

UNITED STATES POSTAL SERVICE

RECEIVED
SEP 24 4 53 PM '01
POSTAL RATE COMMISSION
OFFICE OF THE SECRETARY

LIBRARY REFERENCE USPS-LR-J-18
DOCKET NO. R2001-1

BULK MAIL REVENUE, PIECES AND WEIGHT SYSTEM (BRPW)

COMPUTER SYSTEM DOCUMENTATION

United States Postal Service

Bulk Revenue, Pieces and Weight System (BRPW)
Computer System Documentation

Table of Contents

	<u>Page</u>
Table of Contents	i
Preface	ii

Section Title

1.	General	1
2.	System Methodology	1
3.	Jobstream Description	2
4.	Job Inputs and Outputs	6
5.	Report Titles	7

APPENDIX A: Computer Program Code

Preface

This is a Category 1 library reference. It updates the BRPW computer system documentation previously provided in USPS-LR-I-25/R2000-1. The BRPW is the subject of the testimony of RPW System witness Hunter, USPS-T-4.

United States Postal Service

Bulk Mail Revenue, Pieces and Weight System (BRPW) Computer System Documentation

1. General

This document describes the Bulk Mail Revenue, Pieces and Weight System (BRPW) used by the Postal Service to develop estimates of revenue, pieces, and weight (RPW) totals for the classes and subclasses of bulk mail.

BRPW estimates of RPW totals are constructed from financial revenue accounts information and bulk mailing postage statement data. For the fiscal year (FY) 2000 period, BRPW estimates of RPW totals are developed for the following bulk mail categories: First-Class Mail (FCM), permit imprint Priority Mail, Periodicals, Regular and Nonprofit Standard Mail, and permit imprint Bound Printed Matter (BPM) and Parcel Post Standard Mail.

The BRPW is administered and run locally at Postal Service headquarters using the Service's mainframe computer located at the San Mateo Management Support Service Center (MSSC). The BRPW programs are written in the SAS System programming language (Release 6.09). IBM Job Control Language (JCL) up front in each program identifies job throughputs and job control statements for batch processing under Time Sharing Option (TSO) mode. The BRPW jobstream consists of four jobs run in series. The setting of a "Run Type" switch up front in the programming code of each job allows the same jobstream to be utilized for either postal quarter (PQ) or accounting period (AP) production.

The BRPW jobstream is described in Sections 2 and 3 of this document. Job input and output files, and reports are described in Sections 4 and 5, respectively. The computer program code is provided in Appendix A.

2. System Methodology

Source bulk mail postage statement data in machine readable format are captured each AP and input to the BRPW from the PERMIT System, a national automated bulk mail acceptance and financial reporting system. The PERMIT System provides an ongoing census of its member offices and currently captures over 90 percent of the nation's activity for the major mail categories reported under the BRPW. With respect to bulk mail not captured by the PERMIT System, the target population is represented in the BRPW by a probability-based stratified sample. The BRPW office panel therefore consists of all PERMIT System offices supplemented by a sample of non-automated offices.

Each non-automated panel office reports its postage statement activity for its assigned mail category to headquarters upon the close of an AP. Upon receipt of the postage statement information at headquarters, the data are keyed by a contractor into an electronic data file.

The BRPW input data consist of the postage statement transaction information obtained from the automated and non-automated panel offices aggregated by rate category within finance number. The rate category aggregation is accomplished using a five-character Volume Information Profile (VIP) Code scheme. The VIP Code scheme maps the revenue, volume, and weight data by office to individual rate categories as defined by the line items found on the postage statements for the applicable rate period and form issue date.

Upon input of the revenue, volume, and weight data to the BRPW jobstream, the input data records are matched to master rate file records by VIP Code and rate date to obtain the piece rate, pound rate, and weight per piece limits required for data verification. The estimates of RPW totals are then developed by combining these verified data records with office and stratum level blow-up factors, and then adjusting the combined strata estimates to the postage revenue account totals obtained from the trial balance accounting system.

Program run parameters include the current processing period, run period type (AP or PQ) and allowable rate dates. The parameters are front-end coded for accessibility. The AP or PQ processing mode is selected by setting the RUNTYPE processing period switch within each job. Modular coding is employed in the programs wherein records are grouped by mail class to facilitate maintenance of programming code and data flow comprehension.

Periodic panel updates and related programming code changes are simplified by using standardized input file record layouts to identify the automated and supplemental panel offices and their assigned strata and blowup factors. Rate period based rate files containing revenue per piece, revenue per pound, and weight per piece ratios, by VIP Code, for the numerous bulk mail rate categories, further facilitate periodic maintenance of rate files and related programming code changes. Up to four rate files can be read per mail class covering up to three rate change periods for a run period.

3. Jobstream Description

The BRPW jobstream consists of four jobs run sequentially. The first job collects the current period's postage statement data extracted from the PERMIT System and from the supplemental panel of non-automated offices. The second job performs data verification checks on the source data. This job utilizes individual rate files for each mail class to provide rate category information by rate date. The third job inflates the second job's output using office, stratum based blowup, and national trial balance factors. The final job copies the programming code, input files (except PERMIT System input files which are backed up separately), and output files to a magnetic tape cartridge with a 5-year retention period.

For the selected processing period, JCL file names for job inputs, outputs, and SAS language global variables are updated up front in each program by the operator. Each job is submitted to the mainframe manually in sequence using IBM TSO batch commands. A description of the four component jobs follows.

JOB-1

The following JOB-1 run parameters are specified by the operator:

RUNTYPE = 2-character 'AP' or 'PQ' literal for run period
 FY = 2-digit numeric
 PQ = 1-digit numeric
 AP = 2-digit numeric
 RDATE_n = 6-digit numeric MMDDYY rate date (n=1...4)

The PERMIT System and non-automated office panel files containing VIP Code revenue, volume, and weight data, aggregated by rate date within post office finance number, are read for all AP's of the selected run period. Input records are matched on finance number (FINNO) to the automated office and supplemental panel frames. Records for automated offices not found on the automated office frame are flagged for bypass from downstream processing. Unmatched non-automated office records trigger an automatic job abort for operator action. Delinquent non-automated offices (non-respondents) are tagged. Additional automatic job aborts detect inconsistent AP, PQ, and FY run period parameters, unrecognized rate dates, and duplicate records in the office frame and data source files. Unused foreign mail records are dropped from downstream processing.

Non-automated panel offices are checked for a change in automation status. Because some of the non-automated offices in the panel may be purposefully automated upon their selection in order to manage peak data entry workload, all non-automated panel offices migrating to the PERMIT System maintain their originally assigned strata and blowup factors. Where manual postage statement or negative report (zero volume for an AP) data are inadvertently keyed for a migrating office, these data are automatically overridden by the automated system's data.

All data records are assigned system and mail class level designators for downstream data verification and data editing purposes. The system and class designators are used to match the input data records to the appropriate rate files and VIP Codes within, and to assign AIC and RPW Code designators. Automated offices other than migrating non-automated offices default to the certainty stratum. Negative report offices listed on the non-automated office frame file or identified from the non-automated office data file are assigned a dummy zero volume VIP Code for the applicable AP.

Summary data reports are generated showing, for each AP in the selected run period: non-numeric data, unlisted frame sites, non-respondent offices, and summed revenue, volume, and weight values. A single JOB-1 output file is written containing VIP Code level records by rate date and finance number for each AP in the run period.

JOB-2

The following JOB-2 run parameters are specified by the operator:

RUNTYPE = 2-character 'AP' or 'PQ' literal for run period
 FY = 2-digit numeric
 PQ = 1-digit numeric
 AP = 2-digit numeric
 RDATE_n = 6-digit numeric MMDDYY rate date (up to 4 allowed)
 RTOL = numeric rate tolerance

WTOL = numeric weight tolerance

The JOB-1 output file records are read and checked for a match to the processing period RUNTYPE setting, rate date settings, and run period (FY, PQ and AP). A duplicate record check is made. Job aborts occur if failures are detected.

Depending upon the assigned mail class, each non-zero record is assigned to a class-based module for data verification and editing. The data verification and edit process is hierarchical, wherein, for each record, higher-level validation checks follow prerequisite checks. All other input records, i.e., failed JOB-1 verification, non-respondent and no activity records, are temporarily bypassed to a separate module for subsequent output at the end of the program.

For each mail class, up to four rate tables can be read for up to three rate change periods within the selected run period. The rate tables contain the VIP Code level revenue per piece, revenue per pound, and weight per piece limits that are required to validate the reported revenue, volume, and weight values in each record. Additional program code parameters specific to the rate date are coded at the beginning of each mail class module to facilitate programming code maintenance.

For each VIP Code record, computed revenue per piece, revenue per pound, and weight per piece ratios are constructed and compared at the VIP Code or RPW Code level with the expected ratios found in the matching rate date table. Records with missing or unexpected data, or having computed ratio(s) not within the applicable revenue or weight tolerances specified in the job run parameters section, are flagged.

An EFLAG variable initially set to zero identifies the error status for each input record. The following EFLAG values are used:

- EFLAG = 0000 - None
- EFLAG = 0050 - Migrated site - manual data not needed (Job-1)
- EFLAG = 0100 - Not on automated office frame (Job-1)
- EFLAG = 1000 - No rate table entry found
- EFLAG = 1100 - No RPW Code for RTABLE entry (redundant)
- EFLAG = 1200 - Non-pos. piece- or pound-rate for RTABLE entry
- EFLAG = 2000 - Empty R, P or W
- EFLAG = 2100 - Unexpected R, P or W
- EFLAG = 2500 - Empty revenue per piece or revenue per pound
- EFLAG = 3000 - Revenue tolerance
- EFLAG = 3100 - Weight tolerance

After the data verification and review process is completed, the records from all mail class modules are regrouped by office. If all records for a non-automated office are found to have failed, the office's non-response indicator is set. A non-response factor is computed for each non-automated office in a stratum if there is at least one non-respondent office in the stratum for the entire run period. This factor defaults to unity for all other non-automated and automated offices. If an automated or non-automated office reports activity for one or more AP's for a PQ run, but not for all AP's of the PQ, a missing AP office factor is computed. This factor is automatically set to unity otherwise, including for all offices for an AP run.

Edit reports are generated to show missing RPW codes, negative values, and unused VIP Codes. System and class level totals are generated by stratum. Data verification and edit summary reports are generated at three levels: stratum, VIP Code within stratum, and FINNO within VIP Code, and stratum. Computed and expected revenue per piece, revenue per pound, and weight per piece measures are also reported.

A single JOB-2 output file is written containing VIP Code level records by rate date and finance number for each AP in the run period.

JOB-3

The following JOB-3 run parameters are specified by the operator:

RUNTYPE = 2-character "AP" or "PQ" literal for run period
 FY = 2-digit numeric
 PQ = 1-digit numeric
 AP = 2-digit numeric

The JOB-2 output file records are read and checked for a match to the processing period RUNTYPE setting, rate date settings, and run period (FY, PQ and AP). Failed and unused mail category records are bypassed. Passed records (EFLAG=0000) are checked for missing values and duplicate entries. A failure at this level triggers a job abort.

For each office record, new variables are developed for inflated revenue, volume, weight, and copy (Periodicals) totals upon multiplying the uninflated record values by a stratum level blowup factor, missing AP adjustment factor, and stratum level non-response adjustment factor. For specific indicia and mail class combination records, a final adjustment is made to known revenue totals obtained from the trail balance accounts.

System and class level reports detail inflated revenue, volume, and weight totals along with uninflated totals and stratum office counts. The reports include estimated revenue per piece, revenue per pound, and weight per piece measures.

Passed records containing uninflated and inflated RPW totals by VIP Code, rate date, and FINNO, for each AP in the run period, are output to file. The inflated data are also aggregated by RPW Code and AIC before their output to a second file for input to the RPW Adjustment System (ARPW).

JOB-4

Upon the completion of each job in the BRPW jobstream for a PQ run, the programming code along with all input and output files used to construct the BRPW estimates are copied to tape with a 5-year retention period. All input and output data files are backed up in this manner except for the PERMIT System input files which are backed up separately at the San Mateo ISSC at the time they are created. AP processing period estimates are not backed up to tape, but remain available on disk in archives for a shorter period of time.

This job uses the IBM utility program IEBGENER to copy disk files to tape. With the exception of a standardized first level qualifier and the replacement of 'D01' with 'T01' in the second level qualifier of the dataset name (DSN), the disk and tape file names and content are the same. All backed up files for a PQ run are written to the same tape cartridge.

4. Job Inputs and Outputs

JOB 1-3 input and output files are identified in this section. Data record formats are found within the SAS System programming code provided in Appendix A of this document. All electronic data files are sequential in structure. The PERMIT System frame and input data file names follow the IBM Generation Data Set (GDS) naming convention.

<u>JOB</u>	<u>TYPE</u>	<u>CONTENT</u>	<u>Disk File Name</u>
1	IN	PERMIT System Data - by AP	HSISMN.PS170D01.RPW.FYyyyyap.Dnnnnn(0)
	"	Non-automated Panel - by AP	HSQRAN.BRPWD01.MANUALS.FYyyyyap
	"	PERMIT System Frame - by AP	HSISMN.BRPWD01.OFFLIST.Fyyyyyap(0)
	"	Non-automated Office Frame	HSQRAN.BRPWD01.MANUALS.FRMQnyy
	"	Rate Map for FCM (up to 4)	HSQRAN.BRPWD01.RATES1C.mmmddy
	"	Rate Map for Periodicals (up to 4)	HSQRAN.BRPWD01.RATESPD.mmmddy
	"	Rate Map for STD-A (up to 4)	HSQRAN.BRPWD01.RATESSA.mmmddy
	"	Rate Map for STD-B (up to 4)	HSQRAN.BRPWD01.RATESSB.mmmddy
	OUT	JOB-1 Output	HSQRAN.BRPWD01.EDITDAT1.FYyyyyQn
2	IN	JOB-1 Output	HSQRAN.BRPWD01.EDITDAT1.FYyyyyQn
	"	Rate Map for FCM (up to 4)	HSQRAN.BRPWD01.RATES1C.mmmddy
	"	Rate Map for Periodicals (up to 4)	HSQRAN.BRPWD01.RATESPD.mmmddy
	"	Rate Map for STD-A (up to 4)	HSQRAN.BRPWD01.RATESSA.mmmddy
	"	Rate Map for STD-B (up to 4)	HSQRAN.BRPWD01.RATESSB.mmmddy
	OUT	JOB-2 Output	HSQRAN.BRPWD01.EDITDAT2.FYyyyyQn
3	IN	JOB-2 Output Data	HSQRAN.BRPWD01.EDITDAT2.FYyyyyQn
	"	RPW Code Map	HSQRAN.BRPWD01.RPWLABEL.mmmyy
	"	2-Page Report Map	HSQRAN.BRPWD01.TPLABEL.mmmyy

"	Revenue Control - by AP	HSISMN.RPW.REVCNTL.FYyyyyap
OUT	JOB-3 Output	HSQRAN.BRPWD01.INFLATE1.FyyyyyQn
"	JOB-3 Output - by RPW Code	HSQRAN.BRPWD01.TRANS.FyyyyyQn

5. Report Titles

This section lists the BRPW JOB 1-3 reports by title.

<u>JOB</u>	<u>REPORT No.</u>	<u>REPORT TITLE</u>
No. 1	R1-300/310 R1-400 R1-450	FINNO NOT ON CBCIS FRAME INITIAL EDIT SUMMARY INITIAL EDIT SUMMARY STRATUM TOTALS
No. 2	R2-050A/B R2-100 R2-200 R2-300 R2-400 R2-500 R2-600 R2-800	JOB-1 OUTPUT VERIFICATION UNUSED R-TABLE ENTRIES (NO RPW CODE) EFLAG SUMMARY - BY STRATUM EFLAG SUMMARY - BY VIP CODE EFLAG SUMMARY - BY FINNO F1, F2 FACTORS UNUSED VIP CODES UNINFLATED TOTALS - BY RPW CODE
No. 3	R3-050 R3-200 R3-250 R3-425 R3-475 R3-500 R3-600	JOB-2 OUTPUT VERIFICATION ADJUSTED ESTIMATES STRATUM TOTALS ADJUSTED ESTIMATES - BY PANEL OFFICE ADJUSTED ESTIMATES - BY RPW CODE ESTIMATES BY RPW CODE - BY SYSTEM & AIC ESTIMATES BY 2-PAGE LINE NO.

USPS-LR-J-18

APPENDIX A

BRPW
Computer Program Code

This appendix provides a copy of the IBM JCL operating system and SAS language programming code used by the Postal Service to construct and report estimates of revenue, volume, and weight totals. Record layouts for input and output data files are included in the programming code. Text files containing the machine-readable code for the four jobs comprising the BRPW jobstream are also provided on CD-ROM as part of this library reference. The four job names are:

<u>BRPW JOB</u>	<u>FILE NAME</u>
JOB-1	BRPWJOB1.TXT
JOB-2	BRPWJOB2.TXT
JOB-3	BRPWJOB3.TXT
JOB-4	BRPWJOB4.TXT


```

RDATE=&RDATE1; OUTPUT; RDATE=&RDATE2; OUTPUT;
RDATE=&RDATE3; OUTPUT; RDATE=&RDATE4;
PROC SORT DATA=GLOBALS;          BY RDATE;
      DATA GLOBALS; SET GLOBALS; BY RDATE;
      IF FIRST.RDATE;

```

```

DATA RATES1C;
INFILE RATES1C;
INPUT @1      STAR $1. @;
IF STAR='*' THEN DELETE;
ELSE DO;
      INPUT @1      RDATE  6.
            @8      VIP   $5.
            @9      VIP2345 $4. @8 VIP1 $1. @13 VIPX $1.
            @14     RPWCODE  5.
            @48     AIC    3.
            @79     SYS    $6.
            @86     CLASS  $2.;
      END;
PROC SORT DATA=RATES1C;          BY RDATE;
DATA _NULL_; MERGE RATES1C(IN=A) GLOBALS; BY RDATE;
      IF A=0 THEN ABORT;
DATA RATESPD;
INFILE RATESPD;
INPUT @1      STAR $1. @;
IF STAR='*' THEN DELETE;
ELSE DO;
      INPUT @1      RDATE  6.
            @8      VIP   $5.
            @9      VIP2345 $4. @8 VIP1 $1. @13 VIPX $1.
            @14     RPWCODE  5.
            @77     AIC    3.
            @105    SYS    $6.
            @112    CLASS  $2.;
      END;
PROC SORT DATA=RATESPD;          BY RDATE;
DATA _NULL_; MERGE RATESPD(IN=A) GLOBALS; BY RDATE;
      IF A=0 THEN ABORT;
DATA RATESSA;
INFILE RATESSA;
INPUT @1      STAR $1. @;
IF STAR='*' THEN DELETE;

```

00260000

```

ELSE DO;
  INPUT @1      RDATE  6.
        @8      VIP    $5.
        @9      VIP2345 $4. @8 VIP1  $1. @13 VIPX  $1.
        @14     RPWCODE 5.
        @77     AIC     3.
        @111    SYS    $6.
        @118    CLASS   $2.;
  END;
PROC SORT DATA=RATESSA;                BY RDATE;
DATA _NULL_; MERGE RATESSA(IN=A) GLOBALS; BY RDATE;
  IF A=0 THEN ABORT;
DATA RATESSB;
INFILE RATESSB;
INPUT @1      STAR  $1. @;
IF STAR='*' THEN DELETE;
ELSE DO;
  INPUT @1      RDATE  6.
        @8      VIP    $5.
        @9      VIP2345 $4. @8 VIP1  $1. @13 VIPX  $1.
        @14     RPWCODE 5.
        @77     AIC     3.
        @111    SYS    $6.
        @118    CLASS   $2.;
  END;
PROC SORT DATA=RATESSB;                BY RDATE;
DATA _NULL_; MERGE RATESSB(IN=A) GLOBALS; BY RDATE;
  IF A=0 THEN ABORT;
PROC DELETE DATA=GLOBALS;
DATA SYSMAP; SET RATES1C RATESPD RATESSA RATESSB;
IF RDATE=. THEN DELETE;
*****
* EXPAND MULTI-INDICIA VIPS *
*****;
  IF VIPX=' ' THEN OUTPUT; * OK AS IS;
ELSE IF VIPX='+' THEN DO;
  VIP='1']VIP2345; OUTPUT; *METERED;
  VIP='2']VIP2345; OUTPUT; *STAMPED;
  END;
KEEP RDATE RPWCODE VIP SYS CLASS AIC;
PROC DELETE DATA=RATES1C;

```

```

PROC DELETE DATA=RATESPD;
PROC DELETE DATA=RATESSA;
PROC DELETE DATA=RATESSB;
*****
* EMPTY VIP 44444 *
*****;
DATA SYMAP0; SET SYMAP;
  IF VIP='44444';
*****
***** BEGIN PROGRAM *****
*****;

*****;
* PARAMETER CHECKS *
*****;
DATA _NULL_;
  IF ^(&RUNTYPE='PQ' OR &RUNTYPE='AP') OR
    ^ ( 0<=&FY<=99) OR ^ (1<=&PQ<=4) THEN ABORT;
  IF &RUNTYPE='AP' AND ^ (01<=&AP<=13) THEN ABORT;
  IF &RUNTYPE='PQ' AND &AP ^= . THEN ABORT;
*****
* READ DATA FILES - VERIFY SETTINGS, CHECK DUPS *
*****;
DATA CBCIS; INFILE CBCIS;
  INPUT @1 AP 2.
        @3 PQ 1.
        @4 FY 2.
        @6 FINNO 6.
        @24 RDATE 6.
        @34 VIP $5. @34 VIP1 $1. @35 VIP2 $1. @36 VIP3 $1.
        @39 R 10.2
        @49 P 10.
        @59 W 12.2
        @71 C 10.
        @95 GDEI $1. ; /* DELETE RECORD IF 'Y' */
  IF VIP=' ' THEN DO;
    IF AP=11 & FY= 0 & FINNO=999999 THEN DELETE;
    ELSE IF AP=12 & FY= 0 & FINNO=999999 THEN DELETE;
    ELSE ABORT;
  END;
  IF ^ (RDATE=&RDATE1 OR RDATE=&RDATE2 OR RDATE=&RDATE3

```

```

                                OR RDATE=&RDATE4) THEN ABORT;
IF ^ (10000<=FINNO<=599999)      THEN ABORT;
IF PQ^=&PQ OR FY^=&FY              THEN ABORT; 00260000
IF &RUNTYPE='AP' & AP^=&AP        THEN ABORT; 00260000
IF (PQ=1 & ^ (1<=AP<=3)) OR (PQ=2 & ^ (4<=AP<=6)) OR      00260000
    (PQ=3 & ^ (7<=AP<=9)) OR (PQ=4 & ^ (10<=AP<=13))      THEN ABORT; 00260000
*****                          00260000
* FILTER *                        00260000
*****;                          00260000
IF GDEI=' '; /* IF GDEI='Y' THEN DELETE */                    00260000
IF (VIP1>'9' OR VIP1<'0') OR                                       00260000
    (VIP2>'9' OR VIP2<'0') OR                                       00260000
    (VIP2='2' & VIP1<'6') THEN DELETE;                          00260000
*****                          00260000
* CHECK DUPS *                                                         00260000
*****;                                                                  00260000
PROC SORT DATA=CBCIS; BY AP FINNO RDATE VIP;                      00260000
DATA _NULL_; SET CBCIS; BY AP FINNO RDATE VIP;                    00260000
    IF FIRST.VIP ^=LAST.VIP                                         THEN ABORT; 00260000
DATA MANUALS (DROP=SYSX) MANUALS0; INFILE MANUALS;                00260000
INPUT @1 FINNO 6.                                                  00260000
    @9 FY 2.                                                         00260000
    @11 PQ 1.                                                         00260000
    @12 AP 2.                                                         00260000
    @14 VIP $5. @14 VIP1 $1. @15 VIP2 $1. @16 VIP3 $1.             00260000
    @19 R 10.2                                                       00260000
    @29 C 10.                                                         00260000
    @39 P 10.                                                         00260000
    @49 W 12.2                                                       00260000
    @62 SYSX $6. /*TO EXPAND NEG-RPT 44444 BY AIC/CL W.IN SYS*/ 00260000
    @69 RDATE 6. ;                                                  00260000
IF ^ (RDATE=&RDATE1 OR RDATE=&RDATE2 OR RDATE=&RDATE3
                                OR RDATE=&RDATE4) THEN ABORT;
IF PQ^=&PQ OR FY^=&FY          THEN ABORT; 00260000
IF &RUNTYPE='AP' & AP^=&AP    THEN ABORT; 00260000
IF ^ (10000<=FINNO<=599999)  THEN ABORT;
IF (PQ=1 & ^ (1<=AP<=3)) OR (PQ=2 & ^ (4<=AP<=6)) OR      00260000
    (PQ=3 & ^ (7<=AP<=9)) OR (PQ=4 & ^ (10<=AP<=13))      THEN ABORT; 00260000
IF VIP='44444' THEN OUTPUT MANUALS0;
ELSE OUTPUT MANUALS;
*****                          00260000

```

```

* DUP REC CHECK *                                00260000
*****;                                          00260000
PROC SORT DATA=MANUALS;          BY AP FINNO RDATE VIP; 00260000
  DATA _NULL_; SET MANUALS; BY AP FINNO RDATE VIP;    00260000
  IF FIRST.VIP ^= LAST.VIP THEN ABORT;                00260000
PROC SORT DATA=MANUALS0;        BY AP FINNO SYSX;      00260000
  DATA _NULL_; SET MANUALS0; BY AP FINNO SYSX;        00260000
  IF FIRST.FINNO ^= LAST.FINNO THEN ABORT;            00260000
*****
* VERIFY UNIQUE NEG REPORT *
*****;
PROC SORT DATA=MANUALS;          BY AP FINNO;
  DATA _NULL_; MERGE MANUALS(IN=A) MANUALS0(IN=B); BY AP FINNO;
  IF A=B THEN ABORT;
*****
* READ FRAME FILES - VERIFY RECORDS, NO DUPS. *
*****;
DATA CBCISFRM; INFILE CBCISFRM;                                00260000
  INPUT @5 AP 2.                                               00260000
        @7 PQ 1.                                               00260000
        @8 FY 2.                                               00260000
        @13 FINNO 6.                                           00260000
        @21 CCITY $18.                                          00260000
        @40 CSTATE $2. ;                                        00260000
  IF ^(10000<=FINNO<=599999) THEN ABORT;
  IF PQ^=&PQ OR FY^=&FY THEN ABORT;
  IF &RUNTYPE='AP' AND AP^=&AP THEN ABORT;
  IF &RUNTYPE='PQ' AND
    ^((PQ=1 & 1<=AP<=3) OR (PQ=2 & 4<=AP<=6) OR
      (PQ=3 & 7<=AP<=9) OR (PQ=4 & 10<=AP<=13))
    THEN ABORT; 00260000
    PROC SORT;          BY AP FINNO; 00260000
    DATA _NULL_; SET CBCISFRM; BY AP FINNO; 00260000
    IF FIRST.FINNO ^= LAST.FINNO THEN ABORT; 00260000
DATA MNUALFRM; INFILE MNUALFRM;                                00260000
  INPUT @1 FINNO 6.                                           00260000
        @8 SYS $6. /* SYSTEM (PANEL) IDENTIFIER */ 00260000
        @14 STRATUM 3.                                         00260000
        @18 PNAME $22.                                         00260000
        @40 PSTATE $2.                                         00260000
        @42 PZIP 5.                                           00260000
        @93 BLOWUP 8.                                          00260000

```

```

@101 PNR $1. ; /* PERM NEG-RPT = '*' */ 00260000
AP=&AP;
PQ=&PQ;
FY=&FY;
IF ^(10000<=FINNO<=599999) THEN ABORT;
IF STRATUM<1.0 OR BLOWUP<1.0 THEN ABORT;
PROC SORT; BY SYS FINNO; 00260000
DATA _NULL_; SET MNUALFRM; BY SYS FINNO; 00260000
IF FIRST.FINNO ^= LAST.FINNO THEN ABORT; 00260000
*****
* MERGE CBCIS W. FRAME: TAG FINNO IF NOT ON FRAME: ASSIGN VIP=44444 *
*****;
DATA CBCIS CBCIS0; MERGE CBCIS(IN=A) CBCISFRM(IN=B); BY AP FINNO;
IF A=B THEN OUTPUT CBCIS;
ELSE IF A=0 THEN DO;
VIP='44444'; VIP1='4'; VIP2='4'; VIP3='4';
R=0; P=0; W=0; C=0;
OUTPUT CBCIS0;
END;
ELSE DO;
XFRAME=1; * NO CBCIS FRAME ENTRY (CBCIS DATA N/U);
OUTPUT CBCIS;
END;
PROC DELETE DATA=CBCISFRM;
*****;
* ASSIGN SYS-CLASS-AIC *;
*****;
PROC SORT DATA=CBCIS; BY RDATE VIP;
PROC SORT DATA=SYSMAP; BY RDATE VIP;
DATA CBCIS; MERGE CBCIS(IN=A) SYSMAP(IN=B); BY RDATE VIP;
IF A=1;
* IF B=0 THEN ABORT; * SEE NOTE ABOVE;
*****;
* EXPAND EMPTY VIP 44444 *; *FINNOS ON FRAME - NO ACTIVITY;
*****;
DATA CBCIS0;
SET CBCIS0;
DO I=1 TO K;
SET SYSMAP0 POINT=I NOBS=K;
OUTPUT;
END;

```

```

        IF SYS=' ' OR CLASS=' ' THEN ABORT;
DATA CBCIS; SET CBCIS CBCIS0;
PROC DELETE DATA=CBCIS0;
*****
* MANUALS:  ADD SYS-CLASS-AIC *
*****;
PROC SORT DATA=MANUALS;                                BY RDATE VIP;
DATA MANUALS; MERGE MANUALS(IN=A) SYSMAP(IN=B); BY RDATE VIP;
    IF A=1;
    IF B=0 THEN ABORT;
    IF SYS=' ' OR CLASS=' ' THEN ABORT;
    *****;
    * EXPAND EMPTY VIP 44444 *;*SINGLE ENTRY 44444 EXPANDED BY AIC,CL;
    *****;
DATA MANUALS0;
    SET MANUALS0;
    DO I=1 TO K;
        SET SYSMAP0 POINT=I NOBS=K;
        OUTPUT;
    END;
    IF SYS=' ' OR CLASS=' ' THEN ABORT;
DATA MANUALS0; SET MANUALS0;
    IF SYSX=SYS; * RETAIN VIP=44444 FOR MEMBER PANEL ONLY;
DATA MANUALS; SET MANUALS MANUALS0;
PROC DELETE DATA=MANUALS0;
*****
* MERGE CBCIS, MANUAL & PANEL MASTER (BY AP). *
*****;
* WITHIN SYSTEM - FOR FINNOS COMMON TO 2 OR MORE PANELS;
*****
* EXPAND MANUAL PANEL FRAME *
*****;
DATA MNUALFRM; SET MNUALFRM;
    IF &RUNTYPE='AP' THEN DO; AP=&AP; OUTPUT; END;
    ELSE IF &PQ=1 THEN DO;
        AP=1; OUTPUT; AP=2; OUTPUT; AP=3; OUTPUT; END;
    ELSE IF &PQ=2 THEN DO;
        AP=4; OUTPUT; AP=5; OUTPUT; AP=6; OUTPUT; END;
    ELSE IF &PQ=3 THEN DO;
        AP=7; OUTPUT; AP=8; OUTPUT; AP=9; OUTPUT; END;
    ELSE IF &PQ=4 THEN DO;

```

```

        AP=10; OUTPUT; AP=11; OUTPUT; AP=12; OUTPUT;
        AP=13; OUTPUT; END;
PROC SORT DATA=MNUALFRM;                BY FINNO SYS AP;
PROC SORT DATA=MANUALS;                 BY FINNO SYS AP;
PROC SORT DATA=CBCIS;                   BY FINNO SYS AP;
DATA ABCTABLE;
  * NOTE: MANUAL, CBCIS MATCHED TO ORIG PANEL FRAME (NOT CBCIS FRAME);
  * NOTE: RECORDS FOR UNASSIGNED PANEL (SYSTEM) ARE FLAGGED (ABORT);
  MERGE MNUALFRM(IN=A                      )
        MANUALS(IN=B  KEEP=AP PQ FY FINNO SYS  )
        CBCIS(IN=C  KEEP=AP PQ FY FINNO SYS XFRAME);
        BY FINNO SYS AP;

  IF FIRST.AP;
  *****
  * INITIALIZE MIGRATE & NRESP (MANUALS) VARS *
  *****;
  MIGRATE=0;
  NRESP=0;
  *****
  * ASSIGNMENT TABLE: ABC=000...111 (BY AP) *
  *****;
  ABC=(100*A)+(10*B)+C;
    IF ABC=100 THEN NRESP=1;  *VAR USED FOR MANUALS ONLY;
  ELSE IF ABC=001 THEN DO; STRATUM=1.0; BLOWUP=1; END;
  ELSE IF ABC=010 OR /*FLAG IF REC IS FOR UNASSIGNED PANEL */
    ABC=011 THEN ABORT;
  ELSE IF ABC=111 & STRATUM<2.0 OR
    ABC=110 & STRATUM<2.0 THEN ABORT;
  ELSE IF ABC=111 & STRATUM>=2.0 OR
    ABC=101 & STRATUM>=2.0 THEN DO;
    IF XFRAME^=1 THEN MIGRATE=1; * CBCIS DATA USED;
    ELSE                MIGRATE=0; * MANUAL DATA USED;
  END;
  IF STRATUM=. & (ABC^=001) THEN ABORT;
PROC DELETE DATA=MNUALFRM;
*****
* MERGE "ABC" BY SYSTEM, REMOVE MANUAL MIGRATES *
*****;
DATA EDITDAT0; SET CBCIS(IN=A) MANUALS; /* VERTICAL */
  IF A=1 THEN SOURCE='CBCIS '; * USED FOR MIGRATION;
  ELSE        SOURCE='MANUAL';

```

```

PROC DELETE DATA=CBCIS;
PROC DELETE DATA=MANUALS;
PROC SORT;                                BY FINNO SYS AP;
PROC SORT DATA=ABCTABLE;                 BY FINNO SYS AP;
DATA EDITDAT1; MERGE EDITDAT0(IN=A) ABCTABLE; BY FINNO SYS AP;
RUNTYPE=&RUNTYPE;
EFLAG=0000;
IF SOURCE='CBCIS' & XFRAME=1 THEN EFLAG=0100;
IF SOURCE='MANUAL' & MIGRATE=1 THEN EFLAG=0050;
PROC DELETE DATA=EDITDAT0;
PROC DELETE DATA=ABCTABLE;
*****
* OUTPUT *
*****;
PROC SORT; BY FINNO VIP AP;
DATA _NULL_; SET EDITDAT1;
FILE EDITDAT1;
PUT @1     RUNTYPE  $2.
   @3     CLASS    $2.
   @5     SYS      $6.
   @11    AP       Z2.
   @13    PQ       1.
   @14    FY       2.
   @16    RDATE    Z6.
   @22    FINNO    Z6.
   @28    VIP      $5.
   @33    RPWCODE  Z5.
   @38    R        Z12.2
/* @50-61 */
   @62    P        Z12.
   @74    C        Z12.
   @86    W        Z14.2
   @100   NRESP    1. /* MANUALS ONLY */
   @101   MIGRATE  1.
   @102   EFLAG    Z4.
/* @106   */
   @107   STRATUM  Z3.1
   @110   BLOWUP   Z8.3
   @118   AIC      3. ;

```

00260000

```

* 6.0 REPORTS *                                00260000
*****;                                         00260000
DATA E100; SET EDITDAT1;
  IF EFLAG=0100;
PROC SORT DATA=E100; BY FINNO AP;
PROC SUMMARY DATA=E100; BY FINNO AP;
  VAR R P W C; OUTPUT OUT=E100A SUM=;
  PROC PRINT U;
    FORMAT R P W C COMMA13.;                    000
    VAR FINNO AP _FREQ_ R P W C;
    SUM R P W C;
  TITLE1 "JOB1 ** RPW BULK MAIL SYSTEM ** FOR &RUNTYPE&AP PQ&PQ-&FY";00260000
  TITLE3 'R1-300: CBCIS FINNO NOT ON CBCIS FRAME';
  TITLE4 '(DATA NOT USED - RAW TOTALS SHOWN)';
  TITLE5 'BY FINNO & AP';                       000
  TITLE6 ' ';                                    000
  PROC DELETE DATA=E100A;
PROC SORT DATA=E100; BY SYS CLASS FINNO;
PROC SUMMARY DATA=E100; BY SYS CLASS FINNO;
  VAR R P W C; OUTPUT OUT=E100B SUM=;
  PROC SUMMARY DATA=E100B; BY SYS CLASS;
  VAR R P W C; OUTPUT OUT=E100B SUM=;
  PROC PRINT U; BY SYS CLASS;
    FORMAT R P W C COMMA13.;                    000
    VAR _FREQ_ R P W C;
    SUM R P W C;
    TITLE3 'R1-310: CBCIS FINNO NOT ON CBCIS FRAME';
    TITLE4 '(DATA NOT USED - RAW TOTALS SHOWN)';
    TITLE5 'BY SYSTEM & CLASS';                 000
    TITLE6 ' ';                                    000
  PROC DELETE DATA=E100B;
DATA NRESP RESP; SET EDITDAT1;                  00260000
  IF NRESP=1 THEN OUTPUT NRESP;                 00260000
  ELSE OUTPUT RESP;                             00260000
  PROC DELETE DATA=EDITDAT1;                   00260000
PROC SORT DATA=NRESP; BY EFLAG SYS;           00260000
PROC PRINT DATA=NRESP; BY EFLAG SYS;          00260000
  VAR NRESP EFLAG SYS CLASS AP STRATUM FINNO R P W C; 00260000
  TITLE3 'R1-400: INITIAL EDIT SUMMARY';        002
  TITLES 'NONRESPONSES';                       002
  TITLE6 ' ';                                    002

```

```
PROC DELETE DATA=NRESP;
PROC SORT DATA=RESP; BY SYS CLASS EFLAG AP STRATUM;
PROC SUMMARY DATA=RESP; BY SYS CLASS EFLAG AP STRATUM;
VAR R P W C; OUTPUT OUT=DATISUMS SUM=;
PROC DELETE DATA=RESP;
PROC PRINT DATA=DATISUMS U; BY SYS CLASS; PAGEBY SYS;
FORMAT R P W C COMMA13.;
ID SYS;
VAR CLASS EFLAG AP STRATUM R P W C;
SUM
TITLE3 'R1-450: INITIAL EDIT SUMMARY';
TITLE5 'STRATUM TOTALS (ALL RECS) - BY ERROR FLAG';
TITLE6 ' ';
PROC DELETE DATA=DATISUMS;
```

/*

```

//NNNNNNS JOB (ALA03), 'NNNNNNN NNN NN', CLASS=H, MSGCLASS=H          00010008
/*ROUTE PRINT U5704                                                    00010000
/*****00040000
/*          JOB NAME:  PDS.SASRPW(RPWJOB2) - FORMERLY "J2RRUN97" USED  00060000
/*****00060000
/*                                                    00110008
//SO1      EXEC SAS, REGION=6000K, TIME=60                             00110008
/**ORK     DD SPACE=(CYL, (1500, 1500), RLSE), UNIT=(SYSDA, 3)         00200000
//WORK     DD SPACE=(CYL, (4000)), UNIT=(SYSDA, 3)                       00200000
//SYSOUT   DD DUMMY                                                    00200000
/*                                                    00200000
/*                                                    00200000
/** \\\\// PQ RUN: JOB-1 OUTPUT LAST QUALIFIER:  FY1996QX: X=1...4.    00200000
/** \\\\// AP RUN: JOB-1 OUTPUT LAST QUALIFIER:  FY1996XX: XX=01..13.  00200000
/**                                                    00200000
//EDITDAT1 DD DSN=HSQRAN.BRPWD01.EDITDAT1.FY2000Q4, DISP=SHR          00200000
/** \\\\// PQ/AP RUN:  READ UP TO 4 RATES FILES FOR EACH CLASS.        00200000
/**          *** IMPORTANT: READ 1 FILE FOR EACH GLOBAL R-DATE.        00200000
//RATES1C  DD DSN=HSQRAN.BRPWD01.RATES1C.JAN1099, DISP=SHR
//          DD DSN=HSQRAN.BRPWD01.RATES1C.OCT0498, DISP=SHR
//          DD DSN=HSQRAN.BRPWD01.RATES1C.OCT0597, DISP=SHR
//RATESPD  DD DSN=HSQRAN.BRPWD01.RATESPD.JAN1099, DISP=SHR            00200000
//          DD DSN=HSQRAN.BRPWD01.RATESPD.OCT0498, DISP=SHR            00200000
//          DD DSN=HSQRAN.BRPWD01.RATESPD.OCT0597, DISP=SHR            00200000
//RATESSA  DD DSN=HSQRAN.BRPWD01.RATESSA.JAN1099, DISP=SHR            00200000
//          DD DSN=HSQRAN.BRPWD01.RATESSA.OCT0498, DISP=SHR            00200000
//          DD DSN=HSQRAN.BRPWD01.RATESSA.OCT0597, DISP=SHR            00200000
//RATESSB  DD DSN=HSQRAN.BRPWD01.RATESSB.JAN1099, DISP=SHR            00200000
//          DD DSN=HSQRAN.BRPWD01.RATESSB.OCT0498, DISP=SHR            00200000
//          DD DSN=HSQRAN.BRPWD01.RATESSB.OCT0597, DISP=SHR            00200000
/**                                                    00200000
/** \\\\// PQ RUN: JOB-2 OUTPUT LAST QUALIFIER:  FY1996QX: X=1...4.    00200000
/** \\\\// AP RUN: JOB-2 OUTPUT LAST QUALIFIER:  FY1996XX: XX=01..13.  00200000
/**                                                    00200000
/**DITDAT2 DD DSN=HSQRAN.BRPWD01.EDITDAT2.FY2000Q4, DISP=SHR          00200000
/**DITDAT2 DD DSN=HSQRAN.BRPWD01.EDITDAT2.FY2000Q4,
/**          DISP=(NEW, CATLG), DCB=(RECFM=FB, LRECL=150, BLKSIZE=6000),  00200000
/**          UNIT=SYSDA, SPACE=(CYL, (1000, 750), RLSE)                00200000
/**                                                    00200000
//SYSIN    DD *                                                         00200000
                                                    00210000

```



```

EFLAG=3100; ETYPE='WEIGHT-TOLERANCE----'; OUTPUT;          00210000
EFLAG=.; ETYPE='MISSING-EFLAG-VALUE-'; OUTPUT;             00210000
PROC SORT; BY EFLAG;                                       00210000
*****
* READ EDITDAT1 - VERIFY PARAMETERS - SPLIT INTO CLASS MODULES *
*****;
DATA EDITDAT1; INFILE EDITDAT1;                             00210000
  INPUT @1      RUNTYPE $2.
         @3      CLASS $2.
         @5      SYS $6.
         @11     AP 2.
         @13     PQ 1.
         @14     FY 2.
         @16     RDATE 6.
         @22     FINNO 6.
         @28     VIP $5. @28 VIP1 $1. @29 VIP2 $1.
                   @28 VIP12 $2. @30 VIP3 $1.
                   @30 VIP34 $2. @31 VIP45 $2.
                   @32 VIP5 $1. @29 VIP2345 $4.
                   @28 VIP1234 $4.
         @33     RPWCODE 5.
         @38     R 12.
/* @50-61 */
         @62     P 12.
         @74     C 12.
         @86     W 14.
         @100    NRESP 1. /* MANUALS ONLY */
         @101    MIGRATE 1.
         @102    EFLAG 4.
/* @106 */
         @107    STRATUM 3.
         @110    BLOWUP 8.
         @118    AIC 3. ;
*****
* TEMP FIX * * AP2-00 FOR FINNO 518682;
*****;
IF FINNO=518682 & AP=2 & FY=00 THEN DO;
  IF VIP='04352' THEN DO;
    RHOLD=R; PHOLD=P; WHOLD=W;
    DELETE;
    END;

```

```

IF VIP='04412' THEN DO;
  R=R+RHOLD; P=P+PHOLD; W=W+WHOLD;
  END;
END;
RETAIN RHOLD PHOLD WHOLD;
DROP  RHOLD PHOLD WHOLD;
*****;
* TEMP END *;
*****;
IF NRESP^=1 & (VIP=' ' OR VIP1=' ' OR VIP1='.') THEN ABORT;
IF RUNTYPE^=&RUNTYPE      THEN ABORT;
IF PQ^=&PQ OR FY^=&FY      THEN ABORT;
IF &RUNTYPE='AP' & AP^=&AP THEN ABORT;          00210000
IF ^(10000<=FINNO<=599999) THEN ABORT;
IF ^(RDATE=&RDATE1 OR RDATE=&RDATE2 OR RDATE=&RDATE3
      OR RDATE=&RDATE4) &
^(VIP='44444' OR NRESP=1 OR EFLAG^=0) THEN ABORT;
*****          00210000
* DUP REC CHECK * *DUPS OK IF EFLAG DIFFERS (=0050,0100); 00210000
*****          00210000
PROC SORT;          00210000
  BY AP SYS CLASS AIC FINNO RDATE VIP DESCENDING EFLAG; 00210000
DATA _NULL_; SET EDITDAT1;          00210000
  BY AP SYS CLASS AIC FINNO RDATE VIP DESCENDING EFLAG; 00210000
  IF FIRST.VIP ^= LAST.VIP THEN DO;          00210000
    IF FIRST.VIP THEN HOLDFLAG=EFLAG;          00210000
    IF LAST.VIP & EFLAG=HOLDFLAG THEN ABORT; 00210000
  END;          00210000
  RETAIN HOLDFLAG;          00210000
*****          00210000
* SPLIT INTO CLASS MODULES *          00210000
*****          00210000
DATA MOD0C MOD1C MODPD MODSA MODSB; SET EDITDAT1; 00210000
  IF NRESP =1      OR          002
    EFLAG ^=0     OR          002
    VIP ='44444' THEN OUTPUT MOD0C;          002
ELSE IF CLASS =' ' THEN DO;          002
  EFLAG=1000;          OUTPUT MOD0C;          002
END;          002
ELSE IF CLASS='1C' THEN OUTPUT MOD1C;          002
ELSE IF CLASS='PD' THEN OUTPUT MODPD;          002

```

```

ELSE IF CLASS='SA' THEN OUTPUT MODSA; 002
ELSE IF CLASS='SB' THEN OUTPUT MODSB; 002
ELSE ABORT; 002
PROC DELETE DATA=EDITDAT1; 002
*****
* RPT R2-050: JOB-1 VERIFICATION * * FOR AUDITOR;
*****;
DATA R050; SET MOD0C(IN=A) MOD1C MODPD MODSA MODSB;
IF VIP^=' ';
IF A=1 THEN RETAIN='N';
ELSE RETAIN='Y';
PROC SORT; BY SYS CLASS DESCENDING RETAIN
STRATUM EFLAG AP;
PROC SUMMARY; BY SYS CLASS DESCENDING RETAIN
STRATUM EFLAG AP;
ID BLOWUP; VAR R P W; OUTPUT OUT=R050A SUM=;
PROC PRINT DATA=R050A; BY SYS CLASS DESCENDING RETAIN;
PAGEBY SYS;
ID RETAIN;
VAR SYS CLASS STRATUM EFLAG AP R P W;
SUM R P W;
FORMAT R P W COMMA14.;
TITLE1 "JOB2 ** RPW BULK MAIL SYSTEM ** FOR &RUNTYPE&AP PQ&PQ-&FY";
TITLE3 'R2-050A: JOB-1 OUTPUT VERIFICATION';
TITLE5 '(RETAIN=N DATA UNUSED DOWNSTREAM)';
TITLE7 'SUMMARY';
TITLE9 ' ';
DATA R050B; SET R050;
IF RETAIN='N';
PROC SORT; BY SYS CLASS STRATUM EFLAG RDATE VIP;
PROC SUMMARY; BY SYS CLASS STRATUM EFLAG RDATE VIP;
ID BLOWUP RETAIN;
VAR R P W; OUTPUT OUT=R050B SUM=;
PROC PRINT DATA=R050B; BY SYS CLASS; PAGEBY SYS;
ID RETAIN;
VAR SYS CLASS STRATUM BLOWUP EFLAG RDATE VIP
R P W;
SUM R P W;
FORMAT R P W COMMA14.;
TITLE3 'R2-050B: JOB-1 OUTPUT VERIFICATION';
TITLE5 '(RETAIN=N DATA UNUSED DOWNSTREAM)';

```

```

        TITLE7 'DETAIL FOR RETAIN=N';
        TITLE9 ' ';
        PROC DELETE DATA=R050;
        PROC DELETE DATA=R050S;
        PROC DELETE DATA=R050A;
        PROC DELETE DATA=R050B;
*****
* RATE TABLES (VIP-RPW CODE) *
*****;
DATA RATES1C; INFILE RATES1C;
  LENGTH VIPCAT $48.;
  RTABLE='RATES1C';
  INPUT @1      STAR $1.  @;
  IF STAR='*' THEN DELETE;
  ELSE DO;
    INPUT @1      RDATE  6.
          @8      VIP    $5.
          @9      VIP2345 $4. @8 VIP1 $1. @13 VIPX $1.
          @14     RPWCODE  5.
          @20     VIPCAT $27.
          @48     AIC     3.
          @52     RATEHI  7.
          @60     RATELO  7.
          @68     WPPMAX  9.
          @79     SYS    $6.
          @89     ID     $2. ;
    IF RDATE=. THEN ABORT;
    SPSERV=0;
    IF ID='SS' THEN SPSERV=1;
    IF RPWCODE>0 THEN DO;
      WPPMAX=WPPMAX/16; * CONVERT OZ TO LBS;
      IF WPPMAX<=0 OR RATEHI<=0 OR RATELO< 0 THEN ABORT;
    END;
*****
* EXPAND MULTIPLE INDICIA VIPS *
*****;
    IF VIPX=' ' THEN OUTPUT; /*OK AS IS*/
  ELSE IF VIPX='+' THEN DO;
    VIP='1']VIP2345; VIP1='1'; OUTPUT; *METERED;
    VIP='2']VIP2345; VIP1='2'; OUTPUT; *STAMPED;
  END;

```

```

END;
DROP STAR VIPX;
DATA RATESPD; INFILE RATESPD;
RTABLE='RATESPD';
INPUT @1 STAR $1. @;
IF STAR='*' THEN DELETE;
ELSE DO;
    INPUT @1 RDATE 6.
           @8 VIP $5.
           @14 RPWCODE 5.
           @20 VIPCAT $48.
           @77 AIC 3.
           @81 PCRT 7.
           @89 LBRT 7.
           @97 WPPMAX 7.
           @105 SYS $6.
           @115 ID $2. ;
    IF RDATE=. THEN ABORT;
    IF ID='2D' THEN DISCOUNT='Y';
    ELSE IF ID='2N' THEN DO; DISCOUNT='Y'; DISCNADV='Y'; END;
    ELSE IF ID='2I' THEN DO; DISCOUNT='Y'; FOREIGN='Y'; END;
    ELSE IF ID='2F' THEN FOREIGN='Y';
    ELSE IF ID='2K' THEN KEYRATE='Y';
    ELSE IF ID='2S' THEN SCHARGE='Y';
    OUTPUT RATESPD;
END;
DROP STAR;
DATA RATESSA; INFILE RATESSA;
RTABLE='RATESSA';
INPUT @1 STAR $1. @;
IF STAR='*' THEN DELETE;
ELSE DO;
    INPUT @1 RDATE 6.
           @8 VIP $5.
           @9 VIP2345 $4. @8 VIP1 $1. @13 VIPX $1.
           @14 RPWCODE 5.
           @20 VIPCAT $42.
           @77 AIC 3.
           @81 PCRT 7.
           @89 LBRT 7.
           @97 WPPMIN 6.

```

```

        @104  WPPMAX  6.
        @111  SYS    $6.
        @121  ID     $2.
        @124  MVIP   $5. ;
IF RDATE=. THEN ABORT;
SINGLE=0; PARTIAL=0; MULTI=0;
  IF ID='SP' THEN SINGLE =1;
ELSE IF ID='PP' THEN PARTIAL=1;
ELSE IF ID='MV' THEN MULTI=1; /*PIECE & POUND RATE PAIR*/
IF PCRT=. THEN PCRT=0;
IF LBRT=. THEN LBRT=0;
*****
* EXPAND MULTIPLE INDICIA VIPS * /* VIPS WITH VIP1=. */
*****;
  IF VIPX=' ' THEN OUTPUT; /*OK AS IS*/
ELSE IF VIPX='+' THEN DO;
  VIP='1']VIP2345; VIP1='1'; OUTPUT; *METERED;
  VIP='2']VIP2345; VIP1='2'; OUTPUT; *STAMPED;
END;
END;
DROP STAR VIPX;
DATA RATESSB; INFILE RATESSB;
LENGTH VIPCAT $48.;
RTABLE='RATESSB';
INPUT @1      STAR $1. @;
IF STAR='*' THEN DELETE;
ELSE DO;
  INPUT @1      RDATE  6.
        @8      VIP    $5.
        @9      VIP2345 $4. @8 VIP1 $1. @13 VIPX $1.
        @14     RPWCODE  5.
        @20     VIPCAT   $25.
        @52     RATEHI   7. /* PARCEL POST */
        @60     RATELO   7. /* PARCEL POST */
        @77     AIC      3.
        @81     PCRT     7.
        @89     LBRT     7.
        @97     WPPMIN   6.
        @104    WPPMAX   6.
        @111    SYS     $6.
        @121    ID      $2.

```

```

    @124      MVIP $5. ;
IF RDATE=. THEN ABORT;
SPSERV=0; SINGLE=0; MULTI=0;
    IF ID='DC' THEN DISCOUNT='Y';
ELSE IF ID='SS' THEN SPSEV=1;
ELSE IF ID='SP' THEN SINGLE=1; /* NOTE: ALSO PC-RT M&PCS */
ELSE IF ID='MV' THEN MULTI=1;
IF PCRT=. THEN PCRT=0;
IF LBRT=. THEN LBRT=0;
IF RATEHI=. THEN RATEHI=0;
IF RATELO=. THEN RATELO=0;
IF ABS(PCRT+LBRT)>0 THEN RHILO=0;
ELSE RHILO=1;
*****
* EXPAND MULTIPLE INDICIA VIPS *
*****
    IF VIPX=' ' THEN OUTPUT; /*OK AS IS*/
ELSE IF VIPX='+' THEN DO;
    VIP='1']]VIP2345; VIP1='1'; OUTPUT; *METERED;
    VIP='2']]VIP2345; VIP1='2'; OUTPUT; *STAMPED;
END;

END;
DROP STAR VIPX;
*****
* REPORT R2-100: EMPTY RPWCODE *
*****;
DATA RATESX; SET RATES1C RATESPD RATESSA RATESSB;
    IF RPWCODE=. ;
    IF VIP^='44444';
PROC SORT; BY RTABLE RDATE VIP;
PROC PRINT U; BY RTABLE RDATE; PAGEBY RDATE;
VAR RDATE VIP VIPCAT;
TITLE3 'R2-100: UNUSED R-TABLE ENTRIES (NO RPW CODE)';
TITLE5 ' ';
PROC DELETE DATA=RATESX;
*****
* 1C MODULE: *
*****;
PROC SORT DATA=MOD1C; BY RDATE VIP RPWCODE AIC;
PROC SORT DATA=RATES1C; BY RDATE VIP RPWCODE AIC;
DATA MOD1C RATE1CNU;

```

```

MERGE MOD1C(IN=A) RATES1C(IN=B); BY RDATE VIP RPWCODE AIC;
      IF B=0                      THEN EFLAG=1000;
ELSE IF RPWCODE=.                THEN EFLAG=1100;
      IF EFLAG=1100 OR A=0 THEN OUTPUT RATE1CNU;
ELSE                              OUTPUT MOD1C;
DATA MOD1C; SET MOD1C;
RP=R; RW=0;
IF EFLAG=0 THEN DO;
  IF SPSERV=1 THEN DO;
    IF P=0 OR P=. OR RP=0 OR RP=. THEN EFLAG=2000;
  END;
ELSE DO;
  IF P=0 OR P=. OR W=0 OR W=. OR RP=0 OR RP=.
    THEN EFLAG=2000;
  END;
DROP R;
END;
*****
* RATIOS (VIP LEVEL) *
*****;
RPP=0; RLB=0;
IF P^=0 & P^=. THEN DO;
  RPP=RP/P; WPP=ABS(W/P);
  END;
*****
* EDITS (HIERARCHICAL): *
* 1. MISSING DATA: 2. CLASS-LEVEL: 3. VIP-LEVEL. *
*****;
IF EFLAG=0 THEN DO;
  IF RATEHI<=0 OR RATELO< 0      THEN EFLAG=1200;
  ELSE IF RPP=0 OR RPP=.        THEN EFLAG=2500;
  ELSE IF ^((RATELO*(1-&RTOL)) <=RPP <=(RATEHI*(1+&RTOL)))
    THEN EFLAG=3000;
  ELSE IF WPP>=WPPMAX*(1+&WTOL) THEN EFLAG=3100;
END;
*****
* ADD ETYPE VAR *
*****;
PROC SORT DATA=MOD1C;          BY EFLAG;
DATA MOD1C; MERGE MOD1C(IN=A) ETYPE; BY EFLAG;
  IF A=1;

```

```

PROC DELETE DATA=RATES1C;
*****
* PD MODULE:          *
*****;
PROC SORT DATA=MODPD;          BY RDATE VIP RPWCODE AIC;
PROC SORT DATA=RATESPD;       BY RDATE VIP RPWCODE AIC;
DATA MODPD RATEPDNU;
MERGE MODPD(IN=A) RATESPD(IN=B); BY RDATE VIP RPWCODE AIC;
  IF B=0                      THEN EFLAG=1000;
ELSE IF RPWCODE=.             THEN EFLAG=1100;
  IF EFLAG=1100 OR A=0 THEN OUTPUT RATEPDNU;
ELSE                          OUTPUT MODPD;
DATA MODPD; SET MODPD;
RP=0; RW=0;
*****
* ERROR FLAG *
*****;
IF EFLAG=0 THEN DO;
  IF SCHARGE ='Y' THEN DO;
    RP=R;
    PDISC=C; * RETAIN VOLUME FOR RATE CHECK SECTION;
    P=0;     * ENSURE ZERO (NO DOUBLECOUNT);
    IF (RP=0 OR RP=.) OR
       (PDISC=0 OR PDISC=.) OR
       (W=0 OR W=.) THEN EFLAG=2000;
    ELSE IF P^=0 THEN EFLAG=2100;
  END;
ELSE IF DISCOUNT='Y' & FOREIGN ^= 'Y' THEN DO;
  RP=-R; * NEEDED SINCE CBCIS PD DISC. NOT SIGNED;
  PDISC=P; * RETAIN VOL FOR RATE CHECK SECTION;
  P=0; * ZERO OUT PIECES (AVOID DOUBLECOUNT);
  IF RP=0 OR RP=. THEN EFLAG=2000;
  ELSE IF RP*PDISC ^<0 /* OPPOSITE SIGN CHECK */
  OR P^=0 OR W^=0 THEN EFLAG=2100;
END;
ELSE IF DISCOUNT='Y' & FOREIGN = 'Y' THEN DO;
  RW=-R; * NEEDED SINCE CBCIS PD DISC. NOT SIGNED;
  IF RW=0 OR RW=. OR W=0 OR W=. THEN EFLAG=2000;
  ELSE IF P^=0 OR C^=0 THEN EFLAG=2100;
END;
ELSE IF FOREIGN='Y' THEN DO;

```

```

RW=R;
P=C; /* COPIES = PIECES FOR PUB. PERIODICALS RATE */
IF RW=0 OR RW=. OR W=0 OR W=. OR P=0 OR P=. OR C=.
THEN EFLAG=2000;

END;
ELSE IF KEYSRATE='Y' THEN DO;
RW=R;
IF RW=0 OR RW=. OR W=0 OR W=. THEN EFLAG=2000;
ELSE IF P^=0 THEN EFLAG=2100;
END;
ELSE IF (R=0 OR R=.) OR (P=0 OR P=.) & (W=0 OR W=.) THEN DO;
RP=R;
EFLAG=2000;
END;
ELSE IF P>0 & W>0 THEN DO;
RP=R;
EFLAG=2100;
END;
ELSE IF P^=0 & P^=. THEN RP=R;
ELSE IF W^=0 & W^=. THEN RW=R;

END;
ELSE RP=R; * ALL OTHER CASES;
DROP R;
*****
* ADD RATIOS (VIP LEVEL) *
*****;
RPP=0; RLB=0;
IF P^=0 & P^=. THEN CPP=ABS(C/P);
IF FOREIGN='Y' & DISCOUNT='Y' & W^=0 & W^=. THEN RLB=RW/W;
ELSE IF FOREIGN='Y' & W^=0 & W^=. THEN RLB=RW/W;
ELSE IF DISCOUNT='Y' & ABS(PDISC) >0 THEN RPP=ABS(RP/PDISC);
ELSE IF SCHARGE='Y' & ABS(PDISC) >0 THEN DO;
RPP=ABS(RP/PDISC);
WPP=ABS(W/PDISC);
END;
ELSE IF P^=0 & P^=. THEN RPP=RP/P;
ELSE IF W^=0 & W^=. THEN RLB=RW/W;
*****
* ADD RATIOS (RPW CODE LEVEL) * * NOT VIP LEVEL (FOR PD ONLY!);
*****; * PASSED RECORDS ONLY USED!;
PROC SORT DATA=MODPD; BY AP FINNO RPWCODE RDATE EFLAG;

```

```

PROC SUMMARY DATA=MODPD;          BY AP FINNO RPWCODE RDATE EFLAG;
VAR C P W; OUTPUT OUT=WPPSUMPD SUM=CSUM PSUM WSUM;
DATA WPPSUMPD; SET WPPSUMPD;
    IF EFLAG=0;
    DROP EFLAG;
DATA MODPD; MERGE MODPD WPPSUMPD; BY AP FINNO RPWCODE RDATE;
WPP=0; WPC=0;
IF WSUM^=0 & PSUM^=0 THEN WPP=ABS(WSUM/PSUM);
IF WSUM^=0 & CSUM^=0 THEN WPC=ABS(WSUM/CSUM);
*****
* EDITS (HIERARCHICAL):          *
* 1. MISSING DATA: 2. CLASS-LEVEL: 3. RPW LEVEL *
*****;
DATA MODPD; SET MODPD;
    IF EFLAG=0 THEN DO;
        IF DISCOUNT='Y' & FOREIGN ^= 'Y' THEN DO;
            IF PCRT<=0 OR RPP=0 OR RPP=. OR
            DISCNADV='Y' & ^(PCRT/100<=RPP<=PCRT*(1+&RTOL)) OR
            DISCNADV^='Y' & PCRT>0 & ABS(RPP-PCRT)/PCRT ^<=&RTOL
                THEN EFLAG=3000;
        END;
    ELSE IF DISCOUNT='Y' & FOREIGN='Y' THEN DO;
        IF ABS(RLB-LBRT)/LBRT ^<&RTOL THEN EFLAG=3000;
    END;
    ELSE IF FOREIGN='Y' THEN DO;
        IF ABS(RLB) >LBRT*(1-&RTOL) THEN EFLAG=3000;
    END;
    ELSE IF KEYRATE='Y' THEN DO;
        IF RLB >LBRT*(1+&RTOL) OR RLB<=0
            THEN EFLAG=3000;
        END;
    ELSE IF PCRT<=0 & LBRT<=0 THEN EFLAG=1200;
    ELSE IF RPP+RLB=0 OR RPP+RLB=. THEN EFLAG=2500;
/*
    ELSE IF WPP >WPPMAX*(1+&WTOL) OR
        WPC >WPPMAX*(1+&WTOL) THEN EFLAG=3100;
*/
    ELSE IF PCRT>0 THEN DO;
        IF ABS(RPP-PCRT)/PCRT ^<&RTOL THEN EFLAG=3000;
    END;

```

```

ELSE IF LBRT>0 THEN DO;
  IF ABS(RLB-LBRT)/LBRT ^<&RTOL THEN EFLAG=3000;
  END;

END;

*****;
*****;
*****;
* TEMP *;
*****;
*****;
  IF RDATE=011099 & ('7'<=VIP1<='8') &
    EFLAG=3000 & ((RPP>PCRT) OR
      (DISCOUNT='Y' & ABS(RPP)>PCRT) OR
      (RLB>LBRT)) THEN DO;
    IF VIP34='11' & ABS(RLB/LBRT) <=1.03+&RTOL THEN EFLAG=0;
    ELSE IF VIP34='13' & ABS(RPP/PCRT) <=1.17+&RTOL THEN EFLAG=0;
    ELSE IF VIP34='14' & ABS(RPP/PCRT) <=1.23+&RTOL THEN EFLAG=0;
    ELSE IF VIP34='15' & ABS(RPP/PCRT) <=1.21+&RTOL THEN EFLAG=0;
    ELSE IF VIP34='16' & ABS(RPP/PCRT) <=1.22+&RTOL THEN EFLAG=0;
    ELSE IF VIP34='17' & ABS(RPP/PCRT) <=1.28+&RTOL THEN EFLAG=0;
    ELSE IF VIP34='18' & ABS(RPP/PCRT) <=1.16+&RTOL THEN EFLAG=0;
    ELSE IF VIP34='19' & ABS(RPP/PCRT) <=1.08+&RTOL THEN EFLAG=0;
    ELSE IF VIP34='20' & ABS(RPP/PCRT) <=1.10+&RTOL THEN EFLAG=0;
    ELSE IF VIP34='21' & ABS(RPP/PCRT) <=1.04+&RTOL THEN EFLAG=0;
    ELSE IF VIP34='22' & ABS(RPP/PCRT) <=1.08+&RTOL THEN EFLAG=0;
    ELSE IF VIP34='23' & ABS(RPP/PCRT) <=1.10+&RTOL THEN EFLAG=0;
    ELSE IF VIP34='24' & ABS(RPP/PCRT) <=1.12+&RTOL THEN EFLAG=0;
    ELSE IF VIP34='26' & ABS(RPP/PCRT) <=1.34+&RTOL THEN EFLAG=0;
    ELSE IF VIP34='27' & ABS(RPP/PCRT) <=1.86+&RTOL THEN EFLAG=0;
    ELSE IF VIP34='28' & ABS(RPP/PCRT) <=1.75+&RTOL THEN EFLAG=0;
  END;

*****
* TEMP * * END;
*****;

*****
* ADD ETYPE VAR *
*****;
PROC SORT DATA=MODPD; BY EFLAG;
DATA MODPD; MERGE MODPD(IN=A) ETYPE; BY EFLAG;
  IF A=1;
PROC DELETE DATA=WPPSUMPD;

```

```

PROC DELETE DATA=RATESPD;
*****
* SA MODULE:                *
*****;
PROC SORT DATA=MODSA;          BY RDATE RPWCODE VIP AIC;
PROC SORT DATA=RATESSA;       BY RDATE RPWCODE VIP AIC;
DATA MODSA RATESANU;
MERGE MODSA(IN=A) RATESSA(IN=B); BY RDATE RPWCODE VIP AIC;
  IF B=0 THEN EFLAG=1000;
ELSE IF RPWCODE=. THEN EFLAG=1100;
  IF EFLAG=1100 OR A=0 THEN OUTPUT RATESANU;
ELSE OUTPUT MODSA;
DATA MODSA; SET MODSA;
RP=0; RW=0;
*****;
*****;
*****;
IF SYS='M&PCS' & CLASS='SA' & LBRT ^=0
  THEN W = (R - P*PCRT)/LBRT;

*****
* EDITS *
*****;
IF EFLAG=0 THEN DO;
  IF MULTI=1 THEN DO;
    IF W^=0 & W^=. THEN RW=R;
    ELSE IF P^=0 & P^=. THEN RP=R;
    IF (RP=0 OR RP=.) & (RW=0 OR RW=.) & PCRT^=0 & LBRT^=0
      OR (P=0 OR P=.) & (W=0 OR W=.) THEN DO;
      EFLAG=2000; RP=R;
    END;
    ELSE IF P^=0 & W^=0 THEN DO;
      EFLAG=2100; RP=R;
    END;
  END;
ELSE DO; * ALL OTHER CASES;
  RP=PCRT*P;
  RW=R-RP;
  IF (RP=0 OR RP=.) OR (P=0 OR P=.) OR (W=0 OR W=.)
    THEN EFLAG=2000;
  END;
END;

```

END;

```

ELSE RP=R; * ALL OTHER CASES;
*****
* RPP & RLB RATIOS (VIP LEVEL) *
*****;
RPP=0; RLB=0;
IF PARTIAL=0 THEN DO; /* NO RATIOS FOR PPI */
    IF P^=0 & P^=. & W^=0 & W^=. THEN DO;
        RPP=RP/P;
        RLB=RW/W;
        END;
    ELSE IF P^=0 & P^=. THEN RPP=RP/P;
    ELSE IF W^=0 & W^=. THEN RLB=RW/W;
    END;
DROP R;
*****
* AUX W,P VARS FOR MVIP WPP CALC *
*****;
DATA MODSAM; SET MODSA;
    IF MULTI=1;
    PROC SORT; BY AP FINNO RDATE MVIP;
    PROC SUMMARY; BY AP FINNO RDATE MVIP;
        VAR P W; OUTPUT OUT=WPPMULTI SUM=MP MW;
PROC SORT DATA=MODSA; BY AP FINNO RDATE MVIP;
DATA MODSA; MERGE MODSA WPPMULTI; BY AP FINNO RDATE MVIP;
    IF MULTI=1 &
        (MP=0 OR MP=. OR MW=0 OR MW=.) THEN EFLAG=2000;
*****
* WPP RATIO *
*****;
    IF MULTI=0 & P^=0 & P^=. THEN WPP=ABS(W/P);
    ELSE IF MULTI=1 & MP^=0 THEN WPP=ABS(MW/MP);
*****
* EDITS (HIERARCHICAL): *
* 1. MISSING DATA: 2. CLASS-LEVEL: 3. VIP-LEVEL. *
*****;
IF EFLAG=0 THEN DO;
    * NOTE: BYPASS MULTI PCRT=0, UPDATE AS REQUIRED;
    IF MULTI^=1 & PCRT<=0 & LBRT<=0 THEN EFLAG=1200;
    ELSE IF MULTI^=1 & PARTIAL=0 &
        (RPP+RLB=0 OR RLB+RLB=.) THEN EFLAG=2500;
*****

```

```

* SINGLE PIECE ONLY *
*****;
ELSE IF SINGLE=1 THEN DO;
* NOTE: PCRT INCLUDES SURCHARGE, LBRT IS FOR UNIT OZ;
  OZPP=ROUND(0.4999 + WPP*16); *CHANGE TO UNIT OZ;
  IF RPP < (PCRT+OZPP*LBRT)*(1-&RTOL) OR
    RPP > (PCRT+OZPP*LBRT)*(1+&RTOL)
    THEN EFLAG=3000;
  END;
*****
* ALL OTHER *
*****;
ELSE IF PARTIAL=0 & PCRT>0 & LBRT>0 THEN DO;
  IF ABS((PCRT*P+LBRT*W) -RP-RW) / (RP+RW) ^<&RTOL
    THEN EFLAG=3000;
  END;
ELSE IF PARTIAL=0 & PCRT>0 THEN DO;
  IF ABS(RPP-PCRT) / PCRT ^<&RTOL THEN EFLAG=3000;
  END;
ELSE IF PARTIAL=0 & LBRT>0 THEN DO;
  IF ABS(RLB-LBRT) / LBRT ^<&RTOL THEN EFLAG=3000;
  END;
ELSE IF PARTIAL=1 & RP^=0 THEN DO;
  IF ABS((PCRT*P+LBRT*W) -RP-RW) / (RP+RW) ^<&RTOL
    THEN EFLAG=3000;
  END;
IF ^ (WPPMIN*(1-&WTOL) < WPP <=WPPMAX*(1+&WTOL))
  THEN EFLAG=3100;
END;
*****
* SET EFLAG SAME FOR MVIP PAIRS *
*****;
PROC SORT; BY AP FINNO RDATE MVIP DESCENDING EFLAG;
DATA MODSA; SET MODSA; BY AP FINNO RDATE MVIP;
  IF FIRST.MVIP THEN HIFLAG=EFLAG;
  IF MULTI=1 THEN EFLAG=HIFLAG;
  RETAIN HIFLAG;
*****
* ADD ETYPE VAR *
*****;
PROC SORT DATA=MODSA; BY EFLAG;

```

```

DATA MODSA; MERGE MODSA(IN=A) ETYPE; BY EFLAG;
      IF A=1;
PROC DELETE DATA=MODSAM;
PROC DELETE DATA=RATESSA;
*****
* SB MODULE:          *
*****;
PROC SORT DATA=MODSB;          BY RDATE RPWCODE VIP AIC;
PROC SORT DATA=RATESSB;      BY RDATE RPWCODE VIP AIC;
DATA MODSB RATESBNU;
MERGE MODSB(IN=A) RATESSB(IN=B); BY RDATE RPWCODE VIP AIC;
      IF B=0          THEN EFLAG=1000;
      ELSE IF RPWCODE=. THEN EFLAG=1100;
      IF EFLAG=1100 OR A=0 THEN OUTPUT RATESBNU;
      ELSE          OUTPUT MODSB;
DATA MODSB; SET MODSB;
      RP=0; RW=0;
      *****;
      *****;
      *****;
      IF SYS='M&PCS' & CLASS='SB' & VIP3='7' &
      LBRT ^=0 THEN DO;      * DIV BY ZERO IF PC-RATE VIPS ;
      W = (R - P*PCRT)/LBRT;
      RW= W*LBRT;
      RP= P*PCRT;
      IF ABS(((RP+RW)-R)/R) >&RTOL THEN ABORT;
      END;
      *****
      * EDITS *
      *****;
      IF EFLAG=0 THEN DO;
      IF DISCOUNT='Y' THEN DO;
      RP=R;      * SB DISCOUNT IS SIGNED '-';
      PDISC=P; * RETAIN VOL FOR RATE CHECK SECTION;
      P=0;      * ZERO OUT PIECES (AVOID DOUBLECOUNT);
      IF RP=0 OR RP=. OR PDISC=0 OR PDISC=. THEN EFLAG=2000;
      ELSE IF RP*PDISC ^<0 /* ONE TERM MUST BE NEG */
      OR W^=0 THEN EFLAG=2100;
      END;
      ELSE IF MULTI=1 THEN DO;
      IF W^=0 & W^=. THEN RW=R;

```

```

ELSE IF P^=0 & P^=. THEN RP=R;
IF (RP=0 OR RP=.) & (RW=0 OR RW=.) & PCRT^=0 & LBRT^=0
OR (P=0 OR P=.) & (W=0 OR W=.) THEN DO;
    EFLAG=2000; RP=R;
    END;
ELSE IF P^=0 & W^=0 THEN DO;
    EFLAG=2100; RP=R;
    END;
END;
ELSE IF SPSERV=1 THEN DO;
    RP=R;
    IF (RP=0 OR RP=.) OR (P=0 OR P=.) THEN EFLAG=2000;
END;
ELSE DO; * ALL OTHER EFLAG=0 CASES;
    RP=R;
    IF (RP=0 OR RP=.) OR (P=0 OR P=.) OR (W=0 OR W=.)
        THEN EFLAG=2000;
END;
END;
ELSE RP=R; * ALL OTHER CASES (NON-ZERO EFLAG);
*****
* RPP & RLB RATIOS (VIP LEVEL) *
*****;
RPP=0; RLB=0;
    IF DISCOUNT='Y' & ABS(PDISC) >0 THEN RPP=ABS(RP/PDISC);
ELSE IF P^=0 & P^=. THEN RPP=RP/P;
ELSE IF W^=0 & W^=. THEN RLB=RW/W;
DROP R;
*****
* AUX W,P VARS FOR MVIP WPP CALC *
*****;
DATA MODSBM; SET MODSB;
    IF MULTI=1;
        PROC SORT; BY AP FINNO RDATE MVIP;
        PROC SUMMARY; BY AP FINNO RDATE MVIP;
            VAR P W; OUTPUT OUT=WPPMULTI SUM=MP MW;
PROC SORT DATA=MODSB; BY AP FINNO RDATE MVIP;
DATA MODSB; MERGE MODSB WPPMULTI; BY AP FINNO RDATE MVIP;
    IF MULTI=1 &
        (MP=0 OR MP=. OR MW=0 OR MW=.) THEN EFLAG=2000;
*****

```

```

* WPP RATIO *
*****;
      IF MULTI=0 & P^=0 & P^=. THEN WPP=ABS(W/P);
      ELSE IF MULTI=1 & MP^=0 & MP^=. THEN WPP=ABS(MW/MP);
*****
* EDITS (HIERARCHICAL): *
* 1. MISSING DATA: 2. CLASS-LEVEL: 3. VIP-LEVEL. *
*****;
IF EFLAG=0 THEN DO;
      * NOTE: BYPASS MULTI PCRT=0, UPDATE AS REQUIRED;
      IF MULTI=0 & (RHILO=0 & PCRT<=0 & LBRT<=0 OR
                    RHILO=1 & RATEHI<=0 & RATELO< 0)
          THEN EFLAG=1200;

      ELSE IF MULTI=0 &
            (RPP+RLB=0 OR RLB+RLB=.) THEN EFLAG=2500;
*****
* DISCOUNT ONLY *
*****;
IF DISCOUNT='Y' THEN DO;
      IF PCRT<=0 OR RPP=0 OR RPP=. OR
          ABS(RPP-PCRT)/PCRT ^<=&RTOL THEN EFLAG=3000;
      END;
*****
* SINGLE PIECE (OR PC-RATE ONLY) *
*****;
ELSE IF SINGLE=1 THEN DO;
      IF (RHILO=0 & RPP<(PCRT+WPP*LBRT)*(1-&RTOL)) OR
          (RHILO=0 & RPP>(PCRT+WPP*LBRT)*(1+&RTOL)) OR
          (RHILO=1 & RPP< RATELO*(1-&RTOL)) OR
          (RHILO=1 & RPP> RATEHI*(1+&RTOL))
          THEN EFLAG=3000;

      END;
*****
* ALL OTHER *
*****;
ELSE IF PCRT>0 THEN DO;
      IF ABS(RPP-PCRT)/PCRT ^<&RTOL THEN EFLAG=3000;
      END;
ELSE IF LBRT>0 THEN DO;
      IF ABS(RLB-LBRT)/LBRT ^<&RTOL THEN EFLAG=3000;
      END;

```

```

*****
* ALL *
*****;
IF SPSERV ^=1 & DISCOUNT ^= 'Y' &
  ^(WPPMIN*(1-&WTOL) < WPP <=WPPMAX*(1+&WTOL))
  THEN EFLAG=3100;

END;
*****
* SET EFLAG SAME FOR MVIP PAIRS *
*****;
PROC SORT; BY AP FINNO RDATE MVIP DESCENDING EFLAG;
DATA MODSB; SET MODSB; BY AP FINNO RDATE MVIP;
  IF FIRST.MVIP THEN HIFLAG=EFLAG;
  IF MULTI=1 THEN EFLAG=HIFLAG;
  RETAIN HIFLAG;
*****
* ADD ETYPE VAR *
*****;
PROC SORT DATA=MODSB; BY EFLAG;
DATA MODSB; MERGE MODSB(IN=A) ETYPE; BY EFLAG;
  IF A=1;
PROC DELETE DATA=RATESB;
PROC DELETE DATA=MODSBM;
PROC DELETE DATA=ETYPE;

*****
* COMBINE *
*****;
DATA ERRS; SET MOD1C MODPD MODSA MODSB;
*****
* REPORT R2-150: NEG VALUES *
*****;
DATA NEGVALUE; SET ERRS;
  IF DISCOUNT='Y' & (RP>0 OR PDISC<0 & PDISC^=.) OR
  DISCOUNT^='Y' & (RP<0 & RP^= . OR RW<0 & RW^= . OR
  P<0 & P^= . OR W<0 & W^= . OR C<0 & C^=.);
PROC SORT; BY SYS CLASS EFLAG RDATE VIP;
PROC SUMMARY; BY SYS CLASS EFLAG RDATE VIP;
  ID ETYPE;
  VAR RP RW C P PDISC W;
  OUTPUT OUT=NEGVALUE SUM=;
* PROC PRINT; *BY SYS CLASS;

```

```

* PAGEBY CLASS;
* VAR EFLAG ETYPE RDATE VIP RP RW C P PDISC W;
* SUM RP RW C P W;
* TITLE3 'R2-150: NEGATIVE VALUES';
* TITLES '(FLAGGED AND UNFLAGGED - FOR REVIEW ONLY)';
* TITLE7 '(%RTOL=100*'%RTOL ' %WTOL=100*'%WTOL)';
* TITLE9 ' ';
PROC DELETE DATA=NEGVALUE;
*****
* REPORT R2-200: EFLAG * STRATUM *
*****;
PROC SORT DATA=ERRS; BY SYS CLASS STRATUM EFLAG RDATE;
PROC SUMMARY DATA=ERRS; BY SYS CLASS STRATUM EFLAG RDATE;
ID ETYPE; VAR RP RW C P PDISC W;
OUTPUT OUT=E200 SUM=;
DATA E200; SET E200;
IF P^=0 & P^=. THEN DO;
RPP=RP/P; WPP=ABS(W/P); CPP=ABS(C/P);
END;
IF W^=0 & W^=. THEN RLB=RW/W;
PROC PRINT; BY SYS CLASS STRATUM;
PAGEBY CLASS;
FORMAT RPP RLB WPP 6.3 CPP 5.2 RP RW W 10.;
ID EFLAG;
VAR ETYPE RDATE _FREQ_ RPP RLB WPP CPP
RP RW P PDISC W;
SUM _FREQ_ RP RW P W; * NOT PDISC (DBL CNT);
TITLE3 'R2-200: EFLAG SUMMARY - EFLAG*STRATUM';
TITLE5 '(%RTOL=100*'%RTOL ' %WTOL=100*'%WTOL)';
TITLE7 ' ';
PROC DELETE DATA=E200;
*****
* REPORT R2-300: EFLAG * VIP *
*****;
PROC SORT DATA=ERRS; BY SYS CLASS EFLAG RDATE VIP;
PROC SUMMARY DATA=ERRS; BY SYS CLASS EFLAG RDATE VIP;
ID ETYPE RPWCODE RATELO RATEHI PCRT LBRT WPPMIN WPPMAX;
VAR RP RW P PDISC C W;
OUTPUT OUT=E300 SUM=;
DATA E300; SET E300;
IF P^=0 & P^=. THEN DO;

```

```

RPP=RP/P; WPP=ABS(W/P); CPP=ABS(C/P);
END;
IF PDISC>0 THEN RPP=RP/PDISC;
IF W^=0 & W^=. THEN RLB=RW/W;
PROC PRINT; BY SYS CLASS EFLAG ETYPE;
PAGEBY CLASS;
FORMAT RPP RLB WPP 6.3 CPP 5.2 RP RW W 10.;
ID RDATE;
VAR VIP _FREQ_ PCRT LBRT RATELO RATEHI WPPMIN WPPMAX
RPP RLB WPP CPP RP RW P PDISC W;
SUMBY EFLAG;
SUM _FREQ_ RP RW P W; * NOT PDISC;
TITLE3 'R2-300: EFLAG SUMMARY - EFLAG * VIP';
TITLES5 '(%RTOL=100*'%RTOL ' %WTOL=100*'%WTOL)';
TITLE7 ' ';
PROC DELETE DATA=E300;
*****
* REPORT R2-400: EFLAG FINNO DETAIL *
*****;
DATA E400; SET ERRS;
IF ^(EFLAG=0 OR EFLAG=1100);
PROC SORT; BY SYS CLASS EFLAG RDATE VIP STRATUM FINNO AP;
PROC PRINT; BY SYS CLASS EFLAG ETYPE RDATE; PAGEBY RDATE;
FORMAT RPP RLB WPP WPC 6.3 CPP 5.2;
FORMAT RPP RLB WPP WPC 6.3 CPP 5.2 RP RW W 10.;
ID RDATE;
VAR VIP STRATUM FINNO AP LBRT PCRT RATELO RATEHI
WPPMIN WPPMAX RPP RLB WPP CPP WPC RP RW P PDISC W;
SUMBY EFLAG;
SUM RP RW P W;
TITLE3 'R2-400: EFLAG SUMMARY - FINNO DETAIL';
TITLES5 '(%RTOL=100*'%RTOL ' %WTOL=100*'%WTOL)';
TITLE7 ' ';
PROC DELETE DATA=E400;
PROC DELETE DATA=ERRS;
*****
* COMBINE MODULES, UPDATE NONRESP, CALC F1 & F2 *
*****;
DATA EDITDAT2; SET MOD0C(IN=A) MOD1C MODPD MODSA MODSB;
IF A=1 THEN DO;
RW=0; RP=0;

```

```

IF W>0 & P<=0 THEN RW=R;
ELSE          RP=R;
END;
KEEP RUNTYPE CLASS SYS AP PQ FY FINNO VIP VIP1 VIP2345 RPWCODE
  RP RW P C W NRESP MIGRATE EFLAG STRATUM BLOWUP AIC RDATE
  DISCOUNT PDISC VIPCAT;
PROC DELETE DATA=MOD0C;
PROC DELETE DATA=MOD1C;
PROC DELETE DATA=MODPD;
PROC DELETE DATA=MODSA;
PROC DELETE DATA=MODSB;
*****
* UPDATE NRESP IF ALL RECS FAIL (MNUALS ONLY) * /* BY AP */
*****;
DATA NRESPX; SET EDITDAT2;
  IF STRATUM>=2.0;
  PROC SORT DATA=NRESPX; BY SYS AP FINNO;
  DATA NRESPX; SET NRESPX; BY SYS AP FINNO;
  IF NRESP=0 & EFLAG=0 THEN PASS=1;
  IF LAST.FINNO AND PASS^=1; *ALL RECS FAILED;
  RETAIN PASS;
  KEEP AP SYS FINNO;
PROC SORT DATA=EDITDAT2; BY SYS AP FINNO;
DATA EDITDAT2; MERGE EDITDAT2 NRESPX(IN=B); BY SYS AP FINNO;
  IF B=1 THEN NRESP=1; * CHANGED FROM 0 IF ALL RECS FAILED;
  *****
  * COUNT APS WITH >=1 PASSING RECS *
  *****;
* CBCIS OFFICES;
PROC DELETE DATA=NRESPX;
DATA F1KOUNT1; SET EDITDAT2;
  IF STRATUM<2.0 & EFLAG=0;
  PROC SORT DATA=F1KOUNT1; BY FINNO AP;
  DATA F1KOUNT1; SET F1KOUNT1; BY FINNO AP;
  IF FIRST.FINNO THEN KAP1=0;
  IF FIRST.AP AND NRESP=0 THEN KAP1=KAP1+1;
  IF LAST.FINNO;
  RETAIN KAP1;
  KEEP FINNO KAP1;
* MANUAL OFFICES;
DATA F1KOUNT2; SET EDITDAT2;

```

```

IF STRATUM>=2.0 & EFLAG=0;
PROC SORT DATA=F1KOUNT2;          BY SYS FINNO AP;
DATA F1KOUNT2; SET F1KOUNT2; BY SYS FINNO AP;
  IF FIRST.FINNO THEN KAP2=0;
  IF FIRST.AP AND NRESP=0 THEN KAP2=KAP2+1;
  IF LAST.FINNO;
  RETAIN KAP2;
  KEEP SYS FINNO KAP2;
PROC SORT DATA=EDITDAT2;          BY SYS FINNO;
DATA EDITDAT2; MERGE EDITDAT2 F1KOUNT2; BY SYS FINNO;
PROC SORT DATA=EDITDAT2;          BY FINNO;
DATA EDITDAT2; MERGE EDITDAT2 F1KOUNT1; BY FINNO;
*****
* CALC F1 *      * NOTE: F1=. (N/A) IF NRESP=1 ;
*****;
IF &RUNTYPE='AP' THEN F1=1;
ELSE DO;
  NAP=3; IF &PQ=4 THEN NAP=4; *NUMBER OF APS IN PQ;
  IF STRATUM <2.0 & KAP1 ^=0 THEN F1=NAP/KAP1;
  IF STRATUM >=2.0 & KAP2 ^=0 THEN F1=NAP/KAP2;
  END;
PROC DELETE DATA=F1KOUNT1;
*****
* CALC F2 *
*****;
PROC SORT DATA=EDITDAT2;          BY SYS STRATUM FINNO;
DATA F2KOUNT2; SET EDITDAT2; BY SYS STRATUM FINNO;
  IF STRATUM>=2.0;
  IF FIRST.STRATUM THEN DO;
    NHSAMP=0; KNRESP=0;
  END;
  IF FIRST.FINNO THEN DO;
    NHSAMP=NHSAMP+1;
    IF KAP2=0 OR KAP2=. THEN KNRESP=KNRESP+1;
  END;
  IF LAST.STRATUM;
  RETAIN NHSAMP KNRESP;
  KEEP SYS STRATUM NHSAMP KNRESP;
DATA EDITDAT2; MERGE EDITDAT2 F2KOUNT2; BY SYS STRATUM;
  IF STRATUM <2.0 THEN F2=1;
  ELSE IF NHSAMP>KNRESP THEN F2=NHSAMP/(NHSAMP-KNRESP);

```

```

ELSE F2=0; /* STRATUM EMPTY: MANUAL INTERVENTION */
PROC DELETE DATA=F2KOUNT2;
*****
* F1-F2 REPORT * * FOR AUDITOR;
*****;
* NOTE: LOERR=0 CONFIRMS AT LEAST 1 PASSING RECORD;
PROC SORT DATA=EDITDAT2; BY SYS STRATUM FINNO EFLAG;
DATA FCHECK;SET EDITDAT2; BY SYS STRATUM FINNO EFLAG;
  IF SYS^=' ';
  ZVOL=RP+RW+P+W;
  IF FIRST.FINNO THEN LOERR=EFLAG;
  RETAIN LOERR;
  KEEP SYS STRATUM FINNO F1 F2 NRESP NHSAMP
    KNRESP ZVOL LOERR;
  PROC SUMMARY; BY SYS STRATUM FINNO;
  ID F1 F2 NRESP NHSAMP KNRESP LOERR;
  VAR ZVOL;
  OUTPUT OUT=FCHECK SUM=;
  DATA FCHECK; SET FCHECK;
    X=1;
    IF ABS(ZVOL)=0 THEN ZERO='Y';
    IF STRATUM >1 OR
      F1 ^≠1 OR
      F2 ^≠1 OR
      LOERR^=0 THEN XFINNO=FINNO;
    ELSE IF ZERO='Y' THEN XFINNO=888888;
    ELSE XFINNO=999999;
  PROC SORT; BY SYS STRATUM XFINNO;
  PROC SUMMARY; BY SYS STRATUM XFINNO;
  ID F1 F2 NHSAMP KNRESP ZERO
    NRESP LOERR; VAR X;
  OUTPUT OUT=FCHECK SUM=;
  PROC SORT; BY SYS STRATUM ZERO F1 F2;
  PROC PRINT; BY SYS;
  VAR SYS STRATUM XFINNO _FREQ_ ZERO
    F1 F2 LOERR NRESP NHSAMP KNRESP;
  FORMAT F1 F2 7.4
    NHSAMP KNRESP _FREQ_ COMMA13.;
  TITLE3 'R2-500: F1, F2 FACTORS';
  TITLE4 'AND 0-VOL & 100% FAIL SITES';
  TITLE5 '(DUMMY FIN= 888888,999999)';

```

```

                                TITLE6 ' ';
                                PROC DELETE DATA=FCHECK;

*****
* COLLECT 0-VOL VIPS * /* DUMMY FINNO=0, ASIGN TO STRATUM-1 */
*****;
DATA ZEROVIPS R600;
  SET RATE1CNU(IN=A) RATEPDNU(IN=B) RATESANU(IN=C) RATESBNU(IN=D);
  IF A=1 THEN CLASS='1C';
  ELSE IF B=1 THEN CLASS='PD';
  ELSE IF C=1 THEN CLASS='SA';
  ELSE IF D=1 THEN CLASS='SB';
  AP=&AP; PQ=&PQ; FY=&FY;
  X=1;
  RUNTYPE=&RUNTYPE;
  FINNO=0;
  STRATUM=1;
  F1=1; F2=1; BLOWUP=1;
  RP=0; RW=0; P=0; C=0; W=0;
  IF EFLAG=. THEN EFLAG=0;
  NRESP=0;
  OUTPUT ZEROVIPS;
  *****
  * FILTER *
  *****;
  IF RDATE=&RDATE1;
  OUTPUT R600;
  PROC DELETE DATA=RATE1CNU;
  PROC DELETE DATA=RATEPDNU;
  PROC DELETE DATA=RATESANU;
  PROC DELETE DATA=RATESBNU;
  *****
  * PRINT UNUSED CODES * * MOST RECENT RATE DATE ONLY;
  *****;
  PROC SORT DATA=R600; BY SYS EFLAG RDATE VIP;
  PROC SUMMARY DATA=R600; BY SYS EFLAG RDATE VIP;
  ID VIPCAT RPWCODE PCRT LBRT RATELO RATEHI WPPMIN WPPMAX;
  VAR X; OUTPUT OUT=R600 SUM=;
  PROC SORT DATA=R600; BY DESCENDING EFLAG SYS RDATE;
  PROC PRINT DATA=R600 U; BY DESCENDING EFLAG SYS RDATE;
  PAGEBY SYS;

  ID VIP;

```

```

VAR VIPCAT RPWCODE PCRT LBRT RATELO RATEHI WPPMIN WPPMAX;
TITLE3 'R2-600:  UNUSED VIP CODES';
TITLE4 'FOR RATEDATE ' &RDATE1 ' ONLY (BY SYSTEM)';
TITLE6 'NOTE:  INCLUDES EFLAG=1100';
TITLE7 ' ';
PROC DELETE DATA=R600;
*****
* ADD ZERO RECORDS, OUTPUT *
*****;
DATA EDITDAT2; SET EDITDAT2 ZEROVIPS;
PROC DELETE DATA=ZEROVIPS;
PROC SORT; BY FINNO VIP AP;
DATA _NULL_; SET EDITDAT2;
*****
* OVERWRITE P WITH PDISC *      /* FOR PERIODICALS */
*****;
IF DISCOUNT='Y' THEN P=PDISC;
FILE EDITDAT2;
PUT @1    RUNTYPE  $2.
   @3     CLASS   $2.
   @5     SYS     $6.
   @11    AP      Z2.
   @13    PQ      1.
   @14    FY      2.
   @16    RDATE   Z6.
   @22    FINNO   Z6.
   @28    VIP     $5.
   @33    RPWCODE Z5.
   @38    RP      Z12.2
   @50    RW      Z12.2
   @62    P       Z12.
   @74    C       Z12.
   @86    W       Z14.2
   @100   NRESP   1.
   @101   MIGRATE 1.
   @102   EFLAG   Z4.
   @106   DISCOUNT $1.
   @107   STRATUM Z3.1
   @110   BLOWUP  Z8.3
   @118   AIC     3.
   @121   F1      Z10.8

```

```
)
)
@131      F2  Z15.8 ;
*****
* PRINT RPW TOTALS *
*****;
DATA PASS; SET EDITDAT2;
  IF ^(VIP='44444' OR NRESP=1) & EFLAG=0;
  IF RPWCODE^=.;
  PROC SORT    DATA=PASS; BY SYS AIC CLASS STRATUM RPWCODE;
  PROC SUMMARY DATA=PASS; BY SYS AIC CLASS STRATUM RPWCODE;
    VAR RP RW P C W; OUTPUT OUT=PASS SUM=;
  PROC DELETE DATA=EDITDAT2;
  PROC PRINT DATA=PASS; BY SYS AIC CLASS; PAGEBY CLASS;
    FORMAT RP RW W COMMA15.2 P C COMMA13. STRATUM 3.1;
    ID SYS;
    VAR AIC CLASS STRATUM RPWCODE RP RW P C W;
    SUM                RP RW P C W;
    TITLE3 'R2-800: UNINFLATED RPW: STRATUM * RPWCODE';
    TITLE5 'PASSED RECORDS';
    TITLE6 ' ';
```

/*

-

```

//NNNNNNS JOB (ALA03), 'NNNNNNN NNN NN', CLASS=H, MSGCLASS=H          00010008
/*ROUTE PRINT U5704                                                    00030000
//*****00040000
/**          JOB NAME:  PDS.SASRPW(RPWJOB3) - FORMERLY "J3RRUN97" USED 00060000
//*****00060000
/**                                                    00120008
//SO1      EXEC SAS, REGION=4096K, TIME=60                             00120008
//WORK     DD SPACE=(CYL, (900,700), RLSE)                             00140000
//SYSOUT   DD DUMMY                                                    00250000
/**                                                    00140000
/**                                                    00140000
/** \\\\ PQ RUN: JOB-1 OUTPUT LAST QUALIFIER:  FY1996QX: X=1...4.     00140000
/** \\\\ AP RUN: JOB-1 OUTPUT LAST QUALIFIER:  FY1996XX: XX=01..13.   00140000
//EDITDAT2 DD DSN=HSQRAN.BRPWD01.EDITDAT2.FY2000Q4, DISP=SHR         00140000
/**                                                    00140000
/** \\\\ RPW CODE AND 2-PAGE TITLE FILES.                               00140000
//RPWLABEL DD DSN=HSQRAN.RPW.FY2000.CATEGORY.DIR, DISP=SHR            00140000
//TWOPAGE  DD DSN=HSQRAN.RPW.ADJ.CNTL2000 (SUMDIR), DISP=SHR           00140000
/**                                                    00140000
/** \\\\ TRIAL BALANCE AIC REVENUE (BY AP).                             00140000
/**                                                    00140000
//REVCNTL  DD DSN=HSQRAN.RPW.REVCNTL.FY200013, DISP=SHR               00140000
//          DD DSN=HSQRAN.RPW.REVCNTL.FY200012, DISP=SHR               00140000
//          DD DSN=HSQRAN.RPW.REVCNTL.FY200011, DISP=SHR               00140000
//          DD DSN=HSQRAN.RPW.REVCNTL.FY200010, DISP=SHR               00140000
/**                                                    00140000
/** \\\\ PQ RUN: JOB-2 OUTPUT LAST QUALIFIER:  FY1996QX: X=1...4.     00140000
/** \\\\ AP RUN: JOB-2 OUTPUT LAST QUALIFIER:  FY1996XX: XX=01..13.   00140000
/**                                                    00140000
//*NFLATE  DD DSN=HSQRAN.BRPWD01.INFLATE1.FY2000Q4, DISP=SHR         00140000
//INFLATE  DD DSN=HSQRAN.BRPWD01.INFLATE1.FY2000Q4,                   00140000
//          DISP=(NEW, CATLG), DCB=(RECFM=FB, LRECL=240, BLKSIZE=4800), 00140000
//          UNIT=SYSDA, SPACE=(CYL, (200,250), RLSE)                   00140000
/**                                                    00140000
//*RANS    DD DSN=HSQRAN.BRPWD01.TRANS.FY2000Q4, DISP=SHR              00140000
//TRANS    DD DSN=HSQRAN.BRPWD01.TRANS.FY2000Q4,                       00140000
//          DISP=(NEW, CATLG), DCB=(RECFM=FB, LRECL=80, BLKSIZE=2400), 00140000
//          UNIT=SYSDA, SPACE=(CYL, (5,5), RLSE)                         00140000
/**                                                    00140000
//SYSIN    DD *                                                         00250000
                                                    00260000

```

```

*****;
* \\\\/\\\/\\\/\\\/\\\/\\\/ UPDATE PARAMETERS \\\\/\\\/\\\/\\\/\\\/\\\/ *;
%LET RUNTYPE = 'PQ'; *ENTER 2-CHAR ALPHA RUN TYPE: 'PQ' OR 'AP'. *;00260000
%LET FY = 00; *ENTER 2-CHAR NUMERIC FY (E.G., 96). *;00260000
%LET PQ = 4; *ENTER 1-CHAR NUMERIC QUARTER: 1-4. *;00260000
%LET AP = .; *ENTER 2-CHAR NUMERIC AP: 01-13 (. IF PQ RUN). *;00260000
*****;
*****
* PARAMETER CHECKS *
*****;
DATA _NULL_;                                00260000
  IF ^(&RUNTYPE='PQ' OR &RUNTYPE='AP') OR    00260000
    ^ ( 0<=&FY<=99) OR ^ (1<=&PQ<=4) THEN ABORT; 00260000
  IF &RUNTYPE='AP' AND ^ (01<=&AP<=13) THEN ABORT; 00260000
  IF &RUNTYPE='PQ' AND &AP ^= . THEN ABORT;    00260000
                                                00260000
*****
* MAP AIC TO SYSTEM *                        00260000
*****;                                       00260000
DATA AICMAP; INFILE REVCNTL;                 00260000
  INPUT @1 GLA 5.                             00260000
        @6 AIC 3.                             00260000
        @9 RAIC 13. ;                         00260000
  IF AIC=132 THEN DELETE;
  IF AIC=124 THEN DELETE;
  IF AIC=136 THEN AIC=135;
  IF AIC=224 THEN AIC=135;
  IF AIC=238 THEN DELETE;
  PROC SORT; BY AIC;
  PROC SUMMARY; BY AIC;
  VAR RAIC; OUTPUT OUT=AICMAP SUM=;
*****
* READ EDITDAT2 - VERIFY PARAMETERS *
*****;
DATA EDITDAT2 NRESP; INFILE EDITDAT2;        00260000
  INPUT @1 RUNTYPE $2.
        @3 CLASS $2.
        @5 SYS $6.
        @11 AP 2. @11 APK $2.
        @13 PQ 1.
        @14 FY 2.

```

```

@16      RDATE      6.  @16 RDATEK $6.
@22      FINNO      6.
@28      VIP        $5. @28 VIP1 $1. @29 VIP2 $1. @30 VIP3 $1.
@33      RPWCODE    5.
@38      RP         12.
@50      RW         12.
@62      P          12.
@74      C          12.
@86      W          14.
@100     NRESP      1.
@101     MIGRATE    1.
@102     EFLAG      4.  @102 EFLAGK $4.
@106     DISCOUNT $1.
@107     STRATUM    3.
@110     BLOWUP     8.
@118     AIC        3.
@121     F1         10.
@131     F2         15. ;

```

```

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

```

```

IF AIC=224 THEN DO;
  AIC=135;
  IF VIP='44444' THEN DELETE;
END;

```

```

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

```

```

*****
* ZERO DISCOUNT PIECES *
*****;
PDISC=0;
IF DISCOUNT='Y' THEN DO;
  PDISC=P;
  P=0;
END;
*****

```

```

* FILTER, DROP E-FAILURES *
*****;
IF VIP='44444' OR
  (NRESP=0 & RPWCODE^=. & EFLAG=0) THEN OUTPUT EDITDAT2;
IF NRESP=1 THEN OUTPUT NRESP;
*****
* MISSING CHECK *
*****;
DATA _NULL_; SET EDITDAT2;
IF RUNTYPE^=&RUNTYPE THEN ABORT;
IF &RUNTYPE='AP' & AP^=&AP THEN ABORT; 00260000
IF PQ^=&PQ OR FY^=&FY THEN ABORT;
IF &RUNTYPE='AP' THEN DO; 00260000
  IF (PQ=1 & ^(1<=AP<=3)) OR (PQ=2 & ^(4<=AP<=6)) OR 00260000
    (PQ=3 & ^(7<=AP<=9)) OR (PQ=4 & ^(10<=AP<=13)) THEN ABORT; 00260000
  END; 00260000
ELSE IF &RUNTYPE='PQ' THEN DO; 00260000
  IF ^(PQ=1 OR PQ=2 OR PQ=3 OR PQ=4) THEN ABORT; 00260000
  END; 00260000
IF VIP=' ' THEN ABORT; 00260000
IF ^(0000<=EFLAG<=9999) THEN ABORT; 00260000
IF &PQ=4 & EFLAG=0 & ^(1<=F1<=4) THEN ABORT;
IF &PQ<=3 & EFLAG=0 & ^(1<=F1<=3) THEN ABORT;
IF NRESP^=0 THEN ABORT;
PROC SORT DATA=EDITDAT2; 00260000
  BY AP SYS CLASS FINNO RDATE AIC VIP; 00260000
DATA _NULL_; SET EDITDAT2; 00260000
  BY AP SYS CLASS FINNO RDATE AIC VIP; 00260000
  IF FIRST.VIP ^=LAST.VIP & 00260000
    /* REMOVE FOLLOWING AFTER OLD RATES OBSOLETE */ 00260000
    /* REMOVE FOLLOWING AFTER OLD RATES OBSOLETE */ 00260000
    /* REMOVE FOLLOWING AFTER OLD RATES OBSOLETE */ 00260000
    ^(RDATE=010195 & 00260000
      (VIP='01192' OR VIP='01193' OR VIP='01293' OR 00260000
        VIP='01270' OR VIP='01470' OR VIP='01391' OR 00260000
        VIP='01392' OR VIP='01393')) ) THEN ABORT; 00260000
*****
* JOB-2 VERIFICATION RPT *
*****;
DATA REPO50; SET EDITDAT2;
  IF CLASS^=' ';

```

```

IF VIP^='44444';
PROC SORT; BY SYS AIC CLASS;
PROC SUMMARY DATA=REP050; BY SYS AIC CLASS;
VAR RP RW P PDISC C W; OUTPUT OUT=REP050 SUM=;
PROC PRINT DATA=REP050;
FORMAT RP RW PDISC C W COMMA13. P COMMA14.;
ID SYS;
VAR AIC CLASS RP RW P PDISC C W;
SUM RP RW P C W;
TITLE1 "JOB3 ** RPW BULK MAIL SYSTEM ** FOR &RUNTYPE&AP PQ&PQ-&FY";
TITLE3 'R3-050: JOB-2 OUTPUT VERIFICATION';
TITLE5 'PASSED RECORDS ONLY';
TITLE6 ' ';
***** 00260000
* INFLATE: SYSTEM X STRATUM * 00260000
*****; 00260000
* BLOWUP: INFLATION FACTOR (WITHIN STRATUM); 00260000
* F1: OFFICE FACTOR - ADJUST FOR MISSING AP FOR PQ RUN (=1 IF AP RUN); 00260000
* F2: STRATUM FACTOR - NONRESPONSE ADJUSTMENT; 00260000
* F3: TRIAL BALANCE ADJUSTMENT; 00260000
DATA INFLATE1; SET EDITDAT2; 00260000
R=RP+RW; 00260000
RPHAT=RP*F1*(BLOWUP*F2); 00260000
RWHAT=RW*F1*(BLOWUP*F2); 00260000
RHAT=R*F1*(BLOWUP*F2); 00260000
CHAT=C*F1*(BLOWUP*F2); 00260000
PHAT=P*F1*(BLOWUP*F2); 00260000
WHAT=W*F1*(BLOWUP*F2); 00260000
PDISCHAT=PDISC*F1*(BLOWUP*F2); 00260000
***** 00260000
* ASSIGN $AIC * 00260000
*****; 00260000
PROC SORT DATA=AICMAP; BY AIC; 00260000
PROC SORT DATA=INFLATE1; BY AIC; 00260000
DATA INFLATE1; MERGE INFLATE1(IN=A) AICMAP(IN=B); BY AIC; 00260000
IF A=1; 00260000
***** 00260000
* F3 FACTOR * 00260000
*****; 00260000
PROC SORT DATA=INFLATE1; BY SYS AIC; 00260000
PROC SUMMARY; BY SYS AIC; 00260000

```

```

        VAR RHAT;                                00260000
        OUTPUT OUT=ISUMS SUM=SUMRHAT;           00260000
DATA INFLATE1; MERGE INFLATE1 ISUMS; BY SYS AIC; 00260000
        IF RAIC ^=. THEN F3=RAIC/SUMRHAT;      00260000
        ELSE F3=1;                              00260000
        RPHAT=RPHAT*F3;                          00260000
        RWHAT=RWHAT*F3;                          00260000
        RHAT=RPHAT+RWHAT;                        00260000
        CHAT=CHAT*F3;                            00260000
        PHAT=PHAT*F3;                            00260000
        WHAT=WHAT*F3;                            00260000
        PDISCHAT=PDISCHAT*F3;                   00260000
*****
* REPORTS *
*****;
DATA REP200; SET INFLATE1;
        IF FINNO ^=0;
        IF CLASS ^= ' ';
        IF VIP ^= '44444';
PROC SORT DATA=REP200; BY SYS AIC CLASS STRATUM F1 F2 BLOWUP;
PROC SUMMARY DATA=REP200; BY SYS AIC CLASS STRATUM F1 F2 BLOWUP;
        ID F3; VAR RHAT RPHAT RWHAT CHAT PHAT WHAT;
        OUTPUT OUT=REP200 SUM=;
DATA REP200; SET REP200;
        IF PHAT > 0 THEN RPP=RHAT/PHAT;
        IF PHAT > 0 THEN WPP=WHAT/PHAT;
        IF WHAT > 0 THEN RLB=RHAT/WHAT;
PROC PRINT DATA=REP200 U; BY SYS AIC CLASS; PAGEBY SYS;
        FORMAT RHAT RPHAT RWHAT WHAT CHAT COMMA13. PHAT COMMA14.
                RPP RLB WPP F1 F2 6.3 BLOWUP 8.3 F3 8.5;
        ID STRATUM;
VAR F1 F2 BLOWUP F3 RHAT RPHAT RWHAT PHAT WHAT RPP RLB WPP;
SUM          RHAT RPHAT RWHAT PHAT WHAT          ;
        TITLE3 'R3-200: ADJUSTED ESTIMATES -SUMMARY';
        TITLES ' ';
*****
* COUNT RESPONSES * * NONRESPONDENTS INCLUDED IN POP COUNTS;
*****;
DATA E2KOUNTS; SET EDITDAT2 NRESP;
PROC SORT; BY SYS STRATUM FINNO AP;
DATA E2KOUNTS; SET E2KOUNTS; BY SYS STRATUM FINNO AP;

```

```

IF FINNO^=0;
IF FIRST.FINNO THEN RESP=0;
IF FIRST.AP & NRESP=0 THEN RESP=MAX(RESP,1);
IF LAST.FINNO THEN OUTPUT;
RETAIN RESP;
KEEP SYS STRATUM F1 RESP;
PROC SORT; BY SYS STRATUM F1;
PROC SUMMARY; BY SYS STRATUM F1;
VAR RESP; OUTPUT OUT=E2KOUNTS SUM=;
DATA REP250; SET INFLATE1;
IF FINNO^=0;
IF CLASS^=' ';
PROC SORT DATA=REP250; BY SYS STRATUM F1;
DATA REP250; MERGE REP250(IN=A) E2KOUNTS; BY SYS STRATUM F1;
IF A=1;
IF VIP^='44444'; /* RID EMPTY RECORDS */
NSAMP=INT(0.5 + F2*RESP);
NPOP =INT(0.5 + NSAMP*BLOWUP);
PROC SORT; BY SYS CLASS AIC STRATUM F1;
PROC SUMMARY; BY SYS CLASS AIC STRATUM F1;
ID NSAMP NPOP RESP F2 F3 BLOWUP;
VAR R P W RHAT PHAT WHAT;
OUTPUT OUT=REP250 SUM=;
PROC PRINT DATA=REP250; BY SYS CLASS AIC; PAGEBY SYS;
ID STRATUM;
VAR F1 NPOP NSAMP RESP F2 BLOWUP F3
RHAT PHAT WHAT R P W;
SUM NPOP NSAMP RESP RHAT PHAT WHAT R P W;
FORMAT RHAT WHAT R W COMMA13. PHAT P COMMA14. F2 F3 7.4;
TITLE3 'R3-250: STRATUM COUNTS & SAMP/EST TOTALS';
TITLE4 'BY SYSTEM (COMBINED AICS)';
TITLE6 'NOTE 1: NON-ZERO STRATA ONLY SHOWN';
TITLE7 ' 2: STR-1 SIZE EXCLUDES PANEL MIGRATIONS';
TITLE9 ' ';
*****
* WRITE OUT *
*****;
PROC SORT DATA=INFLATE1; BY FINNO VIP AP;
DATA _NULL_; SET INFLATE1;
*****
* PRDCL DISCOUNT PIECES *

```

*****;

IF DISCOUNT='Y' THEN DO;

 P=PDISC;

 PHAT=PDISCHAT;

 END;

* OUTPUT * /* PASSED RECS ONLY */

*****;

*,

FILE INFLATE;

PUT @1	RUNTYPE	\$2.	
@3	CLASS	\$2.	
@5	SYS	\$6.	
@11	AP	Z2.	
@13	PQ	1.	
@14	FY	2.	
@16	RDATE	Z6.	
@22	FINNO	Z6.	
@28	VIP	\$5.	
@33	RPWCODE	Z5.	
@38	RP	Z12.2	
@50	RW	Z12.2	
@62	P	Z12.	
@74	C	Z12.	
@86	W	Z14.2	
@100	NRESP	1.	
@101	MIGRATE	1.	
@102	EFLAG	Z4.	
@106	DISCOUNT	\$1.	/* PRDCL DISCOUNT */
@107	STRATUM	Z3.1	
@110	BLOWUP	Z8.3	
@118	AIC	3.	
@121	F1	Z10.8	
@131	F2	Z15.8	
@146	F3	Z15.8	
@161	RPHAT	Z12.2	
@173	RWHAT	Z12.2	
@185	PHAT	Z12.2	/*NOT ROUNDED TO INT. */
@197	CHAT	Z12.2	/*NOT ROUNDED TO INT. */
@209	WHAT	Z14.2	;

```

* REPORT R3-425 *
*****;
DATA REP425; SET INFLATE1;
  IF STRATUM^=1.0; * NONAUTOMATED OFFICES (PANEL) ONLY;
  IF CLASS^=' ';
  IF VIP^='44444';
PROC SORT DATA=REP425; BY SYS AIC CLASS STRATUM FINNO;
PROC SUMMARY DATA=REP425; BY SYS AIC CLASS STRATUM FINNO;
  ID NRESP MIGRATE BLOWUP F1 F2 F3;
  VAR RHAT RPHAT RWHAT CHAT PHAT PDISCHAT WHAT;
  OUTPUT OUT=REP425 SUM=;
DATA REP425; SET REP425;
  IF PHAT>0 THEN RPP=RHAT/PHAT;
  IF PHAT>0 THEN WPP=WHAT/PHAT;
  IF WHAT>0 THEN RLB=RHAT/WHAT;
PROC PRINT DATA=REP425 U; BY SYS AIC CLASS; PAGEBY CLASS;
  FORMAT RHAT RPHAT RWHAT PHAT PDISCHAT WHAT CHAT COMMA13.
  RPP RLB WPP 6.3;
  ID STRATUM;
  VAR FINNO RHAT RPHAT RWHAT PHAT PDISCHAT WHAT RPP WPP RLB;
  SUM RHAT RPHAT RWHAT PHAT PDISCHAT WHAT;
  TITLE3 'R3-425: ADJUSTED ESTIMATES - BY PANEL OFFICE';
  TITLE5 '(SYSTEM * AIC * RDATE)';
  TITLE7 ' ';
/*
/*
/*
*****
* REPORT R3-450 *
*****;
PROC SORT DATA=INFLATE1; BY SYS AIC CLASS RDATE VIP;
PROC SUMMARY DATA=INFLATE1; BY SYS AIC CLASS RDATE VIP;
  ID RPWCODE; VAR RHAT RPHAT RWHAT CHAT PHAT PDISCHAT WHAT;
  OUTPUT OUT=REP450 SUM=;
DATA REP450; SET REP450;
  IF CLASS^=' ';
  IF RHAT=0 & PHAT=0 & PDISCHAT=0 & WHAT=0 THEN DELETE;
  IF PHAT>0 THEN RPP=RHAT/PHAT;
  IF PHAT>0 THEN WPP=WHAT/PHAT;
  IF WHAT>0 THEN RLB=RHAT/WHAT;
PROC PRINT DATA=REP450 U; BY SYS AIC CLASS RDATE; PAGEBY RDATE;

```

```

FORMAT RHAT RPHAT RWHAT PDISCHAT WHAT CHAT COMMA13.
      PHAT COMMA14. RPP RLB WPP 6.3;
ID RDATE;
VAR VIP RPWCODE _FREQ_ RHAT RPHAT RWHAT PHAT PDISCHAT WHAT
      RPP WPP RLB;
SUM      _FREQ_ RHAT RPHAT RWHAT PHAT      WHAT;
TITLE3 'R3-450: ADJUSTED ESTIMATES - BY VIP CODE (NON-ZERO)';
TITLE5 '(SYSTEM * AIC * RDATE)';
TITLE7 ' ';

```

*/

*/

*/

* REPORT R3-475 *

*****;

```

PROC SORT DATA=INFLATE1; BY SYS AIC CLASS RPWCODE STRATUM;
PROC SUMMARY DATA=INFLATE1; BY SYS AIC CLASS RPWCODE STRATUM;
VAR RHAT RPHAT RWHAT CHAT PHAT PDISCHAT WHAT;
OUTPUT OUT=REP475 SUM=;
DATA REP475; SET REP475;
      IF CLASS^=' ';
      IF RHAT=0 & PHAT=0 & PDISCHAT=0 & WHAT=0 THEN DELETE;
      IF PHAT>0 THEN RPP=RHAT/PHAT;
      IF PHAT>0 THEN WPP=WHAT/PHAT;
      IF WHAT>0 THEN RLB=RHAT/WHAT;
PROC PRINT DATA=REP475 U; BY SYS AIC CLASS; PAGEBY CLASS;
FORMAT RHAT RPHAT RWHAT PDISCHAT WHAT CHAT COMMA13.
      PHAT COMMA14. RPP RLB WPP 6.3 _FREQ_ COMMA7.;
ID RPWCODE;
VAR STRATUM _FREQ_ RHAT RPHAT RWHAT PHAT PDISCHAT WHAT
      RPP WPP RLB;
SUM      _FREQ_ RHAT RPHAT RWHAT PHAT      WHAT;
TITLE3 'R3-475: ADJUSTED ESTIMATES - BY RPWCODE * STRATUM';
TITLE5 '(COMBINED RDATES)';
TITLE7 ' ';

```

00260000

* LINE NOS, RPW TRANS FILE *

00260000

*****;

00260000

DATA TITLE1; INFILE TWOPAGE;

INPUT @1 LINE 3. @5 LINENAME \$35.;

DATA TITLE2; INFILE RPWLABEL;

```

INPUT @1 RPWCODE 5.   @9 LINE 3.   @28 RPWNAME $35.;
PROC SORT DATA=TITLE1;           BY LINE;
PROC SORT DATA=TITLE2;           BY LINE;
DATA TITLES; MERGE TITLE1(IN=A) TITLE2(IN=B); BY LINE;
  IF A=1;
PROC SORT DATA=TITLES;           BY RPWCODE;
PROC SORT DATA=INFLATE1;        BY RPWCODE;
DATA INFLATE1; MERGE INFLATE1(IN=A) TITLES; BY RPWCODE;
  IF A=1;
PROC SORT DATA=INFLATE1; BY AIC SYS RPWCODE;           00260000
PROC SUMMARY DATA=INFLATE1; BY AIC SYS RPWCODE;       00260000
  ID CLASS RPWNAME LINE LINENAME;                      00260000
  VAR RHAT PHAT CHAT WHAT; OUTPUT OUT=TRANS SUM=;      00260000
DATA TRANS; SET TRANS;
  IF RPWCODE^=.;
  IF PHAT>0 THEN RPP=RHAT/PHAT;
  IF PHAT>0 THEN WPP=WHAT/PHAT;
  IF WHAT>0 THEN RLB=RHAT/WHAT;
  *****                                           00260000
  * OUTPUT *                                           00260000
  *****;                                           00260000
PROC SORT DATA=TRANS; BY SYS AIC RPWCODE;
DATA TRANS; SET TRANS;
  FILE TRANS;
  PUT @1 RPWCODE Z4.
     @5 RHAT Z13.
     @18 PHAT Z11.
     @29 WHAT Z13.
     @43 SYS $6.
     @49 AIC 3.;
*****
* PRINT BY RPWCODE *
*****;
PROC SORT DATA=TRANS; BY SYS AIC CLASS RPWCODE;
PROC PRINT DATA=TRANS U; BY SYS AIC CLASS; PAGEBY CLASS;
  FORMAT RHAT CHAT WHAT COMMA13. RPP RLB WPP 6.3
         PHAT COMMA14.;
  ID AIC;
  VAR CLASS LINE RPWCODE RPWNAME RHAT PHAT WHAT RPP RLB WPP;
  SUM RHAT PHAT WHAT;
TITLE3 'R3-500: ADJUSTED ESTIMATES - BY SYSTEM * AIC * RPWCODE';

```

```

TITLE5 '- TRANS CODE OUTPUT -';
TITLE7 ' ';
*****
* PRINT BY LINE *
*****;
PROC SORT DATA=TRANS; BY SYS AIC LINE;
PROC SUMMARY DATA=TRANS; BY SYS AIC LINE;
OUTPUT OUT=REP600 SUM=;
ID LINENAME;
VAR RHAT PHAT WHAT;
DATA REP600; SET REP600;
IF LINE=. & (RHAT=0 & PHAT=0) THEN DELETE;
IF PHAT>0 THEN RPP=RHAT/PHAT;
IF PHAT>0 THEN WPP=WHAT/PHAT;
IF WHAT>0 THEN RLB=RHAT/WHAT;
PROC PRINT DATA=REP600 U; BY SYS AIC; PAGEBY SYS;
FORMAT RHAT CHAT WHAT COMMA13. RPP RLB WPP 6.3
PHAT COMMA14.;
ID SYS;
VAR AIC LINE LINENAME RHAT PHAT WHAT RPP RLB WPP;
SUM RHAT PHAT WHAT;
TITLE3 'R3-600: ADJUSTED ESTIMATES - BY 2-PAGE LINE NO.';
TITLE5 ' ';

```

/*

00260000

```

//NNNNNNS JOB (ALA03), 'NNNNNNN NNN NN', CLASS=A, MSGCLASS=H          00010008
/*ROUTE PRINT U5704                                                    00030000
//*****00040000
/**          JOB NAME:  PDS.SASRPW(RPWJOB4) - FORMERLY "J4RRUN97" USED 00060000
//*****00060000
/**
/**          I N S T R U C T I O N S          00050003
/**
/**          1. UPDATE DCB LRECL AND BLKSIZE PARAMETERS (IF CHANGED).    00140000
/**          2. ODSN FILES:  ENTER CHANGE COMM:  "CHANGE FY19YYPQ FY19YYPQ ALL" 00140000
/**          "CHANGE FY19YYAP FY19YYAP ALL" 00140000
/**          3. IDSN FILES:  UPDATE INDIVIDUALLY (AS REQUIRED)             00140000
/**          4. ENTER EMPTY (PLACEHOLDER) FILES: COPY "HSQRAN.BRPWD01.EMPTY" TO 00050003
/**          "HSQRAN.BRPWT01.EMPTY.SXX.FY19YYQY" ("SXX" IS STEP NUMBER).    00050003
/**          5. SUBMIT JOB:                                               00140000
/**          (1) CHECK SYSOUT (ALL STEPS) FOR COND CODE 0000;           00050003
/**          FOR RERUN - DELETE FILES WITH TSO DELETE & "NOS" PARAM      00050003
/**          AND CALL TAPE LIBRARY TO "SCRATCH" FROM TMS (OR TEST        00050003
/**          USING RETPD=1 FOR 1-DAY).                                     00050003
/**          (2) "NOT CATLG" INDICATES SAME NAME FILE PREVIOUSLY COPIED.  00050003
/**          (IGNORE IF COND CODE =0000, OR CORRECT AND RERUN).          00050003
/**          (3) USE TSO "TMS" (TAPE MGMT SYS) AND VOLSER NO. TO VERIFY.  00050003
//*****00100000
/**
/**
/**
//COPIES   PROC          00050003
//STEP1    EXEC PGM=IEBGENER 00054000
//SYSPRINT DD DUMMY      00055000
//SYSIN    DD DUMMY      00056000
//SYSUT1   DD DISP=OLD,   00057000
//          DSN=&IDSN     00060000
//SYSUT2   DD UNIT=TAPE,  00070005
//          DISP=(NEW,CATLG), 00080000
//          VOL=(,RETAIN,,REF=*.S01.STEP1.SYSUT2), 00090004
//          LABEL=(&LABNUM.,SL,,RETPD=1825), 00100000
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=1600), 00110000
//          DSN=&ODSN     00120000
//COPIES   PEND          00130000
/**
/**

```

```

//* 00130000
//***** 00360000
//*          BEG PROGRAMS * 00360000
//***** 00360000
//S01      EXEC COPIES, 00150000
//          LABNUM=1, 00160000
//          IDSN='NNNNNN.PDS.SASRPW(RPWJOB1)', 00170000
//          ODSN='HSQRAN.BRPWT01.RPWJOB1.FY2000Q4' 00180000
//STEP1.SYSUT2 DD VOL= 00190004
//*** 00360000
//*** 00360000
//S02      EXEC COPIES, 00200000
//          LABNUM=2, 00210000
//          IDSN='NNNNNN.PDS.SASRPW(RPWJOB2)', 00220000
//          ODSN='HSQRAN.BRPWT01.RPWJOB2.FY2000Q4' 00180000
//*** 00360000
//*** 00360000
//S03      EXEC COPIES, 00200000
//          LABNUM=3, 00210000
//          IDSN='NNNNNN.PDS.SASRPW(RPWJOB3)', 00220000
//          ODSN='HSQRAN.BRPWT01.RPWJOB3.FY2000Q4' 00180000
//*** 00360000
//*** 00360000
//S04      EXEC COPIES, 00320000
//          LABNUM=4, 00330000
//          IDSN='NNNNNN.PDS.SASRPW(RPWJOB4)', 00220000
//          ODSN='HSQRAN.BRPWT01.RPWJOB4.FY2000Q4' 00180000
//***** 00360000
//*          END PROGRAMS * 00360000
//***** 00360000
//* 00360000
//* 00360000
//* 00360000
//***** 00360000
//*          BEG JOB-1 * 00360000
//***** 00360000
//* NOTE: S05-S08 MANUAL OFFICE DATA FILES FOR EACH AP - USE DUMMY. 00370000
//S05      EXEC COPIES, 00360000
//          LABNUM=5, 00370000
//          IDSN='HSQRAN.BRPWD01.MANUALS.FY200010', 00580000
//          ODSN='HSQRAN.BRPWT01.MANUALS.FY200010' 00390000

```

```

//STEP1.SYSUT2 DD DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200) 00831001
//*** 00360000
//*** 00360000
//S06 EXEC COPIES, 00360000
// LABNUM=6, 00370000
// IDSN='HSQRAN.BRPWD01.MANUALS.FY200011', 00580000
// ODSN='HSQRAN.BRPWT01.MANUALS.FY200011' 00390000
//STEP1.SYSUT2 DD DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200) 00831001
//*** 00360000
//*** 00360000
//S07 EXEC COPIES, 00360000
// LABNUM=7, 00370000
// IDSN='HSQRAN.BRPWD01.MANUALS.FY200012', 00580000
// ODSN='HSQRAN.BRPWT01.MANUALS.FY200012' 00390000
//* IDSN='HSQRAN.BRPWD01.EMPTY', 00580000
//* ODSN='HSQRAN.BRPWT01.EMPTY.S07.FY2000Q4' 00390000
//STEP1.SYSUT2 DD DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200) 00831001
//*** 00360000
//*** 00360000
//S08 EXEC COPIES, 00360000
// LABNUM=8, 00370000
// IDSN='HSQRAN.BRPWD01.MANUALS.FY200013', 00580000
// ODSN='HSQRAN.BRPWT01.MANUALS.FY200013' 00390000
//* IDSN='HSQRAN.BRPWD01.EMPTY', 00580000
//* ODSN='HSQRAN.BRPWT01.EMPTY.S08.FY2000Q4' 00580000
//STEP1.SYSUT2 DD DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200) 00831001
//*** 00360000
//*** 00360000
//S09 EXEC COPIES, 00440000
// LABNUM=9, 00450000
//* NOTE: S09 NO CHANGE TO F.Y.1.9.9.6.Q.1 UNLESS PANEL CHANGE MADE. 00380000
// IDSN='HSQRAN.BRPWD01.MANUALS.FRMQ199', 00380000
// ODSN='HSQRAN.BRPWT01.MANUALS.FRMQ199.FY2000Q4' 00380000
//STEP1.SYSUT2 DD DCB=(RECFM=FB,LRECL=120,BLKSIZE=2400) 00831001
//*** 00360000
//*** 00360000
//* NOTE: S10-S13 FRAME FILE FOR EACH AP (USE DUMMY) 00360000
//S10 EXEC COPIES, 00480000
// LABNUM=10, 00490000
// IDSN='HSISMN.BRPWD01.OFFLIST.FY200010(0)', 00380000
// ODSN='HSQRAN.BRPWT01.OFFLIST.FY200010' 00390000

```

```

//STEP1.SYSUT2 DD DCB=(RECFM=FB,LRECL=80,BLKSIZE=1600) 00831001
//*** 00360000
//*** 00360000
//S11 EXEC COPIES, 00480000
// LABNUM=11, 00490000
// IDSN='HSISMN.BRPWD01.OFFLIST.FY200011(0)', 00380000
// ODSN='HSQRAN.BRPWT01.OFFLIST.FY200011' 00390000
//STEP1.SYSUT2 DD DCB=(RECFM=FB,LRECL=80,BLKSIZE=1600) 00831001
//*** 00360000
//*** 00360000
//S12 EXEC COPIES, 00480000
// LABNUM=12, 00490000
// IDSN='HSISMN.BRPWD01.OFFLIST.FY200012(0)', 00380000
// ODSN='HSQRAN.BRPWT01.OFFLIST.FY200012' 00390000
//STEP1.SYSUT2 DD DCB=(RECFM=FB,LRECL=80,BLKSIZE=1600) 00831001
//*** 00360000
//*** 00360000
//S13 EXEC COPIES, 00480000
// LABNUM=13, 00490000
// IDSN='HSISMN.BRPWD01.OFFLIST.FY200013(0)', 00380000
// ODSN='HSQRAN.BRPWT01.OFFLIST.FY200013' 00390000
//* IDSN='HSQRAN.BRPWD01.EMPTY', 00580000
//* ODSN='HSQRAN.BRPWT01.EMPTY.S13.FY2000Q4' 00580000
//STEP1.SYSUT2 DD DCB=(RECFM=FB,LRECL=80,BLKSIZE=1600) 00831001
//*** 00360000
//*** 00360000
//S14 EXEC COPIES, 00520000
// LABNUM=14, 00530000
// IDSN='HSQRAN.BRPWD01.EDITDAT1.FY2000Q4', 00550000
// ODSN='HSQRAN.BRPWT01.EDITDAT1.FY2000Q4' 00550000
//STEP1.SYSUT2 DD DCB=(RECFM=FB,LRECL=125,BLKSIZE=5000) 00831001
//***** 00360000
//* END JOB-1 * 00360000
//***** 00360000
//* 00360000
//* 00360000
//* 00360000
//***** 00360000
//* BEG JOB-2 * 00360000
//***** 00360000
//* NOTE: S15-S26 FOUR FILES FOR EACH MAIL CLASS (USE DUMMY) 00560000

```

```

/*          USE 'HSQRAN.BRPWD01.EMPTY' FOR DUMMY          00580000
//S15      EXEC COPIES,          00560000
//          LABNUM=15,          00570000
//          IDSN='HSQRAN.BRPWD01.RATES1C.JAN1099',        00580000
//          ODSN='HSQRAN.BRPWT01.RATES1C.JAN1099.FY2000Q4' 00580000
//STEP1.SYSUT2 DD DCB=(RECFM=FB,LRECL=132,BLKSIZE=13200) 00831001
//***          00560000
//***          00560000
//S16      EXEC COPIES,          00560000
//          LABNUM=16,          00570000
//          IDSN='HSQRAN.BRPWD01.RATES1C.OCT0498',        00580000
//          ODSN='HSQRAN.BRPWT01.RATES1C.OCT0498.FY2000Q4' 00580000
//STEP1.SYSUT2 DD DCB=(RECFM=FB,LRECL=132,BLKSIZE=13200) 00831001
//***          00560000
//***          00560000
//S17      EXEC COPIES,          00560000
//          LABNUM=17,          00570000
//          IDSN='HSQRAN.BRPWD01.RATES1C.OCT0597',        00580000
//          ODSN='HSQRAN.BRPWT01.RATES1C.OCT0597.FY2000Q4' 00580000
//STEP1.SYSUT2 DD DCB=(RECFM=FB,LRECL=132,BLKSIZE=13200) 00831001
//***          00560000
//***          00560000
//S18      EXEC COPIES,          00560000
//          LABNUM=18,          00570000
//          IDSN='HSQRAN.BRPWD01.EMPTY',                  00580000
//          ODSN='HSQRAN.BRPWT01.EMPTY.S18.FY2000Q4'      00580000
//STEP1.SYSUT2 DD DCB=(RECFM=FB,LRECL=80,BLKSIZE=1600)   00831001
//*TEP1.SYSUT2 DD DCB=(RECFM=FB,LRECL=132,BLKSIZE=13200) 00831001
//***          00560000
//***          00560000
//S19      EXEC COPIES,          00560000
//          LABNUM=19,          00570000
//          IDSN='HSQRAN.BRPWD01.RATESPD.JAN1099',        00580000
//          ODSN='HSQRAN.BRPWT01.RATESPD.JAN1099.FY2000Q4' 00580000
//STEP1.SYSUT2 DD DCB=(RECFM=FB,LRECL=132,BLKSIZE=13200) 00831001
//***          00560000
//***          00560000
//S20      EXEC COPIES,          00560000
//          LABNUM=20,          00570000
//          IDSN='HSQRAN.BRPWD01.RATESPD.OCT0498',        00580000
//          ODSN='HSQRAN.BRPWT01.RATESPD.OCT0498.FY2000Q4' 00580000

```

```

//STEP1.SYSUT2 DD DCB=(RECFM=FB,LRECL=132,BLKSIZE=13200)
//***
//***
//S21 EXEC COPIES,
// LABNUM=21,
// IDSN='HSQRAN.BRPWD01.RATESPD.OCT0597',
// ODSN='HSQRAN.BRPWT01.RATESPD.OCT0597.FY2000Q4'
//STEP1.SYSUT2 DD DCB=(RECFM=FB,LRECL=132,BLKSIZE=13200)
//***
//***
//S22 EXEC COPIES,
// LABNUM=22,
// IDSN='HSQRAN.BRPWD01.EMPTY',
// ODSN='HSQRAN.BRPWT01.EMPTY.S22.FY2000Q4'
//STEP1.SYSUT2 DD DCB=(RECFM=FB,LRECL=80,BLKSIZE=1600)
//*TEP1.SYSUT2 DD DCB=(RECFM=FB,LRECL=132,BLKSIZE=13200)
//***
//***
//S23 EXEC COPIES,
// LABNUM=23,
// IDSN='HSQRAN.BRPWD01.RATESSA.JAN1099',
// ODSN='HSQRAN.BRPWT01.RATESSA.JAN1099.FY2000Q4'
//STEP1.SYSUT2 DD DCB=(RECFM=FB,LRECL=132,BLKSIZE=13200)
//***
//***
//S24 EXEC COPIES,
// LABNUM=24,
// IDSN='HSQRAN.BRPWD01.RATESSA.OCT0498',
// ODSN='HSQRAN.BRPWT01.RATESSA.OCT0498.FY2000Q4'
//STEP1.SYSUT2 DD DCB=(RECFM=FB,LRECL=132,BLKSIZE=13200)
//***
//***
//S25 EXEC COPIES,
// LABNUM=25,
// IDSN='HSQRAN.BRPWD01.RATESSA.OCT0597',
// ODSN='HSQRAN.BRPWT01.RATESSA.OCT0597.FY2000Q4'
//STEP1.SYSUT2 DD DCB=(RECFM=FB,LRECL=132,BLKSIZE=13200)
//***
//***
//S26 EXEC COPIES,
// LABNUM=26,

```

```

00831001
00560000
00560000
00560000
00560000
00570000
00580000
00580000
00831001
00560000
00560000
00560000
00570000
00580000
00580000
00831001
00831001
00560000
00560000
00560000
00570000
00580000
00580000
00831001
00831001
00560000
00560000
00560000
00570000
00580000
00580000
00831001
00831001
00560000
00560000
00560000
00570000
00580000
00580000
00831001
00831001
00560000
00560000
00560000
00570000
00580000
00580000
00831001
00831001
00560000
00560000
00560000
00570000

```

```

//          IDSN='HSQRAN.BRPWD01.EMPTY',                00580000
//          ODSN='HSQRAN.BRPWT01.EMPTY.S26.FY2000Q4'    00580000
//STEP1.SYSUT2 DD DCB=(RECFM=FB,LRECL=80,BLKSIZE=1600) 00831001
//*TEP1.SYSUT2 DD DCB=(RECFM=FB,LRECL=132,BLKSIZE=13200)00831001
//***                                                00560000
//***                                                00560000
//S27      EXEC COPIES,                                00560000
//          LABNUM=27,                                  00570000
//          IDSN='HSQRAN.BRPWD01.RATESSB.JAN1099',      00580000
//          ODSN='HSQRAN.BRPWT01.RATESSB.JAN1099.FY2000Q4'00580000
//STEP1.SYSUT2 DD DCB=(RECFM=FB,LRECL=132,BLKSIZE=13200)00831001
//***                                                00560000
//***                                                00560000
//S28      EXEC COPIES,                                00560000
//          LABNUM=28,                                  00570000
//          IDSN='HSQRAN.BRPWD01.RATESSB.OCT0498',      00580000
//          ODSN='HSQRAN.BRPWT01.RATESSB.OCT0498.FY2000Q4'00580000
//STEP1.SYSUT2 DD DCB=(RECFM=FB,LRECL=132,BLKSIZE=13200)00831001
//***                                                00560000
//***                                                00560000
//S29      EXEC COPIES,                                00560000
//          LABNUM=29,                                  00570000
//          IDSN='HSQRAN.BRPWD01.RATESSB.OCT0597',      00580000
//          ODSN='HSQRAN.BRPWT01.RATESSB.OCT0597.FY2000Q4'00580000
//STEP1.SYSUT2 DD DCB=(RECFM=FB,LRECL=132,BLKSIZE=13200)00831001
//***                                                00560000
//***                                                00560000
//S30      EXEC COPIES,                                00560000
//          LABNUM=30,                                  00570000
//          IDSN='HSQRAN.BRPWD01.EMPTY',                00580000
//          ODSN='HSQRAN.BRPWT01.EMPTY.S30.FY2000Q4'    00580000
//STEP1.SYSUT2 DD DCB=(RECFM=FB,LRECL=80,BLKSIZE=1600) 00831001
//*TEP1.SYSUT2 DD DCB=(RECFM=FB,LRECL=132,BLKSIZE=13200)00831001
//***                                                00560000
//***                                                00560000
//S31      EXEC COPIES,                                00600000
//          LABNUM=31,                                  00610000
//          IDSN='HSQRAN.BRPWD01.EDITDAT2.FY2000Q4',    00630000
//          ODSN='HSQRAN.BRPWT01.EDITDAT2.FY2000Q4'    00630000
//STEP1.SYSUT2 DD DCB=(RECFM=FB,LRECL=150,BLKSIZE=6000)00831001
//***** 00360000

```

```

/*                               END JOB-2                               * 00360000
/******                          * 00360000
/*                               * 00360000
/*                               * 00360000
/******                          * 00360000
/*                               BEG JOB-3                               * 00360000
/******                          * 00360000
//S32      EXEC COPIES,                                               00640000
//          LABNUM=32,                                               00650000
//          IDSN='HSISMN.RPW.FY2000.CATEGORY.DIR',                   00660000
//          ODSN='HSQRAN.BRPWT01.FY2000.CATEGORY.DIR.FY2000Q4'     00660000
//STEP1.SYSUT2 DD DCB=(RECFM=FB,LRECL=110,BLKSIZE=3300)             00831001
//***                                               00360000
//***                                               00360000
//S33      EXEC COPIES,                                               00640000
//          LABNUM=33,                                               00650000
//          IDSN='HSISMN.RPW.ADJ.CNTL2000(SUMDIR)',                   00660000
//          ODSN='HSQRAN.BRPWT01.ADJ.CNTL2000.SUMDIR.FY2000Q4'     00660000
//STEP1.SYSUT2 DD DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120)             00831001
//***                                               00360000
//***                                               00360000
//***                                               00360000
//***                                               00360000
//S34      EXEC COPIES,                                               00680000
//          LABNUM=34,                                               00690000
//          IDSN='HSISMN.RPW.REVCNTL.FY200010',                       00710002
//          ODSN='HSQRAN.BRPWT01.REVCNTL.FY200010'                   00710002
//STEP1.SYSUT2 DD DCB=(RECFM=FB,LRECL=110,BLKSIZE=2200)           00831001
//***                                               00360000
//***                                               00360000
//S35      EXEC COPIES,                                               00680000
//          LABNUM=35,                                               00690000
//          IDSN='HSISMN.RPW.REVCNTL.FY200011',                       00710002
//          ODSN='HSQRAN.BRPWT01.REVCNTL.FY200011'                   00710002
//STEP1.SYSUT2 DD DCB=(RECFM=FB,LRECL=110,BLKSIZE=2200)           00831001
//***                                               00360000
//***                                               00360000
//S36      EXEC COPIES,                                               00680000
//          LABNUM=36,                                               00690000
//          IDSN='HSISMN.RPW.REVCNTL.FY200012',                       00710002

```

```

//          ODSN='HSQRAN.BRPWT01.REVCNTL.FY200012'          00710002
//STEP1.SYSUT2 DD DCB=(RECFM=FB,LRECL=110,BLKSIZE=2200)    00831001
//***                                                    00360000
//***                                                    00360000
//S37      EXEC COPIES,          00680000
//          LABNUM=37,          00690000
//          IDSN='HSISMN.RPW.REVCNTL.FY200013',          00710002
//          ODSN='HSQRAN.BRPWT01.REVCNTL.FY200013'          00710002
//STEP1.SYSUT2 DD DCB=(RECFM=FB,LRECL=110,BLKSIZE=2200)    00831001
//*          IDSN='HSQRAN.BRPWD01.EMPTY',          00580000
//*          ODSN='HSQRAN.BRPWT01.EMPTY.S37.FY2000Q4'          00710002
//*TEP1.SYSUT2 DD DCB=(RECFM=FB,LRECL=80,BLKSIZE=1600)      00831001
//***                                                    00360000
//***                                                    00360000
//S38      EXEC COPIES,          00640000
//          LABNUM=38,          00650000
//          IDSN='HSQRAN.BRPWD01.INFLATE1.FY2000Q4',          00670000
//          ODSN='HSQRAN.BRPWT01.INFLATE1.FY2000Q4'          00670000
//STEP1.SYSUT2 DD DCB=(RECFM=FB,LRECL=240,BLKSIZE=4800)    00831001
//***                                                    00360000
//***                                                    00360000
//S39      EXEC COPIES,          00680000
//          LABNUM=39,          00690000
//          IDSN='HSQRAN.BRPWD01.TRANS.FY2000Q4',          00710002
//          ODSN='HSQRAN.BRPWT01.TRANS.FY2000Q4'          00710002
//STEP1.SYSUT2 DD DCB=(RECFM=FB,LRECL=80,BLKSIZE=2400)      00831001
//*****                                                    00360000
//*          END JOB-3          * 00360000
//*****                                                    00360000
/*          * 00360000

```

-