 1,nshp
         write(*,'(i4,f26.0)') ip,pieces_by_shp(ip)
      end do

      end

C -------------------------------------------------------------------------
      function qind(iap)

      integer*4 qind, iap

      if (iap.ge.1.and.iap.le.3) then
         qind = 1
      else if (iap.ge.4.and.iap.le.6) then
         qind = 2
      else if (iap.ge.7.and.iap.le.9) then
         qind = 3
      else if (iap.ge.10.and.iap.le.13) then
         qind = 4
      else
         stop  ' invalid accounting period '
      end if

      return
      end

C -------------------------------------------------------------------------
      function strind(is)

      integer*4 strind, is

      if (is.ge.1.and.is.le.17) then
         strind = 1
      else if (is.ge.18.and.is.le.18) then
         strind = 2
      else if (is.ge.19.and.is.le.19) then
         strind = 3
      else if (is.ge.20.and.is.le.20) then
         strind = 4

      else
         stop ' bad strata '
      end if

      return
      end
C -------------------------------------------------------------------------
      function shapechk(ip,flag)

      integer*4 shapechk, ip, flag

      if (flag.eq.1) then
         shapechk = 1       ! letter only vip
      else if (flag.eq.2) then
         if (ip.eq.1) then
            shapechk = 2    ! non-letters vip - move letter to flat
         else
            shapechk = ip
         end if
      else if (flag.eq.3) then  ! all shape
         if(ip.gt.3) ip = 3
         shapechk = ip
      else if (flag.eq.4) then  ! flat only vip
         shapechk = 2
      else
         print*, 'Flag ',flag,'  ip,',ip
         stop ' bad shape flag '
      end if

      return
      end

C -------------------------------------------------------------------------
      function iind(vip)

      integer*4 iind
```

```
      character*5 vip

      if (vip(1:1).eq.'0') then
         iind = 1
      else if (vip(1:1).eq.'3') then
         if (vip(5:5).eq.'1') then
            iind = 2
         else if (vip(5:5).eq.'2') then
            iind = 2
         else if (vip(5:5).eq.'3') then
            iind = 2
         else if (vip(5:5).eq.'4') then
            iind = 2
         else if (vip(5:5).eq.'6') then
            iind = 1
         else if (vip(5:5).eq.'7') then
            iind = 1
         else if (vip(5:5).eq.'9') then
            iind = 1
         else
            iind = -1
         end if
      else
         iind = 2
      end if

      return
      end
C -------------------------------------------------------------------------
```

```
      program weststda_20

C     Paul Loetscher 1-06-96
C
C
C
C
C     Estimate third-class volumes by weight increment and class
C     using PERMIT and BRAVIS data

C     Input files:
C     permit rollup matrices: ../pmt/pmt3rd.wi.*
C     bravis rollup matrices: ../brv/brv3rd.wi.*
C     trial balance revenues: /u/mcb/vol96/nctb/strata.41411
C     lists of permit and bravis finance numbers
C
C     Output files:
C     est3rd.csv  -  to be imported into excel
C     est3rd.factors - control factors


      implicit none

C     Parameters

      integer*4  nvip, nshp, nwt, nstr, ntype, nrpw, nind, ncls
      integer*4  npmt, nq, nap

      parameter  (nvip = 669)   ! number of 3rd class regular rate vips
      parameter  (nshp = 6)     ! number of shapes
      parameter  (nwt  = 20)    ! number of weight increments
      parameter  (nstr = 4)     ! number of revenue strata
      parameter  (ntype = 3)    ! number of transaction types
      parameter  (nrpw = 3)     ! r, p, w are held as a dim in array
      parameter  (nind = 2)     ! number of indicia (for ms trans counts)
      parameter  (ncls = 85)    ! number of subclasses
      parameter  (npmt = 2257)  ! number of permit finance numbers
      parameter  (nq = 4)
      parameter  (nap = 13)

C     Real Storage

      real*8   wipr(ncls,nind,nshp,nwt,ntype,nstr,nq) ! r, p, w  - permit
      real*8   wipp(ncls,nind,nshp,nwt,ntype,nstr,nq) ! r, p, w  - permit
      real*8   wipw(ncls,nind,nshp,nwt,ntype,nstr,nq) ! r, p, w  - permit

      real*8   result(nrpw,nwt,nshp,ncls) ! array to write out

      real*8   shrwt(nwt), pishrwt(nwt,ncls,nshp) ! arrays used for redist
      real*8   badwght(nrpw,nshp,ncls)
      real*8   sm_fitted(nstr,nq)

      real*8   pmtrev(nstr,nq)  ! permit revenue by strata and indicia
      real*8   smrev(nstr,nq)   ! permit revenue by strata and indicia
      real*8   nctbpmt(nstr,nq) ! trial balance permit office rev.
      real*8   nctball(nstr,nq) ! trial balance total revenue by strata
      real*8   factpmt(nstr,nq) ! control factors for permit office revenue
      real*8   factall(nstr,nq) ! control factors for total revenue (nctb)
      real*8   factall_sm(nstr,nq) ! control factors for total revenue (nctb)
      real*8   rpwfact(nq)      ! control factor to total rpw revenue
      real*8   revin(nap)
      real*8   r, p, w, wpp,x,xc
      real*8   pieces_by_shp(nshp),rvtst(nq)
      real*8   q1,q2,q3,q4
      real*8   rev_str(20,2)

C     Integer Storage

      integer*4 ir, ic, ii, ip, iw, it, is, iq, if, ia, iv,id
      integer*4 ier, i
      integer*4 searchc, qind, iind
      integer*4 strind          ! function to assign condensed strata
      integer*4 clsmap(nvip)/nvip*0/ ! map from vip to class index
      integer*4 pflags(nap,npmt) ! flag for finance number in permit for ap
      integer*4 minrate(ncls)/ncls*0./ ! flag for minimum rate pieces
      integer*4 shpflag(ncls)/ncls*0./ ! flag for allowable shapes

C     Character Storage

      character*6   pfins(npmt), fin
```

```
      character*2   ap
      character*5   vips(nvip), vip
      real*8        fuzz(20)
C     Logical Storage

C     Intialize arrays

C     RPW revenue = total regular rate revenue from penalty and franked
C     as classes report


      do ip = 1,nshp
         pieces_by_shp(ip) = 0
      end do
      do iq = 1, nq
         do is = 1, nstr
            sm_fitted(iq,is) = 0
            do it = 1, ntype
               do iw = 1, nwt
                  do ip = 1, nshp
                     do ii = 1, nind
                        do ic = 1, ncls
                           wipr(ic,ii,ip,iw,it,is,iq) = 0.
                           wipp(ic,ii,ip,iw,it,is,iq) = 0.
                           wipw(ic,ii,ip,iw,it,is,iq) = 0.
                        end do
                     end do
                  end do
               end do
            end do
         end do
      end do
      do ic = 1, ncls
         do ip = 1, nshp
            do iw = 1, nwt
               pishrwt(iw,ic,ip) = 0.
               do ir = 1, nrpw
                  result(ir,iw,ip,ic) = 0.

               end do
            end do
         end do
      end do
      do ic = 1, ncls
         do ip = 1, nshp
            do ir = 1, nrpw
               badwght(ir,ip,ic) = 0.
            end do
         end do
      end do
      do iq = 1, nq
         rvtst(iq) = 0
         do is = 1, nstr
            pmtrev(is,iq)     = 0.
            smrev(is,iq)      = 0.
            nctbpmt(is,iq)    = 0.
            nctball(is,iq)    = 0.
            factpmt(is,iq)    = 0.
            factall(is,iq)    = 0.
            factall_sm(is,iq)    = 0.
         end do
      end do

      do is = 1,20
         rev_str(is,1) = 0
         rev_str(is,2) = 0
         fuzz(is) = 0
      end do

      do if = 1,npmt
         do ia = 1,nap
            pflags(ia,if) = 0
         end do
      end do

      open(90,file='factors_20.txt')
90    format(i3,i3,f14.2,f14.2,f10.6)

C     Read finance numbers and reported ap's
```

```
      open(15,file='finsbyap.pmt.00')
15    format(a6,13i2)
      do if = 1, npmt
         read (15,15) pfins(if), (pflags(ia,if),ia=1,nap)
      end do
      close (15)
      print *,' PERMIT finance numbers read '



C     Read Trial balance revenue and assign to Permit, Bravis and
C     Total revenue arrays

      open(18,file='strata.41411')
18    format(a6,i3,f15.2,13f12.2)

      if = 0
      ier = 0
      do iq = 1,nq
         rvtst(iq) = 0
      end do
      do while (ier.eq.0)
         read (18,18,iostat=ier,end=180) fin, is, r, revin
         if = if + 1
         id = is
         do ia = 1,nap
            rev_str(id,1) = rev_str(id,1) + revin(ia)
            fuzz(id) = fuzz(id) + revin(ia)
         end do
         ip = 0
         is = strind(is)
         ip = searchc(pfins,npmt,fin)
         do ia = 1, nap
            iq = qind(ia)
            nctball(is,iq) = nctball(is,iq) + revin(ia)
            if (ip.gt.0) then
               rev_str(id,2) = rev_str(id,2) + revin(ia)
               if(pflags(ia,ip).eq.0) then
                  if (is.eq.2) then
                     print*, 'no ap ', pfins(ip),' ia ', ia ,' ', pflags(ia,ip)
                     rvtst(iq) = rvtst(iq) + revin(ia)
                  end if
               else
                  nctbpmt(is,iq) =  nctbpmt(is,iq) + revin(ia)
               end if
            end if
         end do
      end do

180   print *,' Strata.41411 read with ',if,' records, ier = ',ier



      open(19,file='fit_stda_ye.00')
19    format(i3,4f14.2)

      do i = 1, 20
         read(19,19) is, q1, q2, q3,q4
         rev_str(is,1) = rev_str(is,1) + q1 + q2 + q3 + q4
         rev_str(is,2) = rev_str(is,2) + q1 + q2 + q3  + q4

         is = strind(is)
         sm_fitted(is,1) = sm_fitted(is,1) + q1
         sm_fitted(is,2) = sm_fitted(is,2) + q2
         sm_fitted(is,3) = sm_fitted(is,3) + q3
         sm_fitted(is,4) = sm_fitted(is,4) + q4
      end do

      print*, "Stamped/Metered fitted values read"



C     Build class map, shape and min rate flags from vip3rd.94


      open(20,file='vip_stda99.new')
20    format(a5,34x,i2)

      do iv = 1, nvip
         read(20,20) vip, ic
```

```
         vips(iv) = vip
         clsmap(iv) = ic
         if (ic.gt.0) then
            if (ic.lt.62) minrate(ic) = 1
         end if
      end do

      print*,'vips read '


C     Read Permit Transactions

31    format(i2,1x,i1,1x,i3,1x,i1,1x,i3,f14.4,f12.0,f14.4)
32    format(i2,i1,i3,i1,i3,f14.4,2f12.0)

      do ia = 10,13
         iq = qind(ia)
         write (ap,'(i2.2)') ia
         open(30,file='pmt_stda.wi.'//ap)
         i = 0
         ier = 0
         do while (ier.eq.0)
            read (30,31,iostat=ier,end=300) is, it, iw, ip, iv, r, p, w
C            write(*,31) is, it, iw, ip, iv, r, p, w
            id = is
            if (iv.ne.670) then
               if (ip.gt.3) ip = 3
               pieces_by_shp(ip) = pieces_by_shp(ip) + p
               if(iv.gt.0) then
                  ic = clsmap(iv)
C                 print*,'ic is ',ic , ' vip is ',vips(iv), ' iv is ',iv
                  is = strind(is)
                  ii = iind(vips(iv))

                  if (ic.ge.62.and.iw.lt.7) then
                     if ((it.ne.3).and.(ic.ne.0)) then
                        print*,'Bad weight in PERMIT is too small: ',is,
     +                       ' it: ',it,' iw: ',iw,' ip: ',ip,' iv: ',vips(iv),' ic: ',ic,' rpw: ',r,p,w
                     end if
                     it = 3
                  else if (ic.lt.62.and.iw.gt.7) then
                     if ((it.ne.3).and.(ic.ne.0)) then
                        print*,'Bad weight in PERMIT is too large: ',is,
     +                       ' it: ',it,' iw: ',iw,' ip: ',ip,' iv: ',vips(iv),' ic: ',ic,' rpw: ',r,p,w
                     end if
                     it = 3
                  end if

                  if (ic.gt.0) then
                     if (ip.gt.3) ip = 3
                     if (ii.ge.2) then
                        rev_str(id,1) = rev_str(id,1) + r
                        rev_str(id,2) = rev_str(id,2) + r
                     end if
                     wipr(ic,ii,ip,iw,it,is,iq) = wipr(ic,ii,ip,iw,it,is,iq) + r
                     wipp(ic,ii,ip,iw,it,is,iq) = wipp(ic,ii,ip,iw,it,is,iq) + p
                     wipw(ic,ii,ip,iw,it,is,iq) = wipw(ic,ii,ip,iw,it,is,iq) + w
                     if (ii.eq.1) pmtrev(is,iq) = pmtrev(is,iq) + r
                  else
                     if (vips(iv)(2:4).ne.'340') then
                        print*,'No Class ',vips(iv)
                     end if
                  end if

               else
                  print*,'Bad vip remapping ',(iv)
               end if
            end if
         end do
300      print *,' read exit of pmt_stda.wi.',ap,' = ',ier
         close(30)
      end do


C     Calculate Trial Balance Permit and Bravis Own Factors
      write(90,'(a40)') 'Permit/Bravis Own Factors'
      write(90,'(a40)') '   '
      do iq = 1, nq
         do is = 1, nstr
```

```
            if (pmtrev(is,iq).gt.0.) then
               factpmt(is,iq) = nctbpmt(is,iq) / pmtrev(is,iq)
            else
               print *, ' ?!  No PERMIT revenue in strata ',is,' quarter ',iq
            end if
            write(90,90) is,iq,nctbpmt(is,iq),pmtrev(is,iq),factpmt(is,iq)
            pmtrev(is,iq) = 0.
         end do
      end do

400   format(i3,4f8.4,4f14.0)
410   format(i3,8f14.0)
      print *,'  '
      print *,' PERMIT Own Factors and NCTB Revenue: '
      do is = 1, nstr
         write (*,400) is, (factpmt(is,iq),iq=1,nq), (nctbpmt(is,iq), iq=1,nq)
      end do

C     Apply Own Factors and Check

      do iq = 1, nq
         do is = 1, nstr
            do it = 1, ntype
               do iw = 1, nwt
                  do ip = 1, nshp
                     do ii = 1, nind
                        do ic = 1, ncls
                           wipr(ic,ii,ip,iw,it,is,iq) =
     &                        wipr(ic,ii,ip,iw,it,is,iq) * factpmt(is,iq)
                           wipp(ic,ii,ip,iw,it,is,iq) =
     &                        wipp(ic,ii,ip,iw,it,is,iq) * factpmt(is,iq)
                           wipw(ic,ii,ip,iw,it,is,iq) =
     &                        wipw(ic,ii,ip,iw,it,is,iq) * factpmt(is,iq)
                           if (ii.eq.1) then
                              pmtrev(is,iq) = pmtrev(is,iq) +
     &                           wipr(ic,ii,ip,iw,it,is,iq)
                           else
C                             print*,'is ',is
c                             print*,'iq ',iq
c                             print*,'ic ',ic
c                             print*,'ii ',ii
c                             print*,'ip ',ip
c                             print*,'iw ',iw
c                             print*,'it ',it

                              smrev(is,iq) = smrev(is,iq) + wipr(ic,ii,ip,iw,it,is,iq)
                           end if
                        end do
                     end do
                  end do
               end do
            end do
         end do
      end do

      print *,'  '
      print *,' PERMIT Permit Imprint Controlled Revenue and NCTB Revenue: '
      do is = 1, nstr
         write (*,410) is, (pmtrev(is,iq),iq=1,nq), (nctbpmt(is,iq), iq=1,nq)
      end do

C     Calculate Trial Balance Total Factors
      write(90,'(a40)') 'Trial Balance Total Factors'
      write(90,'(a40)') '   '
      do iq = 1, nq
         do is = 1, nstr
            if (pmtrev(is,iq).gt.0.) then
               factall(is,iq) = nctball(is,iq) /
     &            (pmtrev(is,iq))
               write(90,90) is,iq,nctball(is,iq),pmtrev(is,iq),factall(is,iq)
               factall_sm(is,iq) = (sm_fitted(is,iq)+smrev(is,iq)) /
     &            (smrev(is,iq))
               write(90,90) is,iq,(sm_fitted(is,iq) + smrev(is,iq)),smrev(is,iq),factall_sm(is,iq)
            else
               print *, ' ?! No PERMIT or BRAVIS revenue strata ',is,' q ',iq
            end if
            pmtrev(is,iq) = 0.
            smrev(is,iq) = 0.
         end do
      end do
```

```
      print *,'  '
      print *,' All Office Factors and NCTB Revenue: Permit Imprint'
      do is = 1, nstr
         write (*,400) is, (factall(is,iq),iq=1,nq), (nctball(is,iq), iq=1,nq)
      end do
      print *,'  '
      print *,' All Office Factors and NCTB Revenue: Stamped/Metered'
      do is = 1, nstr
         write (*,400) is, (factall_sm(is,iq),iq=1,nq), (sm_fitted(is,iq), iq=1,nq)
      end do


C     Apply Total Factor and Check

      do iq = 1, nq
         do is = 1, nstr
            do it = 1, ntype
               do iw = 1, nwt
                  do ip = 1, nshp
                     do ii = 1, nind
                        do ic = 1, ncls
                           if (ii.eq.1) then
                              wipr(ic,ii,ip,iw,it,is,iq) =
     &                           wipr(ic,ii,ip,iw,it,is,iq) * factall(is,iq)
                              wipp(ic,ii,ip,iw,it,is,iq) =
     &                           wipp(ic,ii,ip,iw,it,is,iq) * factall(is,iq)
                              wipw(ic,ii,ip,iw,it,is,iq) =
     &                           wipw(ic,ii,ip,iw,it,is,iq) * factall(is,iq)
                           else
                              wipr(ic,ii,ip,iw,it,is,iq) =
     &                           wipr(ic,ii,ip,iw,it,is,iq) * factall_sm(is,iq)
                              wipp(ic,ii,ip,iw,it,is,iq) =
     &                           wipp(ic,ii,ip,iw,it,is,iq) * factall_sm(is,iq)
                              wipw(ic,ii,ip,iw,it,is,iq) =
     &                           wipw(ic,ii,ip,iw,it,is,iq) * factall_sm(is,iq)
                           end if
                           if (ii.eq.1) then
                              pmtrev(is,iq) = pmtrev(is,iq) +
     &                           wipr(ic,ii,ip,iw,it,is,iq)
                           else
                              smrev(is,iq) = smrev(is,iq) +
     &                           wipr(ic,ii,ip,iw,it,is,iq)

                           end if
                        end do

                     end do
                  end do
               end do
            end do
         end do
      end do

      print *,'  '
      print *,' PERMIT/BRAVIS Controlled Revenue and NCTB Revenue: Permit Imprint'
      do is = 1, nstr
         write (*,410) is, (pmtrev(is,iq),iq=1,nq), (nctball(is,iq), iq=1,nq)
      end do

      print *,'  '
      print *,' PERMIT/BRAVIS Controlled Revenue and NCTB Revenue: Stamped'
      do is = 1, nstr
         write (*,410) is, (smrev(is,iq),iq=1,nq), (sm_fitted(is,iq), iq=1,nq)
      end do

C     Distribute Bad Weight Data to Identical Weight

      open(50,file='wtstda.csv',recl=500)
51    format(i2,',',i1,',',20(f16.3,','),f16.3,',',f16.3,',',f16.3)

C     Sum identical and non-identical into result array,
C      bad weight into badwght array

      x=0
      xc=0
      do ii = 1, nind
         do iq = 1, nq
            do is = 1, nstr
               do it = 1, 2     ! skip bad weight
```

```
                  do iw = 1, nwt
                     do ip = 1, nshp
                        do ic = 1, ncls
                           result(1,iw,ip,ic) =
     &                        result(1,iw,ip,ic) +
     &                        wipr(ic,ii,ip,iw,it,is,iq)
                           result(2,iw,ip,ic) =
     &                        result(2,iw,ip,ic) +
     &                        wipp(ic,ii,ip,iw,it,is,iq)
                           result(3,iw,ip,ic) =
     &                        result(3,iw,ip,ic) +
     &                        wipw(ic,ii,ip,iw,it,is,iq)

                           xc = xc + wipr(ic,ii,ip,iw,it,is,iq)
                        end do
                     end do
                  end do
                  do iw = 1, nwt
                     do ip = 1, nshp
                        do ic = 1, ncls
                           badwght(1,ip,ic) = badwght(1,ip,ic) +
     &                        wipr(ic,ii,ip,iw,3,is,iq)
                           badwght(2,ip,ic) = badwght(2,ip,ic) +
     &                        wipp(ic,ii,ip,iw,3,is,iq)
                           badwght(3,ip,ic) = badwght(3,ip,ic) +
     &                        wipw(ic,ii,ip,iw,3,is,iq)
                           xc = xc + wipr(ic,ii,ip,iw,3,is,iq)
                           x= x + wipr(ic,ii,ip,iw,3,is,iq)
                        end do
                     end do
                  end do
               end do
            end do
            print*,'sum of bad weight revenues  ', x
            print*,'sum of total revenues ',xc

            x=0

C     Now distribute bad weight pieces

            do ic = 1, ncls
               do ip = 1, nshp
                  do iw = 1, nwt
                     shrwt(iw) = 0.
                  end do
                  p = 0.
                  do iw = 1, nwt
                     p = p + result(2,iw,ip,ic)
                     shrwt(iw) = shrwt(iw) + result(2,iw,ip,ic)
                  end do
                  if (p.gt.0) then
                     do iw = 1, nwt
                        shrwt(iw) = shrwt(iw)/p
                        if (ii.eq.1) then
                           pishrwt(iw,ic,ip) = shrwt(iw)
                        end if
                        if (badwght(2,ip,ic).gt.0) then
C                        write(*,'(3i3,f6.3)') ic,ip,iw,shrwt(iw)
                        end if
                     end do
                  end if
                  if (badwght(2,ip,ic).gt.0) then
                     if (minrate(ic).eq.1) then
                        if (p.gt.0.) then
                           do iw = 1, nwt
                              if (result(2,iw,ip,ic).gt.0) then
                                 wpp = result(3,iw,ip,ic) / result(2,iw,ip,ic)
                              else
                                 wpp = 0.
                              end if
                              do ir = 1, 2
                                 result(ir,iw,ip,ic) =
     &                              result(ir,iw,ip,ic) +
     &                              badwght(ir,ip,ic)*shrwt(iw)
                              end do
                              x=x+badwght(1,ip,ic)*shrwt(iw)
                              result(3,iw,ip,ic) =
     &                           badwght(2,ip,ic)*(shrwt(iw)) * wpp +
     &                           result(3,iw,ip,ic)
```

```
                  end do
               else
                  do iw = 1, nwt
                     print *, "share at wgt inc ", iw, " is ", pishrwt(iw,ic,ip), " with subclass ", ic
                     if (result(2,iw,ip,ic).gt.0) then
                        wpp = result(3,iw,ip,ic) / result(2,iw,ip,ic)
                     else
                        wpp = 0.
                     end if
                     do ir = 1, 2
                        result(ir,iw,ip,ic) =
     &                     result(ir,iw,ip,ic) +
     &                     badwght(ir,ip,ic)*pishrwt(iw,ic,ip)
                     end do
                     x=x+badwght(1,ip,ic)*pishrwt(iw,ic,ip)
                     result(3,iw,ip,ic) =
     &                  badwght(2,ip,ic)*(pishrwt(iw,ic,ip)) * wpp +
     &                  result(3,iw,ip,ic)
                  end do

c                 print *,' No i/ni weight to dist bad ',ic,',',ip,',',ii
c                 print *,' r = ',badwght(1,ip,ic),' p = ',badwght(2,ip,ic)
               end if
            else
               print *,' Bad wght to dist non-minrate ',ic,',',ip,',',ii
            end if
         end if
      end do
   end do
   print*,'sum of badweight revenues after distribution ',x

C     Write Out Data for Importation into Excel


      x=0
      do ic = 1, ncls
         do ip = 1,3
            r = 0.
            p = 0.
            w = 0.
            do iw = 1, nwt
               r = r + result(1,iw,ip,ic)
               x = x + result(1,iw,ip,ic)
               p = p + result(2,iw,ip,ic)
               w = w + result(3,iw,ip,ic)
            end do   ! 1- revenue, 2 pieces 3 weight
            write (50,51) ic, ip, (result(3,iw,ip,ic),iw=1,nwt),
     +           p,r,w
         end do
      end do
      write(50,'(a2)') 'SM'
      do ir = 1,nrpw
         do iw = 1,nwt
            do ip = 1,nshp
               do ic = 1,ncls
                  result(ir,iw,ip,ic) = 0
                  badwght(ir,ip,ic) = 0
               end do
            end do
         end do
      end do
   end do

   print*, 'after writing revenue ',x

C     Write Out Control Factors To File

   open(60,file='eststda_20.control')
61 format(4f10.7)
   do is = 1, nstr
      write (60,61) (factpmt(is,iq),iq=1,nq)
   end do
   do is = 1, nstr
      write (60,61) (factall(is,iq),iq=1,nq)
   end do

   close(60)

   open(60,file='strrev.txt')
```

```
      do is = 1,20
         write(60,'(i3,4f15.2)') is,rev_str(is,1),(rev_str(is,1)-rev_str(is,2)),fuzz(is),rev_str(is,1)-fuzz(is)
      end do

      do ip = 1,nshp
         write(*,'(i4,f26.0)') ip,pieces_by_shp(ip)
      end do

      end

C -------------------------------------------------------------------------
      function qind(iap)

      integer*4 qind, iap

      if (iap.ge.1.and.iap.le.3) then
         qind = 1
      else if (iap.ge.4.and.iap.le.6) then
         qind = 2
      else if (iap.ge.7.and.iap.le.9) then
         qind = 3
      else if (iap.ge.10.and.iap.le.13) then
         qind = 4
      else
         stop  ' invalid accounting period '
      end if

      return
      end

C -------------------------------------------------------------------------
      function strind(is)

      integer*4 strind, is

      if (is.ge.1.and.is.le.17) then
         strind = 1
      else if (is.ge.18.and.is.le.18) then
         strind = 2
      else if (is.ge.19.and.is.le.19) then
         strind = 3
      else if (is.ge.20.and.is.le.20) then
         strind = 4

      else
         stop ' bad strata '
      end if

      return
      end
C -------------------------------------------------------------------------
      function shapechk(ip,flag)

      integer*4 shapechk, ip, flag

      if (flag.eq.1) then
         shapechk = 1       ! letter only vip
      else if (flag.eq.2) then
         if (ip.eq.1) then
            shapechk = 2    ! non-letters vip - move letter to flat
         else
            shapechk = ip
         end if
      else if (flag.eq.3) then  ! all shape
         if(ip.gt.3) ip = 3
         shapechk = ip
      else if (flag.eq.4) then  ! flat only vip
         shapechk = 2
      else
         print*, 'Flag ',flag,'  ip,',ip
         stop ' bad shape flag '
      end if

      return
      end

C -------------------------------------------------------------------------
      function iind(vip)

      integer*4 iind
```

```
      character*5 vip

      if (vip(1:1).eq.'0') then
         iind = 1
      else if (vip(1:1).eq.'3') then
         if (vip(5:5).eq.'1') then
            iind = 2
         else if (vip(5:5).eq.'2') then
            iind = 2
         else if (vip(5:5).eq.'3') then
            iind = 2
         else if (vip(5:5).eq.'4') then
            iind = 2
         else if (vip(5:5).eq.'6') then
            iind = 1
         else if (vip(5:5).eq.'7') then
            iind = 1
         else if (vip(5:5).eq.'9') then
            iind = 1
         else
            iind = -1
         end if
      else
         iind = 2
      end if

      return
      end
C -------------------------------------------------------------------------
```

```
#!/usr/bin/csh
# Shell Script stdanpdisk
foreach ap (01 02 03 04 05 06 07 08 09 10 11 12 13)

gzcat  /u/mcb/permit00/files/permit.00.stdanp.${ap}.gz > tempfile
 stdanp_roll ${ap} > stdanp_roll.out.${ap}


end
```

```
      program stdanp_roll

C     Mike McGrane   9-21-94
C
C     Check PERMIT system transactions for internal consistency
C


      implicit none

C     Parameters

      integer*4  nvip, nshp, nwt, nstr, nms, ntype, nfin,nrpw, nerr, nind
      integer*4  maxln

      parameter  (nvip = 670)   ! number of 3rd class regular rate vips
      parameter  (nshp = 5)     ! number of shapes
      parameter  (nwt  = 16)    ! number of weight increments
      parameter  (nstr = 20)    ! number of revenue strata
      parameter  (ntype = 4)    ! number of transaction types
      parameter  (nfin = 2257)  ! number of permit finance numbers
      parameter  (nrpw = 3)     ! r, p are held as a dim in array
      parameter  (nerr = 29)    ! number of error codes (incl good trans)
      parameter  (nind = 2)     ! number of indicia (for ms trans counts)
      parameter  (maxln = 50)

C     Real storage

      real*8   rerr(nfin,nerr,nshp,nind) ! r, p, w for errors and good trans
      real*8   perr(nfin,nerr,nshp,nind)
      real*8   werr(nfin,nerr,nshp,nind)
      real*8   rerr_x(nerr,nshp) ! r, p, w for errors and good trans
      real*8   perr_x(nerr,nshp)
      real*8   werr_x(nerr,nshp)
      real*8   fintran(2,nfin,nind,4)
      real*8   rv(nvip)          ! r, p, w for a transaction
      real*8   pv(nvip)
      real*8   wv(nvip)
      real*8   rtv(nvip)
      real*8   rts(2,nvip), fxrts(2,nvip) ! rates for each vip code
      real*8   rts96(2,nvip),rts95(2,nvip) ! rates for each vip code
      real*8   rtot, ptot, wtot, rspc
      real*8   r, p, w, rt, t
      real*8   wpp

C     Integer storage

      integer*4 rpttran(nstr)   ! number of trans by strata

      integer*4 iv, ip, iw, it, is, im, il, if, inf, ie, ir, ii, ctold

      integer*4 ier, i, j, k, lines, ierror, n, nt, ctshp, ctn
      integer*4 searchc, msind, day
      integer*4 itype(nvip)     ! array to indicate type of each vip in trans
      integer*4 iwt(nvip)       ! array to indicate weight increment of each vip
      integer*4 strmap(nfin)    ! map from finance number to strata
      integer*4 cnffins(maxln)
      integer*4 cnterr(nfin,nerr,nshp,nind)
      integer*4 cnterr_x(nerr,nshp)
      integer*4 ntran, iind

C     Character strorage

      character*137  rechead, temphead ! header portion of record
      character*48   recarray(maxln) ! array to be equived with rest of record

      character*5008 record, temprec,out_rec
      character*50   foot
      character*48   errorcodes(nerr)
      character*13   tno, newtran,foo
      character*8    trans,tdate
      character*6    fins(nfin), fin, lfin
      character*6    nffins(maxln)
      character*5    vips(nvip), vip
      character*5    vipprn(maxln)
      character*5    perno
      character*4    len,clines
      character*2    viptype(nvip), tcode, seqno, newseq, ap
      character*2    indic
      character*2    nprof
```

```
      character*1    viptyp(maxln) ! identical/non-identical/bad-weight
      character*1    mixid,surflag,cprc

      integer*4      length,badlen,pplen
      integer*4      year
      integer*4      iob,eob

      real*8         rtprn(maxln),pprn(maxln),rprn(maxln),wtprn(maxln)

c      character*20   rec_chk


C     Logicals

      logical   tflag, fflag, first,ppflag

C     Common block for checking subroutine
      common  /check/ fins,rv, pv, wv, rtv, rts, fxrts, rtot, ptot, wtot, rspc,
     &   wpp, itype, iwt, ip,if, viptype, vips, tcode, ppflag,lines

      CALL GETARG(1,ap)

c      open(1, file='bugfile', iointent='output')
c2     format(a20)

      print*, 'Starting bin prog for ap ', ap

      open(40,file='permit.new.stdanp.'//ap,recl=5008)
      open(59,file='tempfile',recl=5008)

      print*, 'Opened good data file'

      open(45,file='badrec.stda.'//ap,recl=5008)

      print*, 'Opened bad data file'

C     Read list of vip codes with rates and flags

      open(15,file='vipnp99.new')
16    format(a5,3x,4f8.3)

      do iv = 1, nvip
         read (15,16) vip, (rts96(i,iv),i=1,2),(rts95(i,iv),i=1,2)
         if (vip(1:1).eq.' ')  vip(1:1)='0'
         vips(iv) = vip
      end do

      print *,' vipnp99.new read '
      close (15)

C     Read list of finance numbers and strata they map to

      open(17,file='finstrata.00')
18    format(a6)
      do if = 1, nfin
         read (17,18) fins(if)
         strmap(if) = 20
         if (strmap(if).eq.99) then
            strmap(if) = 20
            print *,' zero strata for ',fins(if)
         end if
      end do

      print *,' read finance numbers and strata '
      close (17)

C     Read file of error codes

      open(19,file='error.codes')
      do ier = 1, nerr
         read (19,'(a48)') errorcodes(ier)
      end do
      print *,"error codes read"

C     Initialize storage arrays

      do if = 1, nfin
         do ie = 1, nerr
            do ip = 1, nshp
               do ii = 1, nind
```

```
                  rerr(if,ie,ip,ii) = 0.
                  perr(if,ie,ip,ii) = 0.
                  werr(if,ie,ip,ii) = 0.
                  cnterr(if,ie,ip,ii) = 0
               end do
            end do
         end do
      end do

      do ie = 1, nerr
         do ip = 1,nshp
            rerr_x(ie,ip) = 0.
            perr_x(ie,ip) = 0.
            werr_x(ie,ip) = 0.
            cnterr_x(ie,ip) = 0
         end do
      end do

      do ir = 1,2
         do iv = 1,nvip
            rts(ir,iv) = 0
            fxrts(ir,iv) = 0
         end do
      end do

      do ie = 1,2
         do if = 1,nfin
            do ii = 1,nind
               do ip = 1,4
                  fintran(ie,if,ii,ip) = 0
               end do
            end do
         end do
      end do

C     Read through permit file and pigeon hole records

34    format(a137)
31    format(a6,a2,a13,a2,a2,a5,a8,a1,f12.2,a2,f12.2,a13,i1,a1,f8.4,f12.0,f14.4,20x,i3)
32    format(6x,a5,1x,f6.3,f12.0,f18.7)
33    format(8x,a13,a2,16x,f12.2)
      ier = 0
      inf = 0
      ctn = 0
      ctshp = 0
      ntran = 0
      lfin = ' '
      first = .true.


98    do while (ier.eq.0)
         ppflag=.false.
         tflag=.true.
         do il = 1,maxln
            viptyp(il)='X'
            wtprn(il) = 0
            pprn(il) = 0
            rtprn(il) = 0
            rprn(il) = 0
         end do
         do iv = 1, nvip
            rv(iv) = 0.
            pv(iv) = 0.
            wv(iv) = 0.
            rtv(iv) = 0.
         end do
         read (59,'(a)',iostat=ier,end=100) record
         read(record,34)  rechead

c*************************************************************
c         read (record, 2) rec_chk
c         write(1,2) rec_chk
c*************************************************************

         ntran = ntran + 1
c         if (ntran.gt.282530) then
c            print*, record
c         end if
99       read (rechead,31) fin,indic,tno,seqno,tcode,perno,tdate,mixid,rtot,
     &      nprof,rspc,foo,ip,surflag,wpp, ptot, wtot, lines
```

```
badlen = 138 + lines*48

read(tno(5:7),'(i3)') day
read(tno,'(i4)') year
if ((year.eq.1999).and.(day.ge.010)) then
   do iv = 1,nvip
      rts(1,iv) = rts96(1,iv)
      rts(2,iv) = rts96(2,iv)
      fxrts(1,iv) = rts96(1,iv)
      fxrts(2,iv) = rts96(2,iv)
   end do
else
   do iv = 1,nvip
      rts(1,iv) = rts96(1,iv)
      rts(2,iv) = rts96(2,iv)
      fxrts(1,iv) = rts96(1,iv)
      fxrts(2,iv) = rts96(2,iv)
   end do
end if
if (ip.gt.4) then
   print *, "Shape category = ", ip
   print *, "Transaction number ", ntran
   ctshp = ctshp + 1
   ip=4
end if

if (tcode(1:1).eq.'N') then
   ctn = ctn + 1
end if
if (indic.eq.'PI') then
   iind=1
else
   iind=2
end if
if (fin.ne.lfin) then  ! look up finance number
   lfin = fin

   if = searchc(fins,nfin,fin)
   if (if.eq.0) then   ! if finance number not found
      print *, " "
      print *, "Finance Number not found ", fin
      print *, " "
      if (inf.eq.0) then ! first one not found ?
         inf = 1

         nffins(inf) = fin
      else
         fflag = .false.
         do i = 1, inf ! check if this one has been recorded
            if (fin.eq.nffins(i)) then
               fflag = .true.
            end if
         end do
         if (.not.fflag) then ! if a new one add to arrays
            inf = inf + 1
            if (inf.le.-1) then
               nffins(inf) = fin
            else
               print*,'finno mapping error ',fin
               stop  ' ***** ERROR: too many unmapped finance numbers **** '
            end if
         end if
      end if
      is = 0
   else
      is = if          ! if a good finance number, map to strata
   end if
end if
nt = 1

if (if.gt.0) then      ! if a good finance number
   do il=1,lines
      n  = 138+(il-1)*48
      recarray(il) = record(n:n+47)
   end do

   if (lines.gt.0) then ! if there is vip detail
      do il = 1, lines
         read (recarray(il),32)  vip, rt, p, r
         if (recarray(il)(2:5).eq.'6000') then ! Parcel Surcharge
```

```
                  vip = '99999'
                  if (abs(p-(r*10.0)).gt.1.0) then
                     print*,'vip6000 ',p,' ',r
                  end if
               end if
C              Check processing category
               if (vip(5:5).eq.'1') then
                  if (ip.ne.1) then
                     print*, 'Pcat error ',ip,' ',vip
C                     print*,record
                     ip = 1
                  end if
               else if (vip(5:5).eq.'2') then
                  if (ip.ne.2) then
                     print*,'Pcat error ',ip,' ',vip
C                     print*,record
                     ip = 2
                  end if
               else if (vip(5:5).eq.'4') then
                  if (ip.lt.3) then
                     print*,'Pcat error ',ip,' ',vip
C                     print*,record
                     ip = 3
                  end if
               else if (vip(5:5).eq.'6') then
                  if (ip.ne.1) then
                     print*,'Pcat error ',ip,' ',vip
C                     print*,record
                     ip = 1
                  end if
               else if (vip(5:5).eq.'7') then
                  if (ip.lt.2) then
                     print*,'Pcat error ',ip,' ',vip
C                     print*,record
                     ip = 2
                  end if
               else if (vip(5:5).eq.'9') then
                  if (ip.lt.3) then
                     print*,'Pcat error ',ip,' ',vip
C                     print*,record
                     ip = 3
                  end if
               else
                  print*,'No processing category ',vip
                  ierror = 29
                  ip = 1
               end if

               if (vip(1:1).eq.' ')  vip(1:1)= '0'
               if (vip.eq.'03490') vip ='03401'
               vipprn(il) = vip
               rtprn(il) = rt
               if ((vip(1:1).eq.'3').and.(vip(2:4).ne.'340')) then
                  if (vip(5:5).gt.'5') then ! wt portion
                     w=p/10000
                     p=0.
                  else
                     w=0.
                  end if
               end if
               iv = searchc(vips,nvip,vip)
               if (iv.gt.0) then ! look up vip


                  if (vip(1:1).eq.'3') then
                     rv(iv) = rv(iv) + r
                     pv(iv) = pv(iv) + p
                     wv(iv) = wv(iv) + w
                  else
                     rv(iv) = r
                     rtv(iv) = rt
                     if (vip(5:5).lt.'5') then ! piece vip
                        pv(iv) = p

                     else if (vip.ne.'99999') then ! pound vip
                        wv(iv) = p / 10000.0
                     else
                        rv(iv) = r
                        pv(iv) = p
                     end if
```

```
            end if
         else          ! bad vip code
            ! once maps have been updated for 94 this branch
            ! should not be reached
            print *,' vip not in vip_stda96.dat ',vip
            tflag=.false.
         end if
      end do

      if(tflag) then
         call check_stda(tflag,ierror) !perform consistency and error chk
         il = 0
         lines = 0
         do iv = 1,nvip
            if((rv(iv).gt.0).or.(pv(iv).gt.0)) then
               il = il + 1
               lines = lines + 1
               vipprn(il) = vips(iv)
               write(viptyp(il),'(i1.1)') itype(iv)
               wtprn(il) = wv(iv)
               pprn(il) = pv(iv)
               if (vips(iv)(5:5).lt.'5') then
                  rtprn(il) = rv(iv)/pv(iv)
               else if (vips(iv).ne.'99999') then
                  rtprn(il) = rv(iv)/wv(iv)
               else
                  rtprn(il) = rv(iv)/pv(iv)
               end if
               rprn(il) = rv(iv)
            end if
         end do
      end if

      if (tflag) then  ! good transaction
         length = 137
         write(rechead(135:137),'(i3.3)') lines
         write(rechead(79:79),'(i1.1)') ip
         write(rechead(101:114),'(f14.4)') wtot

         rechead(1:6) = fin
         write(out_rec(1:137),34) rechead
         eob = 137

         do il = 1,lines
            iob = eob + 1
            eob = iob + 49
            write(foot,'(1x,a5,a2,f6.3,f12.0,2f12.4)') vipprn(il),viptyp(il),rtprn(il),pprn(il),wtprn(il),rprn(il)
            write(out_rec(iob:eob),'(a49)') foot
            length = length + 50
         end do

         write(40,'(a)') out_rec(1:length)

         i = 0
         i = ierror
         rerr(if,i,ip,iind) = rerr(if,i,ip,iind) + rtot ! good trans are counted under
         perr(if,i,ip,iind) = perr(if,i,ip,iind) + ptot ! error types 1, 2, 3, and 24
         werr(if,i,ip,iind) = werr(if,i,ip,iind) + wtot
         cnterr(if,i,ip,iind) = cnterr(if,i,ip,iind) + 1 ! transaction count
         ! count transactions for mailing size tables
         if (ii.eq.0) then
            ii = 1     ! permit imprint
         else
            ii = 2     ! precan or meter
         end if
      else             ! bad transaction flag
         write(45,'(a)')  record(1:badlen)
C         write(45,'(a)') record(1:badlen)
         if (ppflag) write(45,'(a)') record(1:badlen)
C         write(45,'(a)') temprec(1:pplen)
         rerr(if,ierror,ip,iind) = rerr(if,ierror,ip,iind) + rtot ! store totals under
         perr(if,ierror,ip,iind) = perr(if,ierror,ip,iind) + ptot ! error number
         werr(if,ierror,ip,iind) = werr(if,ierror,ip,iind) + wtot
         cnterr(if,ierror,ip,iind) = cnterr(if,ierror,ip,iind) + 1
      end if
   else                ! lines = 0
      ierror = 26
      rerr(if,ierror,ip,iind) = rerr(if,ierror,ip,iind) + rtot ! store totals under
      perr(if,ierror,ip,iind) = perr(if,ierror,ip,iind) + ptot ! error number
      werr(if,ierror,ip,iind) = werr(if,ierror,ip,iind) + wtot
```

```
            cnterr(if,ierror,ip,iind) = cnterr(if,ierror,ip,iind) + 1
         end if
      else                    ! finance number not in strata map
         ierror = 27
         rerr(if,ierror,ip,iind) = rerr(if,ierror,ip,iind) + rtot ! store totals under
         perr(if,ierror,ip,iind) = perr(if,ierror,ip,iind) + ptot ! error number
         werr(if,ierror,ip,iind) = werr(if,ierror,ip,iind) + wtot
         cnterr(if,ierror,ip,iind) = cnterr(if,ierror,ip,iind) + 1
      end if

      if (nt.eq.2) then       ! process next tran
         record = temprec
         rechead = temphead
         go to 99
      end if


   end do

100   print*,' read exit of ier =  ',ier
      print*,' Number of transactions read = ',ntran
c      print*,' Number of transaction dates before July 1, 1996 ', ctold
      print*,' Number of shape category 5 transactions ', ctshp
      print*,' Number of non-identical records ', ctn

c      write (98, '(a)') 'do we hit this part?'  ---- not bug



C     write out data arrays

C     write out processing summary and error information to file

      do if =1,nfin
         do ie = 1,nerr
            do ip = 1,nshp
               do ii=1,nind
                  cnterr_x(ie,ip)=cnterr_x(ie,ip) + cnterr(if,ie,ip,ii)
                  rerr_x(ie,ip)=rerr_x(ie,ip) + rerr(if,ie,ip,ii)
                  perr_x(ie,ip)=perr_x(ie,ip) + perr(if,ie,ip,ii)
                  werr_x(ie,ip)=werr_x(ie,ip) + werr(if,ie,ip,ii)
               end do
            end do
         end do
      end do

      open(50,file='stda.summary.'//ap)
51    format(i2,2x,i8,2x,f14.2,2f12.0,2x,a48)
52    format(4x,i8,2x,f14.2,2f12.0,2x,' Total All Transactions ')

      do ip = 1, nshp
         write (50,*) ' Third-Class Regular Rate Processing Summary '
         write (50,*) ' AP ',ap,', PFY 1996 '
         write (50,*) ' Processing Category ', ip
         write (50,*) '   # Trans     Revenue         Pieces        Weight      Description '
         write (50,*) '   -------     -------         ------        ------      ----------- '
         rtot = 0.
         ptot = 0.
         wtot = 0.
         ntran = 0.
         do ie = 1, nerr
            write(50,51) ie, cnterr_x(ie,ip), rerr_x(ie,ip), perr_x(ie,ip),
     &         werr_x(ie,ip), errorcodes(ie)
            ntran = ntran + cnterr_x(ie,ip)
            rtot = rtot + rerr_x(ie,ip)
            ptot = ptot + perr_x(ie,ip)
            wtot = wtot + werr_x(ie,ip)
         end do
         write (50,*)
         write (50,52) ntran, rtot, ptot, wtot
         write (50,*)
      end do

      do if =1,nfin
         do ie = 1,nerr
            do ip = 1,nshp
               do ii = 1,nind
                  if (ie.le.3) then
                     fintran(1,if,ii,1) = fintran(1,if,ii,1) + cnterr(if,ie,ip,ii)
                     fintran(1,if,ii,2) = fintran(1,if,ii,2) + rerr(if,ie,ip,ii)
```

```
                  fintran(1,if,ii,3) = fintran(1,if,ii,3) + perr(if,ie,ip,ii)
                  fintran(1,if,ii,4) = fintran(1,if,ii,4) + werr(if,ie,ip,ii)
               else
                  fintran(2,if,ii,1) = fintran(2,if,ii,1) + cnterr(if,ie,ip,ii)
                  fintran(2,if,ii,2) = fintran(2,if,ii,2) + rerr(if,ie,ip,ii)
                  fintran(2,if,ii,3) = fintran(2,if,ii,3) + perr(if,ie,ip,ii)
                  fintran(2,if,ii,4) = fintran(2,if,ii,4) + werr(if,ie,ip,ii)
               end if
            end do
         end do
      end do
   end do

   open(70,file='finrev.dat.'//ap,recl=1000)
55 format(a6,4(f8.0,f16.4,f16.0,f16.4))
   do if = 1,nfin
      write(70,55) fins(if),((fintran(1,if,ii,ip),ip=1,4),(fintran(2,if,ii,ie),ie=1,4),ii=1,nind)
   end do

c     close(1)

   end



C ---------------------------------------------------------------------------
      function msind(ptot)

      integer*4   msind
      real*8      ptot

      if (ptot.lt.200) then
         msind = 1
      else if (ptot.lt.250) then
         msind = 2
      else if (ptot.lt.300) then
         msind = 3
      else if (ptot.lt.350) then
         msind = 4
      else if (ptot.lt.400) then
         msind = 5
      else if (ptot.lt.450) then
         msind = 6
      else if (ptot.lt.500) then
         msind = 7
      else if (ptot.lt.750) then
         msind = 8
      else if (ptot.lt.1000) then
         msind = 9
      else if (ptot.lt.2000) then
         msind = 10
      else if (ptot.lt.5000) then
         msind = 11
      else if (ptot.lt.10000) then
         msind = 12
      else if (ptot.lt.25000) then
         msind = 13
      else if (ptot.lt.50000) then
         msind = 14
      else if (ptot.lt.100000) then
         msind = 15
      else if (ptot.lt.250000) then
         msind = 16
      else if (ptot.lt.500000) then
         msind = 17
      else if (ptot.ge.500000) then
         msind = 18
      end if

      return
      end
C ---------------------------------------------------------------------------
```

```
      program wgt_stdnp_roll2

C     Paul Loetscher
C     Roll up raw PERMIT system transactions for third class
C
C     1)  RPW x vip x shape x wt incr x strata x ntype
C
C       where ntype = 1 - identical
C                     2 - non-identical
C                     3 - bad weight (identical or non-identical)
C     Check PERMIT system transactions for internal consistency
C

      implicit none

      include 'permit_stda.h'

C     Storage for rollup

      integer*4  nvip, nfin, nrpw, ntype,nwt,nshp,nstrata

      parameter  (nvip = 670)   ! number of 3rd class regular rate vips
      parameter  (nshp = 6)     ! number of shapes
      parameter  (nrpw = 3)     ! r, p w are held as a dim in array
      parameter  (ntype = 3)    ! number of transaction types
      parameter  (nwt  = 20)    ! number of weight increments
      parameter  (nfin = 2257)  ! number of permit finance numbers
      parameter  (nstrata  = 20) ! number of NCTB strata


C     Real storage

      real*8         wi(nrpw,nvip,nshp,nwt,ntype,nstrata) ! r,p,w by weight increment

      real*8         vipwpp,viprtx(nvip,2)

      integer*4      ir,iv, ip, iw, if, it,is,il2,iv2
      integer*4      searchc
      integer*4      wind       ! to indicate weight increment of each vip
      integer*4      strata(nfin)
      character*5    vips(nvip),vipx
      character*6    fins(nfin),lastfin
      character*2    ap
      character*1    pc,lb
      integer*4      pci,lbi
      integer*4      pcat_fun


      logical        fndpc, fndwgt,debug

      call getarg(1,ap)

      debug = .false.

      open(30,file='badwt2.stda00',recl=5008)
      open(40,file='pclb2.stda98',recl=5008)
31    format(a4,a6,a2,a13,a2,a2,a5,a1,f12.2,a2,f12.2,a5,a2,a1,a1,a1,a1,a1,a1,a1,a1,f8.4,f12.0,f14.4,a4,
     &   20(a2,a5,f7.3,f12.0,f18.7,f14.4))


      open(15,file='vipnp99.new')
16    format(a5,3x,2f8.3)
      do iv = 1, nvip
         read (15,16) vip,viprtx(iv,1),viprtx(iv,2)
         if (vip(1:1).eq.' ')  vip(1:1)='0'
         vips(iv) = vip
      end do

      print *,' vip_stda.dat read '
      close (15)

      open(17,file='finstrata.00')
18    format(a6,7x,i3)
      do if = 1, nfin
         read (17,18) fins(if),strata(if)
         if (strata(if).eq.99) strata(if)=20
         if (strata(if).eq.0) strata(if)=20
      end do

      print *,' finstrata.00 read '
```

```
      close (17)

C      Initialize arrays

      lastfin='XXXXXX'
      do ir = 1, nrpw
         do iv = 1, nvip
            do ip = 1,nshp
               do iw = 1, nwt
                  do it = 1,ntype
                     do is = 1,nstrata
                        wi(ir,iv,ip,iw,it,is) = 0.
                     end do
                  end do
               end do
            end do
         end do
      end do

      print*, 'Initialized'

C*************READ IN PERMIT TRANSTACTION*********************************
C*************************************************************************

      open(20,file='datafile.dat',recl=5008)
c      open(25, file='datafile.dat', iointent='input')
15    format(a6,a2,a13,a2,a2,a5,a8,a1,f12.2,a2,f12.2,a5,a2,8a1,f8.4,f12.0,
     &   f14.4,20x,a3,a4871)
25    format(100(1x,a5,1x,a1,f6.3,f12.0,f12.4,f11.3,1x))
      ier = 0
      cnt = 0


      do while (ier.eq.0)

C     Read permit file

         include 'permit_read_stda.h'


C *****************END PERMIT READ ************************************

         if (mod(cnt,25000).eq.0) print*, 'Working on transaction ',cnt



         if (fin.ne.lastfin) then
            if = searchc(fins,nfin,fin)
            if (if.gt.0) then
               is = strata(if)
            else
               print*,'Finance number not found ',fin
            end if
         end if

C         if (debug) print*,'lines,',lines
         if (perno(1:1).ne.'G') then ! Insert 'G' if Government transactions are to be shipped
            do il = 1, lines
               iv = searchc(vips,nvip,vipd(il))
               if (iv.gt.0) then
                  read(viptyp(il),'(i1)') it
                  if (it.eq.0) then
C                     print*,'type error ' ,it , ' ' ,vipd(il)

                     if (vipd(il).eq.'99999') then
                        it = 1
                     end if
                  end if
                  ip = pcat_fun(vips(iv)(5:5))
C        Check for parcel Surcharge
                  if (vips(iv)(5:5).eq.'4') then
                     if ((vips(iv)(1:1).eq.'0').or.(vips(iv)(1:1).eq.'3')) then
                        if (abs(1-viprt(il)/viprtx(iv,1)).le.0.05) then
                           ip = 6
                        end if
                     else if (vippc(il).gt.0) then
                        vipwpp = vipwt(il)/vippc(il)
                        if (abs(1.0 - (viprt(il)/(viprtx(iv,1)+(vipwpp*viprtx(iv,2))))).le.0.10) then
                           ip = 6
                        end if
```

```
            end if
         end if

         if (vippc(il).gt.0) then
            vipwpp = vipwt(il)/vippc(il)
         end if
      else
         print*,'vip not found', vipd(il)
         print*, rec
      end if
      fndwgt = .false.
      fndpc = .false.

      if ((vipd(il)(1:1).eq.'0').or.(vipd(il)(1:1).eq.'3')) then
         if ((vipd(il)(3:3).ge.'5').and.(vipd(il)(5:5).le.'4')) then
            if (debug) print*, 'entering first remap'
            vipwpp = 0
            read(vipd(il),'(4x,i1)') pci
            if (pci.eq.3) pci = 1
            lbi = pci + 5
            write(lb,'(i1)') lbi
            vipx = vipd(il)(1:4)//lb
            iv2 = searchc(vips,nvip,vipd(il))
            do il2 = 1, lines
               if ((vipx.eq.vipd(il2))) then
                  if ((vippc(il).gt.0)) then
                     vipwpp = vipwt(il2)/vippc(il)
                     fndwgt = .true.
                  end if
               end if
            end do

            if (fndwgt.eq..false.) then
               write(40,31) len,fin,indic,tno,seqno,tcode,perno,
&                 mix_indic,rtot,non_prof,sp_fees,perno_for,gst_typ,
&                 co_used,cpp_used,op_proc,class,by_for,pre_srt,shape,
&                 surflag,wpp,ptot,wtot,clines,('  ',vipd(il2),
&                 viprt(il2),vippc(il2),vipwt(il2),viprev(il2),
&                 il2 = 1,lines)

            end if

         else if ((vipd(il)(3:3).ge.'5').and.(vipd(il)(5:5).ge.'5')) then
            if (debug) print*,'entering second remap'
            vipwpp = 0
            read(vipd(il),'(4x,i1)') lbi
            pci= lbi - 5
            write(pc,'(i1)') pci
            vipx = vipd(il)(1:4)//pc



            iv2 = searchc(vips,nvip,vipd(il))
            do il2 = 1, lines
               if ((vipx.eq.vipd(il2))) then
                  if ((vippc(il2).gt.0)) then
                     vipwpp = vipwt(il)/vippc(il2)
                     fndpc = .true.
                  end if
               end if
            end do
            if (debug) then
               print*,'vipd(il) ',vipd(il),' vipx ', vipx , 'vipwpp', vipwpp
            end if
            if (fndpc.eq..false.) then
               write(40,31) len,fin,indic,tno,seqno,tcode,perno,
&                 mix_indic,rtot,non_prof,sp_fees,perno_for,gst_typ,
&                 co_used,cpp_used,op_proc,class,by_for,pre_srt,shape,
&                 surflag,wpp,ptot,wtot,clines,('  ',vipd(il2),
&                 viprt(il2),vippc(il2),vipwt(il2),viprev(il2),
&                 il2 = 1,lines)
            end if
         end if
      end if
      iw = wind(vipwpp)
      if ((iw.le.0)) then
         if ((vipd(il)(2:4).ne.'340')) then

            if (ptot.gt.0) then
               vipwpp=wtot/ptot
```

```
                  end if
                  iw = wind(vipwpp)
                  if (iw.lt.0) then
                     print*,'Bad weight 2 ',vipd(il),' ',wtot,' ',ptot
                     it = 3
                     iw = 1
                     print*, rec
                  end if
                  vipwt(il)=vippc(il)*vipwpp
                  wi(1,iv,ip,iw,it,is) = wi(1,iv,ip,iw,it,is) + viprev(il)
                  wi(2,iv,ip,iw,it,is) = wi(2,iv,ip,iw,it,is) + vippc(il)
                  wi(3,iv,ip,iw,it,is) = wi(3,iv,ip,iw,it,is) + vipwt(il)
               else if ((vipd(il)(2:4).eq.'340')) then
                  it=1
                  if (ptot.gt.0) then
                     vipwpp=wtot/ptot
                  end if
                  iw = wind(vipwpp)
                  vipwt(il)=vippc(il)*vipwpp
                  if (iw.lt.0) then
                     print*,'BAD weight 1 ',wtot, 'ptot',ptot,' ',lines
                  else
                     wi(1,iv,ip,iw,it,is) = wi(1,iv,ip,iw,it,is) + viprev(il)
                     wi(2,iv,ip,iw,it,is) = wi(2,iv,ip,iw,it,is) + vippc(il)
                     wi(3,iv,ip,iw,it,is) = wi(3,iv,ip,iw,it,is) + vipwt(il)
                  end if
               end if

            else
               wi(1,iv,ip,iw,it,is) = wi(1,iv,ip,iw,it,is) + viprev(il)
               wi(2,iv,ip,iw,it,is) = wi(2,iv,ip,iw,it,is) + vippc(il)
               wi(3,iv,ip,iw,it,is) = wi(3,iv,ip,iw,it,is) + vipwt(il)
            end if
         end do
      else
         print*,'Skipping Government ',perno
      end if

   end do

100   print*,'read ',cnt
      print*, 'ier = ', ier

      open(50,file='pmt_stda.wi.'//ap)
51    format(i2,1x,i1,1x,i3,1x,i1,1x,i3,f14.4,f12.0,f14.4)

      do is = 1,nstrata
         do it = 1, ntype
            do iw = 1, nwt
               do ip = 1, nshp
                  do iv = 1, nvip
                     if ((wi(1,iv,ip,iw,it,is).gt.0.0).or.(wi(2,iv,ip,iw,it,is).gt.0.0)) then
                        write(50,51) is, it, iw, ip, iv,
     &                     (wi(ir,iv,ip,iw,it,is), ir = 1, nrpw)
                     end if
                  end do
               end do
            end do
         end do
      end do
      close(50)

      close(20)


      end

C ---------------------------------------------------------------------------
C       wind  - return index of weight graduation
C
C       mrm  8-14-92

      function wind(wpp)

      integer*4   wind, ioz
      real*8      wpp, ozs, oz2

      ozs = wpp * 16.0
      ioz = int(ozs)
      if ((ozs.gt.4).and.(ozs.le.16)) then        ! wgt greater than 4 oz, increment by 1
```

```
         if ((ozs-dble(ioz)).gt.0.0) ioz = ioz + 1
         if ((ioz.ge.5).and.(ioz.le.16)) then
            wind = ioz + 4      ! Subsript
         end if
      else if ((ozs.gt.0).and.(ozs.le.4)) then
         oz2 = ozs - ioz
         if (oz2.eq.0) then
            wind = ioz*2
         else if (oz2.gt.0.5) then
            wind = (ioz+1)*2
         else
            wind = (ioz+1)*2-1
         end if

      else
         wind = -9              ! piece is too heavy or is negative.
      end if

      if (wpp.eq.0.0) wind = 0  ! mixed weight transaction

      return
      end

C-------------------------------------------------------------------------------

      function pcat_fun(vip)

      integer*4    pcat_fun
      character*1  vip

      pcat_fun = -9

      if (vip.eq.'1') then
         pcat_fun  = 1
      else if (vip.eq.'2') then
         pcat_fun = 2
      else if (vip.eq.'4') then
         pcat_fun = 3
      else if (vip.eq.'6') then
         pcat_fun = 1
      else if (vip.eq.'7') then
         pcat_fun = 2
      else if (vip.eq.'9') then
         pcat_fun = 3
      end if

      return
      end
```

```
      program fitdatnp

c     Author:  tcg, lpl  (11/21/97)

c     Purpose:  Write out stamped/metered data for nctb data by AP
c               Uses the corrected NCTB data

      implicit none

      integer*4      maxfin, nap
      parameter      (maxfin=30000)
      parameter      (nap=13)

      character*6   finno(maxfin),finx
      character*2   apx,ape
      integer*4     iap

      integer*4     ier,cnt,nfin,if,searchc

      real*8        rev(maxfin,nap)
      real*8        end411(maxfin,nap),end412(maxfin,nap)
      real*8        beg411(maxfin,nap),beg412(maxfin,nap)
      real*8        a411,a412

      do if = 1,maxfin
         do  iap = 1, nap
            end411(if,iap)=0.
            end412(if,iap)=0.
            beg411(if,iap)=0.
            beg412(if,iap)=0.
            finno(if) ='xxxxxx'
            rev(if,iap)=0.
         end do
      end do

      print*, 'Matrices initialized'

      open (10,file='revfile')
C     Sorted strata.41414

10    format(a6,18x,13f12.2)

      ier=0

      cnt = 0
      do while(ier.eq.0)
         cnt = cnt + 1
         read(10,10,iostat=ier,end = 100) finno(cnt), (rev(cnt,iap), iap = 1, nap)
      end do

100   print*,'read revfile.ye with ier ',ier, ' and cnt ',cnt

      nfin=cnt

      do iap = 1, nap

         write(apx,'(i2.2)') iap
         write(ape,'(i2.2)') iap-1

         open(20,file='tb00'//apx//'.dat')
20       format(a6,14x,2f14.2)
         ier = 0
         cnt = 0

         do while(ier.eq.0)
            read(20,20,iostat = ier,end = 200) finx,a411,a412
            cnt = cnt + 1
            if = searchc(finno,nfin,finx)
            if (if.gt.0) then
               end411(if,iap) = -a411
               end412(if,iap) = -a412
            end if
         end do
200      print*, 'tb00',apx,' read with ier = ',ier,' and cnt ',cnt

         if (iap.ge.2) then
            open(30,file='tb00'//ape//'.dat')
            ier = 0
            cnt = 0
```

```
            do while(ier.eq.0)
               read(30,20,iostat = ier,end = 300) finx,a411,a412
               cnt = cnt + 1
               if = searchc(finno,nfin,finx)
               if (if.gt.0) then
                  beg411(if,iap) = -a411
                  beg412(if,iap) = -a412
               end if
            end do
300         print*, 'tb00',ape,' read with ier = ',ier,' and cnt ',cnt
         end if

      end do

      open(40,file='nctb_fit_ye.00',recl=560)
40    format(a6,39f14.2)


      do if = 1,nfin
         write(40,40) finno(if),(end411(if,iap)-beg411(if,iap),end412(if,iap)-beg412(if,iap),
     &      rev(if,iap), iap = 1, nap)
      end do


      end
```

```
      PROGRAM fit_stdanp

C     DESCRIPTION:
C         Estimates SM Revenues for YTD Non-CBCIS sites
C         on AP basis using regression equation

      IMPLICIT NONE

C     Parameter Estimates:

      REAL*8 a1,b1,c1
      REAL*8 a2,b2,c2
      REAL*8 a3,b3,c3
      REAL*8 a4,b4,c4
      REAL*8 a,b,c
      PARAMETER (a1=-0.001207,b1=0.272776,c1=-0.000035)
      PARAMETER (a2= 0.001546,b2=0.165223,c2=-0.000096)
      PARAMETER (a3= 0.002500,b3=0.146355,c3=-0.000102)
      PARAMETER (a4= 0.001010,b4=0.170342,c4=-0.000052)


C     Constants

      INTEGER*4 nfin, iap, nstrata,nq,nap

      parameter (nfin=17269)
      parameter (nstrata=20)
      parameter (nq=4)
      parameter (nap = 13)

      CHARACTER*6 fin
      character*6 fins(nfin)
      CHARACTER*2 ap

      REAL*8    stamp(nap),meter(nap),piprsrt(nap),smhat,total,fit3np(nstrata,nq)
      REAL*8    min,max,tstamp,tmeter,tpiprsrt,pmtrev(nstrata)

      INTEGER*4 ier,i,j,searchc,out,in,nnon,ifin,lastcount,inx,iq
      integer*4 strata(nfin), strata1, fit(nfin,nap)

C     Get Arguments

C      CALL getarg(1,ap)
C      PRINT *, "Working on AP",ap

      OPEN(10,FILE='to_be_fit.41414')
10    FORMAT(a6,i3,13i2)

      do i = 1, nfin
         READ (10,10) fins(i), strata(i), (fit(i,iap),iap = 1,nap)
      END DO

      PRINT *, "NCTB finance numbers read"

      do i = 1, nstrata
         do iq = 1,nq
            fit3np(i,iq) = 0.0
            pmtrev(i) = 0.0
         end do
      end do
      print*,' Arrays initialized '

      ier = 0
      nnon=0
      in = 0

      OPEN(21,FILE='nctb_fit_ye.00',recl=2000)

21    FORMAT(A6,39F14.2)

      DO WHILE (ier.eq.0)
         READ(21,21,iostat=ier,end=100) fin,(stamp(iap),meter(iap),piprsrt(iap),iap=1,nap)
         in = in + 1
         ifin = searchc(fins,nfin,fin)
         do iap = 1,nap
            IF (ifin.gt.0) THEN ! fin found
               IF (fit(ifin,iap).eq.1) THEN ! need to fit revenue
                  nnon = nnon + 1
                  strata1 = strata(ifin)
```

```
                pmtrev(strata1) = pmtrev(strata1) + piprsrt(iap)
                IF ((iap.eq.1).or.(iap.eq.2).or.(iap.eq.3)) THEN
                   iq = 1
                   a=a1
                   b=b1
                   c=c1
                ELSE IF ((iap.eq.4).or.(iap.eq.5).or.(iap.eq.6)) THEN
                   iq = 2
                   a=a2
                   b=b2
                   c=c2
                ELSE IF ((iap.eq.7).or.(iap.eq.8).or.(iap.eq.9)) THEN
                   iq = 3
                   a=a3
                   b=b3
                   c=c3
                ELSE IF ((iap.eq.10).or.(iap.eq.11).or.(iap.eq.12).or.(iap.eq.13)) THEN
                   iq = 4
                   a=a4
                   b=b4
                   c=c4
                END IF

                smhat = a*meter(iap) + b*piprsrt(iap) + c*meter(iap)*meter(iap)/1000000

                if ((meter(iap).gt.0).or.(piprsrt(iap).gt.0)) then

                   if ((meter(iap).le.0).or.(piprsrt(iap).le.0)) then
C                      print*,fin,' ',meter(iap),' ',piprsrt(iap)
                   end if
                end if

                IF (smhat.lt.0) THEN
                   smhat=0
                END IF
                fit3np(strata1,iq) = fit3np(strata1,iq) + smhat
             end if
          else
             print*, 'Fin not found ', fin
             strata1 = 20
             nnon = nnon + 1
             pmtrev(strata1) = pmtrev(strata1) + piprsrt(iap)
             IF ((iap.eq.1).or.(iap.eq.2).or.(iap.eq.3)) THEN
                iq = 1
                a=a1
                b=b1
                c=c1
             ELSE IF ((iap.eq.4).or.(iap.eq.5).or.(iap.eq.6)) THEN
                iq = 2
                a=a2
                b=b2
                c=c2
             ELSE IF ((iap.eq.7).or.(iap.eq.8).or.(iap.eq.9)) THEN
                iq = 3
                a=a3
                b=b3
                c=c3
             ELSE IF ((iap.eq.10).or.(iap.eq.11).or.(iap.eq.12).or.(iap.eq.13)) THEN
                iq = 4
                a=a4
                b=b4
                c=c4
             END IF
             smhat = a*meter(iap) + b*piprsrt(iap) + c*meter(iap)*meter(iap)/1000000
             IF (smhat.lt.0) THEN
                smhat=0
             END IF
             fit3np(strata1,iq) = fit3np(strata1,iq) + smhat
          end if
       end do
    END DO

100 print*, "ier = ", ier
    PRINT *, "Read ",in-1," records"
    PRINT *, nnon," records need to be fitted."

    open(40,file='fit_stda_np.00')
41  format(i3,4f14.2)

    do i = 1, nstrata
```

```
   write(40,41) i, (fit3np(i,iq),iq=1,nq)
   print*, "permit imprint rev for strata ", i , " is ", pmtrev(i)
end do


end
```

```
      program eststdanp_20

C     Paul Loetscher 1-06-96
C
C
C
C
C     Estimate third-class volumes by weight increment and class
C     using PERMIT and BRAVIS data

C     Input files:
C     permit rollup matrices: ../pmt/pmt3rd.wi.*
C     bravis rollup matrices: ../brv/brv3rd.wi.*
C     trial balance revenues: /u/mcb/vol96/nctb/strata.41411
C     lists of permit and bravis finance numbers
C
C     Output files:
C     est3rd.csv  -  to be imported into excel
C     est3rd.factors - control factors


      implicit none

C     Parameters

      integer*4  nvip, nshp, nwt, nstr, ntype, nrpw, nind, ncls
      integer*4  npmt, nq, nap

      parameter  (nvip = 669)   ! number of 3rd class regular rate vips
      parameter  (nshp = 6)     ! number of shapes
      parameter  (nwt  = 20)    ! number of weight increments
      parameter  (nstr = 4)     ! number of revenue strata
      parameter  (ntype = 3)    ! number of transaction types
      parameter  (nrpw = 3)     ! r, p, w are held as a dim in array
      parameter  (nind = 2)     ! number of indicia (for ms trans counts)
      parameter  (ncls = 85)    ! number of subclasses
      parameter  (npmt = 2257)  ! number of permit finance numbers
      parameter  (nq = 4)
      parameter  (nap = 13)

C     Real Storage

      real*8   wipr(ncls,nind,nshp,nwt,ntype,nstr,nq) ! r, p, w  - permit
      real*8   wipp(ncls,nind,nshp,nwt,ntype,nstr,nq) ! r, p, w  - permit
      real*8   wipw(ncls,nind,nshp,nwt,ntype,nstr,nq) ! r, p, w  - permit

      real*8   result(nrpw,nwt,nshp,ncls) ! array to write out

      real*8   shrwt(nwt), pishrwt(nwt,ncls,nshp) ! arrays used for redist
      real*8   badwght(nrpw,nshp,ncls)
      real*8   sm_fitted(nstr,nq)

      real*8   pmtrev(nstr,nq)  ! permit revenue by strata and indicia
      real*8   smrev(nstr,nq)   ! permit revenue by strata and indicia
      real*8   nctbpmt(nstr,nq) ! trial balance permit office rev.
      real*8   nctball(nstr,nq) ! trial balance total revenue by strata
      real*8   factpmt(nstr,nq) ! control factors for permit office revenue
      real*8   factall(nstr,nq) ! control factors for total revenue (nctb)
      real*8   factall_sm(nstr,nq) ! control factors for total revenue (nctb)
      real*8   rpwfact(nq)      ! control factor to total rpw revenue
      real*8   revin(nap)
      real*8   r, p, w, wpp,x,xc
      real*8   pieces_by_shp(nshp),rvtst(nq)
      real*8   q1,q2,q3,q4
      real*8   rev_str(20,2)

C     Integer Storage

      integer*4 ir, ic, ii, ip, iw, it, is, iq, if, ia, iv,id
      integer*4 ier, i
      integer*4 searchc, qind, iind
      integer*4 strind          ! function to assign condensed strata
      integer*4 clsmap(nvip)/nvip*0/ ! map from vip to class index
      integer*4 pflags(nap,npmt) ! flag for finance number in permit for ap
      integer*4 minrate(ncls)/ncls*0./ ! flag for minimum rate pieces
      integer*4 shpflag(ncls)/ncls*0./ ! flag for allowable shapes

C     Character Storage

      character*6   pfins(npmt), fin
```

```
      character*2   ap
      character*5   vips(nvip), vip
      real*8        fuzz(20)
C     Logical Storage

C     Intialize arrays

C     RPW revenue = total regular rate revenue from penalty and franked
C     as classes report


      do ip = 1,nshp
         pieces_by_shp(ip) = 0
      end do
      do iq = 1, nq
         do is = 1, nstr
            sm_fitted(is,iq) = 0
            do it = 1, ntype
               do iw = 1, nwt
                  do ip = 1, nshp
                     do ii = 1, nind
                        do ic = 1, ncls
                           wipr(ic,ii,ip,iw,it,is,iq) = 0.
                           wipp(ic,ii,ip,iw,it,is,iq) = 0.
                           wipw(ic,ii,ip,iw,it,is,iq) = 0.
                        end do
                     end do
                  end do
               end do
            end do
         end do
      end do
      do ic = 1, ncls
         do ip = 1, nshp
            do iw = 1, nwt
               do ir = 1, nrpw
                  result(ir,iw,ip,ic) = 0.
                  pishrwt(iw,ic,ip) = 0.
               end do
            end do
         end do
      end do
      do ic = 1, ncls
         do ip = 1, nshp
            do ir = 1, nrpw
               badwght(ir,ip,ic) = 0.
            end do
         end do
      end do
      do iq = 1, nq
         rvtst(iq) = 0
         do is = 1, nstr
            pmtrev(is,iq)     = 0.
            smrev(is,iq)      = 0.
            nctbpmt(is,iq)    = 0.
            nctball(is,iq)    = 0.
            factpmt(is,iq)    = 0.
            factall(is,iq)    = 0.
            factall_sm(is,iq)    = 0.
            sm_fitted(is,iq) = 0
         end do
      end do

      do is = 1,20
         rev_str(is,1) = 0
         rev_str(is,2) = 0
         fuzz(is) = 0
      end do

      do if = 1,npmt
         do ia = 1,nap
            pflags(ia,if) = 0
         end do
      end do
      do ic = 1, ncls
         do ip = 1, nshp
            do iw = 1, nwt
               pishrwt(iw,ic,ip) = 0.
               shrwt(iw) = 0.
            end do
```

```
      end do
    end do

    open(90,file='factors_20.txt')
90  format(i3,i3,f14.2,f14.2,f10.6)

C   Read finance numbers and reported ap's

    open(15,file='finsbyap.pmt.00')
15  format(a6,13i2)
    do if = 1, npmt
      read (15,15) pfins(if), (pflags(ia,if),ia=1,nap)
    end do
    close (15)
    print *,' PERMIT finance numbers read '


C   Read Trial balance revenue and assign to Permit, Bravis and
C   Total revenue arrays

    open(18,file='strata.41414')
18  format(a6,i3,f15.2,13f12.2)

    if = 0
    ier = 0
    do iq = 1,nq
      rvtst(iq) = 0
    end do
    do while (ier.eq.0)
      read (18,18,iostat=ier,end=180) fin, is, r, revin
      if = if + 1
      id = is
      do ia = 1,nap
        rev_str(id,1) = rev_str(id,1) + revin(ia)
        fuzz(id) = fuzz(id) + revin(ia)
      end do
      ip = 0
      is = strind(is)
      ip = searchc(pfins,npmt,fin)
      do ia = 1, nap
        iq = qind(ia)
        nctball(is,iq) = nctball(is,iq) + revin(ia)
        if (ip.gt.0) then
          rev_str(id,2) = rev_str(id,2) + revin(ia)
          if(pflags(ia,ip).eq.0) then
            if (is.eq.2) then
              print*, 'no ap ', pfins(ip),' ia ', ia ,' ', pflags(ia,ip)
              rvtst(iq) = rvtst(iq) + revin(ia)
            end if
          else
            nctbpmt(is,iq) =  nctbpmt(is,iq) + revin(ia)
          end if
        end if
      end do
    end do

180 print *,' Strata.41411 read with ',if,' records, ier = ',ier


    open(19,file='fit_stda_np.00')
19  format(i3,4f14.2)

    do i = 1, 20
      read(19,19) is,q1,q2,q3,q4
      rev_str(is,1) = rev_str(is,1) + q1 + q2 + q3 + q4
      rev_str(is,2) = rev_str(is,2) + q1 + q2 + q3 +q4

      is = strind(is)
      sm_fitted(is,1) = sm_fitted(is,1) + q1
      sm_fitted(is,2) = sm_fitted(is,2) + q2
      sm_fitted(is,3) = sm_fitted(is,3) + q3
      sm_fitted(is,4) = sm_fitted(is,4) + q4

    end do

    print*, 'Stamped/Metered fitted values read'
```

```
C     Build class map, shape and min rate flags from vip3rd.94


      open(20,file='vipnp99.new')
20    format(a5,73x,i2)

      do iv = 1, nvip
         read(20,20) vip, ic
         vips(iv) = vip
         clsmap(iv) = ic
         if (ic.gt.0) then
            if (ic.lt.62) minrate(ic) = 1
         end if
      end do

      print*,'vips read '


C     Read Permit Transactions

31    format(i2,1x,i1,1x,i3,1x,i1,1x,i3,f14.4,f12.0,f14.4)
32    format(i2,i1,i3,i1,i3,f14.4,2f12.0)

      do ia = 10,13
         iq = qind(ia)
         write (ap,'(i2.2)') ia
         open(30,file='pmt_stda.wi.'//ap)
         i = 0
         ier = 0
         do while (ier.eq.0)
            read (30,31,iostat=ier,end=300) is, it, iw, ip, iv, r, p, w
C            write(*,31) is, it, iw, ip, iv, r, p, w
            id = is
            if (iv.ne.670) then
               if (ip.gt.3) ip = 3
               pieces_by_shp(ip) = pieces_by_shp(ip) + p
               if(iv.gt.0) then
                  ic = clsmap(iv)
C                 print*,'ic is ',ic , ' vip is ',vips(iv), ' iv is ',iv
                  is = strind(is)
                  ii = iind(vips(iv))

                  if (ic.ge.62.and.iw.lt.7) then
                     if ((it.ne.3).and.(ic.ne.0)) then
                        print*,'Bad weight in PERMIT is too small: ',is,
     +                     ' it: ',it,' iw: ',iw,' ip: ',ip,' iv: ',vips(iv),' ic: ',ic,' rpw: ',r,p,w
                     end if
                     it = 3
                  else if (ic.lt.62.and.iw.gt.7) then
                     if ((it.ne.3).and.(ic.ne.0)) then
                        print*,'Bad weight in PERMIT is too large: ',is,
     +                     ' it: ',it,' iw: ',iw,' ip: ',ip,' iv: ',vips(iv),' ic: ',ic,' rpw: ',r,p,w
                     end if
                     it = 3
                  end if

                  if (ic.gt.0) then
                     if (ip.gt.3) ip = 3
                     if (ii.ge.2) then
                        rev_str(id,1) = rev_str(id,1) + r
                        rev_str(id,2) = rev_str(id,2) + r
                     end if
                     wipr(ic,ii,ip,iw,it,is,iq) = wipr(ic,ii,ip,iw,it,is,iq) + r
                     wipp(ic,ii,ip,iw,it,is,iq) = wipp(ic,ii,ip,iw,it,is,iq) + p
                     wipw(ic,ii,ip,iw,it,is,iq) = wipw(ic,ii,ip,iw,it,is,iq) + w
                     if (ii.eq.1) pmtrev(is,iq) = pmtrev(is,iq) + r
                  else
                     if (vips(iv)(2:4).ne.'340') then
                        print*,'No Class ',vips(iv)
                     end if
                  end if

               else
                  print*,'Bad vip remapping ',(iv)
               end if
            end if
         end do
300      print *,' read exit of pmt_stda.wi.',ap,' = ',ier
         close(30)
```

```
      end do


C     Calculate Trial Balance Permit and Bravis Own Factors
      write(90,'(a40)') 'Permit/Bravis Own Factors'
      write(90,'(a40)') '   '
      do iq = 1, nq
         do is = 1, nstr
            if (pmtrev(is,iq).gt.0.) then
               factpmt(is,iq) = nctbpmt(is,iq) / pmtrev(is,iq)
            else
               print *, ' ?!  No PERMIT revenue in strata ',is,' quarter ',iq
            end if
            write(90,90) is,iq,nctbpmt(is,iq),pmtrev(is,iq),factpmt(is,iq)
            pmtrev(is,iq) = 0.
         end do
      end do

400   format(i3,4f8.4,4f14.0)
410   format(i3,8f14.0)
      print *,'  '
      print *,' PERMIT Own Factors and NCTB Revenue: '
      do is = 1, nstr
         write (*,400) is, (factpmt(is,iq),iq=1,nq), (nctbpmt(is,iq), iq=1,nq)
      end do

C     Apply Own Factors and Check

      do iq = 1, nq
         do is = 1, nstr
            do it = 1, ntype
               do iw = 1, nwt
                  do ip = 1, nshp
                     do ii = 1, nind
                        do ic = 1, ncls
                           wipr(ic,ii,ip,iw,it,is,iq) =
     &                        wipr(ic,ii,ip,iw,it,is,iq) * factpmt(is,iq)
                           wipp(ic,ii,ip,iw,it,is,iq) =
     &                        wipp(ic,ii,ip,iw,it,is,iq) * factpmt(is,iq)
                           wipw(ic,ii,ip,iw,it,is,iq) =
     &                        wipw(ic,ii,ip,iw,it,is,iq) * factpmt(is,iq)
                           if (ii.eq.1) then
                              pmtrev(is,iq) = pmtrev(is,iq) +
     &                           wipr(ic,ii,ip,iw,it,is,iq)
                           else
C                             print*,'is ',is
c                             print*,'iq ',iq
c                             print*,'ic ',ic
c                             print*,'ii ',ii
c                             print*,'ip ',ip
c                             print*,'iw ',iw
c                             print*,'it ',it

                              smrev(is,iq) = smrev(is,iq) + wipr(ic,ii,ip,iw,it,is,iq)
                           end if
                        end do
                     end do
                  end do
               end do
            end do
         end do
      end do

      print *,'  '
      print *,' PERMIT Permit Imprint Controlled Revenue and NCTB Revenue: '
      do is = 1, nstr
         write (*,410) is, (pmtrev(is,iq),iq=1,nq), (nctbpmt(is,iq), iq=1,nq)
      end do

C     Calculate Trial Balance Total Factors
      write(90,'(a40)') 'Trial Balance Total Factors'
      write(90,'(a40)') '   '
      do iq = 1, nq
         do is = 1, nstr
            if (pmtrev(is,iq).gt.0.) then
               factall(is,iq) = nctball(is,iq) /
     &            (pmtrev(is,iq))
               write(90,90) is,iq,nctball(is,iq),pmtrev(is,iq),factall(is,iq)
               factall_sm(is,iq) = (sm_fitted(is,iq)+smrev(is,iq)) /
```

```
     &           (smrev(is,iq))
                 write(90,90) is,iq,(sm_fitted(is,iq) + smrev(is,iq)),smrev(is,iq),factall_sm(is,iq)
              else
                 print *, ' ?! No PERMIT or BRAVIS revenue strata ',is,' q ',iq
              end if
              pmtrev(is,iq) = 0.
              smrev(is,iq) = 0.
           end do
        end do

        print *,'  '
        print *,' All Office Factors and NCTB Revenue: Permit Imprint'
        do is = 1, nstr
           write (*,400) is, (factall(is,iq),iq=1,nq), (nctball(is,iq), iq=1,nq)
        end do
        print *,'  '
        print *,' All Office Factors and NCTB Revenue: Stamped/Metered'
        do is = 1, nstr
           write (*,400) is, (factall_sm(is,iq),iq=1,nq), (sm_fitted(is,iq), iq=1,nq)
        end do


C       Apply Total Factor and Check

        do iq = 1, nq
           do is = 1, nstr
              do it = 1, ntype
                 do iw = 1, nwt
                    do ip = 1, nshp
                       do ii = 1, nind
                          do ic = 1, ncls
                             if (ii.eq.1) then
                                wipr(ic,ii,ip,iw,it,is,iq) =
     &                             wipr(ic,ii,ip,iw,it,is,iq) * factall(is,iq)
                                wipp(ic,ii,ip,iw,it,is,iq) =
     &                             wipp(ic,ii,ip,iw,it,is,iq) * factall(is,iq)
                                wipw(ic,ii,ip,iw,it,is,iq) =
     &                             wipw(ic,ii,ip,iw,it,is,iq) * factall(is,iq)
                             else
                                wipr(ic,ii,ip,iw,it,is,iq) =
     &                             wipr(ic,ii,ip,iw,it,is,iq) * factall_sm(is,iq)
                                wipp(ic,ii,ip,iw,it,is,iq) =
     &                             wipp(ic,ii,ip,iw,it,is,iq) * factall_sm(is,iq)
                                wipw(ic,ii,ip,iw,it,is,iq) =
     &                             wipw(ic,ii,ip,iw,it,is,iq) * factall_sm(is,iq)
                             end if
                             if (ii.eq.1) then
                                pmtrev(is,iq) = pmtrev(is,iq) +
     &                             wipr(ic,ii,ip,iw,it,is,iq)
                             else
                                smrev(is,iq) = smrev(is,iq) +
     &                             wipr(ic,ii,ip,iw,it,is,iq)

                             end if
                          end do

                       end do
                    end do
                 end do
              end do
           end do
        end do

        print *,'  '
        print *,' PERMIT/BRAVIS Controlled Revenue and NCTB Revenue: Permit Imprint'
        do is = 1, nstr
           write (*,410) is, (pmtrev(is,iq),iq=1,nq), (nctball(is,iq), iq=1,nq)
        end do

        print *,'  '
        print *,' PERMIT/BRAVIS Controlled Revenue and NCTB Revenue: Stamped'
        do is = 1, nstr
           write (*,410) is, (smrev(is,iq),iq=1,nq), (sm_fitted(is,iq), iq=1,nq)
        end do

C       Distribute Bad Weight Data to Identical Weight

        open(50,file='stdanp.csv',recl=500)
51      format(i2,',',i1,',',20(f16.3,','),f16.3,',',f16.3,',',f16.3)
```

```
C     Sum identical and non-identical into result array,
C      bad weight into badwght array

      x=0
      xc=0
      do iw = 1, nwt
         do ic = 1, ncls
            do ip = 1, nshp
               pishrwt(iw,ic,ip) = 0.
               shrwt(iw) = 0.
            end do
         end do
      end do

      do ii = 1, nind
         x=0
         xc=0
         do iq = 1, nq
            do is = 1, nstr
               do it = 1, 2     ! skip bad weight
                  do iw = 1, nwt
                     do ip = 1, nshp
                        do ic = 1, ncls
                           result(1,iw,ip,ic) =
     &                        result(1,iw,ip,ic) +
     &                        wipr(ic,ii,ip,iw,it,is,iq)
                           result(2,iw,ip,ic) =
     &                        result(2,iw,ip,ic) +
     &                        wipp(ic,ii,ip,iw,it,is,iq)
                           result(3,iw,ip,ic) =
     &                        result(3,iw,ip,ic) +
     &                        wipw(ic,ii,ip,iw,it,is,iq)

                           xc = xc + wipr(ic,ii,ip,iw,it,is,iq)
                        end do
                     end do
                  end do
               end do
               do iw = 1, nwt
                  do ip = 1, nshp
                     do ic = 1, ncls
                        badwght(1,ip,ic) = badwght(1,ip,ic) +
     &                     wipr(ic,ii,ip,iw,3,is,iq)
                        badwght(2,ip,ic) = badwght(2,ip,ic) +
     &                     wipp(ic,ii,ip,iw,3,is,iq)
                        badwght(3,ip,ic) = badwght(3,ip,ic) +
     &                     wipw(ic,ii,ip,iw,3,is,iq)
                        xc = xc + wipr(ic,ii,ip,iw,3,is,iq)
                        x= x + wipr(ic,ii,ip,iw,3,is,iq)
                     end do
                  end do
               end do
            end do
         end do
         print*,'sum of bad weight revenues  ', x
         print*,'sum of total revenues ',xc

         x=0

C     Now distribute bad weight pieces

         do ic = 1, ncls
            do ip = 1, nshp
               do iw = 1, nwt
                  shrwt(iw) = 0.
               end do
               p = 0.
               do iw = 1, nwt
                  p = p + result(2,iw,ip,ic)
                  shrwt(iw) = shrwt(iw) + result(2,iw,ip,ic)
               end do
               do iw = 1, nwt
                  if(p.gt.0) then
                     shrwt(iw) = shrwt(iw)/p
                  end if
                  if (ii.eq.1) then
                     pishrwt(iw,ic,ip) = shrwt(iw)
                  end if
                  if (badwght(2,ip,ic).gt.0) then
C                  write(*,'(3i3,f6.3)') ic,ip,iw,shrwt(iw)
```

```
                  end if
               end do
               if (badwght(2,ip,ic).gt.0) then
                  if (minrate(ic).lt.99) then
                     if (p.gt.0.) then
                        do iw = 1, nwt
                           wpp = 0
                           if (result(2,iw,ip,ic).gt.0) then
                              wpp = result(3,iw,ip,ic) / result(2,iw,ip,ic)
                           else
                              wpp = 0.
                           end if
                           do ir = 1, 2
                              result(ir,iw,ip,ic) =
     &                           result(ir,iw,ip,ic) +
     &                           badwght(ir,ip,ic)*shrwt(iw)
                           end do
                           x=x+badwght(1,ip,ic)*shrwt(iw)
                           result(3,iw,ip,ic) =
     &                        badwght(2,ip,ic)*(shrwt(iw)) * wpp +
     &                        result(3,iw,ip,ic)
                        end do
                     else
                        do iw = 1, nwt
                           print *, "share at wgt inc ", iw, " is ", pishrwt(iw,ic,ip), " with subclass ", ic
                           if (result(2,iw,ip,ic).gt.0) then
                              wpp = result(3,iw,ip,ic) / result(2,iw,ip,ic)
                           else
                              wpp = 0.
                           end if
                           do ir = 1, 2
                              result(ir,iw,ip,ic) =
     &                           result(ir,iw,ip,ic) +
     &                           badwght(ir,ip,ic)*pishrwt(iw,ic,ip)
                           end do
                           x=x+badwght(1,ip,ic)*pishrwt(iw,ic,ip)
                           result(3,iw,ip,ic) =
     &                        badwght(2,ip,ic)*(pishrwt(iw,ic,ip)) * wpp +
     &                        result(3,iw,ip,ic)
                        end do

c                       print *,' No i/ni weight to dist bad ',ic,',',ip,',',ii
c                       print *,' r = ',badwght(1,ip,ic),' p = ',badwght(2,ip,ic)
                     end if
                  else
                     print *,' Bad wght to dist non-minrate ',ic,',',ip,',',ii,' ',badwght(2,ip,ic)
                  end if
               end if
            end do
         end do
         print*,'sum of badweight revenues after distribution ',x

C     Write Out Data for Importation into Excel


         x=0
         do ic = 1, ncls
            do ip = 1,3
               r = 0.
               p = 0.
               w = 0.
               do iw = 1, nwt
                  r = r + result(1,iw,ip,ic)
                  x = x + result(1,iw,ip,ic)
                  p = p + result(2,iw,ip,ic)
                  w = w + result(3,iw,ip,ic)
               end do
               write (50,51) ic, ip, (result(2,iw,ip,ic),iw=1,nwt),
     +              p,r,w
            end do
         end do
         write(50,'(a2)') 'SM'
         do ir = 1,nrpw
            do iw = 1,nwt
               do ip = 1,nshp
                  do ic = 1,ncls
                     result(ir,iw,ip,ic) = 0
                     badwght(ir,ip,ic) = 0
                  end do
               end do
```

```
            end do
         end do
         print*, 'after writing revenue ',x
      end do


C     Write Out Control Factors To File

      open(60,file='eststda_20.control')
61    format(4f20.15)
      do is = 1, nstr
         write (60,61) (factpmt(is,iq),iq=1,nq)
      end do
      do is = 1, nstr
         write (60,61) (factall(is,iq),iq=1,nq)
      end do

      close(60)

      open(60,file='strrev.txt')

      do is = 1,20
         write(60,'(i3,4f15.2)') is,rev_str(is,1),(rev_str(is,1)-rev_str(is,2)),fuzz(is),rev_str(is,1)-fuzz(is)
      end do

      do ip = 1,nshp
         write(*,'(i4,f26.0)') ip,pieces_by_shp(ip)
      end do

      end

C ---------------------------------------------------------------------------
      function qind(iap)

      integer*4 qind, iap

      if (iap.ge.1.and.iap.le.3) then
         qind = 1
      else if (iap.ge.4.and.iap.le.6) then
         qind = 2
      else if (iap.ge.7.and.iap.le.9) then
         qind = 3
      else if (iap.ge.10.and.iap.le.13) then
         qind = 4
      else
         stop  ' invalid accounting period '
      end if

      return
      end

C ---------------------------------------------------------------------------
      function strind(is)

      integer*4 strind, is

      if (is.ge.1.and.is.le.16) then
         strind = 1
      else if (is.ge.17.and.is.le.18) then
         strind = 2
      else if (is.ge.19.and.is.le.19) then
         strind = 3
      else if (is.ge.20.and.is.le.20) then
         strind = 4

      else
         stop ' bad strata '
      end if

      return
      end
C ---------------------------------------------------------------------------
      function shapechk(ip,flag)

      integer*4 shapechk, ip, flag

      if (flag.eq.1) then
         shapechk = 1       ! letter only vip
      else if (flag.eq.2) then
         if (ip.eq.1) then
```

```
            shapechk = 2    ! non-letters vip - move letter to flat
         else
            shapechk = ip
         end if
      else if (flag.eq.3) then  ! all shape
         if(ip.gt.3) ip = 3
         shapechk = ip
      else if (flag.eq.4) then  ! flat only vip
         shapechk = 2
      else
         print*, 'Flag ',flag,'  ip,',ip
         stop ' bad shape flag '
      end if

      return
      end

C ---------------------------------------------------------------------------
      function iind(vip)

      integer*4 iind
      character*5 vip

      if (vip(1:1).eq.'0') then
         iind = 1
      else if (vip(1:1).eq.'3') then
         if (vip(5:5).eq.'1') then
            iind = 2
         else if (vip(5:5).eq.'2') then
            iind = 2
         else if (vip(5:5).eq.'3') then
            iind = 2
         else if (vip(5:5).eq.'4') then
            iind = 2
         else if (vip(5:5).eq.'6') then
            iind = 1
         else if (vip(5:5).eq.'7') then
            iind = 1
         else if (vip(5:5).eq.'9') then
            iind = 1
         else
            iind = -1
         end if
      else
         iind = 2
      end if

      return
      end
C ---------------------------------------------------------------------------
```

```
      program weststdanp_20

C     Paul Loetscher 1-06-96
C
C
C
C
C     Estimate third-class volumes by weight increment and class
C     using PERMIT and BRAVIS data

C     Input files:
C     permit rollup matrices: ../pmt/pmt3rd.wi.*
C     bravis rollup matrices: ../brv/brv3rd.wi.*
C     trial balance revenues: /u/mcb/vol96/nctb/strata.41411
C     lists of permit and bravis finance numbers
C
C     Output files:
C     est3rd.csv  -  to be imported into excel
C     est3rd.factors - control factors


      implicit none

C     Parameters

      integer*4  nvip, nshp, nwt, nstr, ntype, nrpw, nind, ncls
      integer*4  npmt, nq, nap

      parameter  (nvip = 669)   ! number of 3rd class regular rate vips
      parameter  (nshp = 6)     ! number of shapes
      parameter  (nwt  = 20)    ! number of weight increments
      parameter  (nstr = 4)     ! number of revenue strata
      parameter  (ntype = 3)    ! number of transaction types
      parameter  (nrpw = 3)     ! r, p, w are held as a dim in array
      parameter  (nind = 2)     ! number of indicia (for ms trans counts)
      parameter  (ncls = 85)    ! number of subclasses
      parameter  (npmt = 2257)  ! number of permit finance numbers
      parameter  (nq = 4)
      parameter  (nap = 13)

C     Real Storage

      real*8   wipr(ncls,nind,nshp,nwt,ntype,nstr,nq) ! r, p, w  - permit
      real*8   wipp(ncls,nind,nshp,nwt,ntype,nstr,nq) ! r, p, w  - permit
      real*8   wipw(ncls,nind,nshp,nwt,ntype,nstr,nq) ! r, p, w  - permit

      real*8   result(nrpw,nwt,nshp,ncls) ! array to write out

      real*8   shrwt(nwt), pishrwt(nwt,ncls,nshp) ! arrays used for redist
      real*8   badwght(nrpw,nshp,ncls)
      real*8   sm_fitted(nstr,nq)

      real*8   pmtrev(nstr,nq)  ! permit revenue by strata and indicia
      real*8   smrev(nstr,nq)   ! permit revenue by strata and indicia
      real*8   nctbpmt(nstr,nq) ! trial balance permit office rev.
      real*8   nctball(nstr,nq) ! trial balance total revenue by strata
      real*8   factpmt(nstr,nq) ! control factors for permit office revenue
      real*8   factall(nstr,nq) ! control factors for total revenue (nctb)
      real*8   factall_sm(nstr,nq) ! control factors for total revenue (nctb)
      real*8   rpwfact(nq)      ! control factor to total rpw revenue
      real*8   revin(nap)
      real*8   r, p, w, wpp,x,xc
      real*8   pieces_by_shp(nshp),rvtst(nq)
      real*8   q1,q2,q3,q4
      real*8   rev_str(20,2)

C     Integer Storage

      integer*4 ir, ic, ii, ip, iw, it, is, iq, if, ia, iv,id
      integer*4 ier, i
      integer*4 searchc, qind, iind
      integer*4 strind          ! function to assign condensed strata
      integer*4 clsmap(nvip)/nvip*0/ ! map from vip to class index
      integer*4 pflags(nap,npmt) ! flag for finance number in permit for ap
      integer*4 minrate(ncls)/ncls*0./ ! flag for minimum rate pieces
      integer*4 shpflag(ncls)/ncls*0./ ! flag for allowable shapes

C     Character Storage

      character*6   pfins(npmt), fin
```

```
      character*2   ap
      character*5   vips(nvip), vip
      real*8        fuzz(20)
C     Logical Storage

C     Intialize arrays

C     RPW revenue = total regular rate revenue from penalty and franked
C     as classes report


      do ip = 1,nshp
         pieces_by_shp(ip) = 0
      end do
      do iq = 1, nq
         do is = 1, nstr
            sm_fitted(is,iq) = 0
            do it = 1, ntype
               do iw = 1, nwt
                  do ip = 1, nshp
                     do ii = 1, nind
                        do ic = 1, ncls
                           wipr(ic,ii,ip,iw,it,is,iq) = 0.
                           wipp(ic,ii,ip,iw,it,is,iq) = 0.
                           wipw(ic,ii,ip,iw,it,is,iq) = 0.
                        end do
                     end do
                  end do
               end do
            end do
         end do
      end do
      do ic = 1, ncls
         do ip = 1, nshp
            do iw = 1, nwt
               do ir = 1, nrpw
                  result(ir,iw,ip,ic) = 0.
                  pishrwt(iw,ic,ip) = 0.
               end do
            end do
         end do
      end do
      do ic = 1, ncls
         do ip = 1, nshp
            do ir = 1, nrpw
               badwght(ir,ip,ic) = 0.
            end do
         end do
      end do
      do iq = 1, nq
         rvtst(iq) = 0
         do is = 1, nstr
            pmtrev(is,iq)     = 0.
            smrev(is,iq)      = 0.
            nctbpmt(is,iq)    = 0.
            nctball(is,iq)    = 0.
            factpmt(is,iq)    = 0.
            factall(is,iq)    = 0.
            factall_sm(is,iq)    = 0.
            sm_fitted(is,iq) = 0
         end do
      end do

      do is = 1,20
         rev_str(is,1) = 0
         rev_str(is,2) = 0
         fuzz(is) = 0
      end do

      do if = 1,npmt
         do ia = 1,nap
            pflags(ia,if) = 0
         end do
      end do
      do ic = 1, ncls
         do ip = 1, nshp
            do iw = 1, nwt
               pishrwt(iw,ic,ip) = 0.
               shrwt(iw) = 0.
            end do
```

```
         end do
      end do

      open(90,file='factors_20.txt')
90    format(i3,i3,f14.2,f14.2,f10.6)

C     Read finance numbers and reported ap's

      open(15,file='finsbyap.pmt.00')
15    format(a6,13i2)
      do if = 1, npmt
         read (15,15) pfins(if), (pflags(ia,if),ia=1,nap)
      end do
      close (15)
      print *,' PERMIT finance numbers read '



C     Read Trial balance revenue and assign to Permit, Bravis and
C     Total revenue arrays

      open(18,file='strata.41414')
18    format(a6,i3,f15.2,13f12.2)

      if = 0
      ier = 0
      do iq = 1,nq
         rvtst(iq) = 0
      end do
      do while (ier.eq.0)
         read (18,18,iostat=ier,end=180) fin, is, r, revin
         if = if + 1
         id = is
         do ia = 1,nap
            rev_str(id,1) = rev_str(id,1) + revin(ia)
            fuzz(id) = fuzz(id) + revin(ia)
         end do
         ip = 0
         is = strind(is)
         ip = searchc(pfins,npmt,fin)
         do ia = 1, nap
            iq = qind(ia)
            nctball(is,iq) = nctball(is,iq) + revin(ia)
            if (ip.gt.0) then
               rev_str(id,2) = rev_str(id,2) + revin(ia)
               if(pflags(ia,ip).eq.0) then
                  if (is.eq.2) then
                     print*, 'no ap ', pfins(ip),' ia ', ia ,' ', pflags(ia,ip)
                     rvtst(iq) = rvtst(iq) + revin(ia)
                  end if
               else
                  nctbpmt(is,iq) =  nctbpmt(is,iq) + revin(ia)
               end if
            end if
         end do
      end do

180   print *,' Strata.41411 read with ',if,' records, ier = ',ier



      open(19,file='fit_stda_np.00')
19    format(i3,4f14.2)

      do i = 1, 20
         read(19,19) is,q1,q2,q3,q4
         rev_str(is,1) = rev_str(is,1) + q1 + q2 + q3 + q4
         rev_str(is,2) = rev_str(is,2) + q1 + q2 + q3 +q4

         is = strind(is)
         sm_fitted(is,1) = sm_fitted(is,1) + q1
         sm_fitted(is,2) = sm_fitted(is,2) + q2
         sm_fitted(is,3) = sm_fitted(is,3) + q3
         sm_fitted(is,4) = sm_fitted(is,4) + q4

      end do

      print*, 'Stamped/Metered fitted values read'
```

```
C     Build class map, shape and min rate flags from vip3rd.94


      open(20,file='vipnp99.new')
20    format(a5,73x,i2)

      do iv = 1, nvip
         read(20,20) vip, ic
         vips(iv) = vip
         clsmap(iv) = ic
         if (ic.gt.0) then
            if (ic.lt.62) minrate(ic) = 1
         end if
      end do

      print*,'vips read '


C     Read Permit Transactions

31    format(i2,1x,i1,1x,i3,1x,i1,1x,i3,f14.4,f12.0,f14.4)
32    format(i2,i1,i3,i1,i3,f14.4,2f12.0)

      do ia = 10,13
         iq = qind(ia)
         write (ap,'(i2.2)') ia
         open(30,file='pmt_stda.wi.'//ap)
         i = 0
         ier = 0
         do while (ier.eq.0)
            read (30,31,iostat=ier,end=300) is, it, iw, ip, iv, r, p, w
C            write(*,31) is, it, iw, ip, iv, r, p, w
            id = is
            if (iv.ne.670) then
               if (ip.gt.3) ip = 3
               pieces_by_shp(ip) = pieces_by_shp(ip) + p
               if(iv.gt.0) then
                  ic = clsmap(iv)
C                 print*,'ic is ',ic , ' vip is ',vips(iv), ' iv is ',iv
                  is = strind(is)
                  ii = iind(vips(iv))

                  if (ic.ge.62.and.iw.lt.7) then
                     if ((it.ne.3).and.(ic.ne.0)) then
                        print*,'Bad weight in PERMIT is too small: ',is,
     +                     ' it: ',it,' iw: ',iw,' ip: ',ip,' iv: ',vips(iv),' ic: ',ic,' rpw: ',r,p,w
                     end if
                     it = 3
                  else if (ic.lt.62.and.iw.gt.7) then
                     if ((it.ne.3).and.(ic.ne.0)) then
                        print*,'Bad weight in PERMIT is too large: ',is,
     +                     ' it: ',it,' iw: ',iw,' ip: ',ip,' iv: ',vips(iv),' ic: ',ic,' rpw: ',r,p,w
                     end if
                     it = 3
                  end if

                  if (ic.gt.0) then
                     if (ip.gt.3) ip = 3
                     if (ii.ge.2) then
                        rev_str(id,1) = rev_str(id,1) + r
                        rev_str(id,2) = rev_str(id,2) + r
                     end if
                     wipr(ic,ii,ip,iw,it,is,iq) = wipr(ic,ii,ip,iw,it,is,iq) + r
                     wipp(ic,ii,ip,iw,it,is,iq) = wipp(ic,ii,ip,iw,it,is,iq) + p
                     wipw(ic,ii,ip,iw,it,is,iq) = wipw(ic,ii,ip,iw,it,is,iq) + w
                     if (ii.eq.1) pmtrev(is,iq) = pmtrev(is,iq) + r
                  else
                     if (vips(iv)(2:4).ne.'340') then
                        print*,'No Class ',vips(iv)
                     end if
                  end if

               else
                  print*,'Bad vip remapping ',(iv)
               end if
            end if
         end do
300      print *,' read exit of pmt_stda.wi.',ap,' = ',ier
         close(30)
```

```
      end do


C     Calculate Trial Balance Permit and Bravis Own Factors
      write(90,'(a40)') 'Permit/Bravis Own Factors'
      write(90,'(a40)') '   '
      do iq = 1, nq
         do is = 1, nstr
            if (pmtrev(is,iq).gt.0.) then
               factpmt(is,iq) = nctbpmt(is,iq) / pmtrev(is,iq)
            else
               print *, ' ?!  No PERMIT revenue in strata ',is,' quarter ',iq
            end if
            write(90,90) is,iq,nctbpmt(is,iq),pmtrev(is,iq),factpmt(is,iq)
            pmtrev(is,iq) = 0.
         end do
      end do

400   format(i3,4f8.4,4f14.0)
410   format(i3,8f14.0)
      print *,'  '
      print *,' PERMIT Own Factors and NCTB Revenue: '
      do is = 1, nstr
         write (*,400) is, (factpmt(is,iq),iq=1,nq), (nctbpmt(is,iq), iq=1,nq)
      end do

C     Apply Own Factors and Check

      do iq = 1, nq
         do is = 1, nstr
            do it = 1, ntype
               do iw = 1, nwt
                  do ip = 1, nshp
                     do ii = 1, nind
                        do ic = 1, ncls
                           wipr(ic,ii,ip,iw,it,is,iq) =
     &                        wipr(ic,ii,ip,iw,it,is,iq) * factpmt(is,iq)
                           wipp(ic,ii,ip,iw,it,is,iq) =
     &                        wipp(ic,ii,ip,iw,it,is,iq) * factpmt(is,iq)
                           wipw(ic,ii,ip,iw,it,is,iq) =
     &                        wipw(ic,ii,ip,iw,it,is,iq) * factpmt(is,iq)
                           if (ii.eq.1) then
                              pmtrev(is,iq) = pmtrev(is,iq) +
     &                           wipr(ic,ii,ip,iw,it,is,iq)
                           else
C                             print*,'is ',is
c                             print*,'iq ',iq
c                             print*,'ic ',ic
c                             print*,'ii ',ii
c                             print*,'ip ',ip
c                             print*,'iw ',iw
c                             print*,'it ',it

                              smrev(is,iq) = smrev(is,iq) + wipr(ic,ii,ip,iw,it,is,iq)
                           end if
                        end do
                     end do
                  end do
               end do
            end do
         end do
      end do

      print *,'  '
      print *,' PERMIT Permit Imprint Controlled Revenue and NCTB Revenue: '
      do is = 1, nstr
         write (*,410) is, (pmtrev(is,iq),iq=1,nq), (nctbpmt(is,iq), iq=1,nq)
      end do

C     Calculate Trial Balance Total Factors
      write(90,'(a40)') 'Trial Balance Total Factors'
      write(90,'(a40)') '   '
      do iq = 1, nq
         do is = 1, nstr
            if (pmtrev(is,iq).gt.0.) then
               factall(is,iq) = nctball(is,iq) /
     &            (pmtrev(is,iq))
               write(90,90) is,iq,nctball(is,iq),pmtrev(is,iq),factall(is,iq)
               factall_sm(is,iq) = (sm_fitted(is,iq)+smrev(is,iq)) /
```

```
     &           (smrev(is,iq))
                 write(90,90) is,iq,(sm_fitted(is,iq) + smrev(is,iq)),smrev(is,iq),factall_sm(is,iq)
              else
                 print *, ' ?! No PERMIT or BRAVIS revenue strata ',is,' q ',iq
              end if
              pmtrev(is,iq) = 0.
              smrev(is,iq) = 0.
           end do
        end do

        print *,'  '
        print *,' All Office Factors and NCTB Revenue: Permit Imprint'
        do is = 1, nstr
           write (*,400) is, (factall(is,iq),iq=1,nq), (nctball(is,iq), iq=1,nq)
        end do
        print *,'  '
        print *,' All Office Factors and NCTB Revenue: Stamped/Metered'
        do is = 1, nstr
           write (*,400) is, (factall_sm(is,iq),iq=1,nq), (sm_fitted(is,iq), iq=1,nq)
        end do


C       Apply Total Factor and Check

        do iq = 1, nq
           do is = 1, nstr
              do it = 1, ntype
                 do iw = 1, nwt
                    do ip = 1, nshp
                       do ii = 1, nind
                          do ic = 1, ncls
                             if (ii.eq.1) then
                                wipr(ic,ii,ip,iw,it,is,iq) =
     &                             wipr(ic,ii,ip,iw,it,is,iq) * factall(is,iq)
                                wipp(ic,ii,ip,iw,it,is,iq) =
     &                             wipp(ic,ii,ip,iw,it,is,iq) * factall(is,iq)
                                wipw(ic,ii,ip,iw,it,is,iq) =
     &                             wipw(ic,ii,ip,iw,it,is,iq) * factall(is,iq)
                             else
                                wipr(ic,ii,ip,iw,it,is,iq) =
     &                             wipr(ic,ii,ip,iw,it,is,iq) * factall_sm(is,iq)
                                wipp(ic,ii,ip,iw,it,is,iq) =
     &                             wipp(ic,ii,ip,iw,it,is,iq) * factall_sm(is,iq)
                                wipw(ic,ii,ip,iw,it,is,iq) =
     &                             wipw(ic,ii,ip,iw,it,is,iq) * factall_sm(is,iq)
                             end if
                             if (ii.eq.1) then
                                pmtrev(is,iq) = pmtrev(is,iq) +
     &                             wipr(ic,ii,ip,iw,it,is,iq)
                             else
                                smrev(is,iq) = smrev(is,iq) +
     &                             wipr(ic,ii,ip,iw,it,is,iq)

                             end if
                          end do

                       end do
                    end do
                 end do
              end do
           end do
        end do

        print *,'  '
        print *,' PERMIT/BRAVIS Controlled Revenue and NCTB Revenue: Permit Imprint'
        do is = 1, nstr
           write (*,410) is, (pmtrev(is,iq),iq=1,nq), (nctball(is,iq), iq=1,nq)
        end do

        print *,'  '
        print *,' PERMIT/BRAVIS Controlled Revenue and NCTB Revenue: Stamped'
        do is = 1, nstr
           write (*,410) is, (smrev(is,iq),iq=1,nq), (sm_fitted(is,iq), iq=1,nq)
        end do

C       Distribute Bad Weight Data to Identical Weight

        open(50,file='wtstdanp.csv',recl=500)
51      format(i2,',',i1,',',20(f16.3,','),f16.3,',',f16.3,',',f16.3)
```

```
C     Sum identical and non-identical into result array,
C      bad weight into badwght array

      x=0
      xc=0
      do iw = 1, nwt
         do ic = 1, ncls
            do ip = 1, nshp
               pishrwt(iw,ic,ip) = 0.
               shrwt(iw) = 0.
            end do
         end do
      end do

      do ii = 1, nind
         x=0
         xc=0
         do iq = 1, nq
            do is = 1, nstr
               do it = 1, 2     ! skip bad weight
                  do iw = 1, nwt
                     do ip = 1, nshp
                        do ic = 1, ncls
                           result(1,iw,ip,ic) =
     &                        result(1,iw,ip,ic) +
     &                        wipr(ic,ii,ip,iw,it,is,iq)
                           result(2,iw,ip,ic) =
     &                        result(2,iw,ip,ic) +
     &                        wipp(ic,ii,ip,iw,it,is,iq)
                           result(3,iw,ip,ic) =
     &                        result(3,iw,ip,ic) +
     &                        wipw(ic,ii,ip,iw,it,is,iq)

                           xc = xc + wipr(ic,ii,ip,iw,it,is,iq)
                        end do
                     end do
                  end do
               end do
               do iw = 1, nwt
                  do ip = 1, nshp
                     do ic = 1, ncls
                        badwght(1,ip,ic) = badwght(1,ip,ic) +
     &                     wipr(ic,ii,ip,iw,3,is,iq)
                        badwght(2,ip,ic) = badwght(2,ip,ic) +
     &                     wipp(ic,ii,ip,iw,3,is,iq)
                        badwght(3,ip,ic) = badwght(3,ip,ic) +
     &                     wipw(ic,ii,ip,iw,3,is,iq)
                        xc = xc + wipr(ic,ii,ip,iw,3,is,iq)
                        x= x + wipr(ic,ii,ip,iw,3,is,iq)
                     end do
                  end do
               end do
            end do
         end do
         print*,'sum of bad weight revenues  ', x
         print*,'sum of total revenues ',xc

         x=0

C     Now distribute bad weight pieces

         do ic = 1, ncls
            do ip = 1, nshp
               do iw = 1, nwt
                  shrwt(iw) = 0.
               end do
               p = 0.
               do iw = 1, nwt
                  p = p + result(2,iw,ip,ic)
                  shrwt(iw) = shrwt(iw) + result(2,iw,ip,ic)
               end do
               do iw = 1, nwt
                  if(p.gt.0) then
                     shrwt(iw) = shrwt(iw)/p
                  end if
                  if (ii.eq.1) then
                     pishrwt(iw,ic,ip) = shrwt(iw)
                  end if
                  if (badwght(2,ip,ic).gt.0) then
C                  write(*,'(3i3,f6.3)') ic,ip,iw,shrwt(iw)
```

```
                  end if
               end do
               if (badwght(2,ip,ic).gt.0) then
                  if (minrate(ic).lt.99) then
                     if (p.gt.0.) then
                        do iw = 1, nwt
                           wpp = 0
                           if (result(2,iw,ip,ic).gt.0) then
                              wpp = result(3,iw,ip,ic) / result(2,iw,ip,ic)
                           else
                              wpp = 0.
                           end if
                           do ir = 1, 2
                              result(ir,iw,ip,ic) =
     &                           result(ir,iw,ip,ic) +
     &                           badwght(ir,ip,ic)*shrwt(iw)
                           end do
                           x=x+badwght(1,ip,ic)*shrwt(iw)
                           result(3,iw,ip,ic) =
     &                        badwght(2,ip,ic)*(shrwt(iw)) * wpp +
     &                        result(3,iw,ip,ic)
                        end do
                     else
                        do iw = 1, nwt
                           print *, "share at wgt inc ", iw, " is ", pishrwt(iw,ic,ip), " with subclass ", ic
                           if (result(2,iw,ip,ic).gt.0) then
                              wpp = result(3,iw,ip,ic) / result(2,iw,ip,ic)
                           else
                              wpp = 0.
                           end if
                           do ir = 1, 2
                              result(ir,iw,ip,ic) =
     &                           result(ir,iw,ip,ic) +
     &                           badwght(ir,ip,ic)*pishrwt(iw,ic,ip)
                           end do
                           x=x+badwght(1,ip,ic)*pishrwt(iw,ic,ip)
                           result(3,iw,ip,ic) =
     &                        badwght(2,ip,ic)*(pishrwt(iw,ic,ip)) * wpp +
     &                        result(3,iw,ip,ic)
                        end do

c                       print *,' No i/ni weight to dist bad ',ic,',',ip,',',ii
c                       print *,' r = ',badwght(1,ip,ic),' p = ',badwght(2,ip,ic)
                     end if
                  else
                     print *,' Bad wght to dist non-minrate ',ic,',',ip,',',ii,' ',badwght(2,ip,ic)
                  end if
               end if
            end do
         end do
         print*,'sum of badweight revenues after distribution ',x

C     Write Out Data for Importation into Excel


         x=0
         do ic = 1, ncls
            do ip = 1,3
               r = 0.
               p = 0.
               w = 0.
               do iw = 1, nwt
                  r = r + result(1,iw,ip,ic)
                  x = x + result(1,iw,ip,ic)
                  p = p + result(2,iw,ip,ic)
                  w = w + result(3,iw,ip,ic)
               end do
               write (50,51) ic, ip, (result(3,iw,ip,ic),iw=1,nwt),
     +              p,r,w
            end do
         end do
         write(50,'(a2)') 'SM'
         do ir = 1,nrpw
            do iw = 1,nwt
               do ip = 1,nshp
                  do ic = 1,ncls
                     result(ir,iw,ip,ic) = 0
                     badwght(ir,ip,ic) = 0
                  end do
               end do
```

```
            end do
         end do
         print*, 'after writing revenue ',x
      end do


C     Write Out Control Factors To File

      open(60,file='eststda_20.control')
61    format(4f20.15)
      do is = 1, nstr
         write (60,61) (factpmt(is,iq),iq=1,nq)
      end do
      do is = 1, nstr
         write (60,61) (factall(is,iq),iq=1,nq)
      end do

      close(60)

      open(60,file='strrev.txt')

      do is = 1,20
         write(60,'(i3,4f15.2)') is,rev_str(is,1),(rev_str(is,1)-rev_str(is,2)),fuzz(is),rev_str(is,1)-fuzz(is)
      end do

      do ip = 1,nshp
         write(*,'(i4,f26.0)') ip,pieces_by_shp(ip)
      end do

      end

C -------------------------------------------------------------------------
      function qind(iap)

      integer*4 qind, iap

      if (iap.ge.1.and.iap.le.3) then
         qind = 1
      else if (iap.ge.4.and.iap.le.6) then
         qind = 2
      else if (iap.ge.7.and.iap.le.9) then
         qind = 3
      else if (iap.ge.10.and.iap.le.13) then
         qind = 4
      else
         stop  ' invalid accounting period '
      end if

      return
      end

C -------------------------------------------------------------------------
      function strind(is)

      integer*4 strind, is

      if (is.ge.1.and.is.le.16) then
         strind = 1
      else if (is.ge.17.and.is.le.18) then
         strind = 2
      else if (is.ge.19.and.is.le.19) then
         strind = 3
      else if (is.ge.20.and.is.le.20) then
         strind = 4

      else
         stop ' bad strata '
      end if

      return
      end
C -------------------------------------------------------------------------
      function shapechk(ip,flag)

      integer*4 shapechk, ip, flag

      if (flag.eq.1) then
         shapechk = 1       ! letter only vip
      else if (flag.eq.2) then
         if (ip.eq.1) then
```

```
            shapechk = 2    ! non-letters vip - move letter to flat
         else
            shapechk = ip
         end if
      else if (flag.eq.3) then  ! all shape
         if(ip.gt.3) ip = 3
         shapechk = ip
      else if (flag.eq.4) then  ! flat only vip
         shapechk = 2
      else
         print*, 'Flag ',flag,'  ip,',ip
         stop ' bad shape flag '
      end if

      return
      end

C ---------------------------------------------------------------------------
      function iind(vip)

      integer*4 iind
      character*5 vip

      if (vip(1:1).eq.'0') then
         iind = 1
      else if (vip(1:1).eq.'3') then
         if (vip(5:5).eq.'1') then
            iind = 2
         else if (vip(5:5).eq.'2') then
            iind = 2
         else if (vip(5:5).eq.'3') then
            iind = 2
         else if (vip(5:5).eq.'4') then
            iind = 2
         else if (vip(5:5).eq.'6') then
            iind = 1
         else if (vip(5:5).eq.'7') then
            iind = 1
         else if (vip(5:5).eq.'9') then
            iind = 1
         else
            iind = -1
         end if
      else
         iind = 2
      end if

      return
      end
C ---------------------------------------------------------------------------
```

```
      PROGRAM permit5

C     Date: 10-1-97
C     Programmer: JMC
C     Project: Standard B
C
C     Purpose: To write out the PERMIT data in a transaction by the vip code detail.  This
C     program is run for each quarter and the input files are used for each quarter.



      IMPLICIT NONE

C     VARIABLES
      integer*4       l,nvip,nfin,nwgt,flag,flag1,singcat,pwtrans,libvipcnt,liberr
      integer*4       noldvip, noldvipp,wgtflg,errcnt,specvipcnt,specerr,sorttype
      integer*4       iertot,ier1,ier2,ier3,ier4,ier5,ier6,ier7,ier8,jerr
      integer*4       ppvipcnt,pperr,nonpwpp,nonpwlib,nonpwspec,pwvips,totvips
      integer*4       bpmvipcnt,bpmerr,nonpwbpm,matchedbpm,pibpmch,pibpm,sumtr(20)
      parameter       (nvip = 184)
      parameter       (nfin=233)
      parameter       (nwgt=1)
      parameter       (noldvip = 25)
      parameter       (noldvipp = 28) ! number of old vips plus 3
      real*8          rtot(nfin,nvip,4,4), wtot(nfin,nvip,4,4), vtot(nfin,nvip,4,4)
      real*8          rchk, wchk, vchk, wchk2, wchk3,calcwpp
      integer*4       rectot(nfin,nvip,4,4),find2,find4,totviperr
      real*8          revenue(nfin,nvip,2,nwgt), weight(nfin,nvip,2,nwgt), r, w
      real*8          v, p, volume(nfin,nvip,2,nwgt),bpmlbrt(24),bpmpcrt(24)
      real*8          goodtotchkps,goodtotchklbs,goodtotchkrs
      real*8          goodrtots,goodwtots,goodptots
      real*8          goodtotchkpl,goodtotchklbl,goodtotchkrl
      real*8          goodrtotl,goodwtotl,goodptotl
      real*8          goodtotchkpb,goodtotchklbb,goodtotchkrb
      real*8          goodrtotb,goodwtotb,goodptotb
      real*8          goodtotchkpp,goodtotchklbp,goodtotchkrp
      real*8          goodrtotp,goodwtotp,goodptotp
      real*8          goodtotchkp,goodtotchklb,goodtotchkr
      real*8          goodrtot,goodwtot,goodptot,twgt,tpcs,tbpmwgt,tbpmpcs
      real*8          sumrev(20),sumpcs(20),sumwgt(20)
      integer*4       transactions(nfin,nvip,4,4), trancnt(nvip,2), count
      character*2     pmtmthd, reccode,seq,bpmvip2(24),ap
      character*5     vips(nvip), rvip(100), pmtno, lastvip
      character*13    finno,fin(nfin) !was 6
      character*6     finno6
      character*15    transno
      integer*4       i,ier,in,j,k,searchc, totin, inchk, day, lnno, year,cnt
      integer*4       ier9,ier10,ier11,ier12,ier13,ier13a,pwflag,procat,ivip

      integer*4       ier14,ier15,ier16,ier17,ier18,ier19,ier20,ier21
      integer*4       ier22,ier23,ier24,ier25,ier26,ier27,ier28,ier29
      integer*4       ier30,ier31,ier32,ier33,ier34,ier35,ier36,ier37,ier38,ier39
      character*5008  record
      character*137   rechead
      character*48    recarray(100)
      character*31    trans
      character*5     product, lastproduct
      real*8          spcfee, pw, totlb, rrev, rpost(100)
      real*8          totp, rline, rscks, rp(100),rw(100), rlbs, rt(100), trev, tvol
      real*8          tottotp,tottotlb,tottotr,totchkp,totchklb,totchkr
      integer*4       nlines, il, n, rectype, pvip, vdflno, tdflno
      integer*4       check1, check2, find3,sort(100)
      integer*4       gtwy(nvip)
      real*8          disc(nvip)

C     Rate Table Variables

      integer*4       intrawgt(70),internwgt(70),intermwgt(35),dbmcwgt(70)
      integer*4       specwgt(70),libwgt(70),id,idx,zercnt
      real*8          intraz1(70),intraz2(70),intraz3(70),intraz4(70),intraz5(70),intraloc(70)
      real*8          internz1(70),internz2(70),internz3(70),internz4(70),internz5(70)
      real*8          internz6(70),internz7(70),internz8(70),intermz1(35),intermz2(35)
      real*8          intermz3(35),intermz4(35),intermz5(35),intermz6(35),intermz7(35)
      real*8          intermz8(35),dbmc1(70),dbmc2(70),dbmc3(70),dbmc4(70),dbmc5(70)
      real*8          spec1(70),spec2(70),spec3(70),librt(70)
      logical         debug
      character*24    ecode(10)
      logical         found3
      real*8          avewt,avewt2,aprpw(3),specbc
```

```
      integer*4      il2,bcvip


C     READ IN LIST OF INTERNATIONAL VIP CODES
      debug=.false.


C     READ TRANSACTIONS FILE ROLLING TO VIP
20    format(a137)              ! record header
21    format(a31,8x,f12.2,2x,f12.2,13x,i1,1x,f8.4,f12.0,f14.4,20x,i3)
C     transno/permit#,trans amount,special fees,piece weight,total pieces,
C     total pounds,number of lines to follow

22    format(i5,1x,a5,1x,f6.3,f12.0,f18.7)
C     line number, vip code,rate, pieces/pounds,postage
C      open(25,file='/u/mcb/permit96/P/fourth/permit.96f.4th.01',recl=5008,iointent='input')
23    format(a6,a2,a15,a2,a5)
24    format(i4,i3,8x)
C     finno,pmtmthd,transno,reccode,pmtno
      open(35,file='data/data.q4_4')
      open(36,file='datafile4',recl=4500)
35    format(a13,1x,a5,1x,i2,1x,3f14.2)

26    format(a15,",",i2,",",a6,",",a5,",,,",a5,",",i3,",",f16.3,",",f16.0,",",f17.2)
27    format(a15,",",i2,",",a6,",",a5,",,,",a2,",",a2,",",a5,",",f10.2,",",f8.5,",",f14.3,",",
     &   f10.0,",",f14.2,",",f14.3,",",f10.0,",",f14.2)
      totin = 0
      ier = 0
      inchk = 0
      in = 0
      errcnt=0
      specvipcnt=0
      specerr=0
      libvipcnt=0
      liberr=0
      ppvipcnt=0
      pperr=0
      nonpwpp=0
      nonpwspec=0
      nonpwlib=0
      pwtrans=0
      pwvips=0
      totvips=0
      bpmerr=0
      nonpwbpm=0
      bpmvipcnt=0
      matchedbpm=0
      pibpmch=0
      pibpm=0
      totviperr=0
      tottotp=0
      tottotlb=0
      tottotr=0
      totchklb=0
      totchkp=0
      totchkr=0
      goodtotchkps = 0.0
      goodtotchklbs=0.0
      goodtotchkrs=0.0
      goodrtots=0.0
      goodwtots=0.0
      goodptots=0.0
      goodtotchkpl = 0.0
      goodtotchklbl=0.0
      goodtotchkrl=0.0
      goodrtotl=0.0
      goodwtotl=0.0
      goodptotl=0.0
      goodtotchkpb = 0.0
      goodtotchklbb=0.0
      goodtotchkrb=0.0
      goodrtotb=0.0
      goodwtotb=0.0
      goodptotb=0.0
      goodtotchkpp = 0.0
      goodtotchklbp=0.0
      goodtotchkrp=0.0
      goodrtotp=0.0
      goodwtotp=0.0
```

```
      goodptotp=0.0
      goodtotchkp = 0.0
      goodtotchklb=0.0
      goodtotchkr=0.0
      goodrtot=0.0
      goodwtot=0.0
      goodptot=0.0
      do i=1,20
         sumtr(i)=0
         sumrev(i)=0.0
         sumwgt(i)=0.0
         sumpcs(i)=0.0
      end do

30    format(a)

      DO WHILE (ier.EQ.0)
c         print*,'inside the loop'
         pvip = 0
         do i = 1, nvip
            trancnt(i,1) = 0
            trancnt(i,2) = 0
         end do
         read(36,'(a)',iostat=ier,end=29) record


         totin= totin+1
         read(record,20) rechead
         read(rechead,21) trans,r,spcfee,procat,pw,totp,totlb, nlines
         if (procat.le.3) then
            sorttype=1
         else
            sorttype=2
         end if

         tottotp=tottotp+totp
         tottotlb=tottotlb+totlb
         tottotr=tottotr+r
         if (pw.eq.0) then
            pwtrans = pwtrans +1
         end if
         read(trans,23) finno6,pmtmthd,transno,reccode,pmtno
         finno=finno6//pmtmthd//pmtno

         read(transno,24) year,day
         idx=0
         zercnt=0
         do il = 1, nlines
            n = 138+(il-1)*48
            if (record(n+6:n+10).ne.'    0') then
               idx=idx+1
               recarray(idx) = record(n:n+47)
            else
               zercnt=zercnt+1
            end if
         end do
         nlines = nlines-zercnt
         seq=record(22:23)
         if (nlines.gt.0) then  ! if there is vip detail
c        Initialize transaction level variables if vip detail

            rchk = 0.0
            vchk = 0.0
            wchk = 0.0
            trev = 0.0
            tvol = 0.0
            twgt = 0.0
            pvip = 0
            flag1=0
            wgtflg=0
            wchk2=0
            wchk3=0
            singcat=0
            ier1=0
            ier2=0
            ier3=0
            ier4=0
            ier5=0
            ier6=0
            ier7=0
```

```
ier8=0
ier9=0
ier10=0
ier11=0
ier12=0
ier13=0
ier13a=0
ier14=0
ier15=0
ier16=0
ier17=0
ier18=0
ier19=0
ier20=0
ier21=0
ier22=0
ier23=0
ier24=0
ier25=0
ier26=0
ier27=0
ier28=0
ier29=0
ier30=0
ier31=0
ier32=0
ier33=0
ier34=0
ier35=0
ier36=0
ier37=0
ier38=0
ier39=0
pwflag=2
flag1 = 0

find2=0

do i=1,100
   rvip(i)=' '
   rt(i)=0.0
   rp(i)=0.0
   rw(i)=0.0
   rpost(i)=0.0
end do



C     READ IN ALL THE LINES
      do il = 1, nlines


         if (pw.eq.0) then
            pwvips = pwvips + 1
            pwflag = 1
         end if
         totvips = totvips+1

         read(recarray(il),22) lnno, rvip(il), rt(il), rp(il), rpost(il) !rp = pcs or lbs


         if (rvip(il)(1:1).eq.' ') then
            rvip(il)(1:1)='0'
         end if


      end do


C     Add Section to Fix the Special Barcoded RPW

      do il = 1,nlines
         if (rvip(il)(2:5).eq.'4150') then
            bcvip = il
            specbc = rp(il)
            do il2 = 1,nlines
               if (rvip(il2)(2:5).eq.'4140') then
                  if (rp(il2).ge.specbc) then
```

```
            rt(bcvip) = (rpost(il2)/rp(il2))-0.03
            rp(bcvip) = specbc
            rp(il2) = rp(il2) - specbc
            rpost(bcvip) = rp(bcvip)*rt(bcvip)
            rt(il2) = rt(bcvip) + .03
            rpost(il2) = rp(il2) * rt(il2)
            specbc = 0.0
         else
            rp(bcvip) = rp(il2)
            rt(bcvip) = (rpost(il2)/rp(il2))-.03
            rpost(bcvip) = rp(bcvip)*rt(bcvip)
            specbc = specbc - rp(il2)
            rt(il2) = rt(bcvip) + .03
            rp(il2) = 0.0
            rpost(il2) = rp(il2) * rt(il2)
         end if
      end if
   end do

   if (specbc.gt.0) then
      do il2 = 1,nlines
         if (rvip(il2)(2:5).eq.'4100') then
            if (rp(il2).ge.specbc) then
               rt(nlines+1) = (rpost(il2)/rp(il2)) - 0.03
               rp(nlines+1) = specbc
               rvip(nlines+1) = rvip(il2)(1:1)//'4110'
               rpost(nlines+1) = rp(nlines+1)*rt(nlines+1)
               rp(il2) = rp(il2) - specbc
               rt(il2) = rt(nlines+1) + .03
               rpost(il2) = rt(il2) * rp(il2)
               nlines = nlines + 1
            else
               rp(nlines+1) = rp(il2)
               rt(nlines+1) = (rpost(il2)/rp(il2))-.03
               rvip(nlines+1) = rvip(il2)(1:1)//'4110'
               rpost(nlines+1) = rp(nlines+1)*rt(nlines+1)
               specbc = specbc - rp(il2)
               rt(il2) = rt(nlines+1) + .03
               rp(il2) = 0.0
               rpost(il2) = rp(il2) * rt(il2)
               nlines = nlines + 1
               print*,'error in reassigning pcs ',rechead
            end if
         end if
      end do
   end if
end if
if (rvip(il)(2:5).eq.'4110') then
   bcvip = il
   specbc = rp(il)
   do il2 = 1,nlines
      if (rvip(il2)(2:5).eq.'4100') then
         if (rp(il2).ge.specbc) then
            rt(bcvip) = (rpost(il2)/rp(il2))-0.03
            rp(bcvip) = specbc
            rp(il2) = rp(il2) - specbc
            rpost(bcvip) = rp(bcvip)*rt(bcvip)
            rt(il2) = rt(bcvip) + .03
            rpost(il2) = rp(il2) * rt(il2)
            specbc = 0.0
         else
            rp(bcvip) = rp(il2)
            rt(bcvip) = (rpost(il2)/rp(il2))-.03
            rpost(bcvip) = rp(bcvip)*rt(bcvip)
            specbc = specbc - rp(il2)
            rt(il2) = rt(bcvip) + .03
            rp(il2) = 0.0
            rpost(il2) = rp(il2) * rt(il2)
         end if
      end if
   end do
   if (specbc.gt.0) then
      do il2 = 1,nlines
         if (rvip(il2)(2:5).eq.'4180') then
            rpost(il2) = rpost(il2)   rpost(bcvip)
            rpost(bcvip) = 0.0
            rp(bcvip) = 0.0
         end if
      end do
      if (rpost(bcvip).gt.0) then
```

```
                  print*,'error with non-presort ',rechead
               end if
            end if
         end if

      end do
      if (record(76:76).eq.'7') then
         tbpmwgt = 0
         tbpmpcs = 0
         do il = 1,nlines
            if ((rvip(il)(3:3).eq.'7')) then
               rw(il) = 0
               if ((rvip(il)(4:4).eq.'0').or.(rvip(il)(4:4).eq.'1')) then
                  rw(il) = rp(il) * (totlb/totp)
               end if
               tbpmpcs = tbpmpcs + rp(il)
            else if ((rvip(il)(3:3).eq.'8')) then
               rw(il) = rp(il)/10000
               rp(il) = 0
               tbpmwgt = tbpmwgt + rw(il)
            end if
         end do
         if (record(7:8).eq.'PI') then
            if (tbpmwgt.eq.0) then
            end if
         end if
      end if

      if ((record(76:76).eq.'4').or.(record(76:76).eq.'8')) then
         avewt = totlb/totp
         if (avewt.gt.70) then
            print*,'avewt is too large ',avewt
         end if
         do il = 1,nlines
            rw(il) = rp(il) * avewt
         end do
      end if

      if ((totlb/totp.ge.4).and.(procat.eq.2)) then
         procat = 3
C           Parcel if > 4 lbs. per piece.
      end if



      do il = 1,nlines
         write(35,35) finno,rvip(il),procat,rpost(il),rp(il),rw(il)
      end do
   end if



END DO


C     print*,'Exit code from data is: ',ier



END
```

```
      program roll_vip

c     Purpose: To roll up the data by vip code by quarter and by shape
c     Author: JMC
c     Date: 3-21-01

      implicit none
      integer*4   ivip,irpw,ishp,iq,searchc
      integer*4   nvip,nrpw,nshp,nq,ier
      parameter   (nvip=236)
      parameter   (nrpw=3)
      parameter   (nshp=5)
      parameter   (nq=4)
      character*5 vip(nvip),vipx
      real*8      data(nq,nshp,nvip,nrpw),r,p,w


      open(10,file='vipmap')
10    format(a5)

      do ivip = 1,nvip
         read(10,10) vip(ivip)
         do irpw = 1,nrpw
            do ishp = 1,nshp
               do iq = 1,nq
                  data(iq,ishp,ivip,irpw) = 0.0
               end do
            end do
         end do
      end do

      open(20,file='data/data.q1_4')
20    format(14x,a5,i3,1x,3f14.2)

      ier = 0

      do while (ier.eq.0)
         read(20,20,iostat=ier,end=100) vipx,ishp,r,p,w

         ivip = searchc(vip,nvip,vipx)

         if (ivip.gt.0) then
            data(1,ishp,ivip,1) = data(1,ishp,ivip,1) + r
            data(1,ishp,ivip,2) = data(1,ishp,ivip,2) + p
            data(1,ishp,ivip,3) = data(1,ishp,ivip,3) + w
         else
            print*,'vip not found ',ivip
         end if
      end do
100   print*,'loop exited with ier of ',ier
      close(20)
      open(20,file='data/data.q2_4')

      ier = 0

      do while (ier.eq.0)
         read(20,20,iostat=ier,end=200) vipx,ishp,r,p,w

         ivip = searchc(vip,nvip,vipx)

         if (ivip.gt.0) then
            data(2,ishp,ivip,1) = data(2,ishp,ivip,1) + r
            data(2,ishp,ivip,2) = data(2,ishp,ivip,2) + p
            data(2,ishp,ivip,3) = data(2,ishp,ivip,3) + w
         else
            print*,'vip not found ',ivip
         end if
      end do
200   print*,'loop exited with ier of ',ier
      close(20)
      open(20,file='data/data.q3_4')

      ier = 0

      do while (ier.eq.0)
         read(20,20,iostat=ier,end=300) vipx,ishp,r,p,w

         ivip = searchc(vip,nvip,vipx)

         if (ivip.gt.0) then
```

```
            data(3,ishp,ivip,1) = data(3,ishp,ivip,1) + r
            data(3,ishp,ivip,2) = data(3,ishp,ivip,2) + p
            data(3,ishp,ivip,3) = data(3,ishp,ivip,3) + w
         else
            print*,'vip not found ',ivip
         end if
      end do
300   print*,'loop exited with ier of ',ier
      close(20)
      open(20,file='data/data.q4_4')

      ier = 0

      do while (ier.eq.0)
         read(20,20,iostat=ier,end=400) vipx,ishp,r,p,w

         ivip = searchc(vip,nvip,vipx)

         if (ivip.gt.0) then
            data(4,ishp,ivip,1) = data(4,ishp,ivip,1) + r
            data(4,ishp,ivip,2) = data(4,ishp,ivip,2) + p
            data(4,ishp,ivip,3) = data(4,ishp,ivip,3) + w
         else
            print*,'vip not found ',ivip
         end if
      end do
400   print*,'loop exited with ier of ',ier
      close(20)

      open(40,file='data/viprpw_4')
40    format(i2,i2,1x,a5,3f16.2)

      do iq = 1,nq
         do ishp = 1,nshp
            do ivip = 1,nvip
               if ((data(iq,ishp,ivip,1)+data(iq,ishp,ivip,2)+data(iq,ishp,ivip,3)).gt.0.0) then
                  write(40,40) iq,ishp,vip(ivip),(data(iq,ishp,ivip,irpw),irpw=1,nrpw)
               end if
            end do
         end do
      end do

      end
```

```
      PROGRAM ODIS
C
C       PURPOSE:  READ ODIS560 DATA AND FILL MATRIX(INDICIA,TYPE,CLASS,
C                 ODIS,O/D) AND WRITE TO FILE also converts the class/marks
C                 to the previous form
C

      IMPLICIT NONE


      integer*4 lastpos
C
################################################################################
C   ******* These must be updated each Quarter *****************
      parameter      (lastpos=8)     ! Last position on tape with data
      character*11 tapename/"/dev/rmt/2n"/    !   Drive where tape is located ==>
no rewind
      character*1  qtr/"4"/     ! Quarter processing
      character*4  yr/"2000"/   ! Year processing
C
################################################################################


      INTEGER*4       OAO,OAD,CLASS,TYPE,INDICIA,mark,totcats
      integer*4      I,J,ier,numcats,index,pos
      integer*4       searchi
      integer*4       pdco,pdcd,zao,zad,numind,totind,handwritten,FIM,barcode
      integer*4       forward,specserv,totzip
      parameter       (numind=6)     ! # of indicia
      parameter       (totind=numind+1)
      parameter       (totcats=200)
      parameter       (totzip=999)

      integer*4 cat(totcats)/totcats*0/
      REAL*8          VOLUME
      REAL*8          summary(totcats,totind)/1400*0.0/
      REAL*8          totshape(5,totind)/35*0.0/
      REAL*8          totclass(9,totind)/63*0.0/
      REAL*8          totindicia(totind)/totind*0.0/
      character*3     adcd
      character*5     bacsd
      character*6     dfin
      character*34  description(totcats)/totcats*' '/
      character*12  indname(totind)/'    STAMPED','    METERED','         PI',
     &    '  PERMIT BRM','  GOVT METER','  OTHER GOVT','       TOTAL'/
      character*15  classname(9)/'First Class','Periodicals','Priority',
     &   'Standard B','OTHER STD B','STD A SNGL PC','STD A BLK','STD A
NP','FOREIGN'/
      character*15 shapename(5)/'Letters','Cards','IPP','FLATS','PARCELS'/
      character*380  record
      integer*4       unit,tapeopen,taperead,tapeclose


C      ****** Open up map for class, subclass, and shape *******
      OPEN(30,FILE='subclass.map',readonly)
31    FORMAT(i3,1x,a34)
      i = 1
```

```
      ier = 0
      do while (ier.eq.0)
         read(30,31,iostat=ier,end=32) cat(i), description(i)
         i = i + 1
      end do
32    numcats = i - 1
      print*,' Read Subclass map ier = ',ier,' Read ',numcats,' lines'


C      *** OPEN ZIP 560 tape for processing ******


21
FORMAT(2x,2I3,7x,I3,17x,2i3,4x,a3,i3,7x,A5,5x,a6,15x,2i1,3x,4i1,11x,f8.2,29x,i1,
20x,2i1)
22    FORMAT(A380)
      i = 0
      ier = 0
      do unit = 0,lastpos
         ier = tapeopen(20,tapename//char(0),27740,"input",380)
         if (ier.ne.0) then
            print*,'Error opening tape drive ',ier
            stop
         end if

         do while (ier.eq.0)
            ier = taperead(20,record)
            IF (IER.NE.0) goto 200
            read(record,21,iostat=ier,end=200) OAO,pdco,zao,OAD,pdcd,adcd,zad,
     +         bacsd,dfin,type,indicia,mark,barcode,FIM,handwritten,volume
     +            ,class,forward,specserv


            i = i + 1

C ********************* Create National totals ************************
            index = class*100 + mark*10 + type
            pos = searchi(cat,numcats,index)
            if (pos.eq.0) then
               print*,' Category not found!!!! ',index,' ',volume
            else
               if ((indicia.lt.1).or.(indicia.gt.numind)) then
                  print*,' Indicia out of range !!!',indicia,volume
               else

               summary(pos,indicia) = summary(pos,indicia) + volume
               summary(pos,numind+1) = summary(pos,numind+1) + volume
               totshape(type,indicia) = totshape(type,indicia) + volume
               totshape(type,numind+1) = totshape(type,numind+1) + volume
               totclass(class,indicia) = totclass(class,indicia) + volume
               totclass(class,numind+1) = totclass(class,numind+1) + volume
               totindicia(indicia) = totindicia(indicia) + volume
               totindicia(numind+1) = totindicia(numind+1) + volume

            END IF
            END IF
         end do
```

```
200       print *,'Exit code of position ',unit,' is = ',ier,' count = ',i
          if (ier.ne.-11025) stop
          ier = tapeclose (20)
          i  = 0
       end do

       OPEN(50,FILE='hardcopy.q'//qtr//yr(3:4))
51     FORMAT(34x,7a12)
52     FORMAT(a34,7f12.0)
53     format(29x,'Total',7f12.0)
54     format('Q'a1,1x,a4,' AVERAGE DAILY MAIL VOLUMES FROM ODIS 560')


       write(50,54) qtr,yr
       write(50,*)
       write(50,51) (indname(i),i=1,numind+1)

       do i = 1,numcats
          write(50,52) description(i),(summary(i,j),j=1,numind+1)
       end do
       write(50,*)
       do i = 1,9
          write(50,52) classname(i),(totclass(i,j),j=1,numind+1)
       end do
       write(50,*)
       do i = 1,5
          write(50,52) shapename(i),(totshape(i,j),j=1,numind+1)
       end do
       write(50,*)
       write(50,53) (totindicia(i),i=1,numind+1)


       END
```