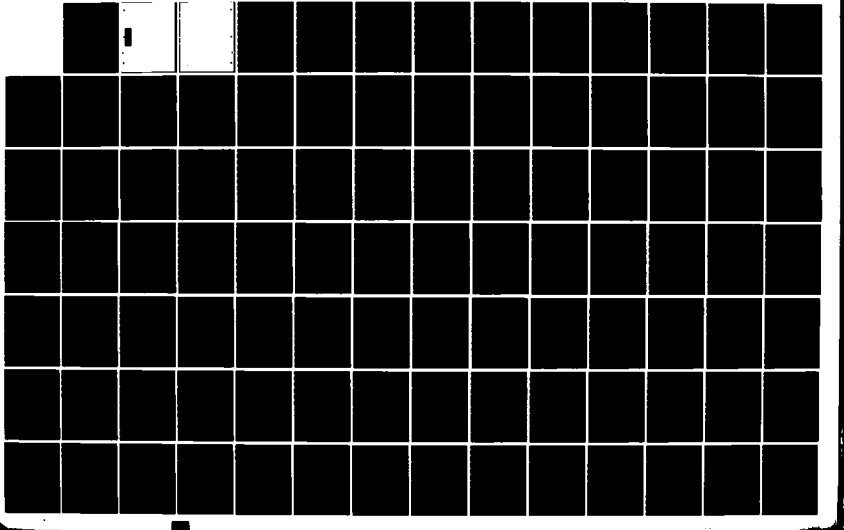
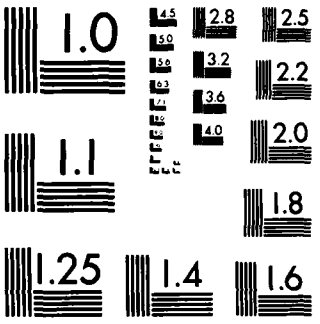
AD-A149 505 PERSONNEL READINESS INDICATOR MODEL (PRIM) DOCUMENTATION PROGRAM MAINTENANCE MANUAL(U) ARMY CONCEPTS ANALYSIS AGENCY BETHESDA MD

1/4

UNCLASSIFIED S J VAN NOSTRAND ET AL. NOV 84 CAA-D-84-2 F/G 9/2 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER CAA-D-84-2	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Personnel Readiness Indicator Model (PRIM) Program Maintenance Manual		5. TYPE OF REPORT & PERIOD COVERED Model Documentation 6. PERFORMING ORG. REPORT NUMBER CAA-D-84-2
7. AUTHOR(s) Sally J. Van Nostrand David Stevens Emma Duffy Adele Narva		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS US Army Concepts Analysis Agency 8120 Woodmont Avenue Bethesda, MD 20814-2797		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Deputy Chief of Staff for Personnel Department of the Army Washington, DC 20310		12. REPORT DATE November 1984 13. NUMBER OF PAGES 402
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) US Army Military Personnel Center 200 Stovall Street Alexandria, VA 22332		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Approved for public release; distribution unlimited.

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Personnel; Manning; Assignment; Model; Policy; Planning

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

PRIM is a computer model which distributes projected personnel inventories to projected Army jobs and measures the resulting readiness. Personnel assignment policies are simulated and the results provide the means for evaluating the long-term effects of the policies. Other PRIM publications are the PRIM Study Report ~~(CAA-SR-84-5)~~; PRIM Functional Description ~~(CAA-D-84-1)~~ and PRIM User Manual ~~(CAA-D-84-3)~~. PRIM has been installed at the US Army Military Personnel Center (MILPERCEN).

p1-1

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

(NOT USED)

PERSONNEL READINESS INDICATOR MODEL
(PRIM)
DOCUMENTATION
PROGRAM MAINTENANCE MANUAL

NOVEMBER 1984

PREPARED BY
FORCE SYSTEMS DIRECTORATE

US ARMY CONCEPTS ANALYSIS AGENCY
8120 WOODMONT AVENUE
BETHESDA, MARYLAND 20814-2797

A-1

CONTENTS

APPENDIX **Page**

FIGURES

FIGURE

TABLES

PERSONNEL READINESS INDICATOR MODEL (PRIM)
PROGRAM MAINTENANCE MANUAL

SECTION 1

GENERAL DESCRIPTION

1 on 1473

1.1 PURPOSE OF PROGRAM MAINTENANCE MANUAL. The objective of the Program Maintenance Manual for the Personnel Readiness Indicator Model (PRIM) is to provide the maintenance programer with the information necessary to effectively maintain the system.

1.2 PROJECT REFERENCES

a. Study Directive, subject: Personnel Readiness Indicator Model (PRIM) Study, 26 September 1983.

b. FORECAST - A study designed to improve and replace many the enlisted and officer inventory projections and distribution models. FORECAST does not directly address personnel readiness.

c. Army Regulation 220-1, Unit Status Reporting.

d. Personnel Readiness Indicator Model (PRIM) User Manual, CAA-D-84-3.

e. Personnel Readiness Indicator Model (PRIM) Study Report, CAA-SR-84-5.

f. The referenced policy guidance memorandums were used to determine the types of policies which were modeled by PRIM. The PRIM design is not dependent upon the specific policies stated in the following:

(1) DCSPER Memorandum, subject: Enlisted Distribution Policy Guidance, 4 October 1982.

(2) DCSOPS Memorandum, subject: Policy Guidance: Personnel Fill for Force Modernization, 13 September 1982.

(3) DCSPER Letter, subject: Officer Distribution Policy Guidance, FY 83, 4 October 1982.

1.3 TERMS AND ABBREVIATIONS

activity	Pay grade of an MOS at a particular demand.
AR 220-1	Unit status reporting regulation for reporting the current status of selected Active and Reserve Component units.

arc	The hypothetical link between the jobs and personnel inventory along which the people can be sent. The number of people assigned to a specific job is called the flow on the arc.
ASCII	American Standard Code for Information Interchange.
assignment code	Major command or DA staff agency to which the unit is assigned. The abbreviation is ASGMT.
assignment policy	see personnel policy.
AUDB	Authorization Data Base is the MILPERCEN-developed data base for authorizations by UIC, MOS/SC, and grade for modification tables of organization and equipment (MTOE), table of distribution and allowances (TDA), and TDA augmentation organizations.
authorized jobs	The number of jobs that should be filled during peacetime. This number is frequently constrained to a smaller number of jobs than the required jobs. Either authorized or required jobs may be specified as the jobs for PRIM.
available MOS percentage	The available MOS-trained strength divided by the structure MTOE strength and converted to a percentage.
available people	Those personnel assigned to an ISSUE and who are available for duty rather than sick, on leave, or unavailable for duty for other specified reasons. For complete list see Appendix B of AR 220-1.
available senior-grade percentage	The total number of available commissioned officers, warrant officers, and E-5 to E-9 enlisted grades, divided by the total structure number of commissioned officers, warrant officers, and grades E-5 to E-9 converted to a percentage.
available strength	The total available strength divided by the structure MTOE strength and converted to a percentage.
CAA	US Army Concepts Analysis Agency.

C-rating	The personnel readiness rating for an ISSUE based upon criteria in AR 220-1. It is the lowest of three ratings: senior grade, available MOS-trained, or available strength.
CAP III	System currently used for enlisted assignments.
demand	Real job or super job to which people must be assigned.
DOPMA	Defense Officers Personnel Management Act.
excess people	The number of people, by MOS and grade for which there was no demand.
fill	The number of assignments made.
flow	The number of people assigned to a job.
goal percentage	The minimum percentage of total aggregate fill.
high 5 or high five	The highest five enlisted grades, E-5, E-6, E-7, E-8, and E-9.
inventory	See projected personnel inventories.
ISSUE	An aggregation of units for the purpose of highlighting Individual Systems, Similar Units, or Equipment.
job data	The jobs which will have personnel assigned to them by the assignment processor. The jobs are of a specific MOS or SC, grade level, and may be either structured or authorized jobs.
MACOM	Major Army command.
MILPERCEN	US Army Military Personnel Center.
MOS	Military occupational specialty is a code representing the type of skill in which enlisted personnel or warrant officers have been trained and should be assigned to perform. Also used in the PRIM job data file to describe the skill the person, whether enlisted, warrant officer, or commissioned officer, assigned to a job must possess.
MTOE	Modification Tables of Organization and Equipment.

number of people	The set of combined enlisted and officer personnel inventory.
ODCSPER	Office of the Deputy Chief of Staff for Personnel.
OFIP	Officer Force Implementation Plan.
P3M	Personnel Policy Projection Model is used by MILPERCEN to develop inventories for use as a personnel data. It computes an MOS and grade level, enlisted personnel inventory, and reenlistment projection while taking into consideration personnel management policy options.
parameters	Data values which are specific to each run and must be set by the user.
personnel data	The number of people (officers and enlisted) by MOS or SC grade that MILPERCEN expects will be available; the output of the P3M Model for enlisted personnel, the OFIP for commissioned officer personnel, and the WOFIP for warrant officer personnel.
personnel policies	In PRIM, policies are the statement of minimum and maximum personnel fill levels which have been in effect or are being tested prior to implementation. Policies may apply to all MOS, specific MOS, or specific grade levels. Policies modeled by PRIM must be converted to computer-readable format. See personnel policies.
PNET	A mathematical programing system for solving network flow problems.
projected MTOE/TDA authorizations	The number of people, by specialty and grade, that will be needed in each unit (3-digit UIC) at a future time.
projected personnel inventories	The number of people, by specialty and grade, who will be in the Active Army at a future time.
required jobs	In PRIM, the jobs that should be filled. The user chooses structure or authorized.
SC	Commissioned code officers specialty.
senior grade	Personnel in grades E-5 through E-9, WO, and O1 through O6.

structure strength	The jobs that must be filled if a unit is to achieve maximum readiness. Structure MTOE strengths are used in PRIM for computation of C-ratings, IAW AR 220-1.
super job	Imaginary job where personnel are assigned when no "real" jobs are left to be filled.
super soldier	Imaginary person assigned to job when no person fitting requirements is available. Also known as super person.
TDA	Table of distribution and allowance.
UIC	Unit identification code. Although the UIC is normally six characters, the PRIM UIC is the "3-digit UIC" (characters 2-4 of UIC) unless otherwise stated.
unavailable factors	A standard factor representing the percentage of people not available for duty, input through the parameter file.
unfilled jobs	The jobs that have not had people assigned by PRIM.
WOFIP	Warrant Officer Force Implementation Plan.

SECTION 2

SYSTEM APPLICATION

2.1 SYSTEM DESCRIPTION. The introduction of new organizational systems into the Army generates major changes in the force structure. Assigning properly trained soldiers in a timely manner, coinciding with the fielding of new equipment, requires intense management and long lead times. The personnel inventories required to operate future systems are, therefore, the product of today's decisions to commit resources; data for these decisions should include knowledge of future requirements. To this end, decisionmakers require a tool with which to assess the Army's ability to provide personnel support for the modernization process. The US Army Military Personnel Center (MILPERCEN) can currently project inventories by military occupational speciality/speciality code (MOS/SC) and grade for the total Active component of the Army. However, there is not an adequate system for evaluating the impact of personnel policy changes on the future levels of personnel fill in units. As a result, conflicting guidance may be issued and personnel policies may not be developed providing the highest potential level of personnel readiness on a global basis.

There is currently no way to link macro-level MOS projections to unit-level predictions for readiness rating in the same period. PRIM will aid in determining the effects of various end strengths and distribution policies on unit readiness postures by showing the increase or decrease of the personnel fills, and the resulting C-ratings based on parameters in AR 220-1 (e.g., top three NCO levels and total fills). The PRIM distributes projections of Active Army personnel strengths by grade and specialty to units and logical groups of units. The PRIM reports provide the personnel readiness measures specified in AR 220-1. In addition, other personnel information such as projected unit fill percentages is provided.

AR 220-1 specifies three types of readiness: training, equipment, and personnel. The Personnel Readiness Indicator Model predicts unit personnel readiness status. Recently, CAA developed a model for the Office of the Deputy Chief of Staff for Logistics (ODCSLOG) called E-DATE to model equipment readiness, and PRIM will model personnel readiness for the Office of the Deputy Chief of Staff for Personnel (ODCSPER). Currently, there is no model for training readiness.

2.2 SECURITY AND PRIVACY

a. Security. All program code and listings are considered UNCLASSIFIED and require no special security considerations. Outputs will be classified at the same level as the classification of the unit aggregation specifications. In normal use, these will be UNCLASSIFIED.

b. Privacy. The personnel data used as input to PRIM is output of an inventory projection model and does not contain information on individual people. Therefore, this section is not applicable.

2.3. GENERAL DESCRIPTION. The PRIM is divided into six separate free-standing processors that are executed sequentially in a batch environment to produce the final reports. Figure 2-1 shows the system organization. Results of all processors are stored in mass storage files. The complete PRIM system consists of 6 program files, 2 runstream files, and 62 data files. Many of the files may be temporary files, and no runstream will require that all files are resident at one time. A complete run of PRIM requires a minimum of 12 separate, sequential runs. The user may choose to run these as one long runstream of executions or to run each separately, correcting input errors and rerunning prior to attempting the next step. A list of the errors recognized by the Preprocessor and the Policy Processor will be found in Appendix E.

a. The initial processor, the Preprocessor, creates the PRIM data base by rolling the UIC-level information on projected jobs into ISSUE-level information to create the MOS-data file which contains the demand (job) data. (ISSUE is further discussed in Appendix C.) The Preprocessor contains four main programs and nine subroutines, plus utility programs. The function of each of these routines is explained in the Program Description, subsection 2.4.

b. The second processor, the Policy Processor, performs three functions. First, it edits the user-prepared Policy file. Then it applies the personnel policies to the MOS-data file, rewriting the applicable records with the minimum and maximum fill levels and assignment values to the Job Assignment Value (JAV) file which is needed by the Assignment Processor. Finally, the remaining MOS-data records are rewritten using information from the Value file. Since the time required to run all segments of the Policy Processor could be an hour or more, depending upon the size of the input MOS-data file and the number of policies, and since the user may want to make changes to the Policy and Value files, the Policy Processor is designed to run as a set of five independent, sequential runs. These runs use 5 main programs and 35 subroutines. The Policy and Value files are described in Appendix C; the Policy Processor is described in detail in section 2.5.

c. The third processor, the Assignment Processor contains 1 main program, 5 major subroutines, and 10 other subroutines. The Assignment Processor uses the Number-of-People file and the Job Assignment Value file to create the data it requires for generating the personnel assignments. The Assignment Processor converts the input files to network input format, makes the personnel assignments, and saves the assignment results for each specialty separately, cycling through all steps for each MOS or SC. The programs are described in section 2.6.

d. The fourth processor, the Substitute Assignment Processor, assigns excess personnel to unfilled jobs. This processor consists of two modules, one which assigns using grade substitutions and one which assigns using specialty substitutions. The processing in each module is similar to the other and to the Assignment Processor; the network portion is duplicated in each. See section 2.7 for the detailed program description.

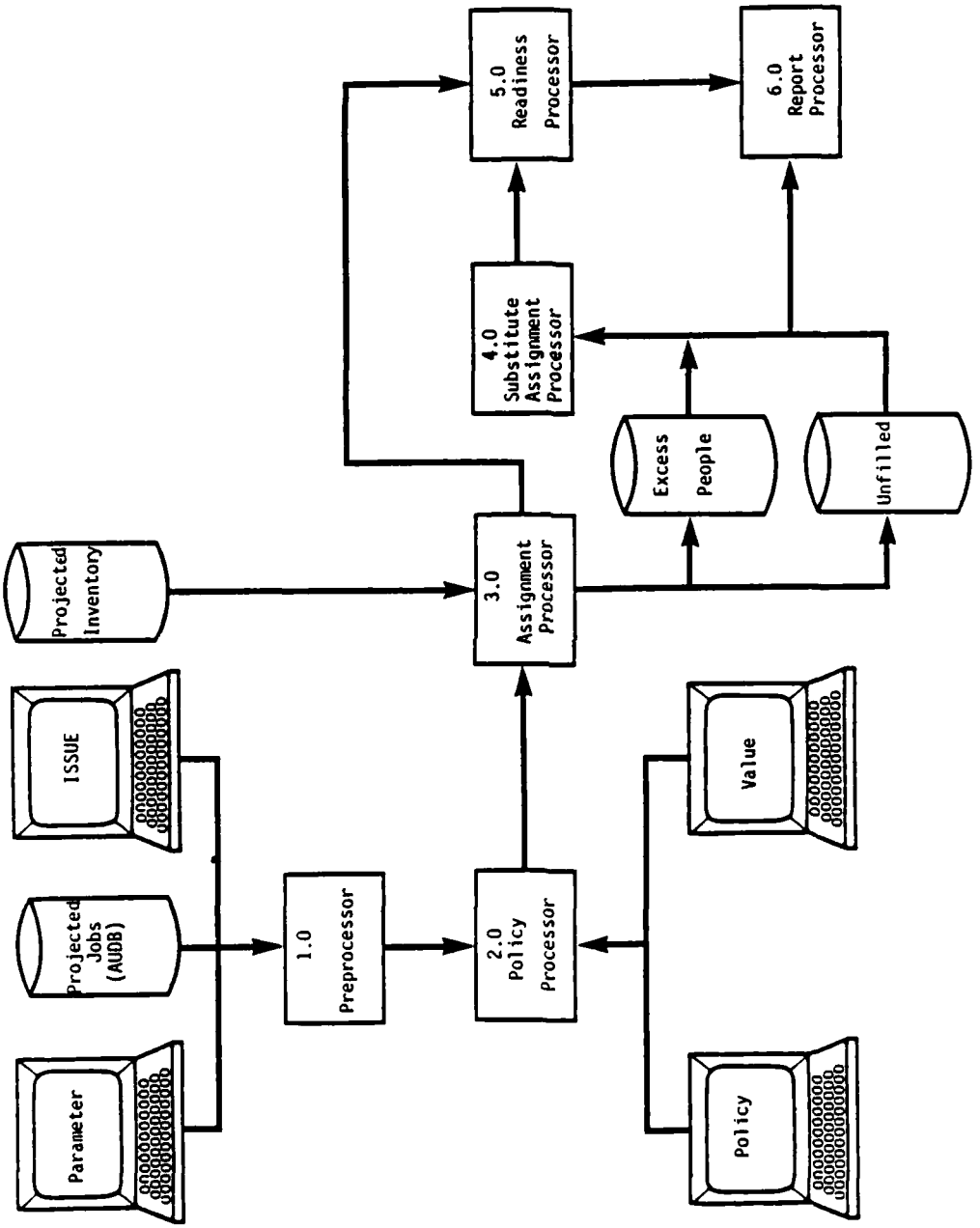


Figure 2-1. System Organization

e. The fifth processor, the Readiness Processor, applies the criteria from AR 220-1 to develop the readiness measures and computes additional indicators for use by MILPERCEN. The processing is explained in the Program Description, subsection 2.8.

f. The final processor, the Report Processor, provides a variety of user designed reports. More detail will be found in section 2.9.

g. Several utility-type subroutines are used by more than one processor. Examples of the functions performed by these subroutines are:

(1) Convert the run date to readable format.

(2) Convert alphanumerics to integer or to a real number.

(3) Write a message to the Error Print file.

(4) Temporarily modify the system to allow ASCII FORTRAN to read and write file numbers from zero through 99.

2.4 PROGRAM DESCRIPTION - PREPROCESSOR. The Preprocessor consists of a set of three subprocessors, one of which has been further divided into two modules. Therefore, a complete run of the processor requires running four separate programs plus three sorts using the SORT utility program. Subsection 2.4 contains a description of the four modules of the Preprocessor. In addition to the four main programs, one for each module, nine subroutines are associated with the third module. A complete listing of the programs discussed in subsection 2.4 is contained below. Subroutines named as called in the program descriptions, but not listed below, are PRIM utility routines which are described in subsection 2.10 or UNIVAC ASCII FORTRAN utilities.

Paragraph	Program	Process	Short description
2.4.1	ROLUIC	1.1.1	Main program for module 1
2.4.2	ROLMOS	1.1.2	Main program for module 2
2.4.3	DRIV12	1.2	Main program for module 3
2.4.3.1	SETTPS	1.2.1	Set TPSN ISSUEs
2.4.3.2	SETUIC	1.2.2	Set UIC ISSUEs
2.4.3.3	SETASG	1.2.3	Set ASGMT ISSUEs
2.4.3.4	AGGUIC	1.2.4	Aggregate UIC-data
2.4.3.5	SETMOS	1.2.5	Set MOS-data ISSUEs
2.4.3.6	CHKNUL	Util	Preprocessor utility
2.4.3.7	DOIT12	Util	Preprocessor utility
2.4.3.8	DOITNO	Util	Preprocessor utility
2.4.3.9	ISSFIN	Util	Preprocessor utility
2.4.4	DRIV13	1.3	Main program for module 4
2.4.4.1	AGGMOS	1.3.1	Aggregate MOS-data

The functions performed by the Preprocessor are:

a. **Process 1.1. Roll or Aggregate Data (two separate modules)**

(1) Process 1.1.1. Roll UIC - This subprocessor rolls (aggregates) the unit information (from the header record of the Authorization Data Base) from six-digit Unit Identification Code (UIC) level to three-digit UIC level. This three-digit UIC information is input to the subprocessor 1.2, Set ISSUE. Prior to this program, the header information should be sorted on three-digit UIC. Be sure to use correct options, i.e., Fielddata or ASCII.

(2) Process 1.1.2. Roll MOS - This subprocessor rolls the detail infor-mation from the Authorization Data Base from the six-digit UIC level to the three-digit UIC level. The information at the three-digit level is input to the subprocessor 1.2, Set ISSUE. The detail information should be sorted on MOS within three-digit UIC prior to this program; the sort must be tailored for the input data, i.e., the options will be different depending upon whether the MOS-data file is Fielddata or ASCII.

b. **Process 1.2. Set ISSUE** - This subprocessor uses the ISSUE file to set the ISSUE code in each UIC-data record. The UIC-data records are then matched with the MOS-data records on the three-digit UIC and the ISSUE is set in the MOS-data records. The ISSUE may be reset if a UIC is included in more than one ISSUE definition during process 1.2.1, Set TPSN, or process 1.2.2, Set UIC. A warning message is written for every UIC that is reset, each time it is reset. Table 2-1 provides a summary of the combinations of variables in UIC-data that may be used to identify ISSUEs. An error message is written for every UIC that did not have the ISSUE set. The process sequence is:

(1) Process 1.2.1. Set TPSN. Set ISSUE based on Troop Program Sequence Number (TPSN) and Assignment Code (ASGMT), TPSN and Location Code (LOCCO), TPSN and Station Code (STACO), and TPSN only.

(2) Process 1.2.2. Set UIC. Set ISSUE based on UIC.

(3) Process 1.2.3. Set ASGMT. Set ISSUE based on ASGMT and LOCCO, ASGMT and STACO, ASGMT only, and character one only of ASGMT.

(4) Process 1.2.4. Aggregate UIC. Aggregate and print UIC information at the UIC level. Outputs are Aggregate Required Strength (AGGREQ), Aggregate Authorized Strength (AGGAUT), and a list of the three-digit UICs included in each ISSUE. Note: AGGREQ and AGGAUT are printed for information purposes but are not the numbers actually used on the final reports. The reports use numbers aggregated from the MOS-data records. These numbers should be the same but are not necessarily equal.

(5) Process 1.2.5. Set MOS. Match the MOS-data file to the UIC-data file on the three-digit UIC. Set the ISSUE in the MOS-data file equal to the ISSUE on the UIC-data file record.

c. **Process 1.3. Aggregate MOS.** This subprocessor aggregates the MOS-data to the ISSUE level and creates print files containing information that may be required to ensure the units have been grouped appropriately. The Aggregate Required Strength and the Aggregate Authorized Strength is computed for each ISSUE and across all ISSUEs. These may be compared with the information from 1.2.4, Aggregate UIC, for an indication of differences between the UIC-data and the MOS-data. A count of the number of specialties found for each ISSUE is printed and a screen-readable file of aggregate strengths for each MOS and ISSUE is created. Another file, containing aggregate strengths for each ISSUE by pay-grade, is also created. The MOS-data file should be sorted on MOS within ISSUE (Major = ISSUE, Minor = MOS) prior to running this program. Since the Mos-data file was output from subprocessor 1.1.2, Roll MOS, the file is ASCII, and the sort options must be ASCII options.

Table 2-1. ISSUE Definition

Processing order	Identification Method 1	Identification Method 2	Warn and reset if not blank
1	TPSN	ASGMT	Yes
2	TPSN	LOCCO	Yes
3	TPSN	STACO	Yes
4	TPSN	--	No
5	UIC3	--	Yes
6	ASGMT	LOCCO	No
7	ASGMT	STACO	No
8	ASGMT	--	No
9	ASGMT1	--	No

A set of figures is provided to assist the reader in better understanding the relationship of the main programs and subroutines in the Preprocessor. Figure 2-2 shows the information flow through the Preprocessor; Figure 2-3 shows the structure of all of the routines within the Preprocessor; Figure 2-3, along with Figure 2-4, which is a chart of the subroutine cross-references allows the reader to see which routines are called by routines other than the main programs. All these figures will be found at the end of section 2.4.

2.4.1 PROGRAM DESCRIPTION

a. Identification

Preprocessor Main Program - ROLUIC
Process Number 1.1.1

b. Function

This program aggregates the UIC-data file from the six-digit UIC level to the three-digit UIC level.

c. Input

UIC-data (six-digit UIC level) file Logical unit 22

d. Processing

Presort the six-digit UIC-data file by characters 2 through 4 of the UIC; sort options must match data type.

Add strengths from all records which match on characters 2 through 4 of the UIC and write one record for each set of matching records.

e. Output

UIC-data (three-digit UIC level) file Logical unit 23

f. Interfaces

Called by: N/A

g. Arguments. N/A

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

i. Error Codes. There are no Error Codes in ROLUIC.

2.4.2 PROGRAM DESCRIPTION

a. Identification

Preprocessor Main Program - ROLMOS
Process Number: 1.1.2

b. Function

This program aggregates the MOS-data file from the six-digit UIC level to the three-digit UIC level.

c. Input

Parameter file	Logical unit 17
MOS - Enlisted file	Logical unit 26
MOS - Officer file	Logical unit 27
MOS - Warrant Officer file	Logical unit 28

d. Processing

Presort each MOS-data file by UIC and by the maximum number of MOS characters (NCCHAR) from the Parameter file; sort optioms must match data type.

Get grades from Parameter file.

Get lengths of MOS from Parameter file.

Roll the MOS-data file to three-digit UIC level and combine all grades within the same MOS and three-digit UIC.

e. Output

MOS-data (three-digit UIC level) file	Logical unit 29
Error Print file	Logical unit 13

f. Interfaces

Called by: N/A

Calls to: WRTERR (NERR)
FSORT (FORTRAN sort utility)

g. Arguments. N/A

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

i. Error Codes. The Error Codes in ROLMOS are 2, 4, 30, 32, 39, 40, 42, 126.

2.4.3 PROGRAM DESCRIPTION

a. Identification

Preprocessor Main Program - DRIV12
Process Number: 1.2

b. Function

This program calls the subroutines that make up the ISSUE file and MOS-data file edit portion of the Preprocessor.

c. Input (via subroutines)

ISSUE Definition file	Logical unit 14
Parameter file	Logical unit 17
UIC-data (three-digit UIC level) file	Logical unit 23
MOS-data (three-digit UIC level) file	Logical unit 29

d. Processing

Get the wall clock time.

Get the day the run is being made.

Get the debug flag.

Set up the Error Print file.

Set the ISSUEs that are based on TPSN.

Set the ISSUEs that are based on specific UICs.

Set the ISSUEs that are based on ASGMT.

If the debug flag is set to greater than 9, do not perform the following two steps:

Aggregate the UIC-data file to the ISSUE level.

Set the ISSUEs in the MOS-data file from the UIC-data file.

e. Output (via subroutines)

Error Print file	Logical unit 13
UIC-data (three-digit with ISSUE added) file	Logical unit 24
UIC-data (three-digit with ISSUE added) file	Logical unit 25
MOS-data (three-digit with ISSUE added) file	Logical unit 30

f. Interfaces

Called by: N/A

Calls to:

GETWCT	(ISH, ISM, ISS, LH, LM, LS)
GETDAY	(RUNDAY)
APRTCA	(PRIM utility)
WRTERR	(NERR)
SETTPS	(IDEBUG, IOK)
SETUIC	(IDEBUG, IOK)
SETASG	(IDEBUG, IOK)
SETMOS	(IDEBUG, IOK)
AGGUIC	(IDEBUG, IOK)
CHKNUL	(CHKFIE, LENG)
DOITNO	(LUNIN, LUNOUT, IDEBUG)
DOIT12	(LUNIN, LUNOUT, IDEBUG)

g. Arguments. N/A

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

h. Error Codes. The Error Code in DRIV12 is 0 (zero).

2.4.3.1 PROGRAM DESCRIPTION

a. Identification

Preprocessor Subroutine - SETTPS (IDEBUG, IOK)
Process Number: 1.2.1

b. Function

This subroutine sets the ISSUE code in the UIC-data file for those units that are grouped on TPSN.

c. Input

ISSUE Definition file	Logical unit 14
UIC-data (three-digit level) file	Logical unit 23

d. Processing

Sort ISSUE file on identification methods.

Sort UIC-data file on TPSN/ASGMT.

Set ISSUE when TPSN and ASGMT match.

Sort UIC-data file on TPSN and LOCCO.

Set ISSUE when TPSN and LOCCO match; write warning messages if not blank.

Sort UIC-data file on TPSN and STACO.

Set ISSUE when TPSN and STACO match; write warning messages if not blank.

Set ISSUE when TPSN (only) matches (do not set if had been set previously).

e. Output

UIC-data (three-digit level with ISSUE added) file	Logical unit 24
UIC-data (three-digit level with ISSUE added) file	Logical unit 25
Error Print file	Logical unit 13

f. Interfaces

Called by: DRIV12

Calls to:

	FSORT	(FORTRAN sort utility)
	WRTERR	(NERR)
	DOITNO	(LUNIN, LUNOUT, IDEBUG)
	CHKNUL	(CHKFIE, LENG)
	DOIT12	(LUNIN, LUNOUT, IDEBUG)
	ISSFIN	(ID1, ID2, LUNISS, IE, ISSEOF)

g. Arguments

IDEBUG - flag for debug print.
IOK - if set other than zero, program cannot continue.

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

i. Error Codes. The Error Codes in SETTPS are 1, 7, 103.

2.4.3.2 PROGRAM DESCRIPTION

a. Identification

Preprocessor Subroutine - SETUIC (IDEBUG, IOK)
Process Number: 1.2.2

b. Function

This subroutine sets the ISSUE in the UIC-data file for the ISSUEs that are grouped on the basis of UIC.

c. Input

UIC-data (three-digit with ISSUE added) file	Logical unit 24
UIC-data (three-digit with ISSUE added) file	Logical unit 25
ISSUE Definition file	Logical unit 14

d. Processing

Sort ISSUE file on identification method.

Sort UIC-data file on characters 2 through 6 of UIC (normally only 2 through 4 will be significant).

Set ISSUE when UIC matches on number of characters specified by ID1.

Write warning message if ISSUE is not blank.

e. Output

UIC-data (three-digit with ISSUE added) file	Logical unit 24
UIC-data (three-digit with ISSUE added) file	Logical unit 25
Error Print file	Logical unit 13

f. Interfaces

Called by: DRIV12

Calls to: WRTERR (NERR)
DOITNO (LUNIN, LUNOUT, IDEBUG)
FSORT (FORTRAN sort utility)
ISSFIN (ID1, ID2, LUNISS, IE, ISSEOF)

g. Arguments

IDEBUG - flag for debug print.
IOK - if set to value other than zero, program cannot continue.

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

i. Error Codes. The Error Codes in SETUIC are 1, 6, 7, 43, 103.

2.4.3.3 PROGRAM DESCRIPTION

a. Identification

Preprocessor Subroutine - SETASG (IDEBUG, IOK)
Process Number: 1.2.3

b. Function

This subroutine sets ISSUE in the UIC-data file at the three-digit level for the ISSUEs that are grouped on the basis of ASGMT.

c. Input

UIC-data (three-digit with ISSUE added) file	Logical unit 24
UIC-data (three-digit with ISSUE added) file	Logical unit 25
ISSUE Definition file	Logical unit 14

d. Processing

Sort UIC-data file on ASGMT/LOCCO.

If ISSUE is blank, set ISSUE when ASGMT and LOCCO match.

Sort UIC-data file on ASGMT/STACCO.

If ISSUE is blank, set ISSUE when ASGMT and STACCO match.

Sort UIC-data file on ASGMT.

If ISSUE is blank, set ISSUE when ASGMT match.

If ISSUE is still blank, set ISSUE when character 1 of ASGMT matches.

Write error message if ISSUE is not set in all records.

e. Output

Error Print file	Logical unit 13
UIC-data (three-digit with ISSUE added) file	Logical unit 24
UIC-data (three-digit with ISSUE added) file	Logical unit 25

f. Interfaces

Called by: DRIV12

Calls to: WRTERR (NERR)
DOITNO (LUNIN, LUNOUT, IDEBUG)
ISSFIN (ID1, ID2, LUNISS, IE, ISSEOF)

g. Arguments

IDEBUG - flag for debug print.
IOK - if set to value other than zero, program cannot continue.

h. Tables and Items. Please refer to Appendix A for Data Dictionary.

i. Error Codes. The Error Codes in SETASG are 5, 7.

2.4.3.4 PROGRAM DESCRIPTION

a. Identification

Preprocessor Subroutine - AGGUIC (IDEBUG, IOK)
Process Number: 1.2.4

b. Function

This subroutine provides the user with a list of the units that will be aggregated into each ISSUE and the aggregated structure and authorized strengths from the UIC-data file.

c. Input

UIC-data (three-digit level with ISSUE added) file — Logical unit 24 or 25

d. Processing

Sort the UIC-data file on three-digit UIC within ISSUE.

Read the UIC-data file and aggregate the structure and authorized strengths for each ISSUE.

Print a list of the three-digit UICs that are included in each ISSUE.

e. Output

Printed listing of UICs in each ISSUE and the aggregated ISSUE authorized and structure strengths

f. Interfaces

Called by: DRIV12

Calls to: APRTCA (PRIM utility)
FSORT (FORTRAN sort utility)

g. Arguments

IDEBUG - flag for debug print.
IOK - if set to value other than zero, program cannot continue.

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

i. Error Codes. There are no Error Codes in AGGUIC.

2.4.3.5 PROGRAM DESCRIPTION

a. Identification

Preprocessor Subroutine - SETMOS (IDEBUG, IOK)
Process Number: 1.2.5

b. Function

This subroutine sets the ISSUE code in the MOS-data file

c. Input

UIC-data (three-digit with ISSUE added) file	Logical unit 24 or 25
MOS-data (three-digit UIC level) file	Logical unit 29

d. Processing

Sort the UIC-data file on UIC.

Match the MOS-data file with the UIC-data file, matching on UIC.

When a match is found, set the ISSUE in the MOS-data file to the ISSUE found in the UIC-data file.

If any records are not matched, write an error message.

e. Output

Error Print file	Logical unit 13
MOS-data (with ISSUE added) file	Logical unit 30

f. Interfaces

Called by: DRIV12

Calls to: CHKNUL (variable to check, number of characters)
DOITNO (LUNIN, LUNOUT, IDEBUG)
FSORT (FORTRAN utility subroutine)
WRTERR (Nerr)

g. Arguments

IDEBUG - flag for debug print.
IOK - If not equal to zero, program cannot continue.

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

i. Error Codes. The Error Code in SETMOS is 5.

2.4.3.6 PROGRAM DESCRIPTION

a. Identification

Preprocessor Subroutine - CHKNUL (CHKFIE, LENG)

b. Function

This subroutine checks the input alphanumeric for the presence of the null character (octal 00) and replaces with a blank (octal 40).

c. Input

Subroutine arguments

d. Processing

Check each character against octal 00.

If a match is found, reset character to a blank (octal 40).

e. Output

Subroutine argument modified - CHKFIE

f. Interfaces

Called by: DRIV12
SETMOS

Calls to: N/A

g. Arguments

CHKFIE - field to be checked
LENG - length of CHKFIE

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

i. Error Codes. There are no Error Codes in CHKNUL.

2.4.3.7 PROGRAM DESCRIPTION

a. Identification

Preprocessor Subroutine - DOIT12 (LUNIN, LUNOUT, IDEBUG)

b. Function

This subroutine puts an end of file mark on the output unit, rewinds both the input and output units, and resets the unit numbers.

c. Input

Subroutine arguments

d. Processing

Write end of file on LUNOUT.

Rewind LUNIN and LUNOUT.

Set LUNIN equal to LUNOUT.

If LUNIN was 24, reset it to 25; if it had been 25, reset it to 24.

e. Output

Subroutine arguments modified - LUNIN, LUNOUT

f. Interfaces

Called by: SETTPS
SETUIC
SETASG

Calls to: N/A

g. Arguments

LUNIN - logical unit for input
LUNOUT - logical unit for output
IDEBUG - flag for debug print

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

i. Error Codes. There are no Error Codes in DOIT12.

2.4.3.8 PROGRAM DESCRIPTION

a. Identification

Preprocessor Subroutine - DOITNO (LUNIN, LUNOUT, IDEBUG)

b. Function

This subroutine rewinds both the input and output units and resets the unit numbers. Similar to DOIT12 but does not write end of file.

c. Input

Subroutine arguments

d. Processing

Rewind LUNIN and LUNOUT.

Set LUNIN equal to LUNOUT.

If LUNIN was 24, reset it to 25; if it had been 25, reset it to 24.

e. Output

Subroutine arguments modified - LUNIN, LUNOUT

f. Interfaces

Called by: SETTPS
SETUIC
SETASG

Calls to: N/A

g. Arguments

LUNIN - logical unit for input
LUNOUT - logical unit for output
IDEBUG - flag for debug print

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

i. Error Codes. There are no Error Codes in DOITNO.

2.4.3.9 PROGRAM DESCRIPTION

a. Identification

Preprocessor - ISSFIN (FIRST, SECOND, LUNISS, IE, ISSEOF)

b. Function

This subroutine will cause an error message to be written when the end of file on the UIC-data file is reached before the end of file on the ISSUE file.

c. Input

Subroutine arguments
ISSUE Definition file Logical unit 14

d. Processing

When EOF is encountered on UIC-data file, and ISSUE file has unused ISSUE definitions, an error message is written for each unused definition.

e. Output

Error Print file Logical unit 13

Subroutine arguments modified - IE, ISSSOF

f. Interfaces

Called by: SETTPS
SETUIC
SETASG

Calls to: WRTERR (NERR)

g. Arguments

FIRST First identification method (ID1) of last ISSUE record processed.

SECOND Second identification method (ID2) of last ISSUE record processed.

LUNISS Logical unit number of the ISSUE file.

IE Integer count of the number of errors found by ISSFIN.

ISSEOF ISSUE file end of file flag.

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

i. Error Codes. The Error Code in ISSFIN is 7.

2.4.4 PROGRAM DESCRIPTION

a. Identification

Preprocessor Main Program - DRIV13
Process Number: 1.3

b. Function

This program calls the subroutine that aggregates the MOS-data file to ISSUE level.

c. Input (via subroutines)

Parameter file	Logical Unit 17
MOS-data (three-digit with ISSUE code added) file	Logical Unit 30

d. Processing

Get day of week.

Get time of day.

Read debug flag.

Call AGGMOS.

Print ending messages.

e. Output (via subroutines)

MOS-data (ISSUE level) file	Logical Unit 31
Strengths-by-grade Print file	Logical Unit 85
Strengths-by-MOS Print file	Logical Unit 86

f. Interfaces

Called by: N/A

Calls to: GETWCT (ISN, ISM, ISS, LH, LM, LS)
GETDAY (RUNDAY)
AGGMOS (IDEBUG, IOK)
WRTERR (NERR)

g. Arguments. N/A

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

i. Error Codes. The Error Code in DRIVE13 is 0 (zero).

2.4.4.1 PROGRAM DESCRIPTION

a. Identification

Preprocessor Subroutine - AGGMOS (IDEBUG, IOK)
Process Number: 1.3.1

b. Function

This subroutine aggregates the MOS-data file to ISSUE level.

c. Input

Parameter file	Logical unit 17
MOS-data (three-digit with ISSUE added) file	Logical unit 30

d. Processing

Get Parameter file data.

For each ISSUE, aggregate strengths over all MOS for each grade and write to file 85.

For each ISSUE, aggregate strengths over all grades for each MOS and write to file 86.

Aggregate strengths by MOS and grade for each ISSUE and write to file 31.

e. Output

Error Print file	Logical unit 13
MOS-data (ISSUE level) file	Logical unit 31
Strengths-by-grade Print file	Logical unit 85
Strengths-by-MOS Print file	Logical unit 86

f. Interfaces

Called by: DRIV13

Calls to: WRTERR (NERR)

g. Arguments

IDEBUG - flag for debug print.
IOK - if set to value other than zero, program cannot continue.

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

i. Error Codes. The Error Codes in AGGMOS are 30, 32, 40, 70, 71, 98, 99.

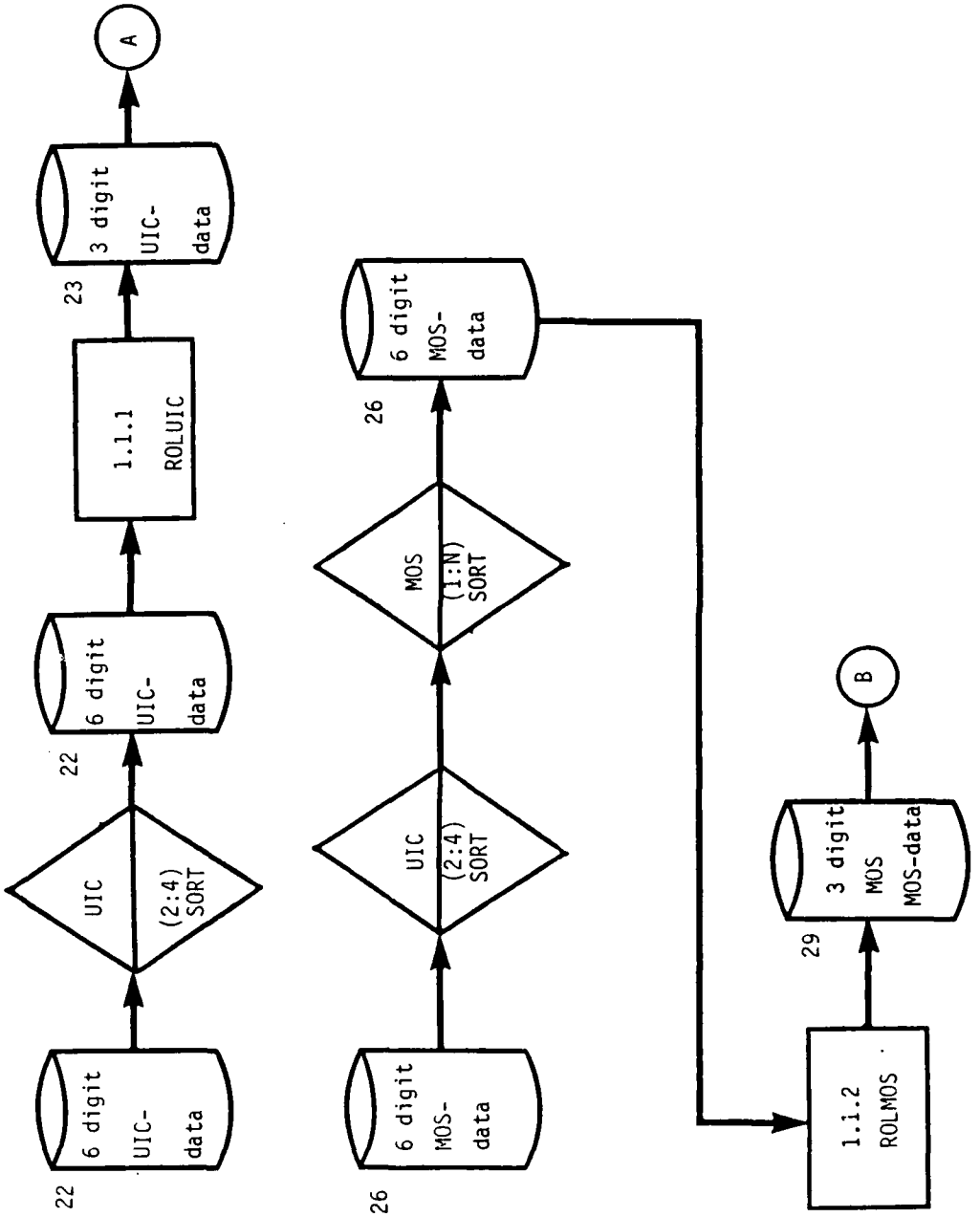


Figure 2-2. Information Flow Through the Preprocessor (page 1 of 3 pages)

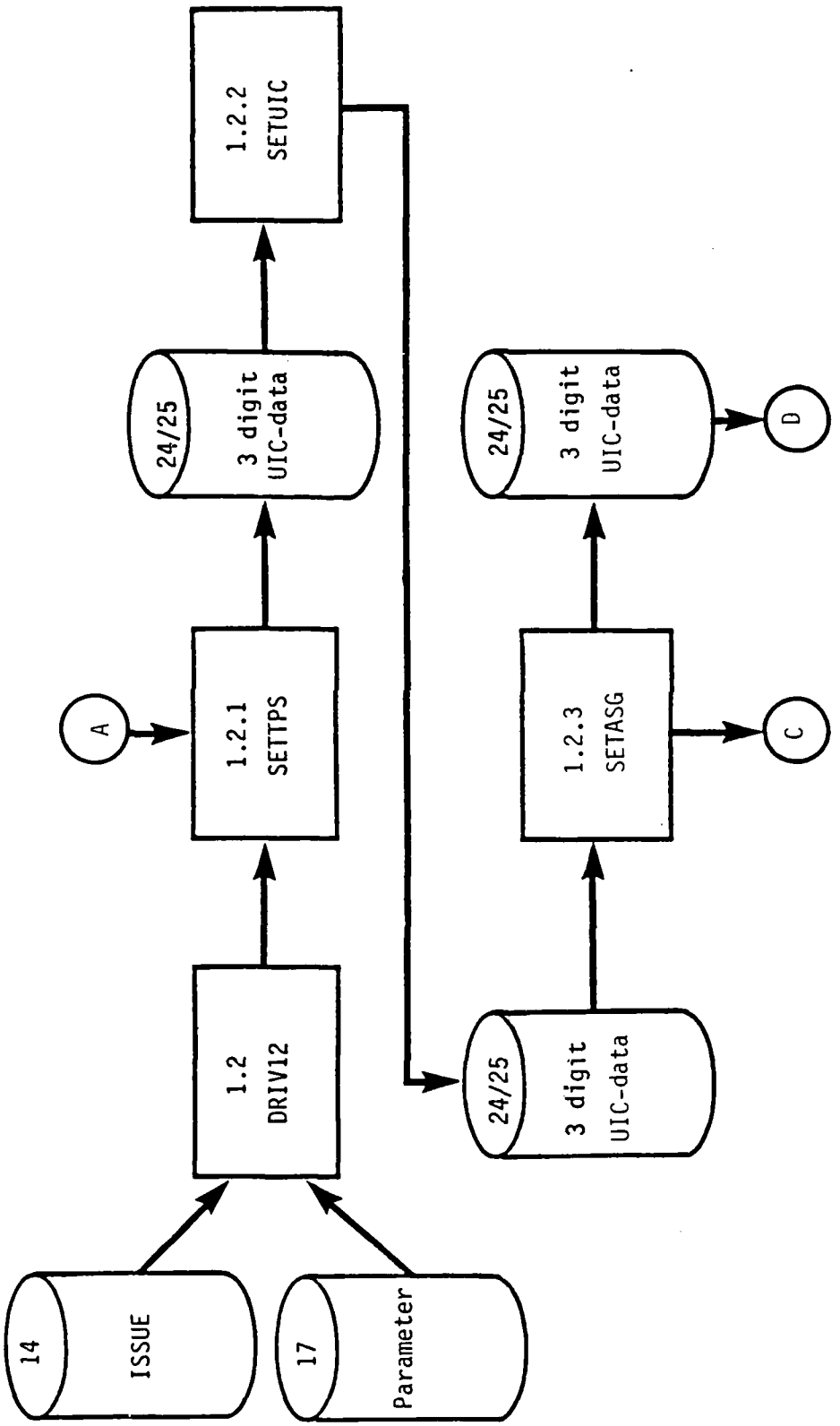


Figure 2-2. Information Flow Through the Preprocessor
(Page 2 of 3 pages)

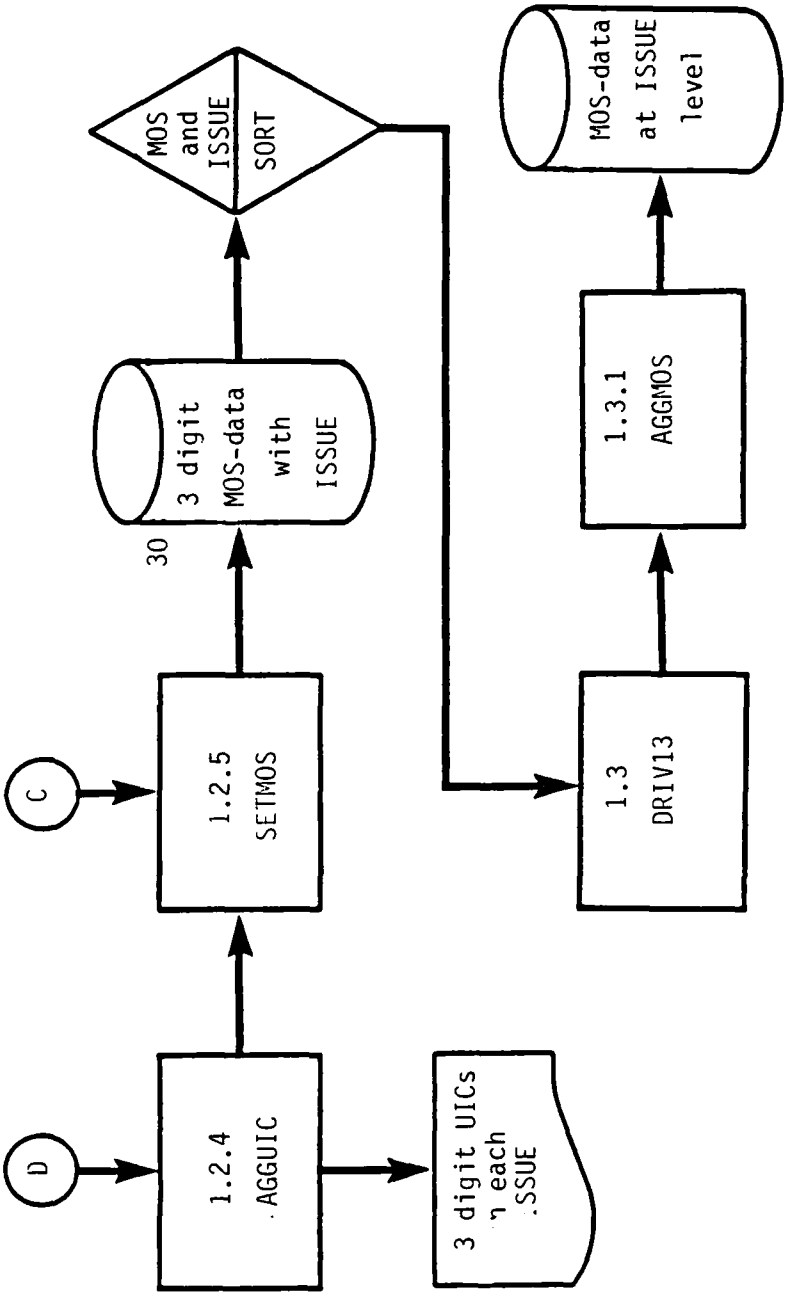


Figure 2-2. Information Flow Through the Preprocessor (page 3 of 3 pages)

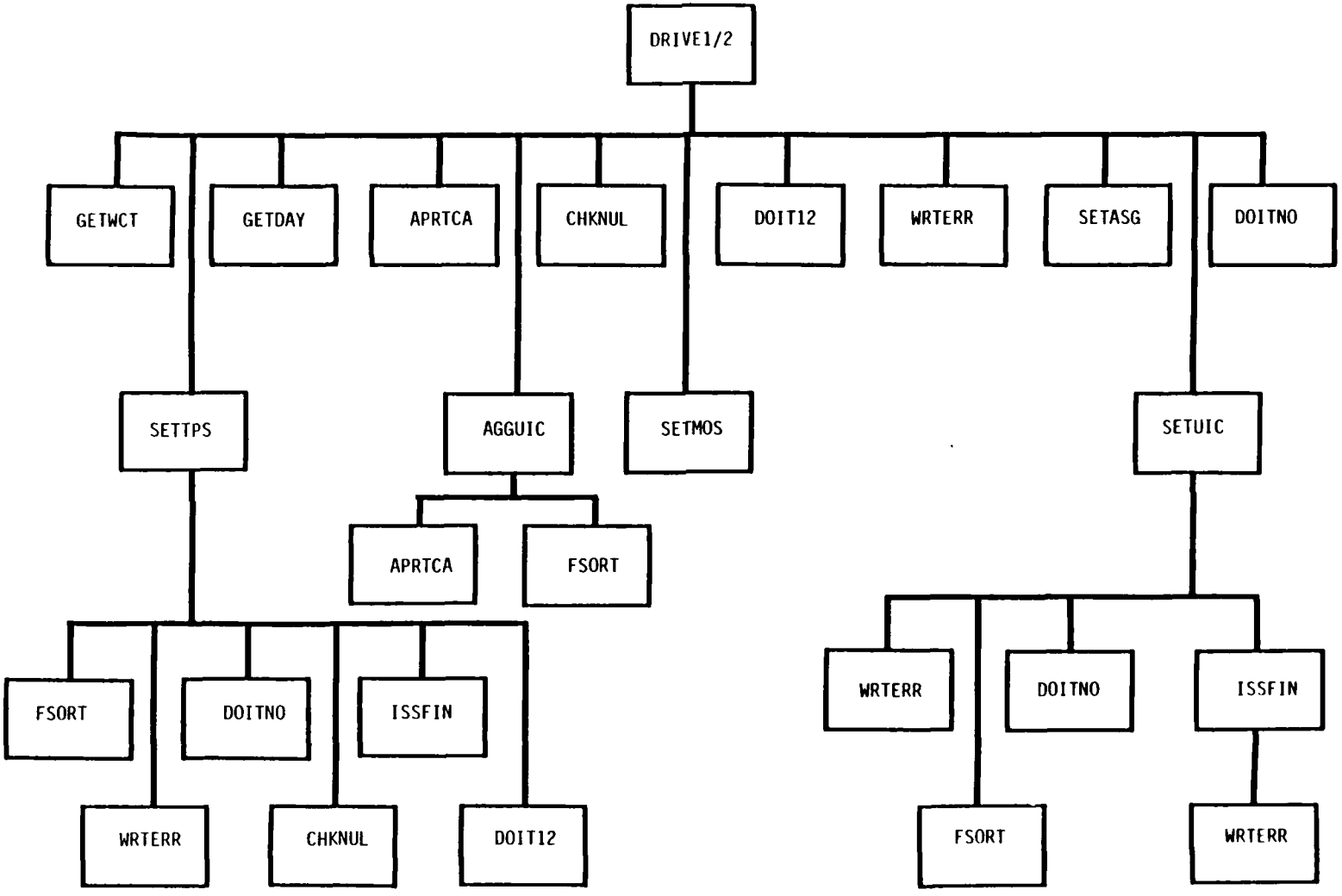


Figure 2-3. Program Unit Hierarchy - Preprocessor

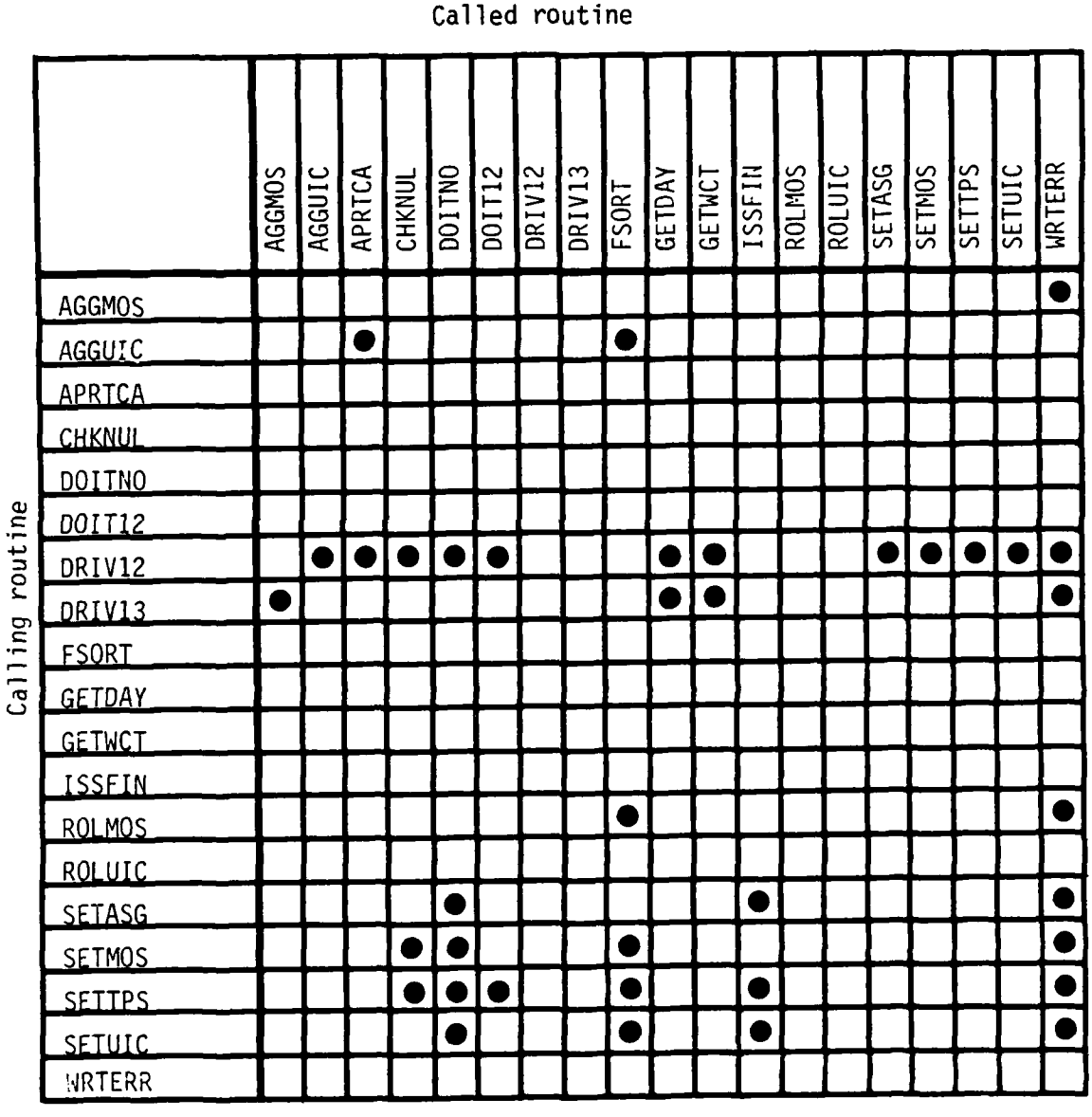


Figure 2-4. Subroutine Cross-reference - Preprocessor

2.5 PROGRAM DESCRIPTION - POLICY PROCESSOR. Subsection 2.5 contains a description of the five modules of the Policy Processor. In addition to the main programs, there are 35 subroutines associated with this processor. Ten of these subroutines are numbered functions; 7 subroutines are PRIM utilities, and 18 are Policy Processor utilities. A complete listing of the programs discussed in subsection 2.5 is shown below; PRIM utilities are discussed in section 2.10.

Paragraph	Program	Process	Short description
2.5.1.1	DRIVE2/1	2.1	Main program for policy edit
2.5.1.2	EDIPOL	2.1.0	Policy edit
2.5.1.3	INSIDE	Utility	Edit subroutine
2.5.2.1	DRIVE2/2-A	2.2/A	Main program for Module A of Apply Policy
2.5.2.2	APP22A	2.2/A	Apply Policy Module 2/A
2.5.2.3	GETPAR	2.2.1	Get parameter data
2.5.2.4	SRTPOL	2.2.2	Sort & separate policy data
2.5.2.5	AGISPO	2.2.3	Set aggregated-ISSUE policy indicators
2.5.2.6	ISMO	2.2.4, 2.2.5, and 2.2.6	Apply ISSUE/MOS policies
2.5.3.1	DRIVE2/2-B	2.2/B	Main program for Module B of Apply Policy
2.5.3.2	APP22B	2.2/B	Apply Policy Module 2/B
2.5.4.1	DRIVE2/2-C	2.2/C	Main program for Module C of Apply Policy
2.5.4.2	APP22C	2.2/C	Apply Policy Module 2/C
2.5.5.1	DRIVE2/3	2.3	Main program for apply base values
2.5.5.2	APP23	2.3	Apply base values
2.5.5.3	SETBAS	2.3.0	Set base values
2.5.6.1	CONPOL	Utility	Convert policy data
2.5.6.2	DOIT	Utility	Do coordination of MATPOL and SETPOL
2.5.6.3	FINDAT	Utility	Find matching data sets
2.5.6.4	FINVAL	Utility	Find EOF on JAV file
2.5.6.5	GETBAS	Utility	Get next value record
2.5.6.6	GETJOB	Utility	Get next MOS-data record
2.5.6.7	GETPOL	Utility	Get next policy record
2.5.6.8	MAKAGG	Utility	Make aggregated grade JAV record
2.5.6.9	MAKIND	Utility	Make individual grade JAV record
2.5.6.10	MATPOL	Utility	Match aggregated-ISSUE policies with job ISSUEs
2.5.6.11	NXTISS	Utility	Compute the next unique demand node name for an ISSUE

2.5.6.12	SETPOL	Utility	Set policy indicators needed by MAKAGG and MAKIND
2.5.6.13	WRTEXT	Utility	Write extra (unused) MOS-data records
2.5.6.14	WRTUNF	Utility	Write unfilled job records
2.5.6.15	WRTVAL	Utility	Write JAV records
2.5.6.16	ZEROIN	Utility	Zero the policy indicators
2.5.6.17	ZILCH	Utility	Zero the output JAV variables

Since this processor was originally designed and programed as one main program which sequentially called the three major functions, an inspection of the program code will show that the five main programs or drivers are still nearly identical. The major difference is the code that has been commented to inhibit processing the entire Policy Processor at one time. If it should be desired, the first driver (DRIVE/2-1) and main subroutine APP22A, APP22B, and APP22C could be easily recompiled to operate the entire processor. The three major functions of the Policy Processor are:

a. **Policy Edit**

b. **Apply Policies** - Apply the personnel policies to the relevant MOS-data file.

c. **Set Base Values** - Edit the Value file and apply the values to the remaining MOS-data.

The second function, Apply Policies, consists of six subfunctions. Several of these could produce prodigious error messages; each assumes that errors previously encountered have been corrected. Therefore, each time a different sort of the data is required in order to continue, this module stops to provide the user with an error correction opportunity. If corrections are made, the user is required to restart this processor with the first subprocessor, Policy Edit; unless the Policy file is very large Policy Edit takes only a few seconds. Since Module A separates the policy types, it must then be rerun. The Apply Policy function consists of these modules:

Module A - Separate the Policy file into the four policy types

- Set the aggregate indicators for personnel policies that apply to the Aggregated-ISSUE-level (00 level)

- Apply the personnel policies that reference specific MOS in specific ISSUEs and apply aggregated policies to these records

Module B - Apply the policies that pertain to specific MOS in all ISSUEs

Module C - Apply the policies that refer to all MOS in specific ISSUEs and apply aggregated policies

All three modules of the Apply Policies function require use of Parameter file data. Therefore, GETPAR, Process 2.2.1, is repeated in each of these modules. Each execution of GETPAR requires less than 1 second. A further complication in the understanding of these modules is the operation of the subroutines ISMO and FINDAT. ISMO controls the application of both ISSUE and MOS policies including the application of combined ISSUE and MOS policies. FINDAT finds the point in the input job file (MOS-data) where the next policy should be applied. Both ISMO and FINDAT are controlled by the calling subroutine (APP22A, APP22B, or APP22C) by the value of the variable named IFLAG.

Prior to execution of Module A, the runstream should cause a sort of the input MOS-data file on the maximum number of characters (NCCHAR in the Parameter file) of MOS within the ISSUE code. As Module A matches the Combined ISSUE and MOS Policy file against the MOS-data file, all MOS records not matched by a policy are written to the extra (unmatched) MOS-data file; matched records are written to the Job Assignment Value file with the appropriate data transformations. More specific information on the transformations may be found in Appendix C, User-defined files, and in the User Manual. The extra MOS-data is sorted on MOS prior to execution of Module B. The same general logic is followed for both Modules B and C; Module B applies the MOS policies, and Module C applies the ISSUE policies.

The third and final function of the Policy Processor is to apply the value data to the job data not previously matched by a specific policy. The same general logic is used as was used in the Apply Policies function, except that all job data not matched in this step will be written to the Unfilled Job file. The data transformations are described in Appendix C, User-defined files.

Demand node names are written to the JAV file and extra job-data file by Module A, Combined ISSUE, and MOS policy application; the extra MOS-data file has demand node names that are made up of the four-character ISSUE code followed by two alphabetic characters. A unique set is used for each MOS. Since all characters from A through Z are used, this aspect of PRIM allows for a maximum of 676 MOS in each ISSUE. The last characters will be set to blank. During the next pass for MOS policies, an additional character is added in the seventh position. The eighth position is added during the pass for ISSUE policies. Finally, the eighth character is changed from alphabetic to numeric during the application of the Value file. Demand node names ending with a numeric one (1) in the eighth position specify the minimum fill levels as both the minimum and maximum levels. The record specifying the desired fill from the minimum to the maximum possible is designated by a two (2) in the last position of the name. These manipulations of the job names are necessitated by the requirement that all demand node names must be unique. After job assignments are made by the Assignment Processor and the Substitute Assignment Processor, the Readiness Processor will aggregate these individual records back up to ISSUE-level.

A set of figures is provided to assist the reader in better understanding the relationship of the main programs and subroutines in the Policy Processor. Figure 2-5 shows the information flow; Figure 2-6 shows the relationship of the routines within the Policy Processor; Figure 2-6, along with Figure 2-7, the subroutine cross-reference allows the reader to see which routines are called by routines other than the main programs. These figures are at the end of section 2.5.

2.5.1.1 PROGRAM DESCRIPTION

a. Identification

Policy Processor Main Program Number 1 - DRIVE2/1
Process Number: 2.1

b. Function

This program controls the processing sequence of the Policy Processor. This particular version calls the Policy file edit subroutine. By removing comments, it could be used to call all the Policy Processor functions.

c. Input (via the main subroutine EDIPOL)

ISSUE Definition file	Logical unit 14
Parameter file	Logical unit 17
Policy file	Logical unit 91

d. Processing

Get and print the time and date.

Initialize the Error Print file.

Call the subroutine which controls the editing of the Policy file.

Print the amount of time used and number of errors and warnings.

e. Output (via the main subroutine)

Edited Policy file	Logical unit 92
Error Print file	Logical unit 13

f. Interfaces

Called by: N/A

Calls to: APRTCA (PRIM utility)
GETWCT (ISN, ISM, ISS, LH, LM, LS)
GETDAY (RUNDAY)
WRTERR (0)
EDIPOL

g. Arguments. N/A

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

i. Error Codes. The Error Code in DRIVE2/1 is 0 (zero).

2.5.1.2 PROGRAM DESCRIPTION

a. Identification

Policy Processor Subroutine - EDIPOL
Process Number: 2.1.0

b. Function

This subroutine edits the Policy file for valid values and compares the ISSUE with ISSUEs from the ISSUE file for acceptability.

c. Input

ISSUE Definition file	Logical unit 14
Parameter file	Logical unit 17
Policy file	Logical unit 91

d. Processing

Read ISSUEs from the ISSUE Definition file and save the codes.

Read the Parameter file.

Read a policy record.

Check for validity of ISSUE, maximum fill level, minimum fill level, low grade, high grade value, aggregation value - if all are valid, write the edited record and return to read the next policy record.

If invalid, write appropriate error messages and read next policy record without writing the policy record.

e. Output

Edited Policy file	Logical unit 92
Error Print file	Logical unit 13

f. Interfaces

Called by: DRIVE2/1

Calls to: GETPAR (IOK)
WRTERR (NERR)
ICHINT (alpha, begin, length, return flag)
RCHFLT (alpha, begin, length, return flag)
INSIDE

g. Arguments. N/A

h. Tables and Items. Please refer to Appendix A for Data Dictionary.

i. Error Codes. Error Codes in EDIPOL are 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 27, 30, 35, 36, 37, 39, 46, 98, 99, 113.

2.5.1.3 PROGRAM DESCRIPTION

a. Identification

Policy Processor Utility Subroutine - INSIDE
Process Number: 2.1.1

b. Function

This subroutine tests the high and low subscripts of grade levels and the maximum and minimum fill levels for acceptability.

c. Input. N/A; this subroutine is internal to subroutine EDIPOL.

d. Processing

Test grade for acceptability.

If grade is not acceptable, write error message and return to calling routine.

Test maximum and minimum fill levels for acceptability.

If fill levels are not acceptable, then write error message and return to calling routine.

e. Output. N/A

f. Interfaces

Called by: EDIPOL

Calls to: N/A

g. Arguments. N/A

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

i. Error Codes. There are no Error Codes in INSIDE.

2.5.2.1 PROGRAM DESCRIPTION

a. Identification

Policy Processor Main Program Number 2 - DRIVE2/2-A
Process Number: 2.2

b. Function

This program controls the processing sequence of the Policy Processor. This version calls the first four subfunctions of the Apply Policy function. This includes get Parameter file data, separate Policy file data, compute aggregated indicators, and apply combined ISSUE and MOS policies.

c. Input (via subroutines)

Parameter file	Logical unit 17
Edited Policy file	Logical unit 92
MOS-data (ISSUE level) file	Logical unit 31

d. Processing

Get and print the time and date.

Initialize the Error Print file.

Call the subroutine which controls the Apply Policy function, subfunctions 1 through 4 (APP22A).

Print amount of time used and number of errors and warnings.

e. Output (via subroutines)

Aggregated-ISSUE Policy file	Logical unit 32
ISSUE Policy file	Logical unit 33
MOS Policy file	Logical unit 34
Combined ISSUE & MOS Policy file	Logical unit 35
Aggregated-ISSUE Indicator file	Logical unit 90
Error Print file	Logical unit 13
Combined ISSUE and MOS Extra Job file	Logical unit 36
Job Assignment Value file	Logical unit 39

f. Interfaces

Called by: N/A

Calls to: APRTCA (PRIM utility)
APP22A (IOK)
GETDAY (RUNDAY)
GETWCT (ISH, ISM, ISS, LH, LM, LS)
WRTERR (O)

g. **Arguments.** N/A

h. **Tables and Items.** Please refer to Appendix A for the Data Dictionary.

i. **Error Codes.** The Error Code in DRIVE2/2-A is 0 (zero).

2.5.2.2 PROGRAM DESCRIPTION

a. Identification

Policy Processor Subroutine - APP22A (IOK)
Process Number: 2.2A

b. Function

This subroutine controls the sequence of the subfunctions included in the Apply Policy function. This version includes get the parameter data, separate the policy data, compute the aggregated-ISSUE indicators, and apply the combined ISSUE and MOS policies. File number definition, REWIND, and DEFINE FILE operations are performed here.

c. Input (via subroutines)

Parameter file	Logical unit 17
MOS-data (ISSUE level) file	Logical unit 31
Edited Policy file	Logical unit 92

d. Processing

Get beginning time.

Get Parameter file data.

Separate Policy file data by policy type.

Set aggregated-ISSUE policy indicators.

Sort the combined ISSUE and MOS Policy file.

Apply combined ISSUE and MOS policies by calling ISMO with IFLAG=2.

Get and print amount of time used and numbers of records processed.

e. Output (via subroutines)

Error Print file	Logical unit 13
Aggregated-ISSUE Policy file	Logical unit 32
ISSUE Policy file	Logical unit 33
MOS Policy file	Logical unit 34
Combined ISSUE and MOS Policy file	Logical unit 35
Aggregated-ISSUE Indicator file	Logical unit 90
Combined ISSUE and MOS Extra Job File	Logical unit 36

f. Interfaces

Called by: DRIVE2/2-A

Calls to: GETDAY (RUNDAY)
GETPAR (IOK)
SRTPOL (IOK)
GETWCT (ISH, ISM, ISS, LH, LM, LS)
AGISPO (IOK)
FSORT (FORTRAN sort utility)
ISMO (IOK)

g. Arguments

IOK - If set to value other than zero, program cannot continue

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

i. Error Codes. The Error Code in APP22A is 0 (zero).

2.5.2.3 PROGRAM DESCRIPTION

a. Identification

Policy Processor Subroutine - GETPAR (IOK)
Process Number: 2.2.1

b. Function

This subroutine gets the as-of-date, grade codes and location, and authorized or required variables from the Parameter file.

c. Input

Parameter file Logical unit 17

d. Processing

Get the as-of-date from the Parameter file.

Get the number of grades from the Parameter file, NGRADE.

Get the integer locations and alphanumeric codes for each grade.

Get the number of characters of the MOS, NCCHAR.

Determine whether authorized or required fill levels are to be used in this run, TYPSTR.

e. Output

Error Print file Logical unit 13

f. Interfaces

Called by: APP22A, APP22B, APP22C

Calls to: ICHINT (alpha, begin, end, return flag)
WRTERR (NERR)

g. Arguments

IOK - if set to value other than zero, program cannot continue.

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

i. Error Codes. The Error Codes in GETPAR are 30, 37, 39, 40, 41, 98, 99.

2.5.2.4 PROGRAM DESCRIPTION

a. Identification

Policy Processor Subroutine - SRTPOL (IOK)
Process Number: 2.2.2

b. Function

This subroutine sorts the Policy file into Aggregated-ISSUE, ISSUE only, MOS only, or combined ISSUE and MOS policies and rewrites the Policy file into four separate files.

c. Input

Edited Policy file	Logical unit 92

d. Processing

Sort Policy file into ISSUE or MOS policies (characters 1-5).

Write specific ISSUE policies to logical unit 33.

Write specific MOS policies to logical unit 34.

Write aggregated-ISSUE policies to logical unit 32.

Write combined ISSUE and MOS policies to logical unit 35.

e. Output

Error Print file	Logical unit 13
Aggregated-ISSUE Policy file	Logical unit 32
ISSUE Policy file	Logical unit 33
MOS Policy file	Logical unit 34
Combined ISSUE and MOS Policy file	Logical unit 35

f. Interfaces

Called by: APP22A

Calls to: FSORT (FORTRAN sort utility)
WRTERR (NERR)

g. Arguments

IOK - if set to value other than zero, program cannot continue.

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

i. Error Codes. The Error Codes in SRTPOL are 42, 99.

2.5.2.5. PROGRAM DESCRIPTION

a. Identification

Policy Processor Subroutine - AGISPO (IOK)
Process Number 2.2.3

b. Function

This subroutine reads the policies that are stated in terms of the aggregated-ISSUE, sets the indicators, and writes all of the indicators to a file to save for later use.

c. Input

Aggregated-ISSUE Policy file	Logical unit 32

d. Processing

Read the Aggregated-ISSUE policies.

Set indicators necessary for computing minimum and maximums.

Write the indicators to a file to save for later use.

e. Output

Error Print file	Logical Unit 13
Aggregated-ISSUE Indicator file	Logical unit 90

f. Interfaces

Called by: APP22A

Calls to: CONPOL (NEWPOL)
GETPOL (IOK)
SETPOL (IOK)
WRTERR (NERR)
ZEROIN

g. Arguments

IOK - if set to value other than zero, program cannot continue.

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

i. Error Codes. The Error Codes in AGISPO are 98, 99.

2.5.2.6 PROGRAM DESCRIPTION

a. Identification

Policy Processor Subroutine - ISMO (IOK)
Process Number: 2.2.4, IFLAG=2, combined ISSUE and MOS policies
2.2.5, IFLAG=3, MOS policies
2.2.6, IFLAG=1, ISSUE policies

b. Function

This subroutine controls the setting of policies that are stated in terms of the ISSUE only, MOS only, or both the ISSUE and MOS. Records that do not match will be written to the "extra" file.

c. Input (via subroutines)

Combined ISSUE and MOS Policy file	Logical unit 35 (process 2.2.4)
MOS Policy file	Logical unit 34 (process 2.2.5)
ISSUE Policy file	Logical unit 33 (process 2.2.6)
MOS-data (ISSUE level) file	Logical unit 31 (process 2.2.4)
Combined ISSUE and MOS Extra Job File	Logical unit 36 (process 2.2.5)
MOS Extra Job file	Logical unit 37 (process 2.2.6)

d. Processing

Get the next policy for FINDAT - after this, SETPOL or FINDAT will have read one record beyond the policy record being processed.

Find the next matching set of policy and job data.

On return from FINDAT, if all flags = 0, set and write the values:

Set the aggregated grades and write a JAV record.

Set any leftover individual grades which have a policy and write a JAV record.

If all policies have been used (IEOF≠0), copy the rest of the MOS-data to the extra file (setting LISSUE and LMOS to all "Z" forces, FINDAT to do the copy).

e. Output (via subroutines)

Combined ISSUE and MOS Extra Job file	Logical unit 36 (process 2.2.4)
MOS Extra Job file	Logical unit 37 (process 2.2.5)
ISSUE Extra Job file	Logical unit 38 (process 2.2.6)
Error Print file	Logical unit 13

f. Interfaces

Called by: APP22A, APP22B, and APP22C

Calls to:
ZILCH
ZEROIN
GETPOL (IOK)
CONPOL (CONPOL)
FINDAT (IOK)
MAKAGG (IOK)
MAKIND (IOK)
DOIT (IOK)

g. Arguments

IOK - if set to value other than zero, program cannot continue.

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

i. Error Codes. There are no Error Codes in ISMO.

2.5.3.1 PROGRAM DESCRIPTION

a. Identification

Policy Processor Main Program Number 3 - DRIVE2/2-B
Process Number: 2.2/B

b. Function

This program controls the processing sequence of the Policy Processor. This version applies the policies that refer to specific MOS across all ISSUEs.

c. Input (via subroutines)

Parameter file	Logical unit 17
MOS Policy file	Logical unit 34
Combined ISSUE and MOS Extra Job file	Logical unit 36
Job Assignment Value file	Logical unit 39

d. Processing

Get and print the time and date.

Initialize the Error Print file.

Call the subroutine which controls the Apply Policy function, subfunction 5.

Print amount of time used and number of errors and warnings.

e. Output (via subroutines)

Error Print file	Logical unit 13
MOS Extra Job file	Logical unit 37
Job Assignment Value file	Logical unit 40

f. Interfaces

Called by: N/A

Calls to: APRTCA (PRIM utility)
APP22B (IOK)
GETDAY (RUNDAY)
GETWCT (ISH, ISM, ISS, LH, LM, LS)
WRTERR (O)

g. Arguments. N/A

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

i. Error Codes. The Error Code in DRIVE2/2-B is 0 (zero).

2.5.3.2 PROGRAM DESCRIPTION

a. Identification

Policy Processor Subroutine - APP22B (IOK)
Process Number: 2.2/B

b. Function

This subroutine controls the sequence of the subfunctions included in the Apply Policy function. This version includes get the parameter data and apply the MOS policies. File definitions and operations such as REWIND are performed here.

c. Input (via subroutines)

Parameter file	Logical unit 17
MOS Policy file	Logical unit 34
Combined ISSUE and MOS Extra Job file	Logical unit 36
Job Assignment Value file	Logical unit 39

d. Processing

Get beginning time.

Get Parameter file data.

Sort the Policy file.

Copy the input Job Assignment Value file to the end of file.

Apply MOS policies by calling ISMO with IFLAG=3.

Get and print amount of time used and number of records processed.

e. Output (via subroutines)

Error Print file	Logical unit 13
MOS Extra Job file	Logical unit 37
Job Assignment Value file	Logical unit 40

f. Interfaces

Called by: DRIVE2/2-B

Calls to: GETDAY (RUNDAY)
GETPAR (IOK)
GETWCT (ISH, ISM, ISS, LH, LM, LS)
FINVAL (LUVIN, LUVOUT, IOK)
FSORT (FORTRAN sort utility)
ISMO (IOK)

g. Arguments

IOK - If set to value other than zero, program cannot continue.

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

i. Error Codes. The Error Code in APP22B is 0 (zero).

2.5.4.1 PROGRAM DESCRIPTION

a. Identification

Policy Processor Main Program Number 4 - DRIVE2/2-C
Process Number: 2.2/C

b. Function

This program controls the processing sequence of the Policy Processor. This version applies the policies that refer to all MOS in specific ISSUEs.

c. Input (via subroutines)

Parameter file	Logical unit 17
ISSUE Policy file	Logical unit 33
MOS Extra Job file	Logical unit 37
Job Assignment Value file	Logical unit 40

d. Processing

Get and print the time and date.

Initialize the Error Print file.

Call the subroutine which controls the Apply Policy function, subfunction 6.

Print amount of time used and number of errors and warnings.

e. Output (via subroutines)

Error Print file	Logical unit 13
ISSUE Extra Job file	Logical unit 38
Job Assignment Value file	Logical unit 39

f. Interfaces

Called by: N/A

Calls to: APRTCA (PRIM utility)
APP22C (IOK)
GETDAY (RUNDAY)
GETWCT (ISH, ISM, ISS, LH, LM, LS)
WRTERR (O)

g. Arguments. N/A

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

i. Error Codes. The Error Code in DRIVE2/2-C is 0 (zero).

2.5.4.2 PROGRAM DESCRIPTION

a. Identification

Policy Processor Subroutine - APP22C (IOK)
Process Number: 2.2/C

b. Function

This subroutine controls the sequence of the subfunctions included in the Apply Policy function. This version includes get the parameter data and apply the ISSUE policies. File definitions and operations such as REWIND are performed here.

c. Input (via subroutines)

Parameter file	Logical unit 17
ISSUE Policy file	Logical unit 33
MOS Extra Job file	Logical unit 37
Job Assignment Value file	Logical unit 40

d. Processing

Get the beginning time.

Get the Parameter file data.

Sort the Policy file.

Copy the input Job Assignment Value file to the end of file.

Apply the ISSUE policies by calling ISMO with IFLAG=1.

Get and print the amount of time used and number of records processed.

e. Output (via subroutines)

Error Print file	Logical unit 13
ISSUE Extra Job file	Logical unit 38
Job Assignment Value file	Logical unit 39

f. Interfaces

Called by: DRIVE2/2-C

Calls to: GETDAY (RUNDAY)
GETPAR (IOK)
GETWCT (ISH, ISM, ISS, LH, LM, LS)
FINVAL (LUVIN, LUVOUT, IOK)
FSORT (FORTRAN sort utility)
ISMO (IOK)

g. Arguments

IOK - If set to value other than zero, program cannot continue.

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

i. Error Codes. There are no Error Codes in APP22C.

2.5.5.1 PROGRAM DESCRIPTION

a. Identification

Policy Processor Main Program Number 5 - DRIVE2/3
Process Number: 2.3

b. Function

This program controls the processing sequence of the Policy Processor. This version applies the Value file to the job records which had not had a specific policy applied.

c. Input (via subroutines)

Parameter file	Logical unit 17
Value file	Logical unit 19
ISSUE Extra Job Data file	Logical unit 38
Job Assignment Value file	Logical unit 39

d. Processing

Get and print the time and date.

Initialize the Error Print file.

Call the subroutine which performs the Set Base Value function.

Print amount of time used and number of errors and warnings.

e. Output (via subroutines)

Error Print file	Logical unit 13
Job Assignment Value file	Logical unit 40
Unfilled Jobs file	Logical unit 50

f. Interfaces

Called by: N/A

Calls to: APRTCA (PRIM utility)
APP23 (IOK)
GETDAY (RUNDAY)
GETWCT (ISH, ISM, ISS, LH, LM, LS)
WRTERR (O)

g. Arguments. N/A

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

i. Error Codes. The Error Code in DRIVE2/3 is 0 (zero).

2.5.5.2 PROGRAM DESCRIPTION

a. Identification

Policy Processor Subroutine - APP23 (IOK)
Process Number: 2.3

b. Function

This subroutine applies the Value file to job records not matched by a specific policy.

c. Input (via subroutines)

Parameter file	Logical unit 17
Value file	Logical unit 19
ISSUE Extra Job file	Logical unit 38
Job Assignment Value file	Logical unit 39

d. Processing

Get the beginning time.

Apply the base values to the MOS data.

Get and print amount of time used and number of records processed.

e. Output (via subroutines)

Error Print file	Logical unit 13
Job Assignment Value file	Logical unit 40
Unfilled Jobs (from Policy Processor) file	Logical unit 50

f. Interfaces

Called by: DRIVE2/2-B

Calls to: GETDAY (RUNDAY)
GETPAR (IOK)
GETWCT (ISH, ISM, ISS, LH, LM, LS)
FINVAL (LUVIN, LUVOUT, IOK)
FSORT (FORTRAN sort utility)
SETBAS (IOK)

g. Arguments

IOK - If set to value other than zero, program cannot continue.

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

i. Error Codes. There are no Error Codes in APP23.

2.5.5.3 PROGRAM DESCRIPTION

a. Identification

Policy Processor Subroutine - SETBAS (IOK)
Process Number: 2.3.0

b. Function

This subroutine computes the minimum and maximum values for the records that are set using the input Value file.

The Value file contains a minimum percentage to which all MOS within an ISSUE should be filled and an associated value. Only those records which had not previously had a specific policy applied are considered. The first record written by SETBAS for each input record contains the number obtained by multiplying the minimum percentage times the strength of both the minimum and maximum values. The second record will contain zero as the minimum number and a number which represents the number of people needed to reach the maximum fill level. This number is the maximum percentage times the strength minus the maximum number from record 1. The maximum percentage may be greater than 100 percent. The maximum percentage must be zero or larger than the minimum percentage. More descriptive information on the Value File may be found in Appendix C, User Files.

c. Input (via subroutines)

Base Value file	Logical Unit 19
ISSUE Extra Job file	Logical unit 38
Job Assignment Value file	Logical unit 39

d. Processing

Copy the input JAV file to end of file.

Get the Value record (also called base values).

If this is the first pass through this subroutine or if a match was found on the ISSUE, and the job record was used, get the next job record.

If the Value ISSUE = Job ISSUE, write JAV records.

e. Output (via subroutines)

Error Print file	Logical unit 13
Job Assignment Value file	Logical unit 40
Unfilled Jobs (from Policy Processor) file	Logical unit 50

f. Interfaces

Called by: DRIVE2/3

Calls to: GETBAS (IOK)
GETJOB (IOK)
WRTVAL (IOK)
ZILCH
WRTERR (NERR)
WRTUNF (IOK)

g. Arguments

IOK - if set to value other than zero, program cannot continue.

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

i. Error Codes. The Error Codes in SETBAS are 3, 26, 125.

2.5.6.1 PROGRAM DESCRIPTION

a. Identification

Policy Processor Utility Subroutine - CONPOL (POL)

b. Function

This subroutine converts the policy stored in the argument (to this subroutine) into the policy variables for use by subroutine SETPOL. Either SAVPOL or NEWPOL could be used as the argument.

c. Input

COMMON variables

d. Processing

Convert policy data into usable values for SETPOL--ISSUE, MOS, LOGRAD, HIGRAD, AGG, PERMIN, PERMAX, and VALUE.

e. Output

COMMON variables

f. Interfaces

Called by: SETPOL
AGISPO

Calls to: ICHINT (alpha, begin, length, return flag)
RCHFLT (alpha, begin, length, return flag)

g. Arguments

POL - Policy to be converted into policy variables usable by subroutine SETPOL. Either SAVPOL or NEWPOL could be used.

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

i. Error Codes. There are no Error Codes in CONPOL.

2.5.6.2 PROGRAM DESCRIPTION

a. Identification

Policy Processor Utility Subroutine - DOIT (IOK)

b. Function

This subroutine makes sure that the aggregated-ISSUE policy has been looked for and that subroutine SETPOL has been called.

c. Input

COMMON variables

d. Processing

If have not obtained aggregated-ISSUE policy, get it by calling MATPOL.

If have not called SETPOL, then call SETPOL.

Reset flags to keep from repeating sequence when neither ISSUE nor MOS have changed.

e. Output

COMMON variables

f. Interfaces

Called by: FINDAT

Calls to: ZEROIN
ZILCH
SETPOL (IOK)
MATPOL (IOK)

g. Arguments

IOK - if set to value other than zero, program cannot continue.

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

i. Error Codes. There are no Error Codes in DOIT.

2.5.6.3 PROGRAM DESCRIPTION

a. Identification

Policy Processor Utility Subroutine - FINDAT (IOK, IFLAG)

b. Function

This subroutine finds the correct job data for the policy in memory.

c. Input (via subroutines)

MOS-data (ISSUE level) file	Logical unit 31
Combined ISSUE and MOS Extra Job file	Logical unit 36
MOS Extra Job file	Logical unit 37

d. Processing

ISSUE check (when IFLAG = 1 or 2):

If job ISSUE is greater than policy ISSUE, check saved policy.

If job ISSUE matches the saved policy, update the policy variables by calling CONPOL, and when IFLAG = 2, check MOS. When IFLAG = 1, return to write the JAV record.

If job ISSUE is greater than saved policy, get next policy.

If there is not a specific policy for the job ISSUE and if this is an ISSUE only search, write the job data to the extra file.

MOS check (when IFLAG = 2 or 3):

If the job MOS is greater than the policy MOS, check the saved policy.

If the job MOS is greater than the saved policy, get the next policy.

If the job MOS matches the new policy, update the policy variables by calling CONPOL and return to write the JAV record.

e. Output (via subroutines)

Error Print file	Logical unit 13
Combined ISSUE and MOS Extra Job file	Logical unit 36
MOS Extra Job file	Logical unit 37
ISSUE Extra Job file	Logical unit 38

f. Interfaces

Called by: ISMO

Calls to: GETJOB (IOK)
CONPOL (NEWPOL)
GETPOL (IOK)
DOIT (IOK)
ICOMP2 (first alpha, beginning character, second alpha, beginning character, number characters)
WRTEXT (IOK)
MATPOL (IOK)

g. Arguments

IOK - if set to value other than zero, program cannot continue.

IFLAG = 1 match on ISSUE
= 2 match on ISSUE and MOS
= 3 match on MOS only

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

i. Error Codes. There are no Error Codes in FINDAT.

2.5.6.4 PROGRAM DESCRIPTION

a. Identification

Policy Processor Utility Subroutine - FINVAL (LUVIN, LUVOUT, IOK)

b. Function

This subroutine finds the correct place on the Job Assignment Value file for this subprocessor to begin adding new records.

c. Input (by calling GETVAL)

Job Assignment Value file (written by previous subprocessor)	Logical unit 39 or 40

d. Processing

Copy the input Job Assignment Value file to the output Job Assignment Value file.

Set the logical unit number for writing new Job Assignment Value records.

e. Output (by calling WRTVAL)

Job Assignment Value file	Logical unit 39 or 40

f. Interfaces

Called by: APP22B, APP22C, and APP23

Calls to: GETVAL (IOK)
WRTVAL (IOK)

g. Arguments

LUVIN - logical unit number for reading JAV records
LUVOUT - logical unit number for writing JAV records
IOK - if this value is not zero, the program cannot continue.

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

i. Error Codes. There are no Error Codes in FINVAL.

2.5.6.5 PROGRAM DESCRIPTION

a. Identification

Policy Processor Utility Subroutine - GETBAS (IOK)

b. Function

This subroutine reads the value records.

c. Input

Value file Logical unit 19

d. Processing

Read the Value file.

Check the format of the value record.

If the format is acceptable, convert the value record to integers or real numbers.

e. Output

Error Print file Logical unit 13

f. Interfaces

Called by: SETBAS

Calls to: ICHINT (alpha, begin, length, return flag)
RCHFLT (alpha, begin, length, return flag)
WRTERR (NERR)

g. Arguments

IOK - if set to value other than zero, program cannot continue.

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

i. Error Codes. The Error Codes in GETBAS are 23, 24, 25, 26, 61, 62, 99.

2.5.6.6 PROGRAM DESCRIPTION

a. Identification

Policy Processor Utility Subroutine - GETJOB(IOK)

b. Function

This subroutine reads all of the job records.

c. Input

MOS-data (ISSUE level) file	Logical unit 31
Combined ISSUE and MOS Extra Job file	Logical unit 36
MOS Extra Job file	Logical unit 37
ISSUE Extra Job file	Logical unit 38

d. Processing

Read the next job data record from the file specified by LUNJOB.

If the ISSUE has changed, a new set of DEMAN names should be started; set the character flags.

Set character flags based on the present demand node name.

If end-of-file has been reached, set the end-of-file flag (JEOF).

e. Output

Error Print file	Logical unit 13

f. Interfaces

Called by: FINDAT
SETBAS

Calls to: WRTERR (NERR)
NXTISS

g. Arguments

IOK - if set to value other than zero, program cannot continue.

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

i. Error Codes. The Error Codes in GETJOB are 98, 99, 125.

2.5.6.7 PROGRAM DESCRIPTION

a. Identification

Policy Processor Utility Subroutine - GETPOL (IOK, IEOF)

b. Function

This subroutine reads the policy data from any of the four files written by SRTPOL.

c. Input

Aggregated-ISSUE Policy File	Logical unit 32
ISSUE Policy file	Logical unit 33
MOS Policy file	Logical unit 34
Combined ISSUE and MOS Policy file	Logical unit 35

d. Processing

Read the next record of the policy file specified by the va e LUNPOL.

If end-of-file has been reached, set the flag (IEOF).

e. Output

Error Print file Logical unit 13

f. Interfaces

Called by: AGISPO
SETPOL
ISMO
FINDAT

Calls to: WRTERR (NERR)

g. Arguments

IOK - if set to value other than zero, program cannot continue.
IEOF - set to 1 if end-of-file found while reading file.

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

i. Error Codes. The Error Code for GETPOL is 99.

2.5.6.8 PROGRAM DESCRIPTION

a. Identification

Policy Processor Utility Subroutine - MAKAGG (IOK)

b. Function

This subroutine computes the values for maximum, minimum, and assignment values for the aggregate and for the grades included in that aggregate.

c. Input

COMMON variables

d. Processing

Create the aggregated JAV record. If all minimums and maximums are zero, do not write new record.

To prevent being written as an "extra" record, set input strengths that are used to zero after storing the aggregate strengths.

Write the JAV records by calling WRTVAL.

e. Output (via subroutines)

Error Print file	Logical unit 13
Job Assignment Value file	Logical unit 39 or 40

f. Interfaces

Called by: ISMO

Calls to: WRTERR (NERR)
WRTVAL (IOK)

g. Arguments

IOK - if set to value other than zero, program cannot continue.

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

i. Error Codes. The Error Codes for MAKAGG are 22, 45.

2.5.6.9 PROGRAM DESCRIPTION

a. Identification

Policy Processor Utility Subroutine - MAKIND (IOK)

b. Function

This subroutine writes any individual grades which should have a policy applied but which were not included in any of the aggregation records.

c. Input

COMMON variables

d. Processing

Create the variables for writing a JAV record which contains all individual grades not included in a grade aggregation policy.

Call WRTVAL to write a JAV record.

Call WRTEXT to write the record of unused data.

e. Output (via subroutines)

Error Print file	Logical unit 13
Combined ISSUE and MOS Extra Job file	Logical unit 36
MOS Extra Job file	Logical unit 37
ISSUE Extra Job file	Logical unit 38
Job Assignment Value file	Logical unit 39 or 40

f. Interfaces

Called by: ISMO

Calls to: WRTVAL (IOK)
NXTISS
WRTEXT (IOK)

g. Arguments

IOK - if set to value other than zero, program cannot continue.

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

i. Error Codes. There are no Error Codes in MAKIND.

2.5.6.10 PROGRAM DESCRIPTION

a. Identification

Policy Processor Utility Subroutine - MATPOL (IOK)

b. Function

To ensure that ISSUE or Combined ISSUE and MOS policies will be set in accordance with policies stated in terms of an Aggregated-ISSUE policy, this subroutine finds the appropriate Aggregated-ISSUE indicators. Processing subsequent to this program will check for inconsistencies between the Aggregated-ISSUE indicators and the ISSUE-specific policies.

c. Input

Aggregated-ISSUE Indicator file Logical unit 90

d. Processing

If the ISSUE from the Aggregated-ISSUE Indicator file is less than the job ISSUE, get the next Aggregated-ISSUE Indicator record.

If the ISSUE from the Aggregated-ISSUE Indicator file is greater than the job ISSUE, there are no Aggregated-ISSUE Indicators; return zero indicators.

If the ISSUE from the Aggregated-ISSUE Indicator file equals the job ISSUE, set the indicators.

e. Output

COMMON variables
Error Print file Logical unit 13

f. Interfaces

Called by: FINDAT

Calls to: WRTERR (NERR)

g. Arguments

IOK - if set to value other than zero, program cannot continue.

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

i. Error Codes. The Error Code for MATPOL is 99.

2.5.6.11 PROGRAM DESCRIPTION

a. Identification

Policy Processor Utility Subroutine - NXTISS

b. Function

This subroutine computes the subscript for setting characters 5, 6, 7, and 8 of the "DEMAN" to assure it is always unique.

c. Input

COMMON variables

d. Processing

When reading the job (MOS-data) file for the first time, create the DEMAN name by making characters 5 and 6 nonblank. All MOS records are given a unique permutation of two alphabetic characters followed by two blank characters.

During the second pass through the job file, use the DEMAN name set during the first pass and make character 7 nonblank.

During the third pass through the job file, use the DEMAN name set during the second pass and make character 8 nonblank.

During the final pass through the job file, the values (Job Assignment Values) are set. Change character 8 from an alpha character to a number. The minimum fill record will end with a 1; the maximum fill record will end with a 2.

e. Output

DEMAN (unique demand node name) in COMMON
Error Print file Logical unit 13

f. Interfaces

Called by: GETJOB
MAKAGG
MAKIND
SETBAS

Calls to: WRTERR (NERR)

g. Arguments. N/A

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

i. Error Codes. The Error Code in NXTISS is 97.

2.5.6.12 PROGRAM DESCRIPTION

a. Identification

Policy Processor Utility Subroutine - SETPOL (IOK)

b. Function

This subroutine reads all of the policy records which pertain to one aggregated-ISSUE, one ISSUE, one MOS, or the ISSUE/MOS combination and creates the set of flags needed to correctly compute the Job Assignment Value data.

c. Input (via subroutines)

COMMON variables	
Aggregated-ISSUE Policy file	Logical unit 32
ISSUE Policy file	Logical unit 33
MOS Policy file	Logical unit 34
Combined ISSUE and MOS Policy file	Logical unit 35

d. Processing

Make sure ISSUE and MOS have not changed.

Convert low grade to integer.

Convert high grade to integer.

Create flags for individual grade minimum and maximum.

If policy is for one grade, make sure this policy is the first for this grade in the set of policy statements.

Set flags if multiple, individual grades.

Set flags if aggregated grades.

Get next Policy record.

e. Output (via subroutines)

COMMON variables	
Aggregated-ISSUE Indicator file	Logical unit 90
Error Print file	Logical unit 13

f. Interfaces

Called by: AGISPO
DOIT

Calls to: GETPOL (IOK)
WRTERR (NERR)
CONPOL (NEWPOL)

g. Arguments

IOK - if set to value other than zero, program cannot continue.

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

i. Error Codes. The Error Codes for SETPOL are 13, 22, 30, 44, 46.

2.5.6.13 PROGRAM DESCRIPTION

a. Identification

Policy Processor Utility Subroutine - WRTEXT (IOK)

b. Function

This subroutine writes all the unmatched job records to the extra file.

c. Input

COMMON variables

d. Processing

Write the unmatched job records to the extra file. Use format for files 26 through 31 but replace UIC and ISSUE with demand node name (DEMAN).

e. Output

Error Print file	Logical unit 13
Combined ISSUE and MOS Extra Job file	Logical unit 36
MOS Extra Job file	Logical unit 37
ISSUE Extra Job file	Logical unit 38

f. Interfaces

Called by: MAKIND
FINDAT

Calls to: WRTERR (NERR)

g. Arguments

IOK - if set to value other than zero, program cannot continue.

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

i. Error Codes. The Error Code in WRTEXT is 99.

2.5.6.14 PROGRAM DESCRIPTION

a. Identification

Policy Processor Utility Subroutine - WRTUNF(IOK)

b. Function

This subroutine writes all of the unfilled job records.

c. Input

COMMON variables

d. Processing

Write to the unfilled jobs file the jobs for which no policy and no value data were available.

e. Output

Error Print file	Logical unit 13
Unfilled Jobs (from Policy Processor) file	Logical unit 50

f. Interfaces

Called by: SETBAS
Calls to: WRTERR (NERR)

g. Arguments

IOK - if set to value other than zero, program cannot continue.

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

i. Error Codes. The Error Code in WRTUNF is 99.

2.5.6.15 PROGRAM DESCRIPTION

a. Identification

Policy Processor Utility Subroutine - WRTVAL (IOK)

b. Function

This subroutine writes all of the Job Assignment Value records.

c. Input

COMMON variables

d. Processing

Write Job Assignment Value record

e. Output

Job Assignment Value file	Logical unit 39 or 40

f. Interfaces

Called by:	MAKIND
	MAKAGG
	SETBAS
Calls to:	WRTERR (NERR)

g. Arguments

IOK - if set to value other than zero, program cannot continue.

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

i. Error Codes. The Error Code in WRTVAL is 99.

2.5.6.16 PROGRAM DESCRIPTION

a. Identification

Policy Processor Utility Subroutine - ZEROIN

b. Function

This subroutine zeroes the policy indicators. It is called prior to starting a new set of policies.

c. Input

Policy indicators in COMMON.

d. Processing

Set all of the policy indicators to zero.

e. Output.

Policy indicators in COMMON

f. Interfaces

Called by: AGISMO
ISMO

Calls to: N/A

g. Arguments

IOK - if set to value other than zero, program cannot continue.

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

i. Error Codes. There are no Error Codes in ZEROIN.

2.5.6.17 PROGRAM DESCRIPTION

a. Identification

Policy Processor Utility Subroutine - ZILCH

b. Function

This subroutine sets the set of policy output variables to zero prior to reading the next job record.

c. Input.

Policy Output variables in COMMON

d. Processing

Zero the set of policy output variables.

e. Output

Policy Output variables in COMMON

f. Interfaces

Called by: ISMO
SETBAS
MAKAGG

Calls to: N/A

g. Arguments. N/A

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

i. Error Codes. There are no Error Codes in ZILCH.

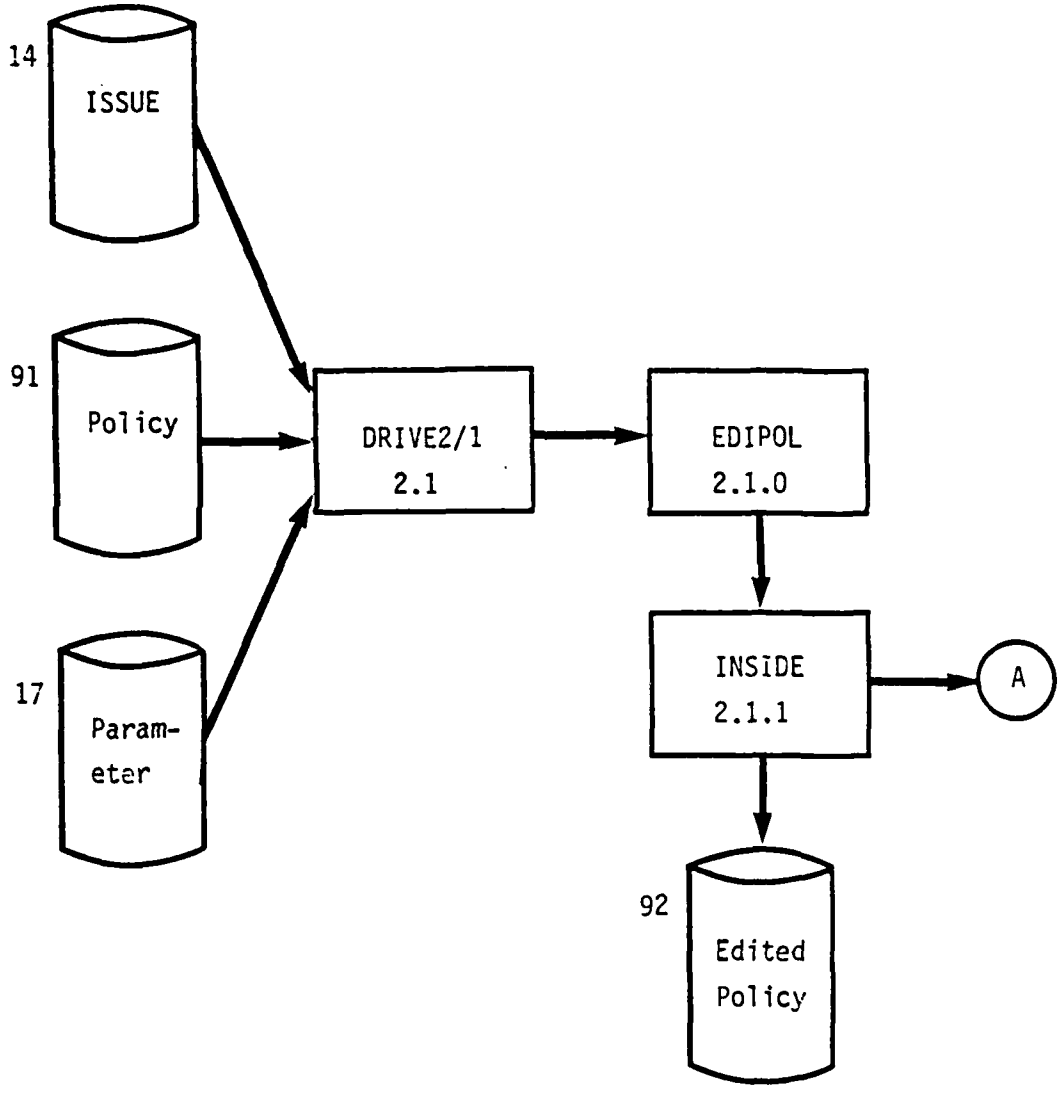


Figure 2-5. Information Flow Through the Policy Processor (page 1 of 5 pages)

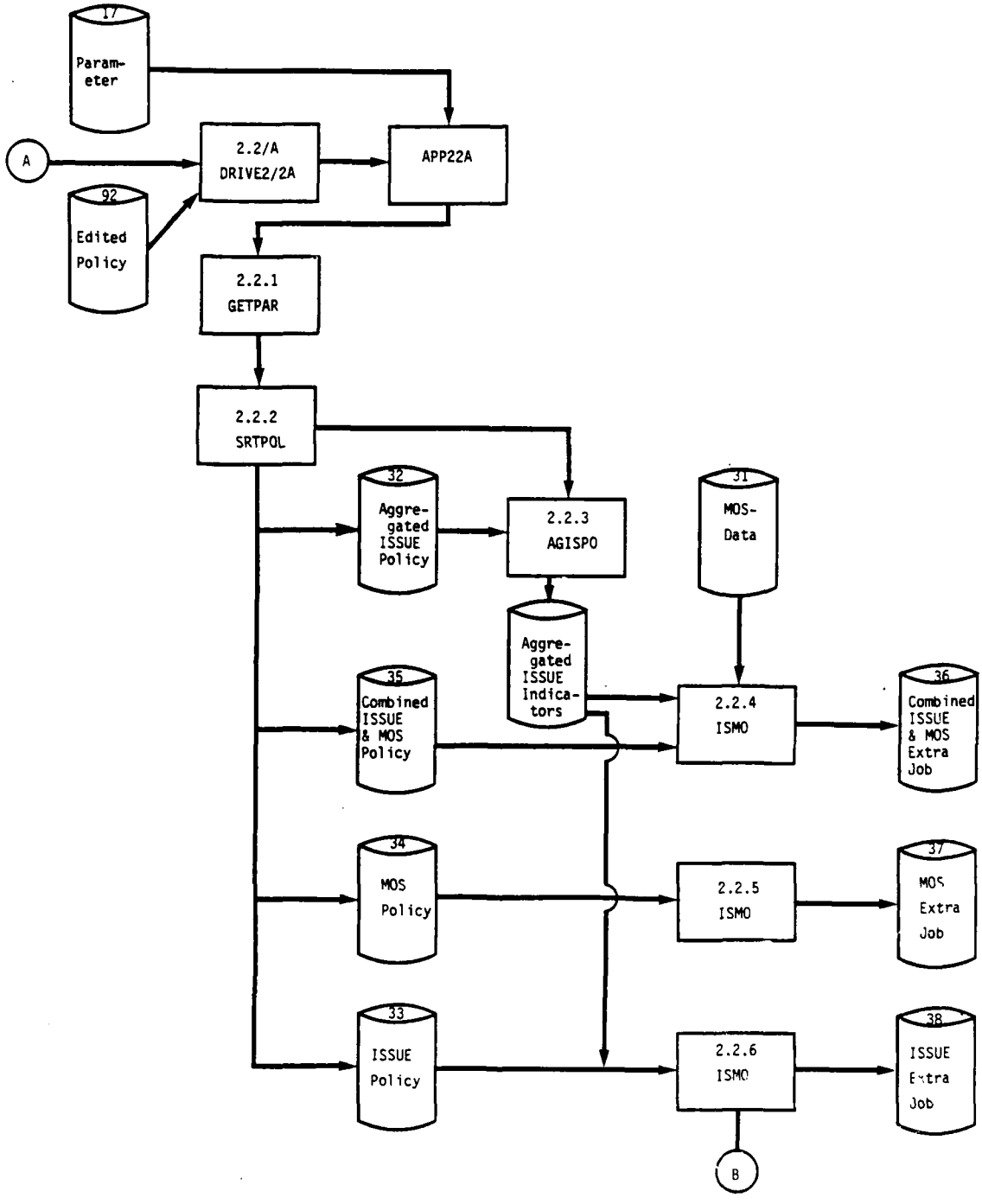
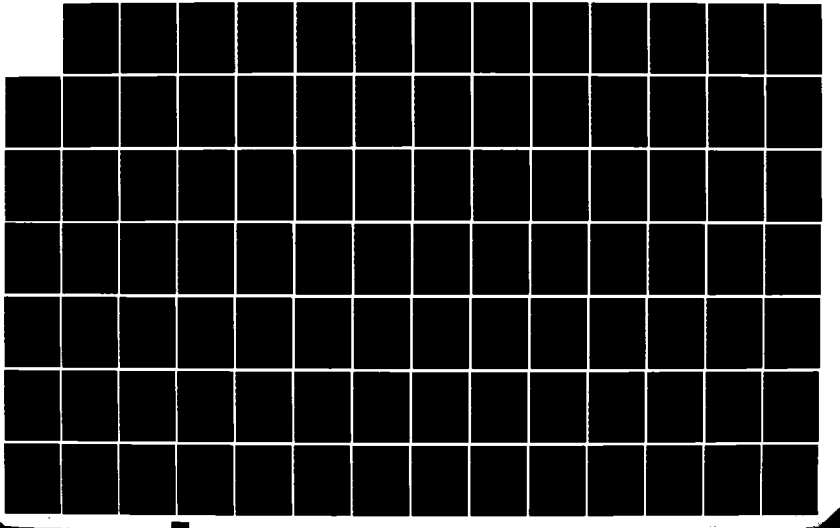


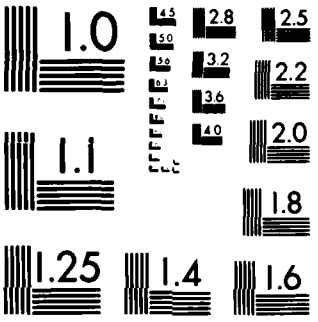
Figure 2-5. Information Flow Through the Policy Processor (page 2 of 5 pages)

AD-A149 505 PERSONNEL READINESS INDICATOR MODEL (PRIM) DOCUMENTATION PROGRAM MAINTENANCE MANUAL(U) ARMY CONCEPTS ANALYSIS AGENCY BETHESDA MD

2/4

UNCLASSIFIED S J VAN NOSTRAND ET AL. NOV 84 CAA-D-84-2 F/G 9/2 NL





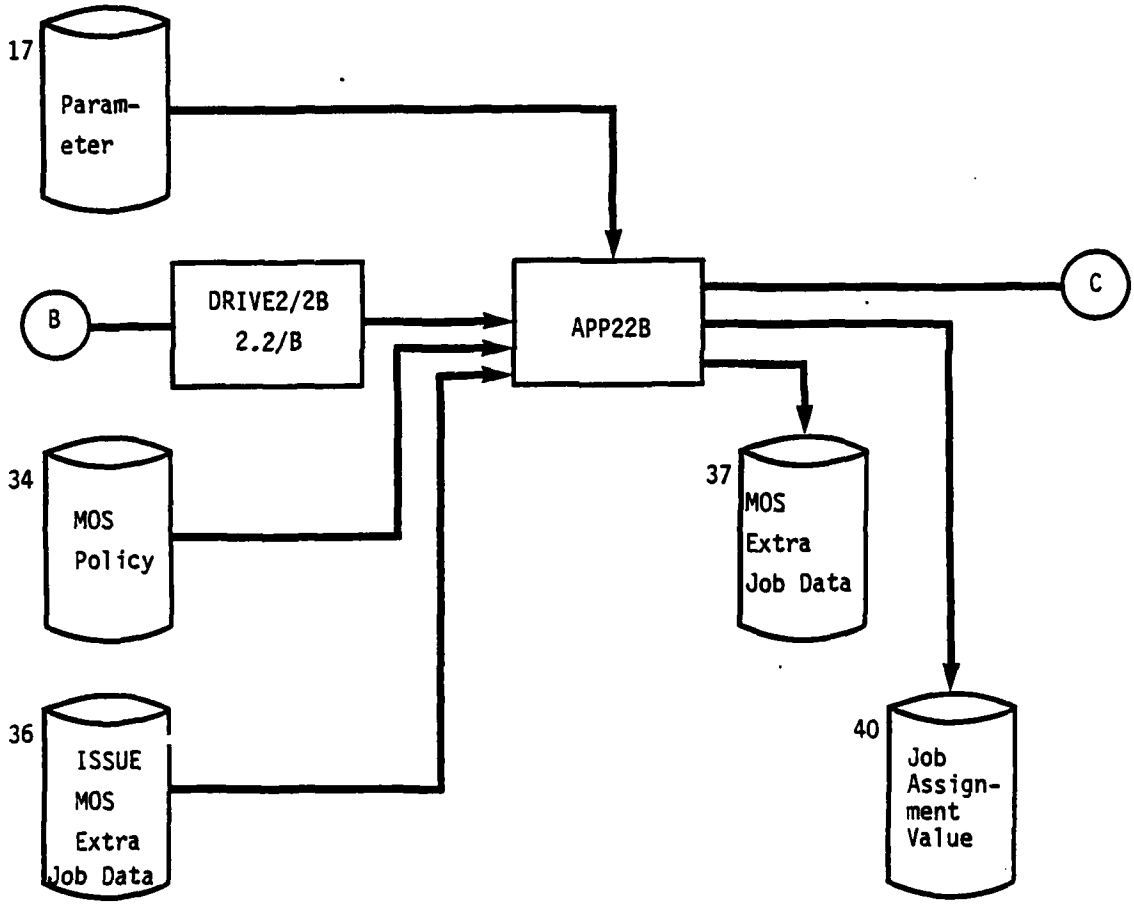


Figure 2-5. Information Flow Through the Policy Processor (page 3 of 5 pages)

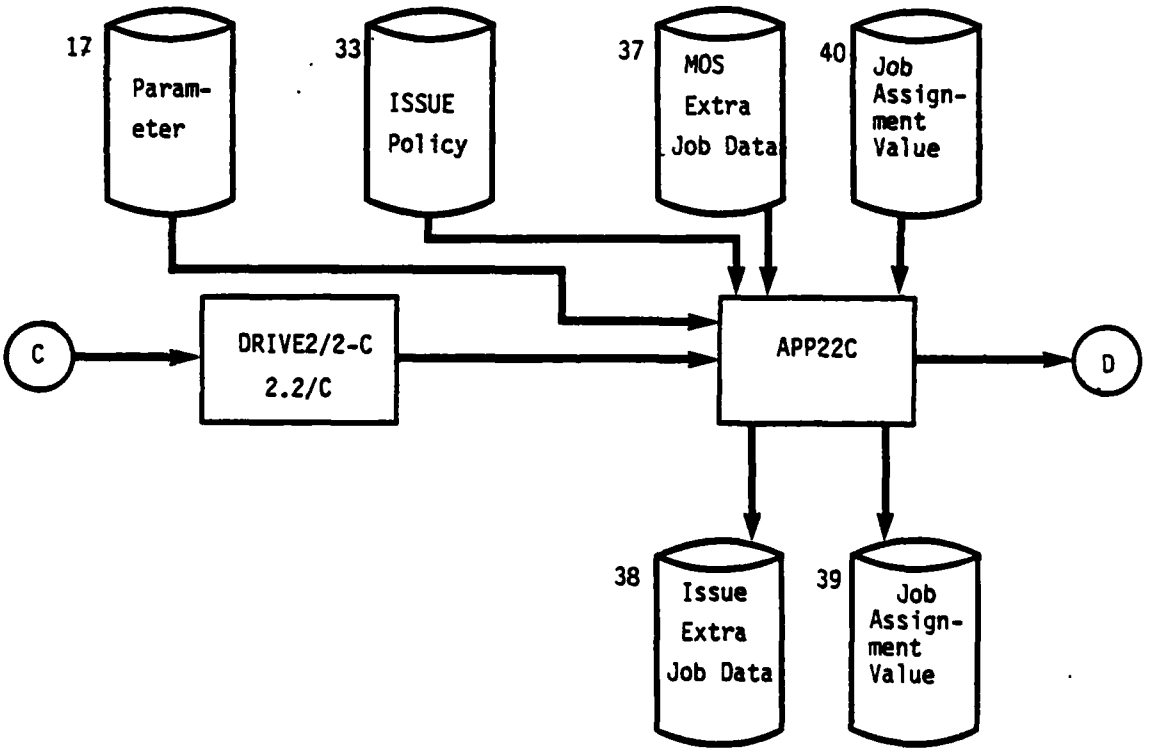


Figure 2-5. Information Flow Through the Policy Processor (page 4 of 5 pages)

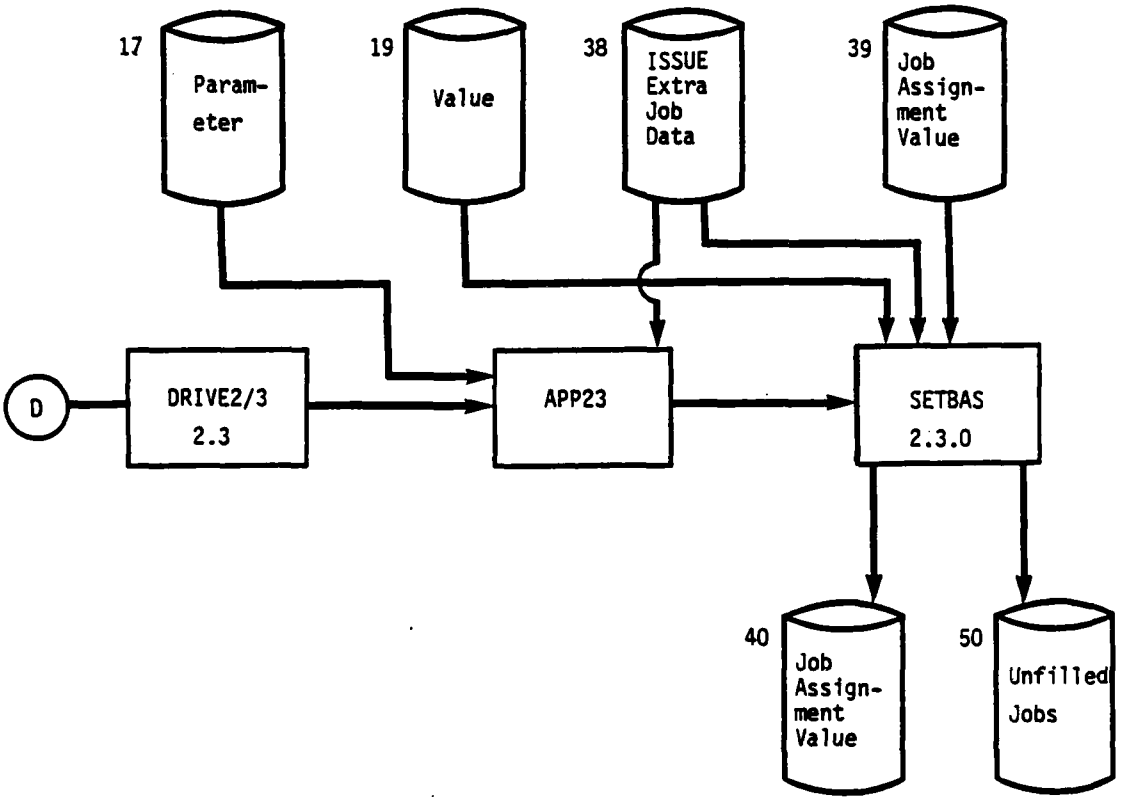


Figure 2-5. Information Flow Through the Policy Processor (page 5 of 5 pages)

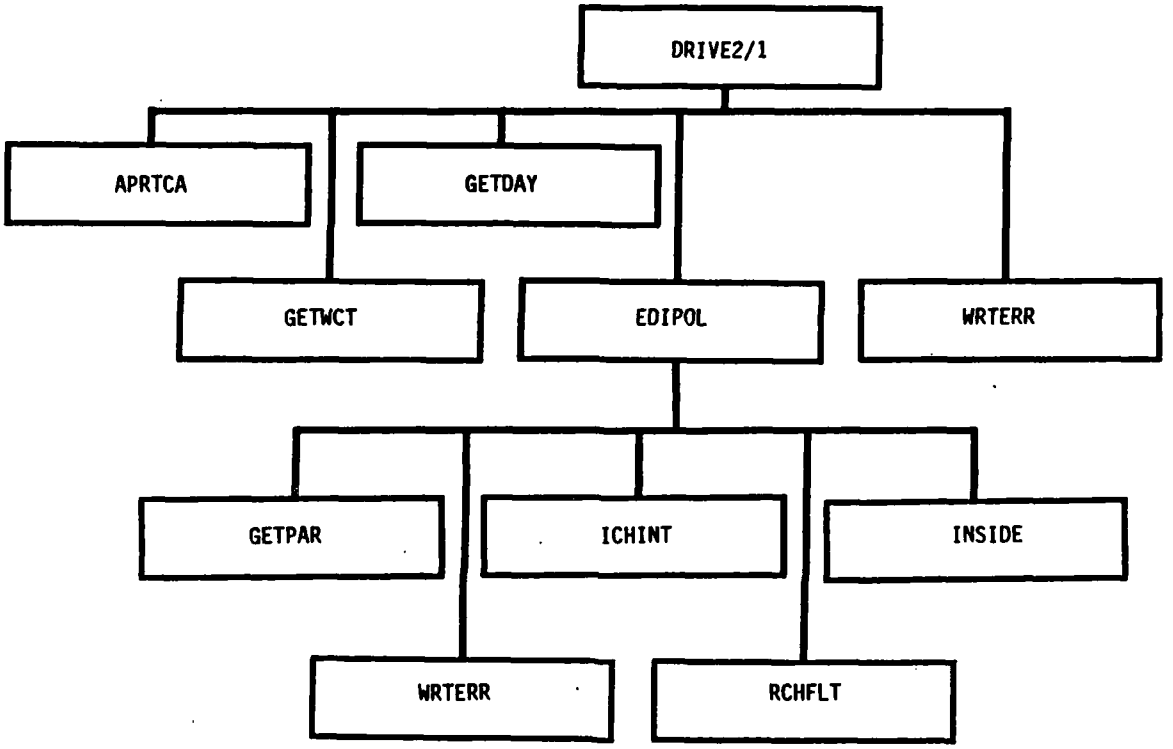


Figure 2-6. Program Unit Hierarchy - Policy Processor (page 1 of 5 pages)

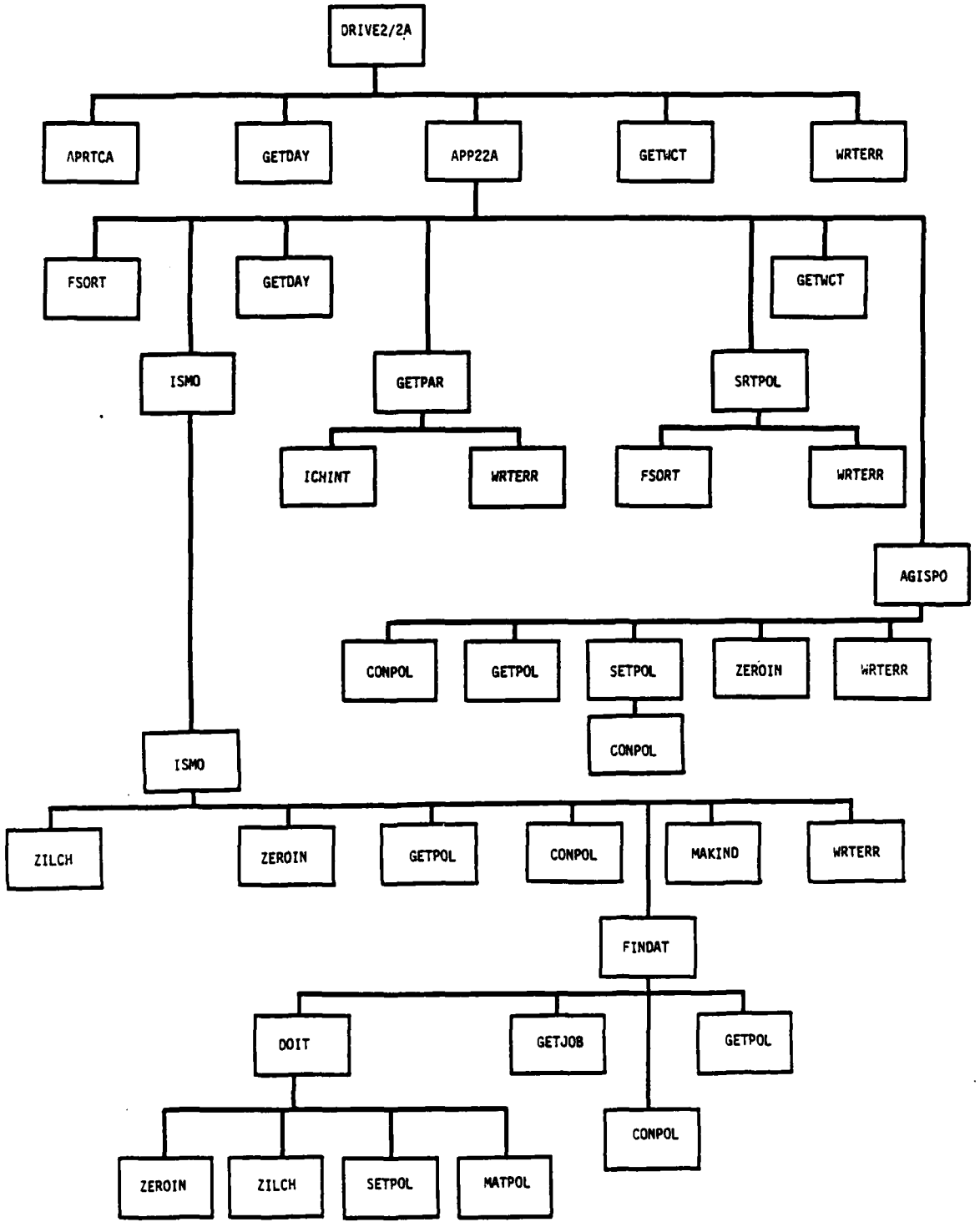


Figure 2-6. Program Unit Hierarchy - Policy Processor (page 2 of 5 pages)

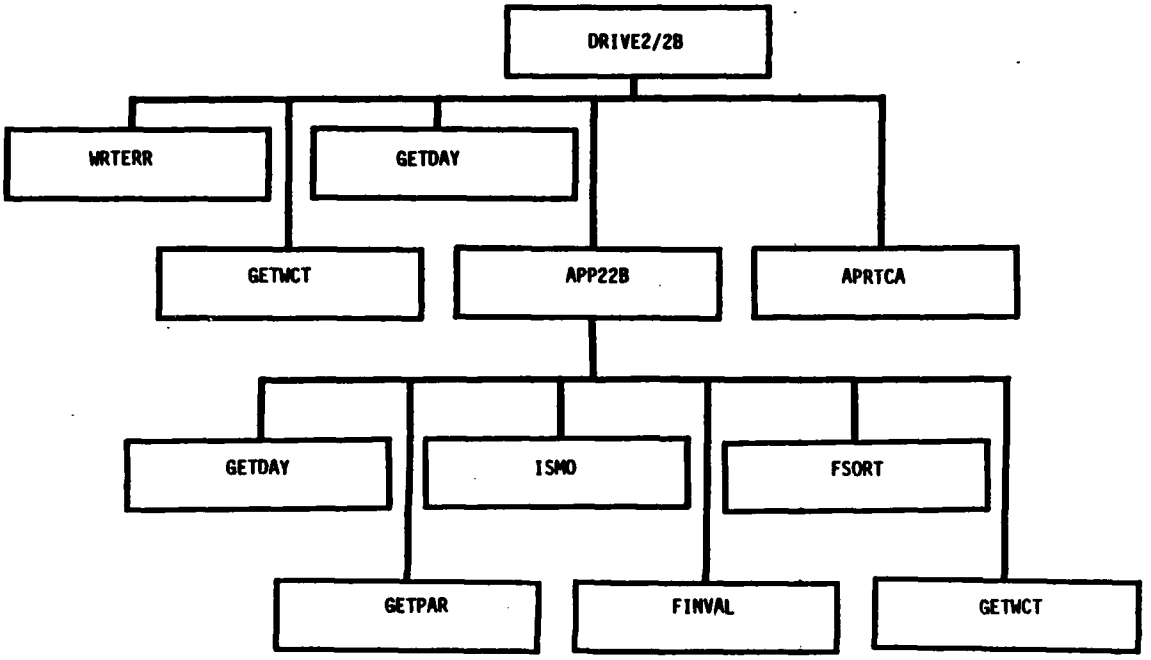


Figure 2-6. Program Unit Hierarchy - Policy Processor (page 3 of 5 pages)

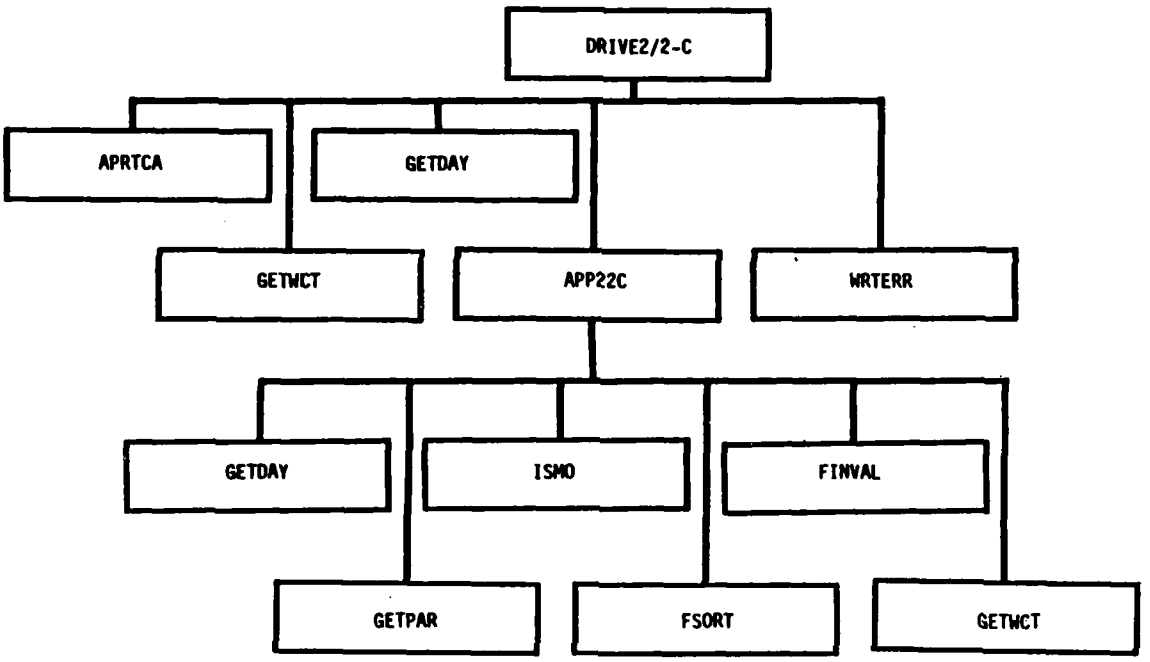


Figure 2-6. Program Unit Hierarchy - Policy Processor (page 4 of 5 pages)

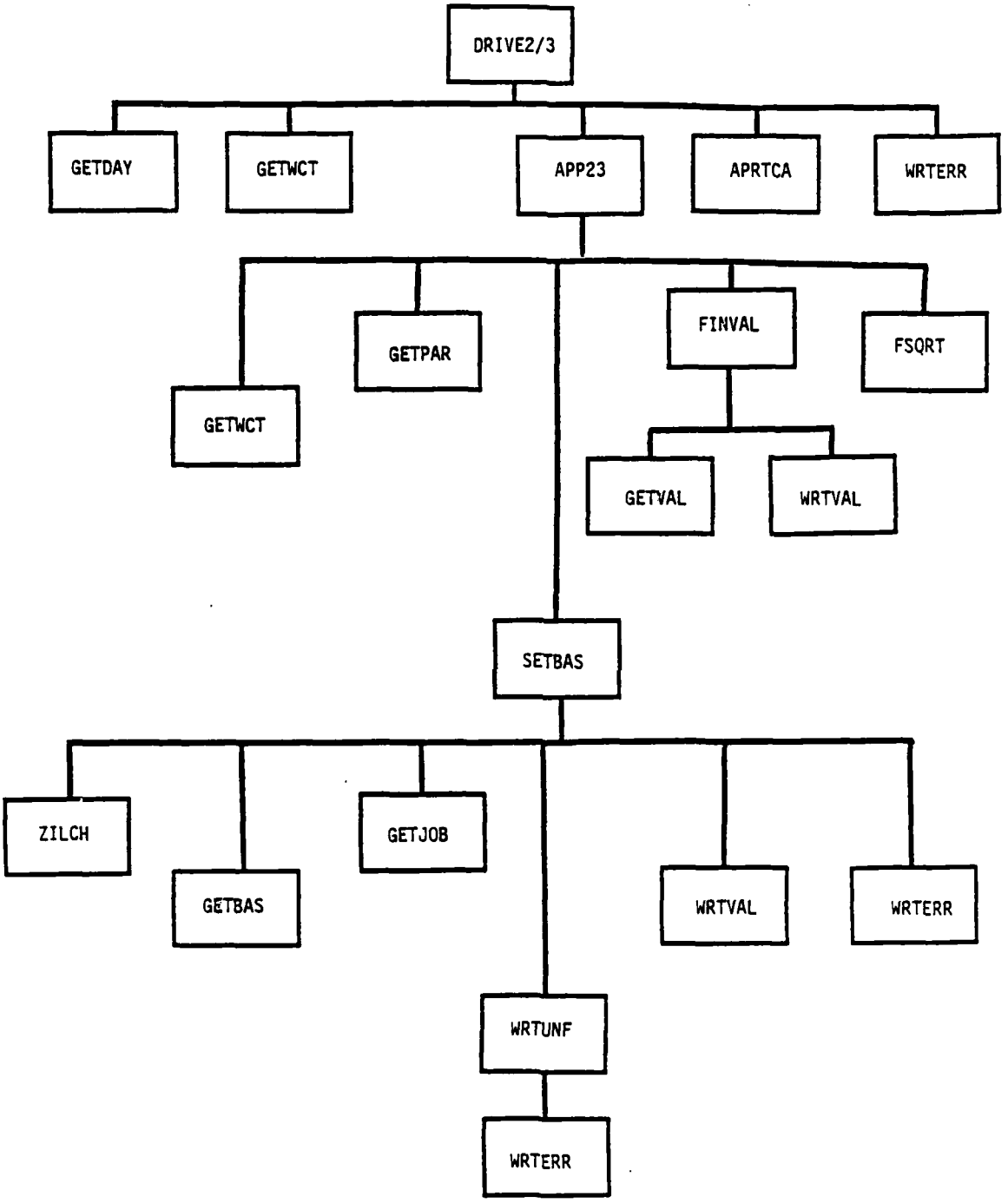


Figure 2-6. Program Unit Hierarchy - Policy Processor (page 5 of 5 pages)

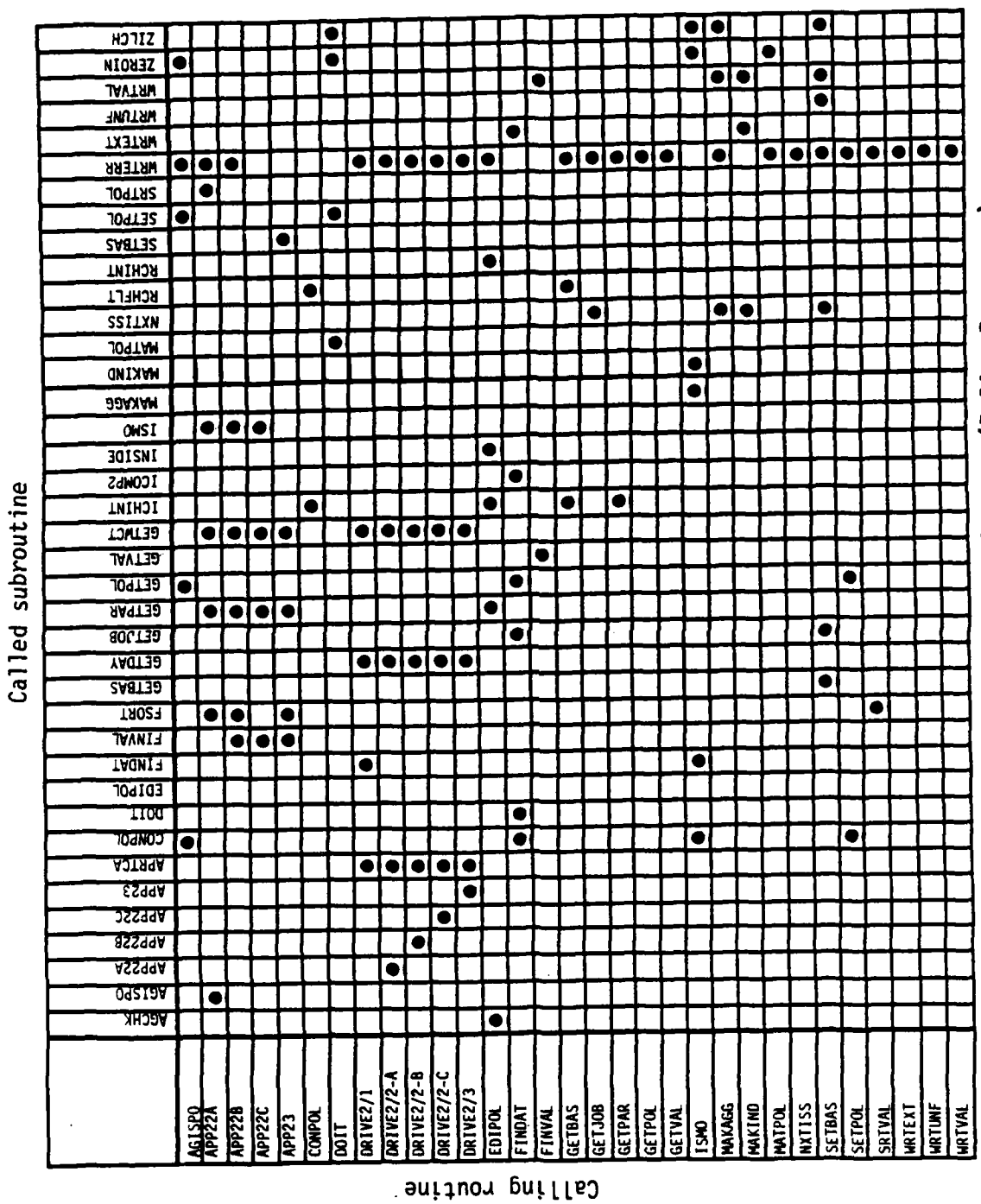


Figure 2-7. Subroutine Cross-reference (Policy Processor)

2.6. PROGRAM DESCRIPTION - Assignment Processor. Subsection 2.6 contains a description of the main programs of the Assignment Processor and the associated subroutines. Since this processor does not use the PRIM standard error writing technique, no error codes exist. A complete listing of the programs discussed in subsection 2.6 is listed below.

Paragraph	Program	Process	Short description
2.6.1	NETWORKMAIN	3.0	Calls the five major functions
2.6.1.1	PRENET	3.1	Prepares data for the network
2.6.1.2	FORM	3.2	Forms the network arcs
2.6.1.3	SNET	3.3	Solves the network
2.6.1.4	REPORT	3.4	Creates the network output file
2.6.1.5	POSTNET	3.5	Saves assignments made by SNET
2.6.2	CREAT51	3.6	Alternate method of computing Unfilled Job file, number 51

Assignments are developed as flows on arcs in a network which defines the personnel inventory as resource nodes and the jobs or authorizations as demand nodes. The flows are the number of assignments made and the arcs are the hypothetical links connecting jobs and inventory. For one MOS, each paygrade that has a number of people in the inventory that is greater than zero becomes a resource node. Similarily, for every ISSUE, each paygrade that has at least one authorization of the same MOS becomes a demand node. Since the number of resources must always match the number of demands, additional resource nodes called super soldiers and additional demand nodes called super jobs are created so that flows may exist on arcs to or from the super nodes as needed for balancing the network. The user's control over the possible amount of flow on each arc is the minimum and maximum percentage fills which can be entered in the Policy file or the Value file. The priority of arc flow is determined by the corresponding values obtained from those files.

The subroutines that are used to solve the network are a variation of the Personnel Network (PNET) presently (1984) used by MILPERCEN for enlisted personnel assignment. Since neither the subroutines which read the input data from unit 15 for use in creating the network arcs nor the network solution subroutines were designed and programed at CAA, they are described in general terms only. The User Manual provided by the vendor for use with these subroutines (Personnel Scheduling Program) will be found in Appendix D. The only changes to these programs have been minor formatting, not to the network logic. In addition, the main program has been changed to allow cycling through all MOS rather than requiring a separate run for each MOS. This has been accomplished by adding a preprocessor (PRENET) prior to the subroutine FORM, and a postprocessor (PSTNET) after the subroutine REPORT. Both PRENET and PSTNET are subroutines which are

called from the main program. PRENET generates the input data required for the FORM subroutine by reading four separate data files and extracting the required information. PSTNET writes the results of the network to three different output files: one for the actual job assignments made; one for the unfilled jobs; and one for excess people. The unfilled job file written by PSTNET is based on the number of jobs filled by super soldiers; the requirement for super soldiers is the difference between the minimum job requirement and the assignments. If the user wants the unfilled jobs based on the maximum job requirement, CREAT51, Process 3.6 must be run.

A set of figures is provided to assist the reader in understanding the relationship of the Assignment Processor programs. Figure 2-8 shows the information flow and Figure 2-9 provides the program hierarchy. The figures are at the end of section 2-6.

Since the description of the inputs to the FORM subroutine in the Personnel Scheduling Program (PSP) manual are difficult to understand, further clarification of several important areas are presented in this section.

a. Super Jobs and Super Soldiers. Super jobs are imaginary jobs to which the network can assign excess people. Super soldiers are imaginary people used to fill jobs when there are not enough projected people to fill all jobs for each MOS. Super jobs must be created by PRENET for each grade level with an inventory greater than zero in file 21, Numbers-of-People file. PRENET also creates super soldiers, one set for each grade level found in file 40, Job Assignment Value file.

b. Supersets and Subsets. The PSP manual frequently uses the term "subsets" to refer to an aggregation level that might better be called a superset since it refers to a higher- or less-detailed level. For example, if a demand node name of E04-AAA were found in File 40, using the example data and files in this manual, one could detemine that the E04- represents the Eighth Infantry Division which is located in Europe; the higher aggregation level is named USAREUR. (Since the demand node name is seven characters long, using information from Section 2.5 one could also determine that the record was written in response to an MOS policy.) The PSP manual refers to E04-AAA as the demand node and USAREUR as the subset; programs and documentations produced by CAA will refer to the aggregation level represented by USAREUR as the superset. In relation to the grades, the MOS will be called a superset by CAA and a subset in the PSP manual. **Do not confuse superset with super jobs or super soldiers; there is no relationship.**

c. Inputs to the FORM Subroutine (file 15)

(1) The initial name list is no longer required in the input file 15. This data is now passed from PRENET to the FORM subroutine via a COMMON block.

(2) The Demand Data Section. The demand data section consists of two parts: real jobs and super jobs. Real jobs are determined by reading

file 40. Super jobs must be created for each grade level for which there are projected people (from file 21) in the MOS being processed.

(a) **Real Jobs.** The first record of the real job indicates an ISSUE identification code and the number of supersets to which it belongs. In PRENET, the number of supersets for any enlisted demand will always be two: The higher aggregation level, "US Army" and the MOS. These two names will appear in the second record. The MOS is requir' to link people, and super soldiers to the demand. Otherwise, the jobs requiring this MOS would not be filled. The third record indicates the aggregate minimum and maximum number of jobs for this demand. The next records indicate the minimum and maximum number of jobs broken out by grade level (activity). The sum of the minimums across all grade levels cannot exceed the aggregate maximum number of jobs for this demand. Likewise, the sum of the maximums across the grade level cannot be less than the aggregate minimum number of jobs for this demand. For each demand (ISSUE) requiring the MOS being processed, the set of four records will be generated.

(b) **Super Jobs.** Super jobs follow the actual jobs. For each grade level for which there are people in the MOS being processed, a super job is created. Again, super jobs are used to assign excess people and do not really exist. Super jobs begin with the key word "SINK." This is followed by the number of supersets to which this imaginary job belongs and the name of the super job. In this implementation, all super jobs begin with "SJ-" followed by the grade level. Thus, "SJ-E4" would indicate a super job to which people with the rank of E4 could be assigned. Also, as with real jobs, only two supersets are used. They are the MOS and the grade level and are listed in the second record of the super jobs. The third record indicates the maximum number of people that may be assigned to the arc pointing to the node from real people with its value and the maximum number of jobs that go toward the total demand with its value. The maximum numbers used will equal the number of people available in each grade level. The value has been set to negative ten (-10).

(3) **Resources.** The resource section consists of two separate parts: real people and super soldiers. Like super jobs, super soldiers are imaginary and are only used by the model to fill jobs for which there are no projected people available.

(a) **Real People.** The first record consists of the MOS being processed, the number of supersets to which the MOS belongs and the number of demands to which people of this MOS can be assigned. In this implementation, the MOS is its own superset. It is used to link the people with this MOS to the demands and

super jobs. The number of demands is determined by counting the number of demands in the demand section. The second record contains the superset names, which in this case is only the MOS. The third record indicates the minimum and maximum number of people that may be assigned to the different jobs in the demand section. In this case, the minimum and maximum numbers will be the same and are equal to the total number of people in the inventory in this MOS. If there are not enough actual jobs, then the excess people will be assigned to super jobs. The fourth record breaks out the total people in the inventory by grade level. Again, the minimum and maximum will be the same because we want to assign all people. The last record indicates the maximum number of people that can be assigned to a demand by grade level along with an associated preference cost or value. There will be one record of this type for each of the demands that were listed in the demand section for this MOS. The number of these records also corresponds to the last value of the first record in the resource section.

(b) **Super Soldiers.** The record format for super soldiers is identical to that of real people, except that super soldiers are identified by placing "SP-" before the MOS in the first record. If jobs do not exist, super soldiers should not be assigned, so the minimum number of people in records 3 and 4 is changed to zero. If this number was not zero, the model could assign super soldiers to super jobs (the network treats super soldiers the same as real people). The preference value in the last record for super soldiers is set to a negative 10 (-10); thereby, when other values are positive, forcing assignment of real people to jobs prior to assignment of super soldiers. The maximum number of super soldiers that could be assigned is made equal to the sum of maximum number of jobs requested for each demand in the demand section.

d. Assignments. This processor attempts to assign the minimum number of people required for each job first. Once all jobs have their minimum requirement level satisfied, the remaining people will be assigned to jobs having the highest maximum fill values until the maximum number of assignments is filled. If jobs have equal value, the job encountered first will be filled to its maximum level prior to other jobs of the same value. The order of the jobs input to the network is not the order actually used; the network programs sort by the size of the fill value.

e. Manual Discrepancies. If any discrepancy is found between the File Descriptions, section 3.3.2, and Personnel Scheduling Program, Appendix D, the correct version for this application is this section and the File Descriptions, section 3.3.2 of the Program Maintenance Manual.

2.6.1 PROGRAM DESCRIPTION

a. Identification

Assignment Processor Main Program - NETWORKMAIN
Process Number: 3.0

b. Function

This program assigns the personnel inventory (Number-of-people file) to jobs requiring the MOS and grade of the inventory personnel; assignments are made to all grade levels of one MOS or SC. All MOS/SC are assigned by cycling through all subroutines for each MOS/SC.

c. Input (via subroutines)

ISSUE Definition file	Logical unit 14
Parameter file	Logical unit 17
Number-of-people file (Personnel Inventory)	Logical unit 21
Job Assignment Value file	Logical unit 40

d. Processing

Prepare input data for network programs for one MOS or SC using the subroutine PRENET.

Build the arcs for the network using subroutine FORM.

Assign people to jobs using subroutine SNET.

Build the Network Output file using subroutine REPORT.

Save the assignments, unfilled jobs, and excess people using subroutine PSTNET.

e. Output (via subroutines)

Excess People from Regular Assignment	Logical unit 41
Unfilled Jobs from Regular Assignment	Logical unit 51
Assignments from Regular Assignment	Logical unit 61

f. Interfaces

Calls to: PRENET
FORM
SNET
REPORT
PSTNET

g. Arguments. N/A

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

2.6.1.1 PROGRAM DESCRIPTION

a. Identification

Assignment Processor Subroutine - PRENET
Process Number: 3.1

b. Function

This subroutine prepares input data for network programs for one MOS or SC each time it is called.

c. Input

ISSUE Definition file	Logical unit 14
Parameter file	Logical unit 17
Number-of-people file	Logical unit 21
Job Assignment Value file	Logical unit 40

d. Processing

Get the superset names from the ISSUE Definition file.

Get grades and number of MOS characters from the Parameter file.

Get the next set of people from the Number-of-people file.

Get the matching set of jobs from the JAV file.

Write people to Excess People file if there are no jobs with that MOS or SC.

Write jobs to Unfilled Jobs file if there were no people with that MOS or SC.

Write Network Input file for a set of people and jobs that match on SC or MOS.

e. Output

Network Input file	Logical unit 15
Excess People from Regular Assignment	Logical unit 41
Unfilled Jobs from Regular Assignment	Logical unit 51

f. Interfaces

Called by: NETWORKMAIN

Calls to: None

g. Arguments. N/A

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

2.6.1.2 PROGRAM DESCRIPTION

a. Identification

Assignment Processor Subroutine - FORM (not written by CAA)
Process Number: 3.2

b. Function

This subroutine creates the COMMON data needed for later network subroutines and forms the arcs.

c. Input

Network Input file	Logical unit 15

d. Processing

Since this subroutine is not a CAA program, the processing is not described in detail. Please refer to the Personnel Scheduling Program User Manual, Appendix D.

e. Output

Network Scratch file	Logical unit 4

f. Interfaces

Called by: NETWORKMAIN

Calls to: Please refer to Appendix D.

g. Arguments. N/A

h. Tables and Items. Please refer to Appendix D.

2.6.1.3 PROGRAM DESCRIPTION

a. Identification

Assignment Processor Subroutine - SNET (not written by CAA)
Process Number: 3.3

b. Function

Assign the inventory of people to the jobs using a network solution algorithm.

c. Input

Network Scratch file	Logical unit 4

d. Processing

Since this subroutine is not a CAA program, the processing is not described in detail. Please refer to the Personnel Scheduling Program User Manual, Appendix D.

e. Output

Network Scratch file	Logical unit 8
Network Scratch file	Logical unit 9

f. Interfaces

Called by: NETWORKMAIN

Calls to: Please refer to Appendix D.

g. Arguments. N/A

h. Tables and Items. Please refer to Appendix D.

2.6.1.4 PROGRAM DESCRIPTION

a. Identification

Assignment Processor Subroutine - REPORT (not written by CAA)
Process Number: 3.4

b. Function

This subroutine creates an output file for the user from the network flows and, when in standalone mode or flags are set differently, creates printed output reports.

c. Input

Network Scratch file	Logical unit 9

d. Processing

Since this subroutine is not a CAA program, the processing is not described in detail. Please refer to the Personnel Scheduling Program User Manual, Appendix D.

e. Output

Network Output file	Logical unit 2

f. Interfaces

Called by: NETWORKMAIN

Calls to: Please refer to Appendix D.

g. Arguments. N/A

h. Tables and Items. Please refer to Appendix D.

2.6.1.5 PROGRAM DESCRIPTION

a. Identification

Assignment Processor Subroutine - PSTNET
Process Number: 3.5

b. Function

This subroutine saves the assignments made by the network program (SNET) and writes additional excess people records and unfilled job records.

c. Input

Network Output file	Logical unit 2

d. Processing

Read the Network Output file which contains pointers to job names and number of assignments made.

Compute the number of excess people (by getting the number of people assigned to super jobs) and write them to the Excess People file.

Compute the number of unfilled jobs at each location (by getting the numbers of jobs filled with super soldiers) and write them to the Unfilled Jobs file.

Write the good assignments to the Assignment file.

e. Output

Excess People from Regular Assignment	Logical unit 41
Unfilled Jobs from Regular Assignment	Logical unit 51
Assignments from Regular Assignment	Logical unit 61

f. Interfaces

Called by: NETWORKMAIN

Calls to: None

g. Arguments. N/A

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

2.6.2 PROGRAM DESCRIPTION

a. Identification

Assignment Processor Main Program - CREAT51
Process Number 3.6

b. Function

This program is executed when the unfilled jobs should be based on the maximum percentages rather than the minimums, where the minimums were based on the flow on the arcs from super soldiers to actual jobs. The flow is the number needed to fill to the minimum percentage only. This program overwrites the Unfilled Job file from process 3.5 and recreates the file using the differences between the maximum percentages and the numbers of real assignments.

c. Input

Parameter file	Logical unit 17
Job Assignment Value file	Logical unit 40
Assignments from Regular Assignment	Logical unit 61

d. Processing

Get grades and number of MOS characters from Parameter file.

Get next job and its associated maximum allowable assignment for each grade level from the JAV file.

Get matching assignments from Assignment file.

Compute any unfilled assignments for each grade level for the job being processed.

If there are unfilled assignments, write job to Unfilled Job file 51.

e. Output

Unfilled jobs from Regular Assignment Logical unit 51

f. Interfaces. N/A

g. Arguments. N/A

h. Tables and Items. Please refer to Appendix A for the Data Dictionary

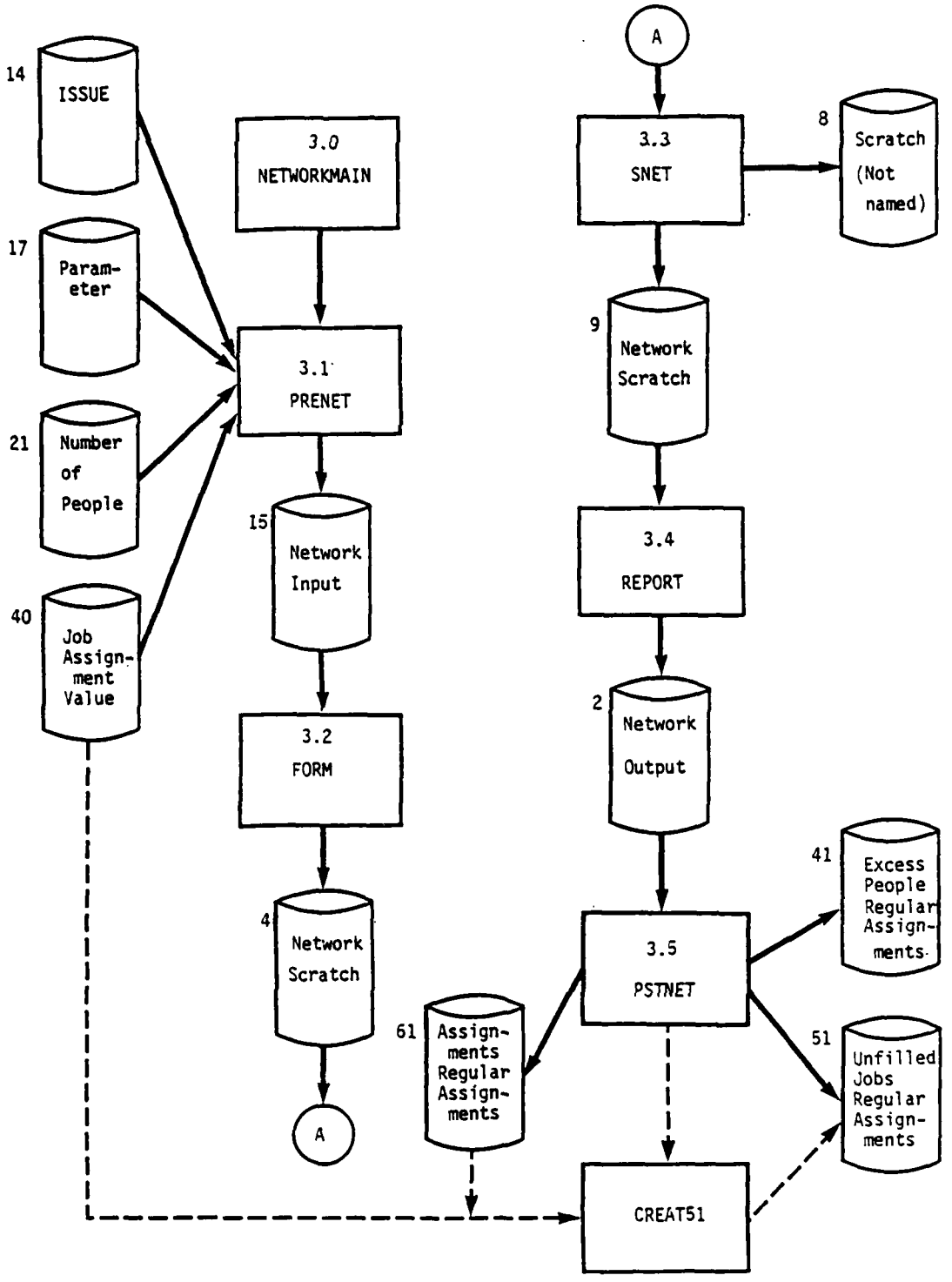


Figure 2-8. Information Flow Through the Assignment Processor

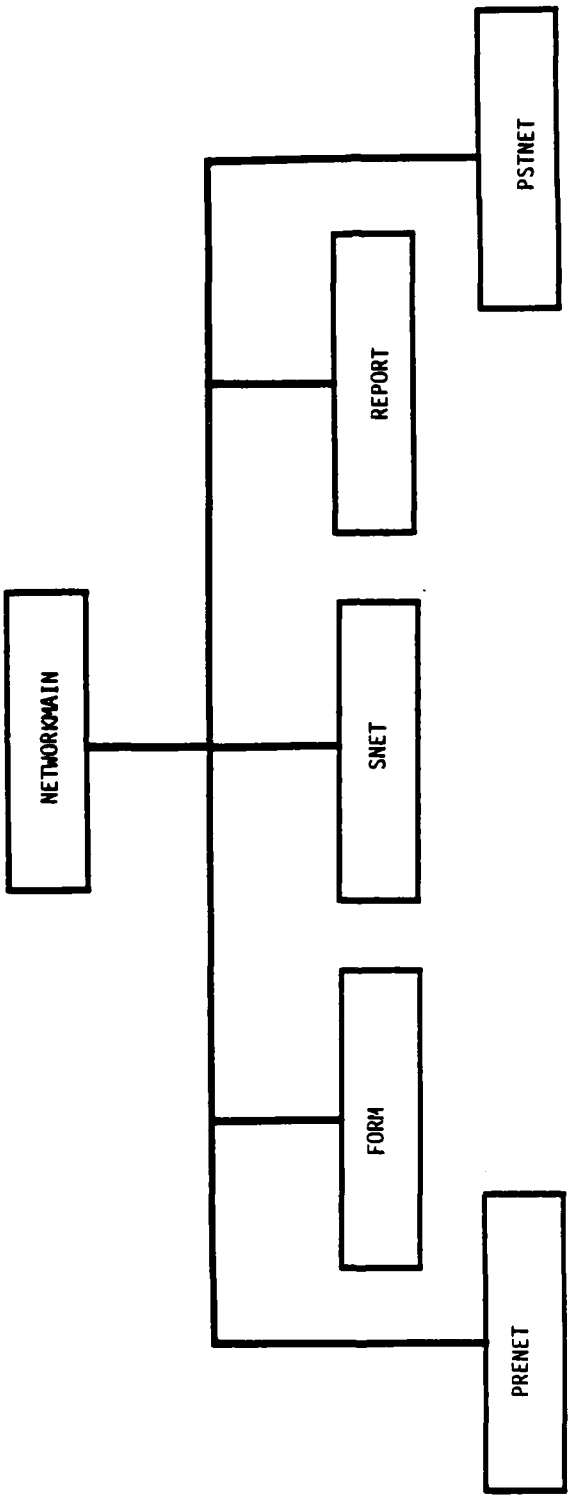


Figure 2-9. Program Unit Hierarchy (Assignment Processor)

2.7 PROGRAM DESCRIPTION - Substitution Assignment Processor. Subsection 2.7 contains a description of the main programs of the Substitution Assignment Processor, Module 1, Module 2, and the associated subroutines. The programs discussed in subsection 2.7 are listed below.

Paragraph	Program	Process	Short description
2.7.1.	GRDMAIN	4.1	Module 1 - grade substitution main program
2.7.1.1	PREGRD	4.1.1	Prepares data for network using grade substitution
2.7.1.2	FORM	4.1.2	Creates data for network using grade substitution
2.7.1.3	SNET	4.1.3	Makes the assignments
2.7.1.4	REPORT	4.1.4	Creates the Network Output file
2.7.1.5	PSTGRD	4.1.5	Saves data from network using grade substitution
2.7.2	MOSMAIN	4.2	Module 2 - MOS substitution main program
2.7.2.1	PREMOS	4.2.1	Prepares data for network using MOS substitution
2.7.2.2	FORM	4.2.2	Creates data for network using MOS substitution
2.7.2.3	SNET	4.2.3	Makes the assignments
2.7.2.4	REPORT	4.2.4	Creates the Network Output file
2.7.2.5	PSTMOS	4.2.5	Saves data from network using MOS substitution

The Substitution Assignment Processor is composed of two modules. The first module, Grade Substitution, controls the processing for assignments within matching MOS but across grades. The second module, MOS Substituition, allows assignment of personnel of a different MOS but of the same grade. These modules may be run in either order. The input for both modules is an Unfilled Job file and an Excess People file which are output from the Assignment Processor (Processor 3) or from the other of these two modules. The correct file number for the desired sequence is specified in the Parameter file. It is possible to execute either module several times, using the output from one run as the input to the next run by controlling the file numbers via both the Parameter file and the @USE statements. The major requirement is that the file numbers match Table 2-2. All comments in Section 2.6 regarding network assignment also apply to this processor.

Table 2-2. Substitution File Numbers

Run Sequence	Substitution	Input		Output	
		Job	People	Job	People
1	Grade	51	41	53	43
2	Grade	52	42	53	43
1	MOS	51	41	52	42
2	MOS	53	43	52	42

Figure 2-10 shows the information flow, and Figure 2-11 shows the relationship of the programs within the Substitution Assignment Processor. Both figures are at the end of section 2.7.

2.7.1 PROGRAM DESCRIPTION

a. Identification

Substitution Assignment Processor, Main Program for Module One - GRDMAIN
Process Number: 4.1

b. Function

This program assigns the excess people to unfilled jobs by making the grade substitutions that are specified in the Parameter file. One MOS or SC is assigned on each pass through the network subroutine, SNET.

c. Input (via subroutines)

ISSUE Definition file	Logical unit 14
Parameter file	Logical unit 17
Excess People file	Logical unit 41 or 42
Unfilled Job file	Logical unit 51 or 52

d. Processing

Prepare input data for network programs for one MOS or SC using the subroutine PREGRD.

Build the arcs for the network using the subroutine FORM.

Assign excess people to unfilled jobs using subroutine SNET.

Build the Network Output file using subroutine REPORT.

Save the assignments, the new set of unfilled jobs, and the new excess people using subroutine PSTGRD.

e. Output (via subroutines)

Excess People from Grade Substitution	Logical unit 43
Unfilled Jobs from Grade Substitution	Logical unit 53
Assignments from Grade Substitution	Logical unit 63

f. Interfaces

Calls to: PREGRD
FORM
SNET
REPORT
PSTGRD

g. Arguments. N/A

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

2.7.1.1 PROGRAM DESCRIPTION

a. Identification

Substitute Assignment Processor, Module 1 Subroutine - PREGRD
Process Number: 4.1.1

b. Function

This subroutine prepares input data for network programs for one MOS or SC with grade substitution.

c. Input

ISSUE Definition file	Logical unit 14
Parameter file	Logical unit 17
Excess People file	Logical unit 41 or 42
Unfilled Jobs file	Logical unit 51 or 52

d. Processing

Get the superset names from the ISSUE Definition file.

Get the grades, number of MOS characters, and grade substitutions from the Parameter file.

Get the next set of people from the Excess People file.

Get the matching set of jobs from the Unfilled Jobs file.

Write people to the Excess People file if there are no jobs with the MOS or SC.

Write jobs to the Unfilled Jobs file if there are no people with that MOS or SC.

Write the Network Input file for grade substitution using the set of people and jobs that match on SC or MOS.

e. Output

Network Input file	Logical unit 15
Excess People from Grade Substitution	Logical unit 43
Unfilled Jobs from Grade Substitution	Logical unit 53

f. Interfaces

Called by: GRDMAIN

Calls to: None

g. Arguments. N/A

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

2.7.1.2 PROGRAM DESCRIPTION

a. Identification

Substitute Assignment Processor, Module 1 Subroutine - FORM (not written by CAA)
Process Number: 4.1.2

b. Function

This subroutine creates the COMMON data needed for later network programs and forms the arcs.

c. Input

Network Input file Logical unit 15

d. Processing

Since this subroutine is not a CAA program, the processing is not described in detail. Please refer to the Personnel Scheduling Program User Manual, Appendix D.

e. Output

Network Scratch file Logical unit 4

f. Interfaces

Called by: GRDMAIN

Calls to: Please refer to Appendix D.

g. Arguments. N/A

h. Tables and Items. Please refer to Appendix D.

2.7.1.3 PROGRAM DESCRIPTION

a. Identification

Substitute Assignment Processor Subroutine - SNET (not written by CAA)
Process Number: 4.1.3

b. Function

This subroutine assigns the inventory of people to the jobs using a network solution algorithm.

c. Input

Network Scratch file	Logical unit 4

d. Processing

Since this subroutine is not a CAA program, the processing is not described in detail. Please refer to the Personnel Scheduling Program User Manual, Appendix D.

e. Output

Network Scratch file	Logical unit 8
Network Scratch file	Logical unit 9

f. Interfaces

Called by: GRDMAIN

Calls to: Please refer to Appendix D.

g. Arguments. N/A

h. Tables and Items. Please refer to Appendix D.

2.7.1.4 PROGRAM DESCRIPTION

a. Identification

Substitute Assignment Processor, Module 1 Subroutine - REPORT (not written by CAA)
Process Number: 4.1.4

b. Function

This subroutine creates an output file for the user from the network flows, and, when in standalone mode or flags are set differently, creates printed output reports.

c. Input

Network Scratch file	Logical unit 9

d. Processing

Since this subroutine is not a CAA program, the processing is not described in detail. Please refer to the Personnel Scheduling Program User Manual, Appendix D.

e. Output

Network Output file	Logical unit 2

f. Interfaces

Called by: GRDMAIN

Calls to: Please refer to Appendix D.

g. Arguments. N/A

h. Tables and Items. Please refer to Appendix D.

2.7.1.5 PROGRAM DESCRIPTION

a. Identification

Substitute Assignment Procesor, Module 1 Subroutine - PSTGRD.
Process Number: 4.1.5

b. Function

This subroutine saves the assignments made by the network program and writes additional excess people records and unfilled job records as required.

c. Input

Network Output file	Logical Unit 2

d. Processing

Read the Network Output file which contains pointers to job names and number of assignments made.

Compute the number of excess people (by getting the number of people assigned to super jobs) and write them to the Excess People file.

Compute the number of unfilled jobs at each location (by getting the numbers of jobs filled with super soldiers) and write them to the Unfilled Jobs file.

Write the good assignments to the Assignment file.

e. Output

Excess People from Grade Substitution	Logical unit 43
Unfilled Jobs from Grade Substitution	Logical unit 53
Assignments from Grade Substitution	Logical unit 63

f. Interfaces

Called by: GRDMAIN

Calls to: None

g. Arguments. N/A

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

2.7.2 PROGRAM DESCRIPTION

a. Identification

Substitute Assignment Processor, Main Program for Module Two - MOSMAIN
Process Number: 4.2

b. Function

This program assigns the excess people to unfilled jobs by making the MOS substitutions that are specified in the Parameter file. One MOS or SC is assigned at one time; up to five MOS may be substituted.

c. Input (via subroutines)

ISSUE Definition file	Logical unit 14
Parameter file	Logical unit 17
Excess People file	Logical unit 41 or 43
Unfilled Job file	Logical unit 51 or 53

d. Processing

Prepare input data for network programs for one MOS or SC from the Unfilled Job file using subroutine PREMOS.

Build the arcs for the network using the subroutine FORM.

Assign excess people to unfilled jobs using subroutine SNET.

Build the network output file using subroutine REPORT.

Save the assignments, the new set of unfilled jobs and the new excess people using subroutine PSTMOS.

e. Output (via subroutines)

Excess People from MOS Substitution	Logical unit 42
Unfilled Jobs from MOS Substitution	Logical unit 52
Assignments from MOS Substitution	Logical unit 62

f. Interfaces

Calls to: PREMOS
FORM
SNET
REPORT
PSTMOS

g. Arguments. N/A

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

2.7.2.1 PROGRAM DESCRIPTION

a. Identification

Substitute Assignment Processor, Module 2 Subroutine - PREMOS
Process Number: 4.2.1

b. Function

This subroutine prepares input data for network programs for one MOS or SC with MOS/SC substitution each time it is called.

c. Input

ISSUE Definition file	Logical unit 14
Parameter file	Logical unit 17
Excess People file	Logical unit 41 or 43
Unfilled Jobs file	Logical unit 51 or 53

d. Processing

Get superset names from ISSUE Definition file.

Get grades, number of MOS characters, and MOS and SC substitution from Parameter file.

Get the next set of people from the Excess People file.

Get the matching set of jobs from the Unfilled Jobs file.

Write people to Excess People file if there are no jobs with the substitute MOS or SC.

Write jobs to Unfilled Jobs file if there were no people who can be substituted for that MOS or SC.

Write Network Input file for MOS substitution for the set of people and jobs that match on SC or MOS with substitution.

e. Output

Network Input file	Logical unit 15
Excess People from MOS Substitution	Logical unit 42
Unfilled Jobs from MOS Substitution	Logical unit 52

f. Interfaces

Called by: MOSMAIN

Calls to: None

g. Arguments. N/A

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

2.7.2.2 PROGRAM DESCRIPTION

a. Identification

Substitute Assignment Processor, Module 2 Subroutine - FORM (not written by CAA)
Process Number: 4.2.2

b. Function

This subroutine creates the COMMON data needed for later network programs and forms the arcs.

c. Input

Network Input file — Logical unit 15

d. Processing

Since this subroutine is not a CAA program, the processing is not described in detail. Please refer to the Personnel Scheduling Program User Manual, Appendix D.

e. Output

Network Scratch file — Logical unit 4

f. Interfaces

Called by: MOSMAIN

Calls to: Please refer to Appendix D.

g. Arguments. N/A

h. Tables and Items. Please refer to Appendix D.

2.7.2.3 PROGRAM DESCRIPTION

a. Identification

Substitute Assignment Processor Subroutine - SNET (not written by CAA)
Process Number: 4.2.3

b. Function

This subroutine assigns the inventory of people to the jobs using a network solution algorithm.

c. Input

Network Scratch file	Logical unit 4

d. Processing

Since this subroutine is not a CAA program, the processing is not described in detail. Please refer to the Personnel Scheduling Program User Manual, Appendix D.

e. Output

Network Scratch file	Logical unit 8
Network Scratch file	Logical unit 9

f. Interfaces

Called by: MOSMAIN

Calls to: Please refer to Appendix D.

g. Arguments. N/A

h. Tables and Items. Please refer to Appendix D.

2.7.2.4 PROGRAM DESCRIPTION

a. Identification

Substitute Assignment Processor, Module 2 Subroutine - REPORT (not written by CAA)
Process Number: 4.2.4

b. Function

This subroutine creates an output file for the user from the network flows, and, when in standalone mode or flags are set differently, creates printed output reports.

c. Input

Network Scratch file Logical unit 9

d. Processing

Since this subroutine is not a CAA program, the processing is not described in detail. Please refer to the Personnel Scheduling Program User Manual, Appendix D.

e. Output

Network Output file Logical unit 2

f. Interfaces

Called by: MOSMAIN

Calls to: Please refer to Appendix D.

g. Arguments. N/A

h. Tables and Items. Please refer to Appendix D.

2.7.2.5 PROGRAM DESCRIPTION

a. Identification

Substitute Assignment Processor, Module 2 Subroutine - PSTMOS
Process Number: 4.2.5

b. Function

This subroutine saves the assignments made by the network program and writes additional excess people records and unfilled job records as required.

c. Input

Network Output file	Logical unit 2

d. Processing

Read the Network Output file which contains pointers to job names and number of assignments made.

Compute the number of excess people (by getting the number of people assigned to super jobs) and write them to the Excess People file.

Compute the number of unfilled jobs at each location (by getting the numbers of jobs filled with super soldiers) and write them to the Unfilled Jobs file.

Write the good assignments to the Assignment file.

e. Output

Excess People from MOS Substitution	Logical unit 42
Unfilled Jobs from MOS Substitution	Logical unit 52
Assignments from MOS Substitution	Logical unit 62

f. Interfaces

Called by: MOSMAIN

Calls to: None

g. Arguments. N/A

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

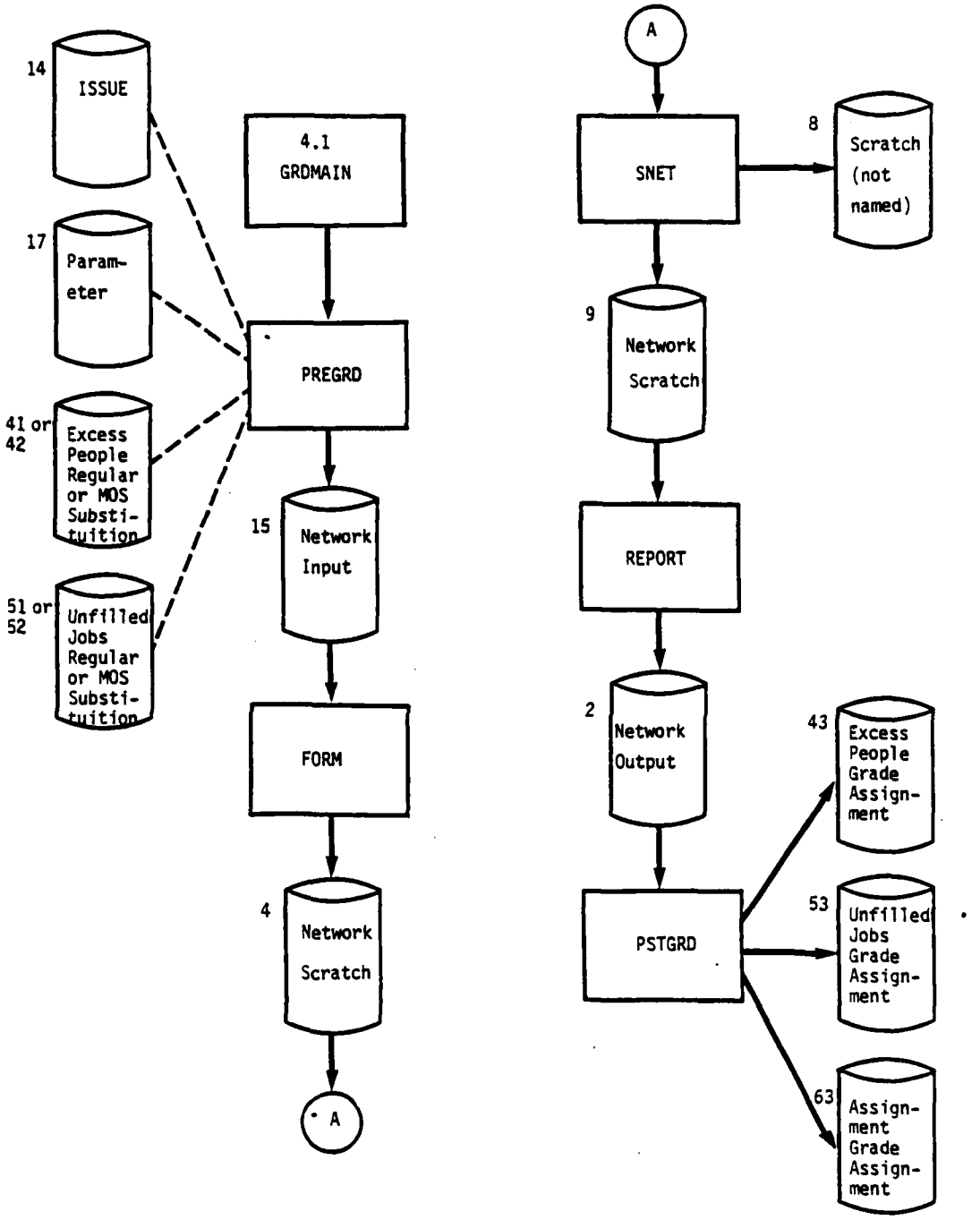


Figure 2-10. Information Flow Through the Substitute Assignment Processor (page 1 of 2 pages)

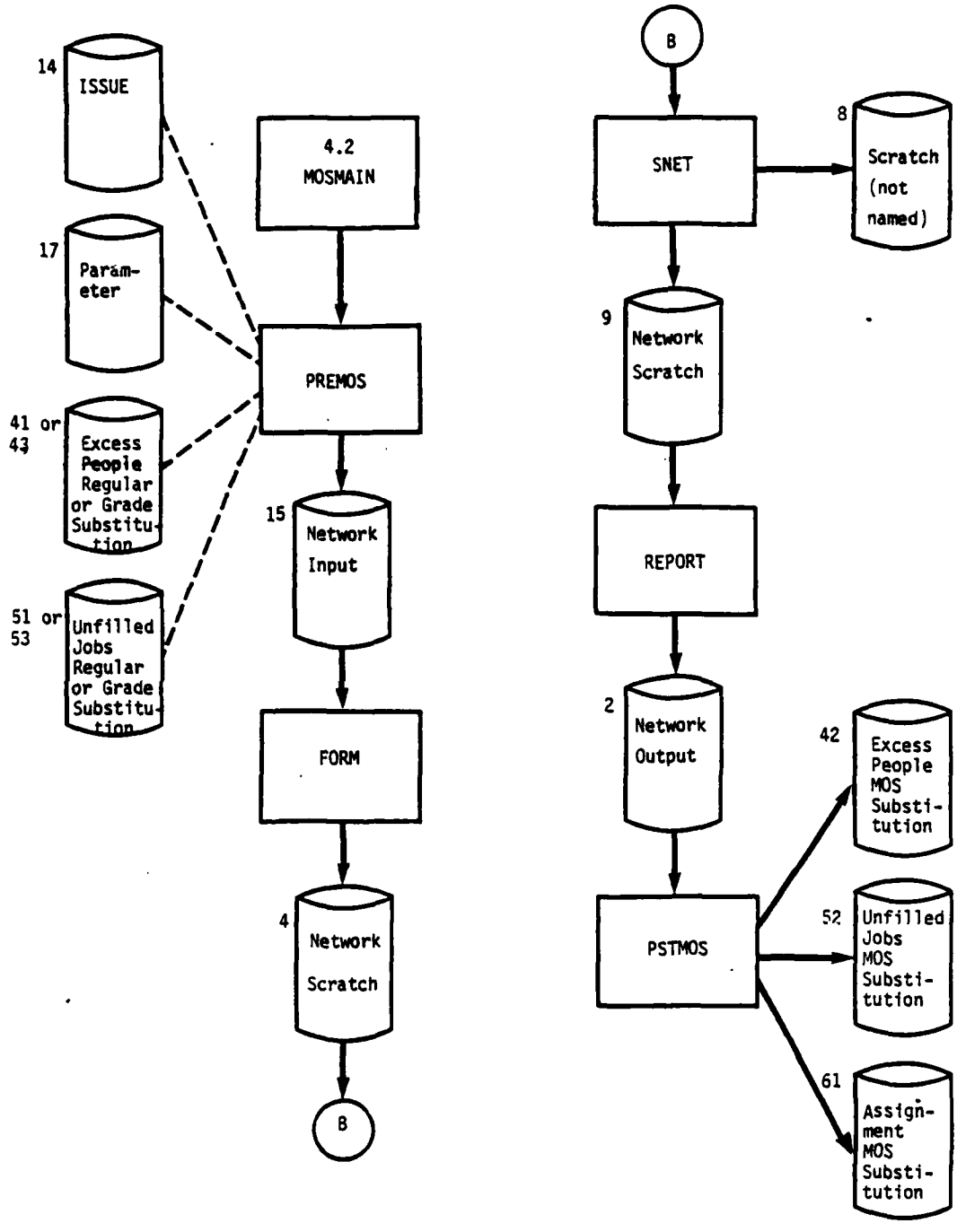


Figure 2-10. Information Flow Through the Substitute Assignment Processor (page 2 of 2 pages)

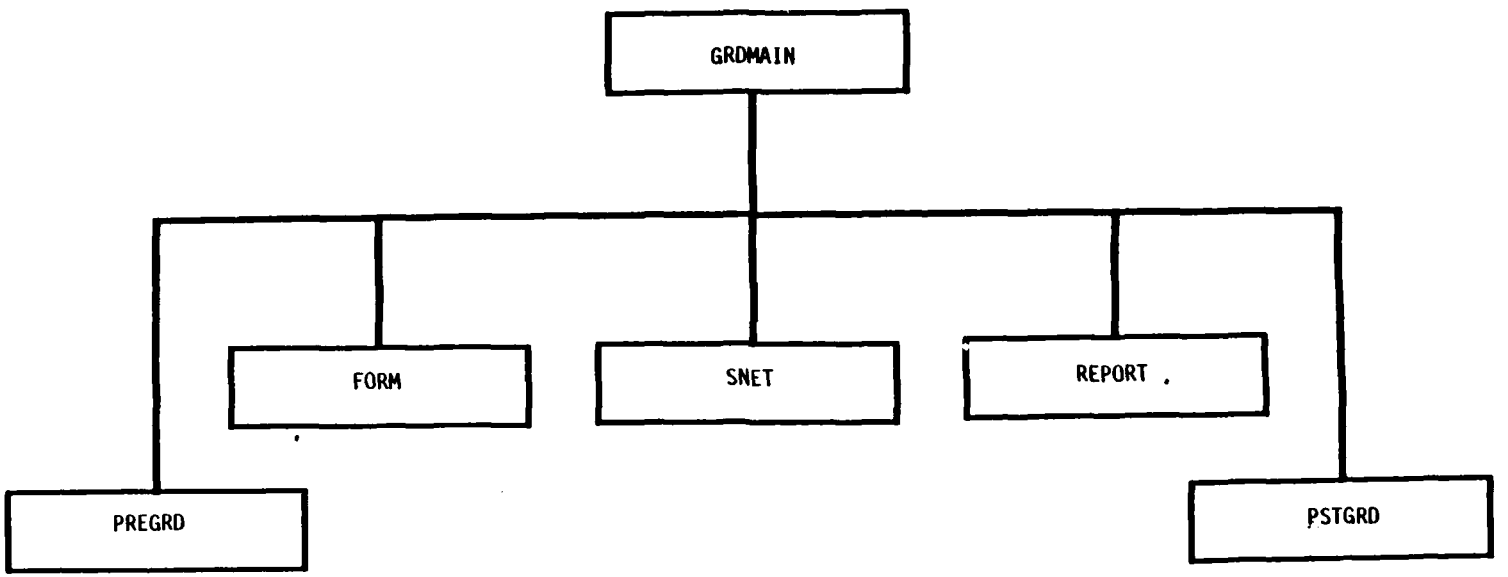


Figure 2-11. Program Unit Hierarchy Substitution Assignment Processor (page 1 of 2 pages)

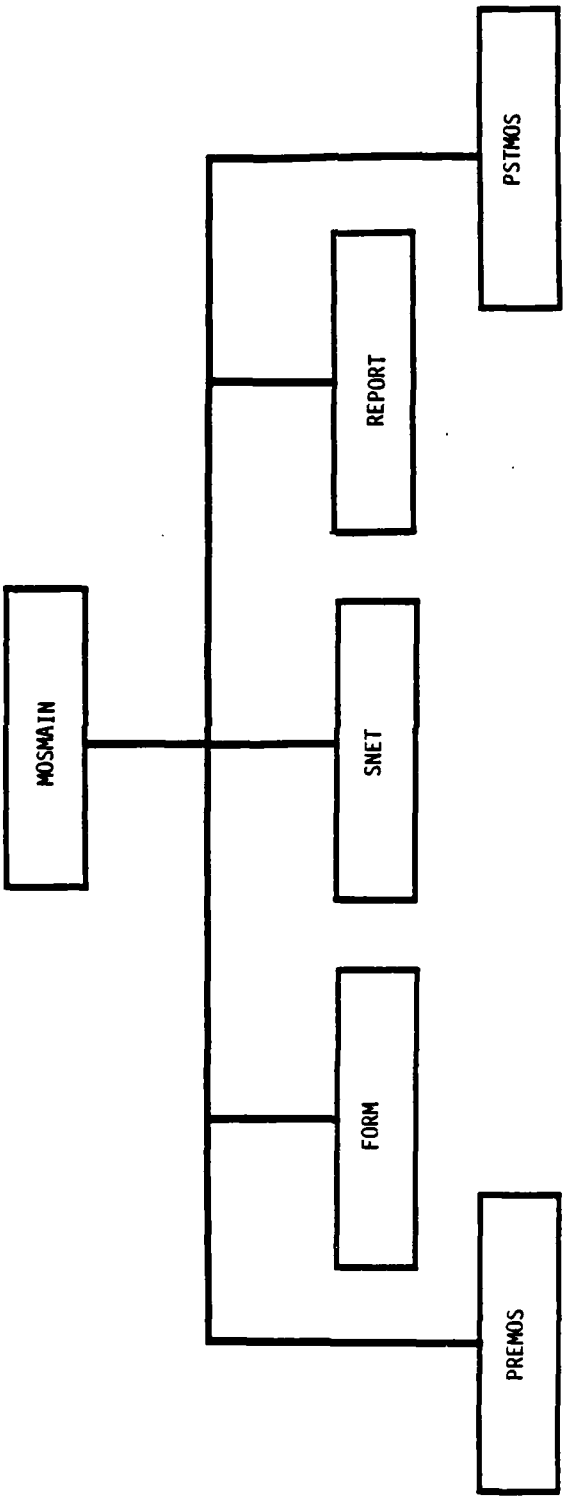


Figure 2-11. Program Unit Hierarchy Substitution Assignment Processor (page 2 of 2 pages)

2.8 PROGRAM DESCRIPTION - READINESS PROCESSOR. Subsection 2.8. contains a description of the main program of the Readiness Processor. The Readiness Processor consists of one main program. Since all input files will have been created by previous processors, the assumptions are made that they will have all been created correctly and that there is no need for error messages to the user. The Readiness Processor combines the input authorized and structure strengths with the assignments from both of the previous processors. During this process, the Readiness Processor reaggregates the multiple records produced by the other processors back up to the ISSUE level. Its final output is the set of files needed for the Report Processor. These output files have been formatted so that they may be easily scanned or printed using the system editor. Figure 2-12, at the end of section 2.8, shows the information flow through the Readiness Processor.

2.8.1 PROGRAM DESCRIPTION

a. Identification

Readiness Processor Main Program, READYMAIN
Process Number 5.0

b. Function

The Readiness Processor computes the available strengths for each MOS and grade in each ISSUE, computes the various percentages required by the reports, and computes the ISSUE C-ratings using the criteria from the Parameter file.

c. Input

Parameter file	Logical unit 17
Value file	Logical unit 19
MOS-data (ISSUE level) file	Logical unit 31
Assignments from Regular Assignment	Logical unit 61
Assignments from MOS Substitution	Logical unit 62
Assignments from Grade Substitution	Logical unit 63

d. Processing

Get the Parameter file data.

Get all ISSUEs from the MOS-data file.

Get the minimum and maximum percentages from the Value file for all ISSUEs found in the MOS-data file.

For each MOS, get the required and authorized strengths from the MOS-data file and compute the sums.

Get assignments from regular assignment and compute available sums and percentages.

Repeat for assignments from grade substitution and then for MOS substitution.

Compute C-ratings.

Write Readiness files.

e. Output

Detailed Readiness Indicators	Logical unit 64
Detailed Readiness Indicators Using MOS Substitution	Logical unit 65
Detailed Readiness Indicators Using Grade Substitituion	Logical unit 66
Total Results including MOS Substitution	Logical unit 67
Aggregated Readiness Indicator file	Logical unit 70
Aggregated Percentage and C-rating file	Logical unit 72
Grade Readiness and Percentage file	Logical unit 73
MOS Readiness, Percentage, and C-rating file	Logical unit 74

f. Interfaces

N/A

g. Arguments. N/A

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

CAA-D-84-2

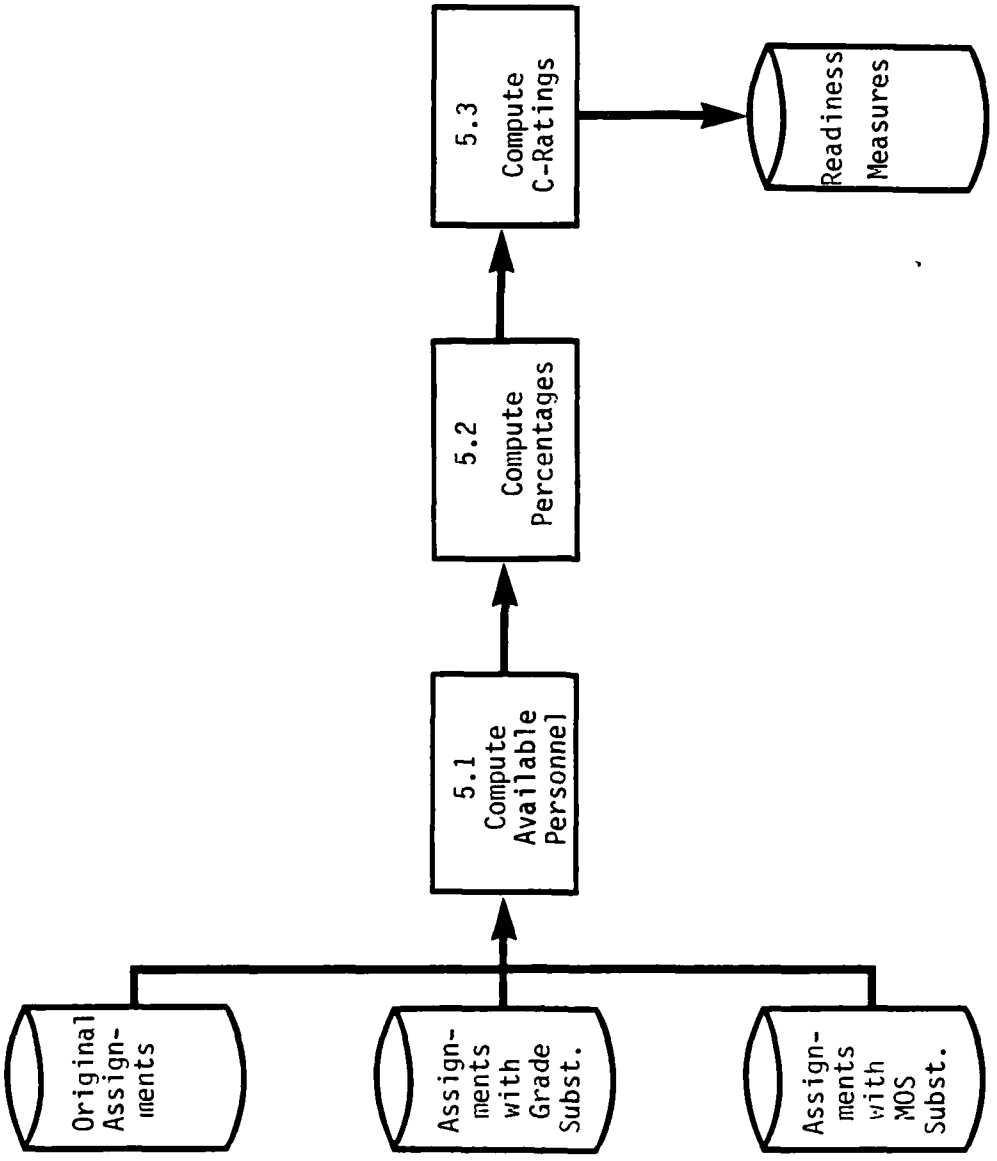


Figure 2-12. Information Flow Through the Readiness Processor

2.9 PROGRAM DESCRIPTION - REPORT PROCESSOR. This section contains a description of the programs in the Report Processor. The Report Processor uses the date and valid grade codes from the Parameter file and the Readiness files that were output from the Readiness Processor to produce printed readiness reports that are formatted and labeled for readability. Some readiness reports are produced at the highest aggregated level. Since the Report Processor can be run as frequently as desired, and any number of reports can be requested on each run, the user may request aggregated level reports first, then choose detail reports for only certain ISSUEs or MOS. See Table 2-3 for a list of the available readiness reports. The main program that controls the Report Processor and the 11 report subroutines, one for each readiness report type, are listed below. Two utility subroutines, used by the Report Processor subroutines, are also described. Figure 2-13 shows the information flow of the Report Processor, and Figure 2-14 shows the relationship of the programs within the Report Processor. Both figures are at the end of section 2.9.

Paragraph	Program	Process	Short description
2.9.1	DRIVE6	6.0	Report Processor main program
2.9.1.1	REPT01	6.1	Subroutine for report type 1
2.9.1.2	REPT02	6.2	Subroutine for report type 2
2.9.1.3	REPT03	6.3	Subroutine for report type 3
2.9.1.4	REPT04	6.4	Subroutine for report type 4
2.9.1.5	REPT05	6.5	Subroutine for report type 5
2.9.1.6	REPT06	6.6	Subroutine for report type 6
2.9.1.7	REPT07	6.7	Subroutine for report type 7
2.9.1.8	REPT08	6.8	Subroutine for report type 8
2.9.1.9	REPT09	6.9	Subroutine for report type 9
2.9.1.10	REPT10	6.10	Subroutine for report type 10
2.9.1.11	REPT11	6.11	Subroutine for report type 11
2.9.1.12	ISSUNA	utility	Gets ISSUE name
2.9.1.13	ZERODV	utility	Performs division

Table 2-3. Readiness Report Types

Report type	Report subtype	Report name
1		ISSUE Summary Report
2		Specialty Summary by Aggregate-ISSUE
3	ALL	Specialty Summary by ISSUE - All Specialities
	MOS code	Specialty Summary by ISSUE - Specific MOS
4		Specialty Summary by Grade
5		Grade Summary by Aggregate-ISSUE
6		Grade Summary by ISSUE
7		High Five Summary
8	ALL	ISSUE Listing - All ISSUEs
	ISSUE code	Specific ISSUE Listing
9		C-ratings
10	ALL	Excess Personnel - All Types
	ORIGINAL	Excess Personnel from Regular Assignment
	MOS SUB	Excess Personnel from MOS Substitution
	GRADE SUB	Excess Personnel from Grade Substitution
11	ALL	Unfilled Jobs - All Types
	ORIGINAL	Unfilled Jobs from Regular Assignment
	MOS SUB	Unfilled Jobs from MOS substitution
	GRADE SUB	Unfilled Jobs from Grade Substitution

2.9.1 PROGRAM DESCRIPTION

a. Identification

Report Processor Main Program - DRIVE6
Process Number: 6.0

b. Function

This program controls the production of all PRIM formatted readiness reports.

c. Input

ISSUE Definition file	Logical unit 14
Parameter file	Logical unit 17
Report Request file	Logical unit 93

d. Processing

Get ISSUE names from the ISSUE Definition file.

Get the as-of-date and grade parameters from the Parameter file.

Get the report number from the Report Request file.

Call the appropriate report subroutine.

e. Output

Count of reports requested
Reports requested

f. Interfaces

Calls to: REPT01
REPT02
REPT03
REPT04
REPT05
REPT06
REPT07
REPT08
REPT09
REPT10
REPT11

Called by: N/A

g. Arguments. N/A

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

i. Error Codes. There are no Error Codes in DRIVE6.

2.9.1.1 PROGRAM DESCRIPTION

a. Identification

Report Processor Subroutine - REPT01 (IFILE)
Process Number: 6.1

b. Function

This subroutine produces report type 1, ISSUE Summary.

c. Input

Aggregated Readiness Indicator file Logical unit 70

d. Processing

Presort the Aggregated Readiness Indicator file on ISSUE.

Read the Aggregated Readiness Indicator file.

Compute subtotals for both ISSUE aggregation levels.

Print the formatted report.

e. Output

Issue Summary Report - Report #1

f. Interfaces

Called by: DRIVE6

g. Arguments

IFILE - Input file logical unit number (70)

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

i. Error Codes. There are no Error Codes in REPT01.

2.9.1.2 PROGRAM DESCRIPTION

a. Identification

Report Processor Subroutine - REPT02 (IFILE)
Process Number: 6.2

b. Function

This subroutine writes report type 2, Specialty Summary by Aggregated ISSUE.

c. Input

MOS Readiness, Percentage, and C-rating file Logical unit 74

d. Processing

Presort the MOS Readiness file on ISSUE within major aggregation level.

Read the MOS Readiness file, aggregate the sums to the highest aggregation level, and compute the percentages.

Print the sums and percentages for each major aggregation, and print the sums across all ISSUEs for each MOS.

e. Output

Specialty Summary by Aggregated-ISSUE Report - (Report #2)

f. Interfaces

Called by: DRIVE6

g. Arguments

IFILE - Input file logical unit number (74)

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

i. Error Codes. There are no Error Codes in REPT02.

2.9.1.3 PROGRAM DESCRIPTION

a. Identification

Report Processor Subroutine - REPT03 (IFILE, MOSPAR)
Process Number: 6.3

b. Function

This subroutine writes report type 3, Specialty Summary by ISSUE.

c. Input

MOS Readiness, Percentage, and C-rating file Logical unit 74

d. Processing

Sort the MOS Readiness file by ISSUE within MOS.

Read and print the MOS Readiness file.

Compute and print the strengths and percentages for the lowest ISSUE level, the highest aggregation level, and the total Army.

If the report subtype is ALL, all specialties are listed. Otherwise, only the specified MOS or SC is listed.

e. Output

Specialty Summary by ISSUE Report - (Report #3)

f. Interfaces

Called by: DRIVE6

g. Arguments

IFILE - Input file logical unit number (74)
MOSPAR - The MOS which should be printed; may be ALL

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

i. Error Codes. There are no Error Codes in REPT03.

2.9.1.4 PROGRAM DESCRIPTION

a. Identification

Report Processor Subroutine - REPT04 (IFILE)
Process Number: 6.4

b. Function

This subroutine writes report type 4, Specialty Summary by Grade.

c. Input

Total Results Including MOS Substitution Logical unit 67

d. Processing

Read aggregated strengths by grade within MOS.

Print aggregated strengths by grade within MOS.

e. Output

Specialty Summary by Grade Report - (Report #4)

f. Interfaces

Called by: DRIVE6

g. Arguments

IFILE - Input file logical unit number (67)

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

i. Error Codes. The Error Code in REPT04 is 30.

2.9.1.5 PROGRAM DESCRIPTION

a. Identification

Report Processor Subroutine - REPT05 (IFILE)
Process Number: 6.5

b. Function

This subroutine writes report type 5, Grade Summary by Aggregated-ISSUE.

c. Input

Grade Readiness and Percentage file Logical unit 73

d. Processing

Read Grade Readiness file.

Aggregate strengths by both aggregation levels and total Army.

Compute percentages for aggregation levels.

Print the report.

e. Output

Grade Summary by Aggregated ISSUE Report - (Report #5)

f. Interfaces

Called by: DRIVE6

g. Arguments

IFILE - Input file logical unit number (73)

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

i. Error Codes. The Error Code in REPT05 is 33.

2.9.1.6 PROGRAM DESCRIPTION

a. Identification

Report Processor Subroutine - REPT06 (IFILE)
Process Number: 6.6

b. Function

This subroutine writes report type 6, Grade Summary by ISSUE.

c. Input

Grade Readiness file Logical unit 73

d. Processing

Read Grade Readiness and Percentage file.

Aggregate strengths for both aggregation levels.

Print the report.

e. Output

Grade Summary by ISSUE Report - (Report #6)

f. Interfaces

Called by: DRIVE6

g. Arguments

IFILE - Input file logical unit number (73)

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

i. Error Codes. The Error Code in REPT06 is 33.

2.9.1.7 PROGRAM DESCRIPTION

a. Identification

Report Processor Subroutine - REPT07 (IFILE)
Process Number: 6.7

b. Function

This subroutine writes report type 7, High-5 Summary.

c. Input

Aggregated Readiness Indicator file Logical unit 70

d. Processing

Read Aggregated Readiness Indicator file.

Compute totals and subtotals of high five enlisted grades fill information for each ISSUE.

Print the report.

e. Output

High-5 Summary Report - (Report #7)

f. Interfaces

Called by: DRIVE6

g. Arguments

IFILE - Input file logical unit number (70)

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

i. Error Codes. There are no Error Codes in REPT07.

2.9.1.8 PROGRAM DESCRIPTION

a. Identification

Report Processor Subroutine - REPT08 (IFILE1, IFILE2, IFILE3, IFILE4, FIELD)
Process Number: 6.8

b. Function

This subroutine writes report type 8, ISSUE Listing. For each ISSUE, the ISSUE Listing report contains data aggregated across all grades and MOS, aggregates of the High-5 enlisted grades, senior grade aggregates, data for each grade aggregated across all MOS, and detailed information by grade and MOS.

c. Input

Aggregated Readiness Indicator file	Logical unit 70
Grade Readiness and Percentage file	Logical unit 73
MOS Readiness, Percentage, and C-rating file	Logical unit 74
Detailed Readiness Indicators	Logical unit 64

d. Processing

Read files.

Sort files by ISSUE, grade, and specialty.

Read from files 70, 73, 74, and 64.

Print aggregated data for the ISSUE (aggregate, High-5, senior grade).

Print aggregated grade data.

For each MOS, print data aggregated over all grades and detail for each grade.

If subtype is ALL, all ISSUEs are listed. Otherwise, only the specified ISSUE is listed.

e. Output

ISSUE Listing Report - (Report #8)

f. Interfaces

Called by: DRIVE6

g. Arguments

IFILE1=70
IFILE2=73 logical unit numbers of input files
IFILE3=74
IFILE4=64

FIELD=report subtype. Valid entries are ALL or a valid ISSUE code.

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

i. Error Codes. There are no Error Codes in REPT08.

2.9.1.9 PROGRAM DESCRIPTION

a. Identification

Report Processor Subroutine - REPT09 (IFILE)
Process Number: 6.9

b. Function

This subroutine writes report type 9, C-rating report.

c. Input

Aggregated Percentage and C-rating file Logical unit 72

d. Processing

Read Aggregated Percentage and C-rating file.

List C-ratings.

e. Output

C-rating Report - (Report #9)

f. Interfaces

Called by: DRIVE6

g. Arguments

IFILE - Input file logical unit number (72)

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

i. Error Codes. There are no Error Codes in REPT09.

2.9.1.10 PROGRAM DESCRIPTION

a. Identification

Report Processor Subroutine - REPT10 (IFILE)
Process Number: 6.10

b. Function

This subroutine writes report type 10, Excess Personnel.

c. Input

Excess People from Regular Assignment	Logical unit 41
Excess People from MOS Substitution	Logical unit 42
Excess People from Grade Substitution	Logical unit 43

d. Processing

Read appropriate file.

Print out personnel specified.

e. Output

Excess Personnel Report - (Report #10)

f. Interfaces

Called by: DRIVE6

g. Arguments

IFILE - Input file logical unit number (41, 42, or 43)

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

i. Error Codes. There are no Error Codes in REPT10.

2.9.1.11 PROGRAM DESCRIPTION

a. Identification

Report Processor Subroutine - REPT11 (IFILE)
Process Number: 6.11

b. Function

This subroutine writes report type 11, Unfilled Jobs by grade, MOS, and ISSUE.

c. Input

Unfilled Jobs from Policy Processor	Logical unit 50
Unfilled Jobs from Regular Assessment	Logical unit 51
Unfilled Jobs from MOS Substitution	Logical unit 52
Unfilled Jobs from Grade Substituion	Logical unit 53

d. Processing-

Read appropriate file.

Print unfilled jobs.

e. Output

Unfilled Jobs Report - (Report #11)

f. Interfaces

Called by: DRIVE6

g. Arguments

IFILE - Input file logical unit number (50, 51, 52, 53)

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

i. Error Codes. There are no Error Codes in REPT11.

2.9.1.12 PROGRAM DESCRIPTION

a. Identification

Report Processor Utility Subroutine - ISSUNA (Issue, ISSNAM, KERR)

b. Function

This subroutine finds the ISSUE name that matches the input ISSUE code.

c, Input

Subroutine Arguments
ISSUE Definition file Logical Unit 14

d. Processing

Read the ISSUE file.

If an ISSUE code is found that matches the input ISSUE, set ISSNAM to the corresponding ISSUE name.

If an ISSUE code is not found that matches the input ISSUE, write error message number 9.

Rewind the ISSUE file and return.

e. Output

Subroutine Argument - ISSNAME

f. Interfaces

Called by: Report Processor Subroutines

Calls to: None

g. Arguments

ISSUE - Character*4 ISSUE code to be matched

ISSNAM - Character*12 ISSUE name

KERR - Integer error indicator

h. Tables and Items. Please refer to Appendix A for the Data Dictionary.

2.9.1.13 PROGRAM DESCRIPTION

a. Identification

Report Processor Utility Subroutine - ZERODV (A, B, C)

b. Function

This subroutine is used to assure that a percentage computation does not divide by zero.

c. Input

Subroutine Arguments - A and B

d. Processing

Set quotient to zero.

If number to be divided is greater than zero and divisor is zero, set quotient to very large number (to provide visual signal on the output).

If both the number to be divided and the divisor are greater than zero, perform the division.

e. Output

Subroutine Argument - C

f. Interfaces

Called by: Report Processor subroutines

Calls to: None

g. Arguments

A - Real The number to be divided

B - Real The divisor

C - Real The quotient

h. Tables and Items. No variable names are used except A, B, and C.

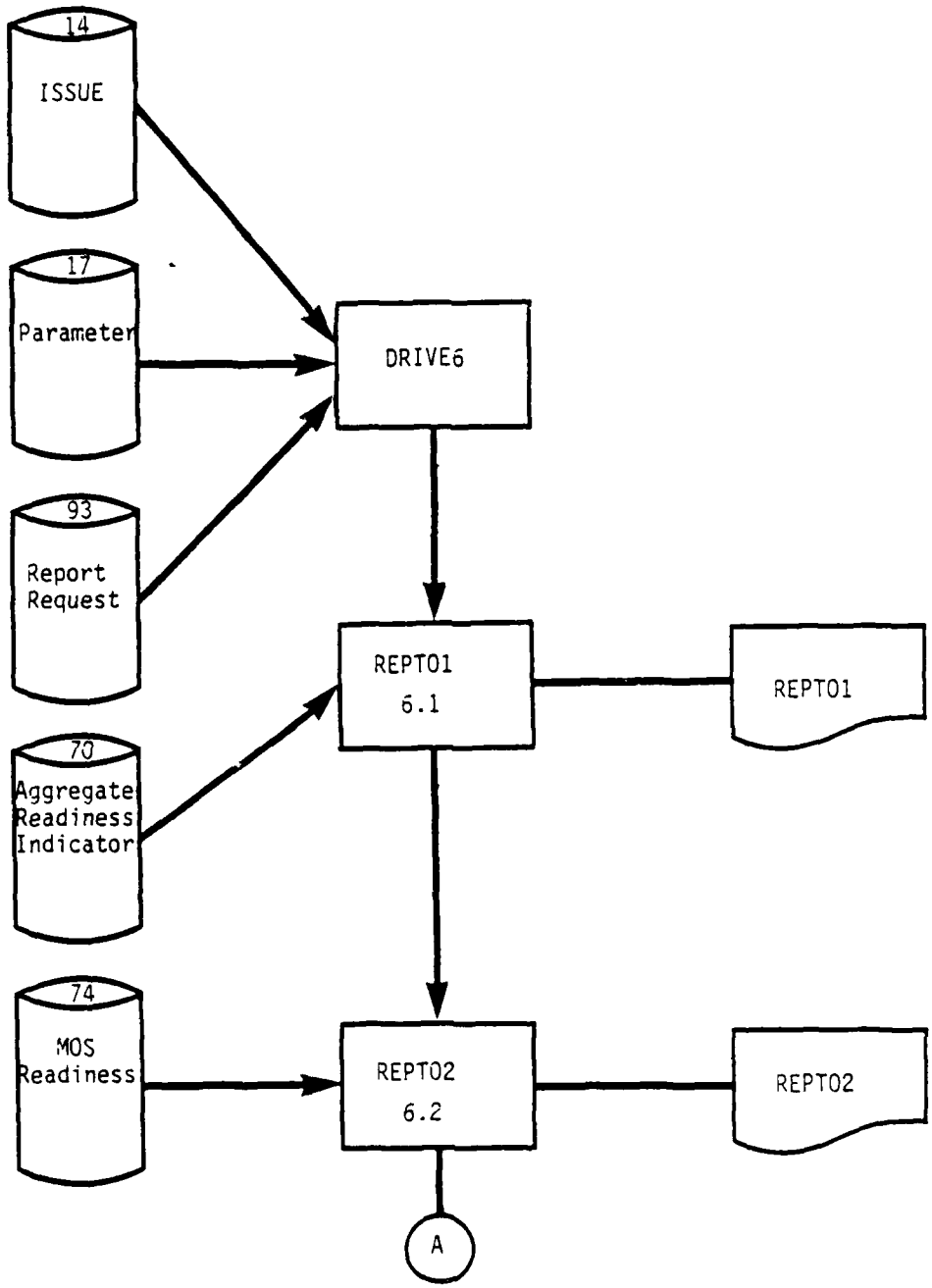


Figure 2-13. Information Flow Through the Report Processor
(page 1 of 4 pages)

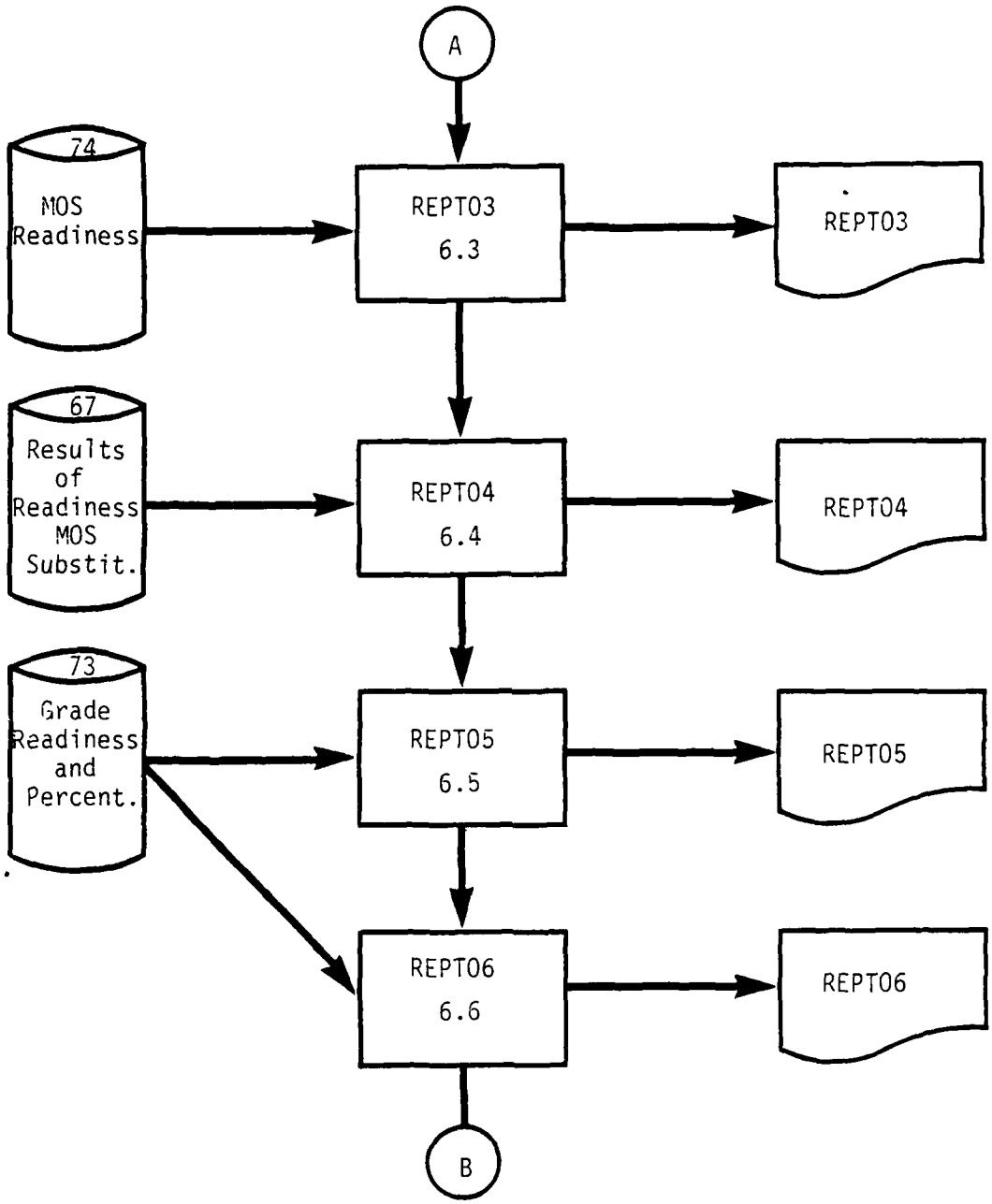


Figure 2-13. Information Flow Through the Report Processor (page 2 of 4 pages)

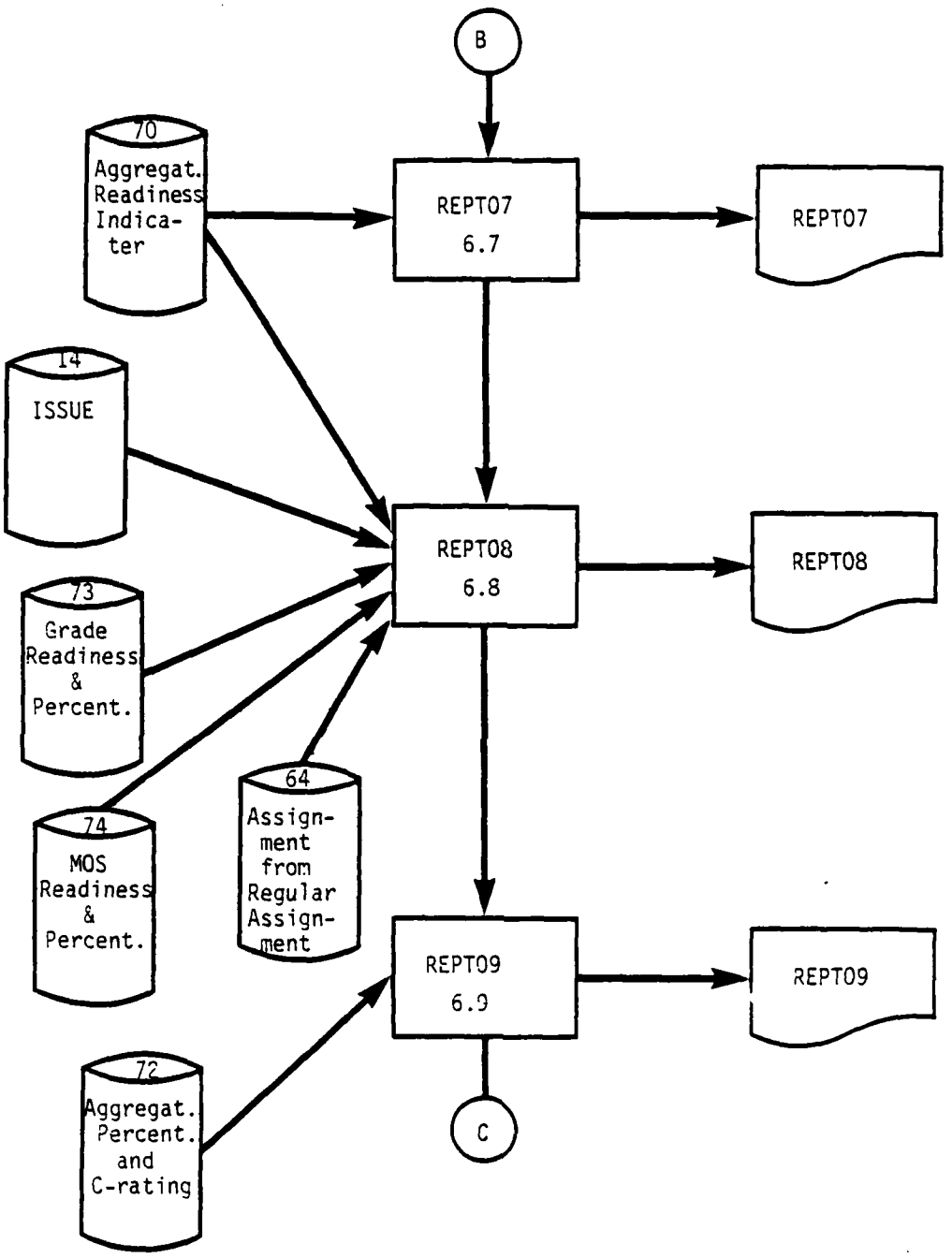


Figure 2-13. Information Flow Through the Report Processor
(page 3 of 4 pages)

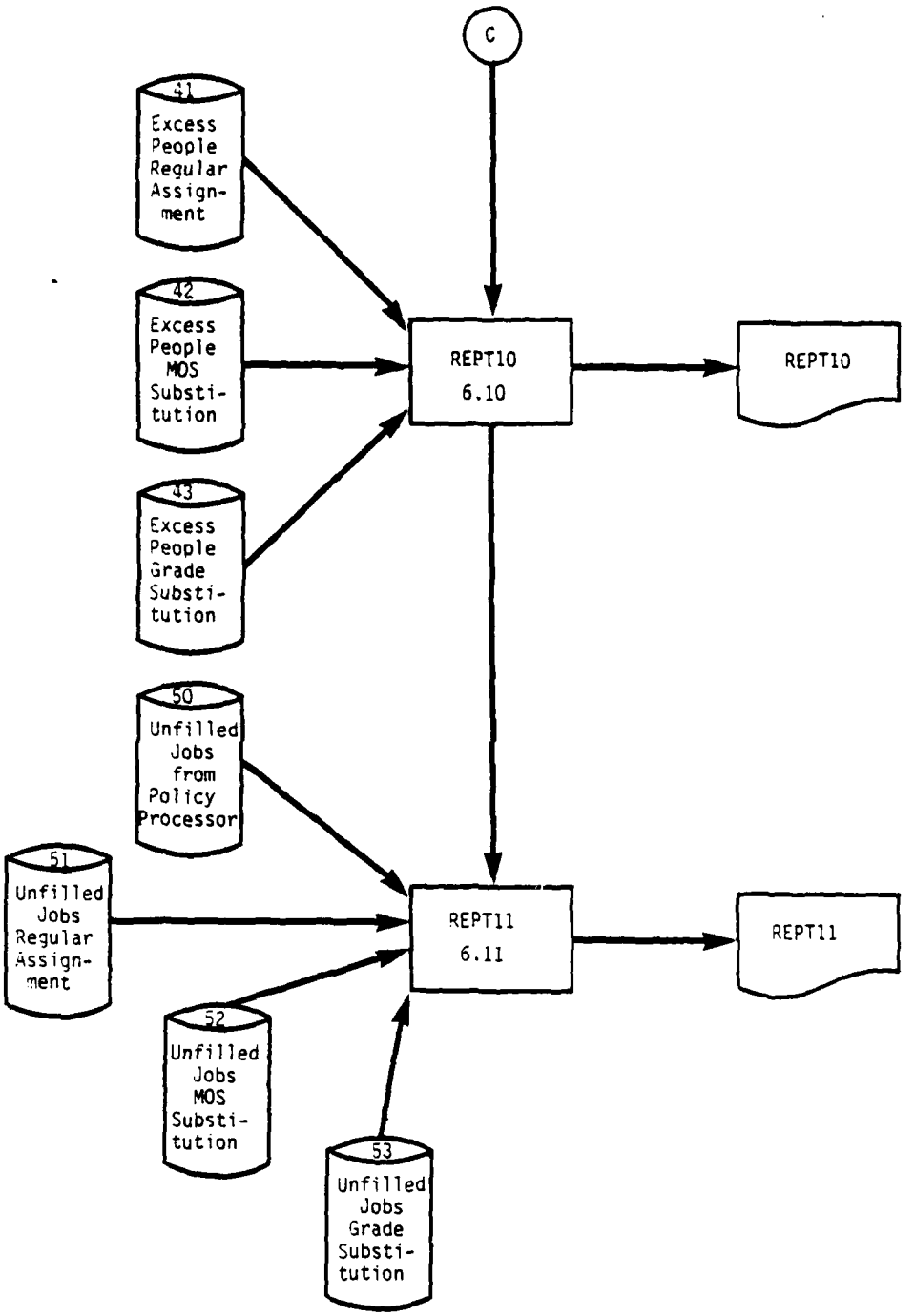


Figure 2-13. Information Flow Through the Report Processor (page 4 of 4 pages)

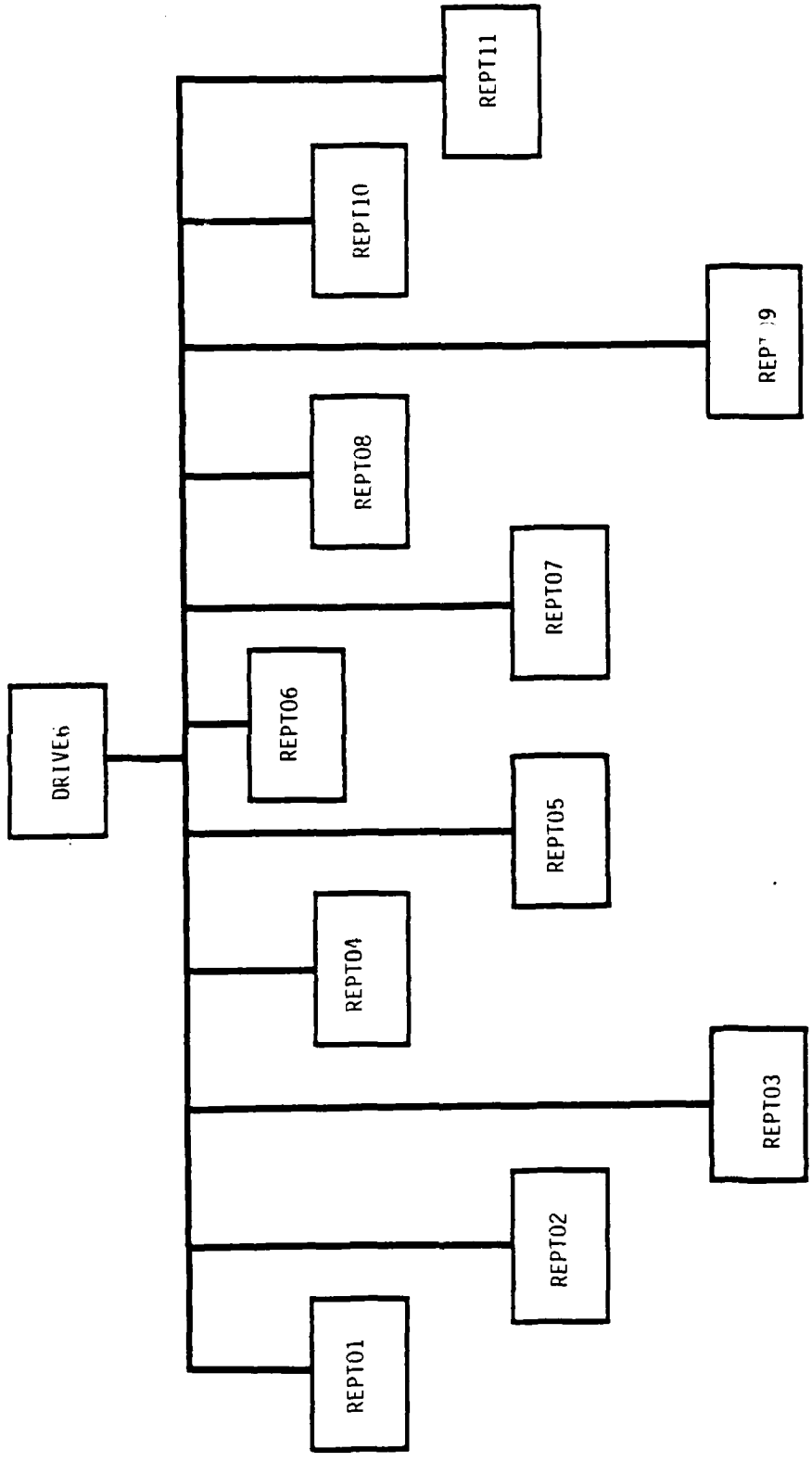


Figure 2-14. Program Unit Hierarchy-Report Processor

2.10 PROGRAM DESCRIPTION UTILITIES. The programs described in this section are the PRIM utility subroutines. They are considered utility routines because they perform their function for more than one processor. Most of the utility routines would be useful in other programs. In fact, many of these were originally programed for use in different models and have been included in PRIM because of their utility.

Paragraph	Program	Function
2.10.1	APRTCA	Print heading on each page of alternate print file
2.10.2	F2FRT$	Temporarily modify operating system to allow files numbered to 99
2.10.3	GETDAY	Get the run date and convert to DD MMM YY
2.10.4	GETWCT	Get wall clock time or elapsed time
2.10.5	ICHINT	Convert character to integer
2.10.6	ICOMP	Compare two alphanumeric variables
2.10.7	RCHFLT	Convert character to real
2.10.8	TITLE	Print title page
2.10.9	WRTERR	Write an error message to file 13

2.10.1 PROGRAM DESCRIPTION

a. Identification

Utility subroutine APRTCA (file number, heading)

b. Function

This subroutine creates a heading on an alternate print file such as would have been created as a result of @HDG. In PRIM, APRTCA is used for the alternate print file for error and warning messages since HDG cannot be used.

c. Format Provided

Relocatable

d. Arguments

File number	Integer	The number of the file that has been defined as an alternate print file
Heading	Alphanumeric	"H,U,, characters actually printed" The first six characters are specifically "H,U,, followed by the line you want printed as the heading followed by a closing quote (")

e. Arguments changed by this program

None

f. Example

```
Call APRTCA(13,"H,U,,P R I M  E R R O R S")
```

2.10.2 PROGRAM DESCRIPTION

a. Identification

Utility program F2FRT$

b. Function

This program temporarily, for the length of the run only, modifies the computer operating system to allow ASCII FORTRAN to read and write file numbers from zero through 99.

c. Format Provided

Symbolic Assembly Language Program

d. Accessed by: (1) Compile using the @MASM statement, and (2) In the mapping element, between the @MAP and END statements, include the following image:

```
IN        PFPRIM-REL.F2FRT
```

e. Code

```
@MASM, IS          PFPRIM-SYMB.F2FRT, PFPRIM-REL.F2FRT
 F$FRT             99
 PR                 6
 PU                 1
 CR                 5
 RR                 0
 END
@EOF
```

2.10.3 PROGRAM DESCRIPTION

a. Identification

Utility subroutine GETDAY (RUNDAY)

b. Function

This subroutine returns the run date in the form DD MMM YY when provided the date in the form YYMMDD.

c. Format Provided

Symbolic

d. Arguments

RUNDAY	CHARACTER*9

e. Arguments changed by this program

RUNDAY

f. Example

Input RUNDAY = 840817

Output RUNDAY = 17 Aug 84

2.10.4 PROGRAM DESCRIPTION

a. Identification

Utility subroutine GETWCT (ISH, ISM, ISS, EH, EM, ES)

b. Function

When ISH, ISM, and ISS all equal zero on input, wall clock time is returned in EH, EM, and ES.

When ISH, ISM, and ISS are not zero, elapsed time since the time represented by ISH, ISM, and ISS is returned in EH, EM, and ES.

c. Format Provided

Symbolic

d. Arguments

ISH	Integer	The hour of the start time for this computation
ISM	Integer	The minute of the start time for this computation
ISS	Integer	The second of the start time for this computation
EH	Integer	The elapsed time number of hours
EM	Integer	The elapsed time number of minutes
ES	Integer	The elapsed time number of seconds

e. Arguments changed by this program

EH, EM, and ES

2.10.5 PROGRAM DESCRIPTION

a. Identification

Utility function ICHINT (ALPHA, BEGIN, LENGTH, FLAG)

b. Function

The utility converts a character variable to an integer.

c. Format Provided

Symbolic

d. Arguments

ALPHA	Character	The alphanumeric representation of the integer
BEGIN	Integer	The number of the beginning character for conversion
LENGTH	Integer	The number of characters, including BEGIN, to convert
FLAG	Integer	A flag which indicates whether conversion concluded normally FLAG=0, Conversion O.K. FLAG≠0, Conversion not O.K., usually because invalid characters found

e. Arguments changed by this program

FLAG

f. Example

```
CHARACTER*4 ALPHA

INTEGER FLOW

FLOW = ICHINT(ALPHA, 1, 4, IFLAG)

IF(IFLAG .NE. 0) PRINT *,'ERROR IN CONVERSION OF', ALPHA
```

2.10.6 PROGRAM DESCRIPTION

a. Identification

Utility function ICOMP2 (ALPHA1, BEGIN1, ALPHA2, BEGIN2, LENGTH)

b. Function

The utility compares two alphanumeric variables. Value returned is negative if the first variable is smallest, zero if they are equal, positive if the first is the largest.

c. Format Provided

Symbolic

d. Arguments

ALPHA1	Alphanumeric	First variable
BEGIN1	Integer	Character number within ALPHA1 to start comparison
ALPHA2	Alphanumeric	Second variable
BEGIN2	Integer	Character number within ALPHA2 to start comparison
LENGTH	Integer	Number of characters to compare

e. Arguments changed by this program

None

f. Example

```
CHARACTER*6 ALPHA1, ALPHA2

IVAL = ICOMP2 (ALPHA1, 1, ALPH2, 1, 3)

IF(IVAL   .GE. 0)
   THEN do computations desired when the first three characters of
      ALPHA2 are greater than or equal to the first three characters
         of ALPHA1.

   ELSE, do computations desired when the first three characters of
      ALPHA2 are less than the first three characters of ALPHA1.

ENDIF
```

2.10.7 PROGRAM DESCRIPTION

a. Identification

Utility function RCHFLT (ALPHA, BEGIN, LENGTH, FLAG)

b. Function

The utility converts a character variable to a real number (floating point).

c. Format Provided

Symbolic

d. Arguments

ALPHA	Character	The alphanumeric representation of the real variable
BEGIN	Integer	The number of the beginning character for conversion
LENGTH	Integer	The number of characters, including BEGIN, to convert
FLAG	Integer	A flag which indicates whether conversion concluded O.K. FLAG=0 = O.K. FLAG≠0 = Not O.K.

e. Arguments changed by this program

FLAG

f. Example

```
CHARACTER*9 ALPHA

REAL NUMBER

NUMBER = RCHFLT(ALPHA, 1, 9, IFLAG)

IF(IFLAG .NE. 0) PRINT *, ' ERROR IN CONVERSION OF', ALPHA
```

2.10.8 PROGRAM DESCRIPTION

a. Identification

Utlity program @TITLE,option A1,A2,A3,A4,A5

b. Function

The program prints a page of large characters as specified in A1 through A5.

c. Format Provided

Executable (absolute) element

d. Options

D = Insert the run date in the location specified by two adjacent commas

e. Arguments

A1 through A5	Each variable will be a full line of print and may be words or any other combination of alphabetic characters, blank, dash (-), or numbers. The maximum length of each is 10 characters. When a blank should be inserted within the 10 characters, signify the place by inserting a dollar sign ($)

f. Example

```
@TITLE,D  THIS$RUN,IS$AS$OF,,PLF
```

will print in very large letters, if this is 30 June 1984:

```
THIS RUN
IS AS OF
84 06 30
  PLF
```

2.10.9 PROGRAM DESCRIPTION

a. Identification

Utility subroutine WRTERR (NERR)

b. Function

The subroutine writes an appropriate, PRIM-specific message for each error found in the Preprocessor, Policy Processor, or Report Processor.

c. Format Provided

Symbolic

d. Arguments

NERR - the error number. See list at Appendix E.

e. Arguments changed by this program

None

SECTION 3

ENVIRONMENT

3.1 EQUIPMENT ENVIRONMENT

The PRIM is resident on the UNIVAC 1100/82 Timesharing Multiprocessing System at the US Army Military Personnel Center (MILPERCEN). Access to the MILPERCEN system is provided through terminal devices located in the work areas of MILPERCEN analysts. The PRIM draws significantly on available system main and mass memory resources during execution. As a result, it will be necessary to anticipate and schedule the run workload into the overall run mix in order to maintain satisfactory throughput.

3.2 SUPPORT SOFTWARE

In the event that any changes to the model code are to be made, the PRIM requires the availability of the UNIVAC ASCII FORTRAN compiler, the associated subroutines, and the system collector. Each PRIM module that uses a file number larger than 30 makes a temporary modification to the file reference table. Assuming upward compatibility of the software system revisions, any level subsequent to the 10R1 level of the FORTRAN compiler should be compatible with the code. In addition, the PRIM requires the use of the UNIVAC utility processor, SORT, level 13, or a version compatible with level 13.

3.3 DATA BASES

The major input data bases are the available inventories and the job authorizations or requirements. Both of these data bases can be created from other data available at MILPERCEN. The available inventories should be developed as a summary by MOS and grade of every MOS or SC in the format shown for the Numbers-of-people file, file number 21. The job authorizations or requirements should be specified by MOS, grade, and UIC in the format shown for the Input MOS file, file number 8.

- The Preprocessor aggregates the MOS or job data and the unit data to the levels of aggregation specified by the user in the ISSUE file and the Parameter file. The ISSUE file specifies the desired aggregation for units; the Parameter file specifies the specialty aggregation levels. Files created by the Preprocessor are the UIC-data (three-digit) files and the MOS-data files at six-digit, three-digit, and ISSUE levels.

- The Policy Processor applies the user-supplied policies from the Policy file and the Value file to create the Job Assignment Value file and the first Unfilled Job file. Interim files produced are the Aggregated-ISSUE Policy file, ISSUE Policy file, MOS Policy file,

Combined ISSUE and MOS Policy file, Combined ISSUE and MOS Extra Job file, MOS Extra Job file, ISSUE Extra Job file, Aggregated-ISSUE Indicator file, and Edited Policy file.

- The Assignment Processor combines the Job Assignment Value file and the Numbers-of-people file to create the interim file, Network Input file. The final output files from this processor are the Unfilled Jobs from Regular Assignment file, the Excess People from Regular Assignment file, and the Assignments from Regular Assignment file. Interim files used by the network module are the Network Output file, three Network Scratch files, and the Network Input file. In addition, binary files are used for passing integer information from one module to another in the Assignment processor. The use of these files decreases the amount of memory required by the programs.

- The Substitute Assignment Processor uses the output files from the Assignment Processor (Excess People, Unfilled Jobs and Assignments) and the network module revises the same interim files as were used by the Assignment Processor. Output files are the Excess People from MOS Substitution file, Excess People from Grade Substitution file, Unfilled Jobs from MOS Substitution file, Unfilled Jobs from Grade Substitution file, Assignments from MOS Substitution file, and Assignments from Grade Substitution file.

- The Readiness Processor uses the assignment files, excess people files, and the unfilled jobs files created by both the Assignment Processor and the Substitute Assignment Processor plus the MOS-data ISSUE-level file to create the Readiness files--Aggregated Readiness Indicator file, Aggregate Percentage and C-Rating file, Grade Readiness file, and MOS Readiness file.

- The Report Processor uses the Readiness files to create the formated readiness reports using directions from the Report Request file.

3.3.1 General Characteristics. Numerous files are created by PRIM to pass information from one module or processor to another as described below:

- **Network Input.** This file is created by the assignment processors to feed the demand and resource data to the network module. See Table 3-1 for the file identification of mass storage file number 2.

- **Error Print.** This file is an alternate print file on mass storage. See Table 3-1 for the file identification of mass storage file number 13.

- **ISSUE.** This file specifies the aggregation levels for the units and provides the identification method and values for the aggregation. The ISSUE file is created by the user on mass storage. See Table 3-1 for the file identification of mass storage file number 14.

- **Value.** The user creates this mass storage file to specify the minimum percentage fills and the values of filling to that level. A separate input is required for each ISSUE specified on the ISSUE file. This file also contains the maximum percentage fill for each ISSUE and the value of each assignment between the minimum level and the maximum level. See Table 3-1 for the identification of mass storage file number 19.

- **Number-of-People.** This file contains the entire personnel inventory by grade and specialty code. It may contain enlisted, officer, or any combination of enlisted and officer data. The location of data for each pay grade on this mass storage file must be specified on the Parameter file. See Table 3-1 for the identification of mass storage file number 21.

- **UIC-data.** The data for this set of mass storage files are information about the units that will be used to aggregate them to ISSUE level. The source could be the header record from the Authorization Data Base. See Table 3-1 for the identification of these four mass storage files, file numbers 22, 23, 24, and 25.

- **MOS-data (six digits).** These files contain the required and authorized personnel by specialty and grade for every unit. See Table 3-1 for the identification of these four files, mass storage file numbers 8, 26, 27, and 28.

- **MOS-data (three-digit and ISSUE).** These files are aggregations of the six-digit MOS-data file, first at the three-digit UIC level, then at the ISSUE level. All pay grades are combined and similar records are combined so that the final MOS-data file contains one record for each unique combination of ISSUE and specialty. See Table 3-1 for the identification of these three files, mass storage file numbers 29, 30, and 31.

- **Extra Job Data.** When the Policy Processor is able to match a policy with an MOS-data record, the data needed by the Assignment Processor is created and written to the Job Assignment Value file. However, the MOS-data records that cannot be matched to a policy are considered extra or unused job data for that pass. This extra data is written to an extra job data file (in a format similar to the MOS-data files) for use as MOS-data on the next pass through the Policy Processor. The Value file is applied to the final extra job data file with all unmatched MOS-data written to an Unfilled Job file. See Table 3-1 for the identification of mass storage files numbered 36, 37, and 38.

- **Job Assignment Value (JAV).** Two files are used by the Policy Processor to store the minimum and maximum number of assignments and the value of making these assignments for each job. They are used alternately, with the output file on one pass becoming the input file for the next pass. See Table 3-1 for the identification of the mass storage JAV files numbered 39 and 40.

- **Excess People.** This set of files is created as outputs from the Assignment Processor (file 41) and the Substitute Assignment Processor (file 42 and 43), then as inputs to the Readiness Processor, and to Report. See Table 3-1 for identification of mass storage files numbered 41 through 44.

- **Unfilled Jobs.** This set of files is first created as outputs from the Policy Processor (file 50) and the Assignment Processor (file 51). These are then used as input to the Substitute Assignment Processor which creates files 52 and 53. These are input to the Readiness Processor and to the Report Processor. See Table 3-1 for identification for mass storages files numbered 50 through 54.

- **Assignments.** These files are first created from the Assignment Processor (file 61) and the Substitute Assignment Processor (files 62 and 63). These are then used as input to the Readiness Processor. See Table 3-1 for identification of mass storage files 61 through 63.

- **Readiness Indicators.** This set of files contains outputs from the Readiness Processor for use in formatted readiness reports. Files numbered 64 through 67 contain the detail data for each MOS and grade in every ISSUE. Files numbered 70 and 72 are data aggregated to ISSUE level; file 73 contains the data for each grade level in each ISSUE, aggregated across all MOS; and file 74 contains the data for each MOS in each ISSUE, aggregated across grades. See Table 3-1 for identification of mass storage files 64 through 67, 70, 72, 73, and 74.

- **Aggregated-ISSUE Indicator.** This is an interim file produced by the Policy Processor. It contains the Aggregated-ISSUE Policies converted to the flags that are used within the Policy Processor. See Table 3-1 for the identification of mass storage file 90.

- **Policy.** The unedited Policy file is created by the user and edited by the Policy Processor by comparing data from the Parameter and ISSUE files with the input Policy file. The edited Policy file is written to file 92. See Table 3-1 for identification of mass storage files 91 and 92.

- **Report Request.** This is the user's interface with the Report Processor. If this file is empty, no formatted readiness reports will be produced. See Table 3-1 for identification of mass storage file 93.

Table 3-1. File Identification

Logical unit number	Description	Mass storage name
2	Network Output File	PFNETOUT2
8	MOS File	PFINPUTMOS8
13	Error Print File	PFERROR13
14	ISSUE Definition File	PFISSUE14
15	Network Input File	PFNETIN15
17	Parameter File	PFPARAMET17
19	Value File	PFVALUE19
21	Number of People File	PFNUMBPEOP21
22	UIC-data (6-digit level) File	PFUIC6DIG22
23	UIC-data (3-digit level) File	PFUIC3DIG23
24	UIC-data (3-digit with ISSUE added) File	PFUICDATA24
25	UIC-data (3-digit with ISSUE added) File	PFUICDATA25
26	MOS-Enlisted File	PFMOS-ENL-26
27	MOS-Officer File	PFMOS-OFF-27
28	MOS-Warrant Officer File	PFMOS-WO-28
29	MOS-data (3-digit level) File	PFMOSDATA29
30	MOS-data (3-digit with ISSUE added) File	PFMOSDATA30
31	MOS-data (ISSUE level) File	PFMOSISSUE31
32	Aggregated ISSUE Policy File	PFAGISPO32
33	ISSUE Policy File	PFISSPOL33
34	MOS Policy File	PFMOSPOL34
35	Combined ISSUE and MOS Policy File	PFISMOPO35
36	Combined ISSUE and MOS Extra Job File	PFISMOEX36
37	MOS Extra Job File	PFMOSEXT37
38	ISSUE Extra Job File	PFISSEXT38
39	Job Assignment Value File (scratch)	PFJOBASVAL39
40	Job Assignment Value File (final)	PFJOBASVAL40
41	Excess People (from Regular Assignment) File	PFEXCPEOP41
42	Excess People (from MOS Substitution) File	PFEXPEOMOS42
43	Excess People (from Grade Substitution) File	PFEXPEOGRA43
44	Excess People (Scratch) File	PFEXPEOMOS44
50	Unfilled Jobs (from Policy Processor) File	PFUNFILLED50
51	Unfilled Jobs (from Regular Assignment) File	PFUNFILLED51
52	Unfilled Jobs (from MOS Substitution) File	PFUNFILLMO52
53	Unfilled Jobs (from Grade Substitution) File	PFUNFILLGR53
54	Unfilled Jobs (Scratch) File	PFUNFILLED54
61	Assignment (from Regular Assignment) File	PFASSIGNED61
62	Assignment (from MOS Substitution) File	PFASSIGNMO62
63	Assignment (from Grade Substitution) File	PFASSIGNGR63
64	Assignment (from Regular Assignment Plus Percentages) File	PFASSIGNED64
65	Assignment (from MOS Substitition Plus Percentages) File	PFASSIGNMO65
66	Assignment (from Grade Substition Plus Percentages) File	PFASSIGNGR66
67	Total Results including MOS Substitution File	PFASSIGNED67
70	Aggregated Readiness Indicator File	PFISSREADI70
72	Aggregated Percentage and C-Rating File	PFISSPERCR72
73	Grade Readiness and Percentage File	PFGRAREADI73
74	MOS Readiness, Percentage, and C-Rating File	PFMOSREADI74
85	Strength-by-grade Print File	PFGRASTREN85
86	Strength-by-MOS Print File	PFMOSSTREN86
90	Aggregated ISSUE Indicator File	PFAGISIND90
91	Policy File	PFPOLICY91
92	Edited Policy File	PFEDIPOL92
93	Report Request File	PFREPORT93

3.3.2 Organization and Detailed Description. The description of the layout of every mass storage file described in Section 3.3.1, General Characteristics, is provided in this section. The files are listed in the same general sequence as Section 3.3.1. The sequence is the same as the logical unit number used in the programs; the unit file number is provided with each description. Each file is described on a separate page.

NETWORK OUTPUT FILE
FILE NUMBER 2

Record Length: N/A
Storage Medium: Mass Storage
Source File: 15

First Use: Output from Assignment Processor, Subroutine SNET

Name	Description	Type
IDIFF	Resource (personnel) Category: points to MOS in (NAMES) array	Binary
IDIFF2	Activity: points to grade in (IACT) array	Binary
IDIFF3	Demand (job identifier): points to job in (NAMES) array	Binary
IDIFF4	Number of people assigned to job People assigned to super job = excess people Super people assigned to job = unfilled job	Binary

INPUT MOS FILE
FILE NUMBER 8

Record Length: 42 characters
Storage Medium: Mass Storage
Source File: AUDB

First Use: Input to Preprocessor, Module 1.1.2.1, Separate MOS data

Name	Description	Position	Format
--	Filler	1	A1
UIC	6-Digit unit identification code	2-7	A6
--	Filler	8-14	A7
GRADE	Pay grade	15-16	A2
CMF	Career Management Field	17-18	A2
MOS	Military occupational specialty code	19-27	A9
--	Filler	28-31	A4
IDENT	Person identity code	32	A1
REQSTR	Required (structure) strength	33-35	I3
AUSTR	Authorized strength	36-38	I3
--	Filler	39-42	I4

ERROR PRINT FILE
FILE 13

Record Length: 132 characters
Storage Medium: Mass Storage
Source File: N/A

Name	Description	Position	Format
*	Error description	1-132	A-132

*Each record is variable in length and contains error number and description of error found.

ISSUE DEFINITION FILE
FILE NUMBER 14

Record Length: 62 characters
Storage Medium: Mass Storage
Source File: PFPRIM-FILES

First Use: Input to Preprocessor, Module 1.2, Set ISSUE

Name	Description	Position	Format
--	ISSUE=	1-6	A6
ISSUE	ISSUE code	7-10	A4
--	ID1=	11-14	A4
ID1	First identification Method Valid entries are: TPSN UIC3 UIC6 ASGMT ASGMT1	15-20	A6
--	= sign	21	A1
IDIVAL	First identification method value	22-27	A6
--	ID2=	28-31	A4
ID2	Second identification method. Valid entries are: LOCCO STACO ASGMT	32-37	A6
--	= sign	38	A1
ID2VAL	Value of second identification method	39-44	A6
--	NAME=	45-49	A5
ISSNAM	Name of ISSUE for report purposes	50-62	A12

NETWORK INPUT FILE
FILE 15

Source File: 21, 40

Created by: PRETEST (3.1), PREMOS (4.1.1), PREGRD (4.2.1)

Note: This file contains six sets of records. Within each set are one to four record types. Records must be input in the following order.

Set one: Demand -- Type 1, 2, 3, and 4

Set two: Demand Sink (Super Jobs) -- Types 1, 2, and 3

Set three: End Demand -- Type 1

Set four: Resource -- Type 1, 2, 3, 4, and 5

Set five: Resource Sink (Super Soldiers) -- Type 1, 2, 3, 4, and 5

Set six: End Resource -- Type 1

MNEMONIC	DESCRIPTION	TYPE	EXAMPLE
	//////////////////////////		
	SET ONE: DEMAND RECORDS		
	//////////////////////////		
	--DEMAND RECORD TYPE ONE		
	One per Job Assignment Value Record for an MOS		
DEMAN	Demand node name (job identifier) Created by adding two to four alpha-numeric characters to the ISSUE code	A8	F01-QEA2
NSUPST	Number of supersets that include this DEMAN	I2	
	--DEMAND RECORD TYPE TWO		
	One for each demand record type one		

NETWORK INPUT FILE
FILE 15, cont.

MNEMONIC	DESCRIPTION	TYPE	EXAMPLE
SUBSET	Superset names: US ARMY and MOS	A8	11A1
	These names are used to link the job (DEMAN) to the available inventory i.e., if the job and the inventory have the same MOS as a super set name (see inventory resource section, record type 2 below), then a link is created		
	--DEMAND RECORD TYPE THREE		
	One for each demand record type one		
AGGMIN	Minimum number of people that must be assigned to the DEMAN	I10	
AGGMAX	Maximum number of people that may be assigned to the DEMAN	I10	
	--DEMAND RECORD TYPE FOUR		
	One to four for each demand record type one		
GRAMIN	Minimum number of people that must be assigned to this DEMAN; one minimum for each grade	I10	
GRAMAX	Maximum number of people that may be assigned to this DEMAN; one maximum for each grade	I10	
	NOTE: These numbers are input as pairs. One GRAMIN and one GRAMAX for the first valid grade, then one GRAMIN and one GRAMAX for the next valid grade, etc., until the number of pairs is equal to NGRADE in the parameter file. Four sets fit on each record		

NETWORK INPUT FILE
FILE 15, cont.

MNEMONIC	DESCRIPTION	TYPE	EXAMPLE
	////////////////////		
	SET TWO: DEMAND SINK		
	////////////////////		
	--SINK RECORD TYPE ONE		
	One per valid grade level		
'SINK'	Identifies beginning of sink area ('SINK' is followed by 4 blanks)	A8	SINK
NSUPST	Number of supersets to which the sink belongs	I3	
'SJ-xxx'	Name of the super job. For regular assignment, the name is created by concatinating SJ- with the grade code, followed by two blanks. For MOS substitution, the blank is replaced with '-A' or '-F'	A8	SJ-ES SJ-EE5-A
	--SINK RECORD TYPE TWO		
	One per superset record type one		
SUPSET	Superset names: grade and MOS The number of names listed must be equal to NSUPST	A8	E6 100A
	--SINK RECORD TYPE THREE		
	One per MOS		
MOS	MOS of the super job (will be the same as the MOS of the jobs listed above	A8	11B 54
NUMPEO	The number of people that may be assigned to this super job. The 'PRE---' programs set this to the number of people available in the inventory for this grade level and MOS	I9	

NETWORK INPUT FILE
FILE 15, cont.

MNEMONIC	DESCRIPTION	TYPE	EXAMPLE
VALUE	The value associated with assigning personnel to this imaginary job. The 'PRE---' programs set this value to -10	I3	-10
VALUE	Same as above	I3	-10
NUMPEO	Same as above	I9	

////////////////////,/////////

SET THREE: END DEMAND RECORDS

////////////////////////////

--END DEMAND RECORD

One per MOS

'END DEM'	Signal that the end of the demand section has been reached	A8	END DEM

//////////////////

SET FOUR: RESOURCE

//////////////////

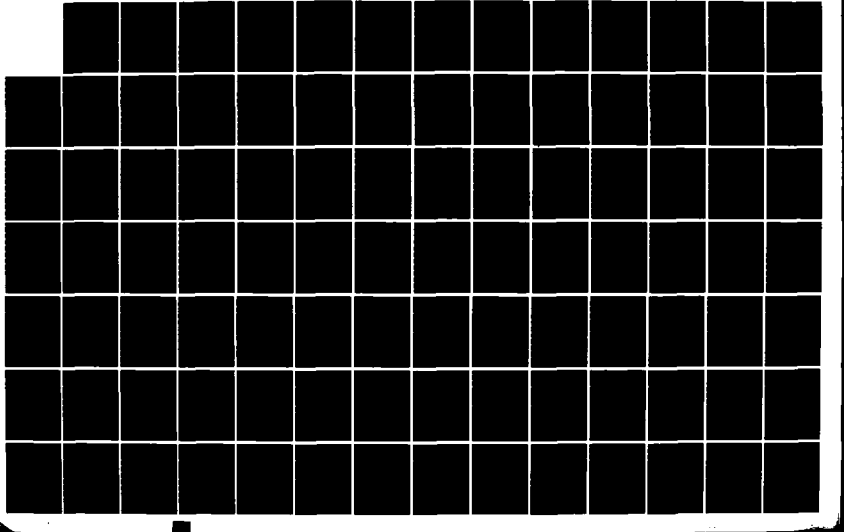
--RESOURCE RECORD TYPE ONE

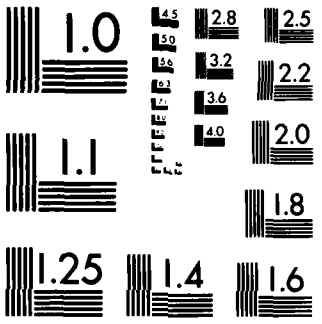
One per MOS

MOS	Name of resource (for grade substitution; '-GRADE' is concatenated to MOS)	A8	76Y 11B-GRADE
NSUPST	Number of supersets to which this resource belongs; usually one	I3	
NJOBS	Number of jobs (ISSUEs) to which this MOS can be assigned	I10	

AD-A149 505 PERSONNEL READINESS INDICATOR MODEL (PRIM) DOCUMENTATION PROGRAM MAINTENANCE MANUAL(U) ARMY CONCEPTS ANALYSIS AGENCY BETHESDA MD 3/4

UNCLASSIFIED S J VAN NOSTRAND ET AL. NOV 84 CAA-D-84-2 F/G 9/2 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

NETWORK INPUT FILE
FILE 15, cont.

MNEMONIC	DESCRIPTION	TYPE	EXAMPLE
	--RESOURCE RECORD TYPE TWO		
SUPSET	Name of Superset (normally MOS) for linking MOS to jobs	A8	11B 54 76Y
	--RESOURCE RECORD TYPE THREE		
NUMBEO	Minimum number of people in the aggregate that are available for assignment	I10	
NUMPEO	Maximum number of people in the aggregate that are available for assignment; normally equal to NUMBEO	I10	
	--RESOURCE RECORD TYPE FOUR		
NBY6	Minimum number of people in the aggregate that are available for assignment in the grade	I10	
NBY6	Maximum number of people in the aggregate that are available for assignment in the grade	I10	
	--RESOURCE RECORD TYPE FIVE		
	One for each possible real job to which this MOS can be assigned		
DEMAND	Job name	A8	AO1-AQC2 FO1-QEA2
GRAVAL	Value associated with job	I9	
GRAMAX	Maximum number of people by grade that can be assigned to this job	I9	
	NOTE: These numbers are input as pairs similar to demand record type four		

MNEMONIC	DESCRIPTION	TYPE	EXAMPLE
//////////////////////			
SET FIVE: SUPER SOLDIERS			
//////////////////////			
	--SUPER SOLDIERS RECORD TYPE ONE		
	One set per MOS		
'SP'-MOS	Name of super soldiers. 'SP' is concatenated to the front of MOS	A8	SP-11B
NSUPST	Number of supersets to which this resource belongs	I3	1
NJOBS	Number of jobs requiring this MOS	I10	
	--SUPER SOLDIERS RECORD TYPE TWO		
SUPSET	Name of superset (MOS)	A8	11B 76Y
	--SUPER SOLDIERS RECORD TYPE THREE		
MINPEO	Minimum number of super soldiers that must be assigned (normally zero)	I10	
MAXPEO	Maximum number of super soldiers that may be assigned. The 'PRE---' programs set this value equal to the total number of jobs in DEMAN	I10	
	--SUPER SOLDIERS RECORD TYPE FOUR		
	One set for each grade		
MINPEO	Minimum number of super soldiers by grade (zero)	I10	

NETWORK INPUT
FILE 15, cont.

MNEMONIC	DESCRIPTION	TYPE	EXAMPLE
MAXPEO	Maximum number of super soldiers by grade. Usually equal to GRAMAX	I10	
	--SUPER SOLDIERS RECORD TYPE FIVE		
	One for each possible job (DEMAN)		
DEMAND	Job name	A8	F01-QEA2
GRAVAL	Value associated with filling this job with super soldiers	I9	-10
MAXNEO	Maximum number of super soldiers by grade (GRAMAX)	I9	
//////////////////////			
SET SIX: END RESOURCE			
//////////////////////			
	--END RESOURCE RECORD TYPE ONE		
'END RES'	Signal that the end of the resource section has been reached	A7	END RES

CAA-D-84-2

PARAMETER FILE
FILE NUMBER 17

Record Length: 66 characters
Storage Medium: Mass Storage
Use: Input to all modules of all processors
Source File: PFPRIM-FILES

This file contains three record types.

Name	Description	Position	Format
	(RECORD TYPE 1 - REQUIRED)		
	Mnemonic searched for:	1-6	A6
	ASOF MOSSUB MOSJOB C3VALA NGRADE SCDSUB GRDPEO C1VALM NCHENL ENLSUB GRDJOB C2VALM NCHOFF OFFSYB NONAVA C3VALM NCHWOF TYPSTR C1VALA NCCHAR MOSPEO C2VALA		
--	= sign	7	A1
ALPHA	Alphanumeric of 1 to 6 characters, left justified	8-13	A6
--	Filler	14	A1
DESCRI	Description	15-80	A66
	(RECORD TYPE 2 - REQUIRED - GRADE LOCATIONS)		
--	Filler	1-4	A4
GRADE	Two-character grade designation, the first of these should be the second after NGRADE	5-6	A2
--	= sign	7	A1
IGRADE	Relative location of grade data. If pay grade will not be specifically in the input files, this location should be set to zero	8-9	I2
--	Filler	10-14	A5
DESCRI	Description	15-80	A66
	(RECORD TYPE 3 - OPTIONAL - SUBSTITUTION INFORMATION)		
--	Filler	1	A1
FROM	'GRADE FROM' or ' MOS FROM' or ' SC FROM'	2-11	A10
--	= sign	12	A1
GRDFRM or MOSFRM	Grade or specialty the substitution is from	13-21	A9
TO	'GRADE TO' or ' MOS TO' or ' SC TO'	22-29	A8
--	= sign	30	A1
GRDTO or MOSTO	Grade or MOS the substitution is to	31-39	A9

VALUE FILE
FILE NUMBER 19

Record Length: 62 characters
Storage Medium: Mass Storage
Source File: PFPRIM-FILES

First Use: Input to Policy Processor, Module 2.3, Set Base Values

Mnemonic	Description	Position	Format
ISSALP	'ISSUE='	1-6	A6
ISSUE	ISSUE code	7-10	A4
	Filler (blank)	11	A1
--	'MINIMUM='	12-19	A8
PERMIN	Minimum percentage fill stated as decimal rate	20-24	F5.3
--	Filler (blank)	25	A1
--	'MINVAL='	26-32	A7
VALUE	Value of filling each job up to minimum percentage	33-35	I3
--	Filler (blank)	36-38	A3
--	'MAXIMUM='	39-46	A8
PERMAX	Maximum percentage fill	47-51	F5.3
--	Filler (blank)	52	A1
--	'MAXVAL='	53-59	A7
VALUE2	Value of filling each job above the minimum percentage (PERMIN) up to the maximum (PERMAX)	60-62	I3

NUMBER OF PEOPLE FILE
FILE NUMBER 21

Record Length: 143 characters
Storage Medium: Mass Storage
Source File: N/A

First Use: Input to Assignment Processor, Subroutine PRENET

Name	Description	Position	Format
ASDAT	As of date of data	1-6	I6
CMF	Career management field	7-8	A2
MOS	MOS if enlisted data; SC if officer. As of Sep 83, only data for first 3 characters is available; space is provided for all 9	9-17	A9
NE1	Number of people available in each enlisted grade of E1 through E7	18-24	I7
NE2		25-31	I7
NE3		32-38	I7
NE4		39-45	I7
NE5		46-52	I7
NE6		53-59	I7
NE7		60-66	I7
NE8		67-73	I7
NE9		74-80	I7
NENLIS	Total number of enlisted personnel E1 through E9	81-87	I7
NWO	Total number of warrant officers	88-94	I7
NO1	Number of officers in grade O1 through O6	95-101	I7
NO2		102-108	I7
NO3		109-115	I7
NO4		116-122	I7
NO5		123-129	I7
NO6		130-136	I7
NOFF	Total number of officers in grades O1 through O6	137-143	I7

UIC-DATA (6-DIGIT LEVEL) FILE
FILE NUMBER 22

Record Length: 97 characters
Storage Medium: Mass Storage
Source File: AUDB

First Use: Input to Preprocessor Module 1.1.1, Roll UIC data to 3-digit UIC level

Name	Description	Position	Format
--	Filler	1	A1
UIC	UIC code	2-7	A6
--	Filler	8-11	A4
COMPO	Component	12	A1
UNTDS	Unit description	13-33	A21
TYPCO	Unit type	34	A1
ASGMT	MACOM or organization	35-36	A2
TPSN	Troop program sequence number	37-41	A5
STACO	Station code	42-46	A5
STNNM	Station name	47-55	A9
LOCCO	Location code	56-58	A3
STSOFF	Structure strength officers	59-62	I4
STSWOF	Structure strength warrant officers	63-66	I4
STSENL	Structure strength enlisted	67-70	I4
STSAGG	Structure strength aggregate military	71-74	I4
STSCIV	Structure strength civilian	75-78	I4
AUSOFF	Authorized strength officers	79-82	I4
AUSWOF	Authorized strength warrant officers	83-86	I4
AUSENL	Authorized strength enlisted	87-90	I4
AUSAGG	Authorized strength aggregate military	91-94	I4
AUSCIV	Authorized strength civilian	95-98	I4

UIC-DATA (3-DIGIT LEVEL) FILE
FILE NUMBER 23

Record Length: 97 characters
Storage Medium: Mass Storage
Source File: 22

First Use: Output from Preprocessor, Module 1.1.1, Roll UIC data to 3-digit UIC level

Name	Description	Position	Format
UIC	UIC code	1-6	A6
--	Filler	7-10	A4
COMPO	Component	11	A1
UNTDS	Unit description	12-32	A21
TYPCO	Unit type	33	A1
ASGMT	MACOM or organization	34-35	A2
TPSN	Troop program sequence number	36-40	A5
STACO	Station code	41-45	A5
STNNM	Station name	46-54	A9
LOCCO	Location code	55-57	A3
STSOFF	Structure strength officers	58-62	I5
STSWOF	Structure strength warrant officers	63-67	I5
STSENL	Structure strength enlisted	68-72	I5
STSAGG	Structure strength aggregate military	73-77	I5
STSCIV	Structure strength civilian (set to zero)	78-82	I5
AUSOFF	Authorized strength officers	83-87	I5
AUSWOF	Authorized strength warrant officers	88-92	I5
AUSENL	Authorized strength enlisted	93-97	I5
AUSAGG	Authorized strength aggregate military	98-102	I5
AUSCIV	Authorized strength civilian (set to zero)	103-107	I5

UIC-DATA (3-DIGIT WITH ISSUE ADDED) FILE
FILE NUMBER 24

Record Length: 97 characters
Storage Medium: Mass Storage
Source File: 23, 25

First Use: Output from Preprocessor, Module 1.2, Set ISSUE

Name	Description	Position	Format
UIC	UIC code	1-6	A6
ISSUE	ISSUE code from ISSUE file	7-10	A4
COMPO	Component	11	A1
UNTDS	Unit description	12-32	A21
TYPCO	Unit type	33	A1
ASGMT	MACOM or organization	34-35	A2
TPSN	Troop program sequence number	36-40	A5
STACO	Station code	41-45	A5
STNNM	Station name	46-54	A9
LOCCO	Location code	55-57	A3
STSOFF	Structure strength officers	58-62	I5
STSWOF	Structure strength warrant officers	63-67	I5
STSENL	Structure strength enlisted	68-72	I5
STSAGG	Structure strength aggregate military	73-77	I5
STSCIV	Structure strength civilian (set to zero)	78-82	I5
AUSOFF	Authorized strength officers	83-87	I5
AUSWOF	Authorized strength warrant officers	88-92	I5
AUSENL	Authorized strength enlisted	93-97	I5
AUSAGG	Authorized strength aggregate military	98-102	I5
AUSCIV	Authorized strength civilian (set to zero)	103-107	I5

UIC-DATA (3-DIGIT WITH ISSUE ADDED) FILE
FILE NUMBER 25

Record Length: 97 characters
Storage Medium: Mass Storage
Source File: 24

First Use: Output from Preprocessor, Module 1.2, Set ISSUE

Name	Description	Position	Format
UIC	UIC code	1-6	A6
ISSUE	ISSUE code from ISSUE file	7-10	A4
COMPO	Component	11	A1
UNTDS	Unit description	12-32	A21
TYPCO	Unit type	33	A1
ASGMT	MACOM or organization	34-35	A2
TPSN	Troop program sequence number	36-40	A5
STACO	Station code	41-45	A5
STNNM	Station name	46-54	A9
LOCCO	Location code	55-57	A3
STSOFF	Structure strength officers	58-62	I5
STSWOF	Structure strength warrant officers	63-67	I5
STSENL	Structure strength enlisted	68-72	I5
STSAGG	Structure strength aggregate military	73-77	I5
STSCIV	Structure strength civilian (set to zero)	78-82	I5
AUSOFF	Authorized strength officers	83-87	I5
AUSWOF	Authorized strength warrant officers	88-92	I5
AUSENL	Authorized strength enlisted	93-97	I5
AUSAGG	Authorized strength aggregate military	98-102	I5
AUSCIV	Authorized strength civilian (set to zero)	103-107	I5

MOS-ENLISTED FILE
FILE NUMBER 26

Record Length: 42 characters
Storage Medium: Mass Storage
Source File: 8

First Use: Output from Preprocessor, Module 1.1.2.1, Separate MOS Data

Name	Description	Position	Format
--	Filler	1	A1
UIC	6-Digit unit identification code	2-7	A6
--	Filler	8-14	A7
GRADE	Pay grade	15-16	A2
CMF	Career Management Field	17-18	A2
MOS	Military occupational specialty code	19-27	A9
--	Filler	28-31	A4
IDENT	Person identity code	32	A1
REQSTR	Required (structure) strength	33-35	I3
AUTSTR	Authorized strength	36-38	I3
--	Filler	39-42	A4

MOS-OFFICER FILE FILE NUMBER 27

Record Length: 42 characters
Storage Medium: Mass Storage Source File: 8

First Use: Output from Preprocessor, Module 1.1.2.1, Separate MOS Data

Name	Description	Position	Format
--	Filler	1	A1
UIC	6-Digit unit identification code	2-7	A6
--	Filler	8-14	A7
GRADE	Pay grade	15-16	A2
CMF	Career Management Field	17-18	A2
MOS	Specialty code	19-27	A9
--	Filler	28-31	A4
IDENT	Person identity code	32	A1
REQSTR	Required (structure) strength	33-35	I3
AUTSTR	Authorized strength	36-38	I3
--	Filler	39-42	A4

MOS-WARRANT OFFICER FILE FILE NUMBER 28

Record Length: 42 characters
Storage Medium: Mass Storage Source File: 8

First Use: Output from Preprocessor, Module 1.1.2.1, Separate MOS Data

Name	Description	Position	Format
--	Filler	1	A1
UIC	6-Digit unit identification code	2-7	A6
--	Filler	8-14	A7
GRADE	Pay grade	15-16	A2
CMF	Career Management Field	17-18	A2
MOS	Specialty code	19-27	A9
--	Filler	28-31	A4
IDENT	Person identity code	32	A1
REQSTR	Required (structure) strength	33-35	I3
AUTSTR	Authorized strength	36-38	I3
--	Filler	39-42	A4

MOS-DATA (AGGREGATED TO 3-DIGIT UIC LEVEL) FILE
FILE NUMBER 29

Record Length: 325 characters
Storage Medium: Mass Storage
Source File: 26, 27, 28

First Use: Output from Preprocessor, Module 1.1.2.2, Roll MOS to 3-digit UIC level

Name	Description	Position	Format
--	Filler	1	A1
UIC26	Characters 2-6 of UIC (5 and 6 are blank)	2-6	A5
--	Filler	7-10	A4
CMF	Career management field	11-12	A2
MOS	MOS	13-21	A9
GRDGRP	Grade group; repeats 16 items GRDGRP contains:	22-325	(see below)
GRADE	E1-E9, W0, O1-O6	(1-2)	A2
ASI	Additional skill identifier	(3-4)	A2
LIC	Language identification code	(5-6)	A2
IDENT	Person identity (M/F/O & grade)	(7)	A1
REQSTR	Required (Structure) strength	(8-11)	I4
AUTSTR	Authorized strength	(12-15)	I4
--	Filler	(16-19)	I4

MOS-DATA (3-DIGIT WITH ISSUE CODE ADDED) FILE
FILE NUMBER 30

Record Length: 325 characters
Storage Medium: Mass Storage
Source File: 29

First Use: Output from Preprocessor, Module 1.2, Set ISSUE

Name	Description	Position	Format
--	Filler	1	A1
UIC24	Characters 2-4 of UIC (5 and 6 are blank)	2-6	A5
ISSUE	ISSUE code	7-10	A4
CMF	Career management field	11-12	A2
MOS	MOS	13-21	A9
GRDGRP	Grade group; repeats 16 items GRDGRP contains:	22-325	(see below)
GRADE	E1-E9, W0, O1-O6	(1-2)	A2
ASI	Additional skill identifier	(3-4)	A2
LIC	Language identification code	(5-6)	A2
IDENT	Person identity (M/F/O & grade)	(7)	A1
REQSTR	Required (Structure) strength	(8-11)	I4
AUTSTR	Authorized strength	(12-15)	I4
--	Filler	(16-19)	I4

MOS-DATA (AGGREGATED TO ISSUE LEVEL) FILE
FILE NUMBER 31

Record Length: 325 characters
Storage Medium: Mass Storage
Source File: 30

First Use: Output from Preprocessor, Module 1.3, Aggregate MOS data to ISSUE level

Name	Description	Position	Format
--	Filler	1-6	A6
ISSUE	ISSUE code	7-10	A4
CMF	Career management field	11-12	A2
MOS	Specialty code	13-21	A9
GRDGRP	Grade group; repeats 16 items GRDGRP contains:	22-389	(see below)
GRADE	E1-E9, W0, O1-O6	(1-2)	A2
ASI	Additional skill identifier	(3-4)	A2
LIC	Language identification code	(5-6)	A2
IDENT	Person identity (M/F/O & grade)	(7)	A1
REQSTR	Required (Structure) strength	(8-13)	I6
AUTSTR	Authorized strength	(14-19)	I6
--	Filler	(20-23)	A4

AGGREGATED ISSUE POLICY FILE
FILE NUMBER 32

Record Length: 80
Storage Medium: Mass Storage
Source File: 92

First Use: Output from Policy Processor, Module 2.2A

Name	Description	Position	Format
POLTYP	Policy type='ISSUE'	1-5	A5
--	= sign	6	A1
ISSUE	ISSUE code of '00' level only	7-10	A4
--	'LO='	11-13	A3
LOGRAD	Lowest grade to which policy applies	14-15	A2
--	'HI='	16-18	A3
HIGRAD	Highest grade to which policy applies	19-20	A2
--	'AG='	21-23	A3
AGG	'YES' if policy is to aggregate 'NO ' if policy is not to aggregate	24-26	A3
--	'VALUE='	27-32	A6
VALUE	Fill value	33-35	I3
--	Filler	36-38	A3
--	'MIN='	39-42	A4
PERMIN	Minimum percent fill	43-47	F5.3
--	'MAX='	48-51	A4
PERMAX	Maximum percent fill	52-56	F5.3
--	'MOS='	57-60	A4
--	Blank	61-69	A9
--	Filler	70-80	A11

CAA-D-84-2

ISSUE POLICY FILE
FILE NUMBER 33

Record Length: 80
Storage Medium: Mass Storage
Source File: 92

First Use: Output from Policy Processor, Module 2.2A

Name	Description	Position	Format
POLTYP	Policy type='ISSUE'	1-5	A5
--	= sign	6	A1
ISSUE	ISSUE code	7-10	A4
--	'LO='	11-13	A3
LOGRAD	Lowest grade to which policy applies	14-15	A2
--	'HI='	16-18	A3
HIGRAD	Highest grade to which policy applies	19-20	A2
--	'AG='	21-23	A3
AGG	'YES' if policy is to aggregate 'NO ' if policy is not to aggregate	24-26	A3
--	'VALUE='	27-32	A6
VALUE	Fill value	33-35	I3
--	Filler	36-38	A3
--	'MIN='	39-42	A4
PERMIN	Minimum percent fill	43-47	F5.3
--	'MAX='	48-51	A4
PERMAX	Maximum percent fill	52-56	F5.3
--	'MOS='	57-60	A4
--	Blank	61-69	A9
--	Filler	70-80	A11

MOS POLICY FILE
FILE NUMBER 34

Record Length: 80
Storage Medium: Mass Storage
Source File: 92

First Use: Output from Policy Processor, Module 2.2A

Name	Description	Position	Format
POLTYP	Policy type='MOSSC'	1-5	A4
--	= sign	6	A1
--	Blank	7-10	A4
--	'LO='	11-13	A3
LOGRAD	Lowest grade to which policy applies	14-15	A2
--	'HI='	16-18	A3
HIGRAD	Highest grade to which policy applies	19-20	A2
--	'AG='	21-23	A3
AGG	'YES' if policy is to aggregate 'NO ' if policy is not to aggregate	24-26	A3
--	'VALUE='	27-32	A6
VALUE	Fill value	33-35	I3
--	Filler	36-38	A3
--	'MIN='	39-42	A4
PERMIN	Minimum percent fill	43-47	F5.3
--	'MAX='	48-51	A4
PERMAX	Maximum percent fill	52-56	F5.3
--	'MOS='	57-60	A4
MOS	MOS or specialty code	61-69	A9
--	Filler	70-80	A11

CAA-D-84-2

COMBINED ISSUE & MOS POLICY FILE
FILE NUMBER 35

Record Length: 80
Storage Medium: Mass Storage
Source File: 92

First Use: Output from Policy Processor, Module 2.2A, Apply Policies
(Reread as input to same module)

Name	Description	Position	Format
POLTYP	Policy type='ISSUE'	1-5	A5
--	= sign	6	A1
ISSUE	ISSUE	7-10	A4
--	'LO='	11-13	A3
LOGRAD	Lowest grade to which policy applies	14-15	A2
--	'HI='	16-18	A3
HIGRAD	Highest grade to which policy applies	19-20	A2
--	'AG='	21-23	A3
AGG	'YES' if policy is to aggregate 'NO ' if policy is not to aggregate	24-26	A3
--	'VALUE='	27-32	A6
VALUE	Fill value	33-35	I3
--	Filler	36-38	A3
--	'MIN='	39-42	A4
PERMIN	Minimum percent fill	43-47	F5.3
--	'MAX='	48-51	A4
PERMAX	Maximum percent fill	52-56	F5.3
--	'MOS='	57-60	A4
MOS	MOS or specialty code	61-69	A9
--	Filler	70-80	A11

COMBINED ISSUE & MOS EXTRA JOB FILE
FILE NUMBER 36

Record Length: 325 characters
Storage Medium: Mass Storage
Source File: 31

First Use: Output from Policy Processor, Module 2.2A, Apply Combined Policies

Name	Description	Position	Format
DEMAN	Demand node name (ISSUE is first 4 characters)	1-9	A9
--	Filler	10	A1
CMF	Career management field	11-12	A2
MOS	Specialty code	13-21	A9
GRDGRP	Grade group; repeats 16 items GRDGRP contains:	22-389	(see below)
GRADE	E1-E9, W0, 01-06	(1-2)	A2
ASI	Additional skill identifier	(3-4)	A2
LIC	Language identification code	(5-6)	A2
IDENT	Person identity (M/F/O & grade)	(7)	A1
REQSTR	Required (Structure) strength	(8-13)	I6
AUTSTR	Authorized strength	(14-19)	I6
--	Filler	(20-23)	A4

MOS EXTRA JOB FILE
FILE NUMBER 37

Record Length: 325 characters
Storage Medium: Mass Storage
Source File: 36

First Use: Output from Policy Processor, Module 2.2B, Apply MOS Policies

Name	Description	Position	Format
DEMAN	Demand node name (ISSUE is first 4 characters)	1-9	A9
--	Filler	10	A1
CMF	Career management field	11-12	A2
MOS	Specialty code	13-21	A9
GRDGRP	Grade group; repeats 16 items GRDGRP contains:	22-389	(see below)
GRADE	E1-E9, W0, O1-O6	(1-2)	A2
ASI	Additional skill identifier	(3-4)	A2
LIN	Language identification code	(5-6)	A2
IDENT	Person identity (M/F/O & grade)	(7)	A1
REQSTR	Required (Structure) strength	(8-13)	I6
AUTSTR	Authorized strength	(14-19)	I6
--	Filler	(20-23)	A4

ISSUE EXTRA JOB FILE
FILE NUMBER 38

Record Length: 325 characters
Storage Medium: Mass Storage
Source File: 37

First Use: Output from Policy Processor, Module 2.2C, Apply ISSUE Policies

Name	Description	Position	Format
DEMAN	Demand node name (ISSUE is first 4 characters)	1-9	A9
--	Filler	10	A1
CMF	Career management field	11-12	A2
MOS	Specialty code	13-21	A9
GRDGRP	Grade group; repeats 16 items GRDGRP contains:	22-389	(see below)
GRADE	E1-E9, W0, O1-O6	(1-2)	A2
ASI	Additional skill identifier	(3-4)	A2
LIC	Language identification code	(5-6)	A2
IDENT	Person identity (M/F/O & grade)	(7)	A1
REQSTR	Required (Structure) strength	(8-13)	I6
AUTSTR	Authorized strength	(14-19)	I6
--	Filler	(20-23)	A4

JOB ASSIGNMENT VALUE FILE
FILE NUMBER 39

Record Length: 452 characters
Storage Medium: Mass Storage
Source File: 19, 32-35, 36-28

First Use: Output from Policy Processor, Module 2.2A, Apply Combined Policies

Name	Description	Position	Format
DEMAN	Demand node name (ISSUE is first 4 characters)	1-9	A9
JMOS	Job MOS	10-18	A9
CMF	Job career management field	19-20	A2
LOGRAD	Lowest grade used in this record	21-22	A2
HIGRAD	Highest grade used in this record	23-24	A2
AGGFIL	Desired aggregate fill; sum of GRAFIL	25-30	I6
AGGMIN	Minimum aggregate fill	31-36	I6
AGGMAX	Maximum aggregate fill	37-42	I6
AGGVAL	Value of filling to this aggregate level	43-48	I6
GRVAGR	Grade value group - repeats 16 times Included in this designation are:	49-452	16(4I6)
GRAFIL	Desired fill for this grade (I6)-may be authorized or structure		
GRAMIN	Minimum fill for this grade (I6)		
GRAMAX	Maximum fill for this grade (I6)		
GRAVAL	Value of fill in this grade (I6)		

JOB ASSIGNMENT VALUE FILE
FILE NUMBER 40

Record Length: 452 characters
Storage Medium: Mass Storage
Source File: 19, 32-35, 36-39

First Use: Output from Policy Processor, Module 2.2B, Apply MOS Policies

Name	Description	Position	Format
DEMAN	Demand node name (ISSUE is first 4 characters)	1-9	A9
JMOS	Job MOS	10-18	A9
CMF	Job career management field	19-20	A2
LOGRAD	Lowest grade used in this record	21-22	A2
HIGRAD	Highest grade used in this record	23-24	A2
AGGFIL	Desired aggregate fill; sum of GRAFIL	25-30	I6
AGGMIN	Minimum aggregate fill	31-36	I6
AGGMAX	Maximum aggregate fill	37-42	I6
AGGVAL	Value of filling to this aggregate level	43-48	I6
GRVAGR	Grade value group - repeats 16 times Included in this designation are:	49-452	16(4I6)
GRAFIL	Desired fill for this grade (I6)-may be authorized or structure		
GRAMIN	Minimum fill for this grade (I6)		
GRAMAX	Maximum fill for this grade (I6)		
GRAVAL	Value of fill in this grade (I6)		

EXCESS PEOPLE (FROM REGULAR ASSIGNMENT) FILE
FILE NUMBER 41

Record Length:
Storage Medium: Mass Storage
Source File: 2

First Use: Output from Assignment Processor, Subroutine PRENET and PSTNET

Name	Description	Position	Format
MOS	MOS or specialty code	1-9	A9
NUMEXC	Number of unassigned people by grade (location of each grade is that specified by Parameter File)	10-105	16(I6)

EXCESS PEOPLE (FROM MOS SUBSTITUTION) FILE
FILE NUMBER 42

Record Length:
Storage Medium: Mass Storage
Source File: 2

First Use: Output from Substitute Assignment Processor, Module 4.2, MOS Substitution

Name	Description	Position	Format
MOS	MOS or specialty code	1-9	A9
NUMEXC	Number of unassigned people by grade (location of each grade is that specified by Parameter File)	10-105	16(I6)

EXCESS PEOPLE (FROM GRADE SUBSTITUTION) FILE
FILE NUMBER 43

Record Length:
Storage Medium: Mass Storage
Source File: 2

First Use: Output from Substitute Assignment Processor, Module 4.1, Grade Substitution

Name	Description	Position	Format
MOS	MOS or specialty code	1-9	A9
NUMEXC	Number of unassigned people by grade (location of each grade is that specified by Parameter File)	10-105	16(I6)

EXCESS PEOPLE (SCRATCH) FILE
FILE NUMBER 44

Record Length:
Storage Medium: Mass Storage
Source File: 2

First Use: Scratch file for Assignment Processor

Name	Description	Position	Format
MOS	MOS or specialty code	1-9	A9
NUMEXC	Number of unassigned people by grade (location of each grade is that specified by Parameter File)	10-105	16(I6)

UNFILLED JOBS (FROM POLICY PROCESSOR) FILE
FILE NUMBER 50

Record Length: 98 characters
Storage Medium: Mass Storage
Source File: 38

First Use: Output from Policy Processor, Module 2.3, Set Base Values

Name	Description	Position	Format
DEMAN	Demand node name (ISSUE code is first 4 characters)	1-9	A9
MOS	MOS or specialty code	10-18	A9
NUMUNF	Number of unfilled jobs by grade	19-114	16(I6)

UNFILLED JOBS (FROM REGULAR ASSIGNMENT) FILE
FILE NUMBER 51

Record Length: 98 characters
Storage Medium: Mass Storage
Source File: 2, 40, 61

First Use: Output from Assignment Processor, Subroutines PRENET and PSTNET

Name	Description	Position	Format
DEMAN	Demand node name (ISSUE code is first 4 characters)	1-9	A9
MOS	MOS or specialty code	10-18	A9
NUMUNF	Number of unfilled jobs by grade	19-114	16(I6)

UNFILLED JOBS (FROM MOS SUBSTITUTION) FILE
FILE NUMBER 52

Record Length: 98 characters
Storage Medium: Mass Storage
Source File: 2, 51, 53

First Use: Output from Substitute Assignment Processor, Module 4.2, MOS Substitution

Name	Description	Position	Format
DEMAN	Demand node name (ISSUE code is first 4 characters)	1-9	A9
MOS	MOS or specialty code	10-18	A9
NUMUNF	Number of unfilled jobs by grade	19-114	16(I6)

UNFILLED JOBS (FROM GRADE SUBSTITUTION) FILE
FILE NUMBER 53

Record Length: 98 characters
Storage Medium: Mass Storage
Source File: 2, 51, 52

First Use: Output from Substitute Assignment Processor, Module 4.1, Grade Substitution

Name	Description	Position	Format
DEMAN	Demand node name (ISSUE code is first 4 characters)	1-9	A9
MOS	MOS or specialty code	10-18	A9
NUMUNF	Number of unfilled jobs by grade	19-114	16(I6)

UNFILLED JOBS (SCRATCH) FILE
FILE NUMBER 54

Record Length: 98 characters
Storage Medium: Mass Storage
Source File: 2, 51, 52, 53

First Use: Scratch File for Substitute Assignment Processor

Name	Description	Position	Format
DEMAN	Demand node name (ISSUE code is first 4 characters)	1-9	A9
MOS	MOS or specialty code	10-18	A9
NUMUNF	Number of unfilled jobs by grade	19-114	16(I6)

ASSIGNMENT FROM REGULAR ASSIGNMENT FILE
FILE NUMBER 61

Record Length: 43 characters
Storage Medium: Mass Storage
Source File: 2

First Use: Output from Assignment Processor, Subroutine PSTNET

Name	Description	Position	Format
DEMAN	Demand node name (ISSUE code is first 4 characters)	1-9	A9
--	Filler	10	1X
MOSREQ	MOS required for demand	11-19	A9
--	Filler	20	1X
MOSACT	Actual MOS filling demand	21-29	A9
--	Filler	30	1X
GRDREQ	Grade required for demand	31-32	A2
--	Filler	33	1X
GRDACT	Actual grade filling demand	34-35	A2
--	Filler	36	1X
ASSSTR	Assigned strength	37-43	I7

ASSIGNMENT (FROM MOS SUBSTITUTION) FILE
FILE NUMBER 62

Record Length: 43 characters
Storage Medium: Mass Storage
Source File: 2

First Use: Output from Substitute Assignment Processor, Module 4.2, MOS Substition

Name	Description	Position	Format
DEMAN	Demand node name (ISSUE code is first 4 characters)	1-9	A9
--	Filler	10	1X
MOSREQ	MOS required for demand	11-19	A9
--	Filler	20	1X
MOSACT	Actual MOS filling demand	21-29	A9
--	Filler	30	1X
GRDREQ	Grade required for demand	31-32	A2
--	Filler	33	1X
GRDACT	Actual grade filling demand	34-35	A2
--	Filler	36	1X
ASSSTR	Assigned strength	37-43	I7

ASSIGNMENT (FROM GRADE SUBSTITUTION) FILE
FILE NUMBER 63

Record Length: 43 characters
Storage Medium: Mass Storage
Source File: 2

First Use: Output from Substitute Assignment Processor, Module 4.1, Grade Substitution

Name	Description	Position	Format
DEMAN	Demand node name (ISSUE code is first 4 characters)	1-9	A9
--	Filler	10	1X
MOSREQ	MOS required for demand	11-19	A9
--	Filler	20	1X
MOSACT	Actual MOS filling demand	21-29	A9
--	Filler	30	1X
GRDREQ	Grade required for demand	31-32	A2
--	Filler	33	1X
GRDACT	Actual grade filling demand	34-35	A2
--	Filler	36	1X
ASSSTR	Assigned strength	37-43	I7

DETAILED READINESS INDICATORS FILE
FILE NUMBER 64

Record Length: 68 characters
Storage Medium: Mass Storage
Source File: 61

First Use: Output from Readiness Processor

Name	Description	Position	Format
ISSUE	ISSUE code	1-4	A4
--	Filler	5-10	6X
MOS	MOS required for demand	11-19	A9
--	Filler	20	1X
GRDREQ	Grade required for demand	21-22	A2
REQSTR	Required (Structure) strength	23-28	I6
AUTSTR	Authorized strength	29-34	I6
ASSSTR	Assigned strength	35-40	I6
AVASTR	Available people	41-46	I6
ASPEAU	ASSSTR/AUTSTR*100	47-53	F7.1
AVPEAU	AVASTR/AUTSTR*100	54-60	F7.1
AVPERE	AVASTR/REQSTR*100	61-67	F7.1

DETAILED READINESS INDICATORS USING MOS SUBSTITUTION FILE
FILE NUMBER 65

Record Length: 67 characters
Storage Medium: Mass Storage
Source File: 62

First Use: Output from Readiness Processor

Name	Description	Position	Format
ISSUE	ISSUE code	1-4	A4
--	Filler	5-10	6X
MOS	MOS required for demand	11-19	A9
--	Filler	20	1X
GRDREQ	Grade required for demand	21-22	A2
REQSTR	Required (Structure) strength	23-28	I6
AUTSTR	Authorized strength	29-34	I6
ASSSTR	Assigned strength	35-40	I6
AVASTR	Available people	41-46	I6
ASPEAU	ASSSTR/AUTSTR*100	47-53	F7.1
AVPEAU	AVASTR/AUTSTR*100	54-60	F7.1
AVPERE	AVASTR/REQSTR*100	61-67	F7.1

NOTE: For File 65, REQSTR, AUTSTR, ASPEAU, AVPEAU, AVPERE equal zero; a matching record in file 64 holds these data.

DETAILED READINESS INDICATORS USING GRADE SUBSTITUTION FILE
FILE NUMBER 66

Record Length: 67 characters
Storage Medium: Mass Storage
Source File: 63

Name	Description	Position	Format
ISSUE	ISSUE code	1-4	A4
--	Filler	5-10	6X
MOS	MOS required for demand	11-19	A9
--	Filler	20	1X
GRDREQ	Grade required for demand	21-22	A2
REQSTR	Required (Structure) strength	23-28	I6
AUTSTR	Authorized strength	29-34	I6
ASSSTR	Assigned strength	35-40	I6
AVASTR	Available people	41-46	I6
ASPEAU	ASSSTR/AUTSTR*100	47-53	F7.1
AVPEAU	AVASTR/AUTSTR*100	54-60	F7.1
AVPERE	AVASTR/REQSTR*100	61-67	F7.1

NOTE: For File 66, REQSTR, AUTSTR, ASPEAU, AVPEAU, AVPERE equal zero; a matching record in file 64 holds these data.

TOTAL RESULTS INCLUDING MOS SUBSTITUTION FILE FILE NUMBER 67

Record Length: 67 characters
Storage Medium: Mass Storage
Source File: 31, 61, 62

Name	Description	Position	Format
ISSUE	ISSUE code	1-4	A4
--	Filler	5-10	6X
MOS	MOS required for demand	11-19	A9
--	Filler	20	1X
GRDREQ	Grade required for demand	21-22	A2
REQSTR	Required (Structure) strength	23-28	I6
AUTSTR	Authorized strength	29-34	I6
ASSSTR	Assigned strength	35-40	I6
AVASTR	Available people	41-46	I6
ASPEAU	ASSSTR/AUTSTR*100	47-53	F7.1
AVPEAU	AVASTR/AUTSTR*100	54-60	F7.1
AVPERE	AVASTR/REQSTR*100	61-67	F7.1

NOTE: This file is equivalent to the sum of file 61 and file 62.

AGGREGATED READINESS INDICATOR FILE
FILE NUMBER 70

Record Length: 128 characters
Storage Medium: Mass Storage
Source File: 31, 61, 62, 63

First Use: Output from Readiness Processor

Name	Description	Position	Format
ISSUE	ISSUE code	1-4	A4
AGGREQ	Aggregate required (structure) strength	5-10	I6
AGGAUT	Aggregate authorized strength	11-16	I6
AGGASS	Aggregate assigned strength	17-22	I6
AGGAVA	Aggregate available strength	23-28	I6
H5REAG	High 5 required (structure) aggregate	29-34	I6
H5AUAG	High 5 authorized aggregate	35-40	I6
H5ASAG	High 5 assigned aggregate	41-46	I6
H5AVAG	High 5 available aggregate	47-52	I6
AGRESE	Senior grade required (structure) aggregate	53-58	I6
AGAUSE	Senior grade authorized aggregate	59-64	I6
AGASSE	Senior grade assigned aggregate	65-70	I6
AGAVSE	Senior grade available aggregate	71-76	I6
AGREOK	Correct MOS required (structure) aggregate	77-82	I6
AGAUOK	Correct MOS authorized aggregate	83-88	I6
AGASOK	Correct MOS assigned aggregate	89-94	I6
AGAVOK	Correct MOS available aggregate	95-100	I6
NUMPER	MINPER*AGGAUT	101-106	I6
MISGOR	NUMPER-AGGREQ	107-112	I6
MISGOA	NUMPER-AGGAUT	113-118	I6
PERMIN	PERMIN from base Value File	119-122	F4.2
PERMAX	PERMAX from base Value File	123-126	F4.2

AGGREGATED PERCENTAGE AND C-RATING FILE
FILE NUMBER 72

Record Length: 80 characters
Storage Medium: Mass Storage
Source File: 70

First Use: Output from Readiness Processor

Name	Description	Position	Format
ISSUE	ISSUE code	1-4	A4
--	Filler	5	A1
	Percentages		
REASPE	(AGGASS/AGGREQ)*100	6-12	F7.1
REAVPE	(AGGAVA/AGGREQ)*100	13-19	F7.1
AUASPE	(AGGASS/AGGAUT)*100	20-26	F7.1
H5REPE	(H5AVAG/H5REAG)*100	27-33	F7.1
H5AUPE	(H5ASAG/H5AUAG)*100	34-40	F7.1
AVSEPE	(AGAVSE/AGRESE)*100	41-47	F7.1
ASSEPE	(AGASSE/AGAUSE)*100	48-54	F7.1
APREOK	(AGAVOK/AGGREQ)*100	55-61	F7.1
APAUOK	(ASASOK/AGGAUT)*100	62-68	F7.1
	C-ratings		
REAVC	Based on REAVPE (aggregate)	69-70	I2
REASC	Based on REASPE (aggregate)	71-72	I2
AUASC	Based on AUASPE (aggregate)	73-74	I2
AVSEC	Based on ASEPE (senior grade)	75-76	I2
OKMOSC	Based on APREOK (correct MOS)	77-78	I2
ISCRAT	Overall C-rating. Based on lowest of REAVC, REASC, AUSEC, and MOSC	79-80	I2

GRADE READINESS AND PERCENTAGE FILE
FILE NUMBER 73

Record Length: 44 characters
Storage Medium: Mass Storage
Source File: 64, 65, 66

First Use: Output from Readiness Processor

Name	Description	Position	Format
ISSUE	ISSUE code	1-4	A4
GRADE	Grade code	5-6	A2
AGREGR	Aggregate required (structure) over all MOS	7-12	I6
AGAUGR	Aggregate authorized over all MOS	13-18	I6
AGASGR	Aggregate assigned over all MOS	19-24	I6
AGAVGR	Aggregate available over all MOS	25-30	I6
PEREGR	(AGAVGR/AGREGR)*100	31-37	F7.1
PEAUGR	(AGASGR/AGAUGR)*100	38-44	F7.1

MOS READINESS, PERCENTAGE, AND C-RATING FILE
FILE NUMBER 74

Record Length: 87 characters
Storage Medium: Mass Storage
Source File: 31, 61, 63

First Use: Output from Readiness Processor

Name	Description	Position	Format
ISSUE	ISSUE code	1-4	A4
MOS	MOS or specialty code	5-13	A9
--	Filler	14-15	A2
AGREMO	Aggregate required (structure) over all grades		16-21
AGAUMO	Aggregate authorized over all grades	22-27	I6
AGASMO	Aggregate assigned over all grades	28-33	I6
AGAVMO	Aggregate available over all grades	34-39	I6
MPERRE	(AGAVMO/AGREMO)*100	40-46	F7.1
MPERAU	(AGASMO/AGAUMO)*100	47-53	F7.1
H5REMO	High five required (structure)	54-59	I6
H5AUMO	High five authorized	60-65	I6
H5ASMO	High five assigned	66-71	I6
H5AVMO	High five available	72-77	I6
C-ratings			
RMOSC	Based on MPERRE	78-79	I2
AMOSC	Based on MPERAU	80-81	I2

STRENGTH-BY-GRADE PRINT FILE
FILE NUMBER 85

Record Length: Formated for printer, maximum length = 132 characters
Storage Medium: Mass Storage
Source File: 30

First Use: Output from Preprocessor, Module 1.3, Aggregate MOS-data to ISSUE level

Name	Description	Position	Format
--	Carriage control	1	A1
ISSUE	ISSUE code	2-5	A4
--	Filler	6	1X
AGGREQ	Aggregate required (structure) strength	7-15	I9
AGGAUT	Aggregate authorized strength	16-24	I9
--	Filler		2X
GRADE	Pay grade or rank, E1-E9, W0, O1-O6	repeated	A2
AGREGR	Aggregate required (structure) for grade	as	I5
AGAUGR	Aggregate authorized for grade	necessary	I5

STRENGTH-BY-MOS PRINT FILE
FILE NUMBER 86

Record Length: Formated for printer - maximum length = 132 characters
Storage Medium: Mass Storage
Source File: 30

First Use: Output from Preprocessor, Module 1.3, Aggregate MOS data to ISSUE level

Name	Description	Position	Format
--	Carriage control	1	A1
ISSUE	ISSUE code	2-5	A4
--	Filler	6-8	3X
CMF	Career Management Field	9-10	A2
--	Filler	11-13	3X
MOS	Military occupational specialty	14-22	A9
--	Filler	23	1X
AGREMO	Aggregate required (structure) for MOS	24-31	I12
AGAUMO	Aggregate authorized for MOS	32-40	I12

CAA-D-84-2

AGGREGATED-ISSUE INDICATOR FILE
FILE NUMBER 90

Record Length: 405 characters
Storage Medium: Mass Storage
Source File: 32

First Use: Output from Policy Processor, Module 2.2 A, Apply Combined Policies

Name	Description	Position	Format
ISSUE	ISSUE code. In this file, characters 2-3 = 00 and character 4 = blank or -	1-4	A4
AGIND	Grade aggregations dimension (16, 3) ((I,J), J=1,3), I = 1,8) (I,1) = low grade (I,2) = high grade (I,3) = value of assignment	5-76	24(I3)
AGPER	Aggregate percentages (I,1) = minimum percent (I,2) = maximum percent	77-156	16(F5.3)
USEGRD	Indicator of inclusion, one per grade dimension (16)	157-172	16(I1)
COMPUT	Indicator for individual grade policy dimension (16)	173-188	16(I1)
VALGRD	Value of assignment at this grade dimension (16)	189-284	16(I6)
PERIND	Percentages for each grade dimension (16, 2) (I,1) = minimum percentage (I,2) = maximum percentage	285-444	32(F5.3)

POLICY FILE
FILE NUMBER 91

Record Length: 80 characters
Storage Medium: Mass Storage
Source File: PFPRIM-FILES

First Use: Input to Policy Processor, Module 2.1, Edit Policy File

Name	Description	Position	Format
POLTYP	Policy type. One of: 'ISSUE' 'MOSSC'	1-5	A5
--	= sign	6	A1
ISSUE	ISSUE code	7-10	A4
--	'LO='	11-13	A3
LOGRAD	Lowest grade to which policy applies	14-15	A2
--	'HI='	16-18	A3
HIGRAD	Highest grade to which policy applies	19-20	A2
--	'AG='	21-23	A3
AGG	'YES' if policy is to aggregate 'NO ' if policy is not to aggregate	24-26	A3
--	'VALUE='	27-32	A6
VALUE	Fill value	33-35	I3
--	Filler	36-38	A3
--	'MIN='	39-42	A4
PERMIN	Minimum percent fill	43-47	F5.3
--	'MAX='	48-51	A4
PERMAX	Maximum percent fill	52-56	F5.3
--	'MOS='	57-60	A4
MOS	MOS or specialty code	61-69	A9
--	Filler	70-80	A11

EDITED POLICY FILE
FILE NUMBER 92

Record Length: 80 characters
Storage Medium: Mass Storage
Source File: 91

First Use: Output from Policy Processor, Module 2.1, Edit Policy

Name	Description	Position	Format
POLTYP	Policy type. One of: 'ISSUE' 'MOSSC'	1-5	A5
--	= sign	6	A1
ISSUE	ISSUE code	7-10	A4
--	'LO='	11-13	A3
LOGRAD	Lowest grade to which policy applies	14-15	A2
--	'HI='	16-18	A3
HIGRAD	Highest grade to which policy applies	19-20	A2
--	'AG='	21-23	A3
AGG	'YES' if policy is to aggregate 'NO ' if policy is not to aggregate	24-26	A3
--	'VALUE='	27-32	A6
VALUE	Fill value	33-35	I3
--	Filler	36-38	A3
--	'MIN='	39-42	A4
PERMIN	Minimum percent fill	43-47	F5.3
--	'MAX='	48-51	A4
PERMAX	Maximum percent fill	52-56	F5.3
--	'MOS='	57-60	A4
MOS	MOS or specialty code	61-69	A9
--	Filler	70-80	A11

REPORT REQUEST FILE
FILE NUMBER 93

Record Length:
Storage Medium: Mass Storage
Source File: PFPRIM-FILES

First Use: Input to Report Processor

Name	Description	Position	Format
REPNUM	Report number	1-2	A2
--	Filler	3	A1
YES/NO	'Yes' or 'no'	4-6	A3
--	Filler	7	A1
WHICH	Report type: 'MOS', 'All', 'ORIGINAL', , 'MOS SUB', 'Grade Sub', or ISSUE code when report number is 03, 08, 10, or 11; WHICH=blank when report number is 01, 02, 04, 05, 06, 07 or 09	8-19	A12
--	Filler	20	A1
REPNAM	Report name	21-81	A61

*Please refer to section 2.4.6 and Appendix C, section 5 for a full description on the use of this file.

SECTION 4

PROGRAM MAINTENANCE PROCEDURES

4.1 CONVENTIONS

4.1.1 Process Numbering Conventions. Process numbers have been assigned to all main programs and subroutines which perform a function described in the PRIM Functional Description (CAA-D-84-1). Typically, these numbered programs are called upon to perform the stated function one time during a complete set of model executions. If the numbered program is a subroutine, it will be called by one main program or by one subroutine that is higher in the hierarchical organization, and will never be called by any other program. Subroutines that have not been assigned a process number are utility-type subroutines that perform a function that is required by the program design but which is transparent to the user. These subroutines are usually called by more than one main program or subroutine, and some are called by programs in more than one processor. The subroutines that are called by multiple programs, but within one processor only, are described in alphabetical order in the section that describe that processor. Subroutines that are used by more than one processor are described in Section 2.10, PRIM Utility Program Descriptions.

a. Translating The Process Number. Each PRIM process has been assigned an integer number in the range of one to six which represents the order of that process in the sequential processing (see Table 4.1). The next digits represent the module or subroutine order in the processing order within the process. The individual process digits are described below:

- First digit — The process number.
- Second digit — The integer represented by statements in the Functional Description such as first, second, and third. When this digit is zero, there are no other main programs for the processor.
- Third digit — A subsequence number representing the logical order in which this self-contained sub-function should be processed. When not present, there were not multiple sub-functions.
- Fourth digit — If present, this digit represents the logical sequence within the hierarchy represented by the first three digits.

b. Example of Process Number: In the process number 1.2.3:

1 represents the processor, Preprocessor,

2 represents the second function of the Preprocessor "Set the Issue code" described in the Functional Description,

3 represents the third method of setting the ISSUE code, "Set the ISSUE based on ASGMT code."

Table 4-1. PRIM Process

Process order	Process name
1	Preprocessor
2	Policy Processor
3	Assignment Processor
4	Substitute Assignment Processor
5	Readiness Processor
6	Report Processor

4.1.2 File Name Conventions. File names are assigned as follows (note that file names may be 12 or fewer characters):

Character 1-2	PF, which represents the Force Plans division at MILPERCEN.
Character 3-10	Mnmemonic for the type of data contained in the file. When possible, the entire word is used as in PFISSUE14 and PFPOLICY91. Otherwise, mnemonics are the first one to three characters of the words that describe it, such as PFAGISP032 (aggregated-ISSUE policy).
Character 11-12 (or last two characters)	File number used in program and for which an @USE statement wil be needed in the runstream.

4.1.3 Data Element Names. Data elements have the same name in all files (see Appendix A, Data Dictionary). In general, all programs use these same names. However, when the same element must be read from more than one file for comparison or for addition, a variation on the same name is used which changes the first or last character.

4.2. VERIFICATION PROCEDURES

a. The processors which are most likely to require future modifications are the Preprocessor and the Policy Processor. Most of the programs in these processors contain print statements that are executed or not based on the value of a variable called IDEBUG. The larger the IDEBUG value is set, the more debug print is produced. The programer may trace all the processing of input data, the parameter values, and the process interactions by running the program with the debug flag set to its maximum value or trace certain areas by selectively setting it. A table ordered by program name which shows the debug flag settings that affect each program will be found in Table 4-2, Debug Flag Settings by Program Name. Table 4-3, Debug Flags by Process shows the valid debug flags for each process.

b. The debug flag in process 1.2, Set ISSUE Code controls whether the output file of MOS-data will be written. When the flag is set to nine (9) or greater and the number of serious errors is greater than zero, the ISSUE code is not set in the MOS-data and the output file is not written. This option allows the user to begin testing the ISSUE definitions prior to completion of process 1.1.2, Roll MOS-data to three-digit UIC level.

c. The Report Processor is designed as a main program (DRIVE6) which reads the report requests and calls the appropriate subroutine. A different formatted report is produced by each subroutine. This processor can be easily modified by adding a report number to the Report Request file, file number 93 and adding an additional call to the end of program DRIVE6, or by replacing a report subroutine with a different subroutine.

d. The Assignment Processors and the Readiness Processor are not likely to require modification and no special procedures have been developed for verification of future modifications.

Table 4-2. Debug Flag Settings by Program Name

Program name	Flags						
	1/8	2/9	3/10	4/11	5/12	6/13	7/14
AGGMOS	1	2	6	10	20	30	
AGGUIC	5	19	98				
AGISPO	7						
CONPOL	4						
DOIT	3	8					
DOIT12	6						
DOTINO	6						
DRIVE1/1-2	9	10	15				
DRIVE1/1-3	10						
EDIPOL	1	2	3	8			
FINDAT	7						
FINVAL	4						
GETBAS	2	3					
GETJOB	2	3	5	8			
GETPAR	2	11					
GETPOL	2	11					
GETVAL	11						
ISMO	6	8					
MAKAGG	5	8					
MAKIND	7						
MATPOL	6						
NXTISS	5						
ROLUIC/1-1-1	3	9					
SETASG	5	11	31	34	37	40	69
	71	72	74	75	78	79	
SETBAS	3	4					
SETMOS	5	65	66				
SETPOL	4	6					
SETTPS	5	11	46	49	52	55	88
	92	95	98				
SETUIC	5	11	43	84	85	86	
SRTPOL	1						
WRTEXT	3	7					
WRTUNF	4						
WRTVAL	3	7					

Table 4-3. Debug Flag Settings by Process

Process	Flags						
	1/8	2/9	3/10	4/11	5/12	6/13	7/14
1.1.1	3	9					
1.2	6	9	10	15			
1.2.1	5	11	46	49	52	55	88
	92	95	98				
1.2.2	5	11	43	84	85	86	
1.2.3	5	11	31	34	37	40	69
	71	72	74	75	78	79	
1.2.4	5	19	98				
1.2.5	5	65	66				
1.3	1	2	6	10	20	30	
2.1	1	8					
2.2.1	2	11					
2.2.2	1						
2.2.3	7						
2.2.4 thru 2.2.	2	3	4	5	6	7	8
	11						
2.3	2	3	4	5	8	11	

4.3 ERROR CONDITIONS

The majority of the errors are found by the programs that read and reformat the user-provided files. These programs are in the Preprocessor and Policy Processor. Errors identified by these processors are written to file 13 which is defined as an alternate print file. The errors can be sent to the printer with the @SYM command, printed on a local printer with the @ED command, or scanned from the demand terminal. A complete listing of the error messages, and a description of the reason for each error and the action taken by the program will be found is Appendix E. Table 4-4 associates an error number with the programs which write that error. Errors originating from the computer operating system should be handled in accordance with standard operations procedures.

Table 4-4. Error Numbers and Programs

Message or Error	From Programs			
0	DRIV12	DRIVE13	DRIVE 2/2-A	
	DRIVE2/2-B	DRIVE2/2-C	DRIVE2/3	
	APP22A	APP22B		
1	SETTPS	SETUIC		
2	ROLMOS			
3	SETBAS			
4	ROLMOS			
5	SETASG	SETMOS		
6	SETUIC			
7	SETTPS	SETUIC	SETASG	
	ISSFIN			
9	ISSUNA			
10	EDIPOL			
11	EDIPOL			
12	EDIPOL			
13	EDIPOL	SETPOL		
14	EDIPOL			
15	EDIPOL			
16	EDIPOL			
17	EDIPOL			
18	EDIPOL			
19	EDIPOL			
20	EDIPOL			
21	EDIPOL			
22	EDIPOL	MAKAGG	SETPOL	
23	GETBAS			
24	GETBAS			
25	GETBAS			
26	GETBAS	SETBAS		
30	ROLMOS	AGGMOS	EDIPOL	
	GETPAR	SETPOL		
32	ROLMOS	AGGMOS	PAR	
37	EDIPOL	GETPAR		
39	ROLMOS	EDIPOL	GETPAR	
40	ROLMOS	AGGMOS	GETPAR	
41	GETPAR			
42	ROLMOS	SRTPOL		
43	SETUIC			
44	SETPOL			
45	MAKAGG			
46	EDIPOL	SETPOL		
61	GETBAS			
62	GETBAS			
70	AGGMOS			
71	AGGMOS			
97	NXTISS			
98	AGGMOS	EDIPOL	GETPAR	AGISPO
	GETJOB			
99	AGGMOS	EDIPOL	GETPAR	SRTPOL
	AGISPO	GETBAS	GETJOB	GETPOL
	MATPOL	WRTEXT	WRTUNF	WRTVAL
103	SETTPS	SETUIC		
125	GETJOB			
126	ROLMOS			

4.4 SPECIAL MAINTENANCE PROCEDURES

a. Executive Control Language (ECL)

The ECL associated with each of the 16 runstreams used to run the 6 processors is shown in Figures 4-1 through 4-16. Although there are only 14 different modules, there are two extra runstreams for the Substitute Assignment Processor; one for performing grade substitution first, and one for performing MOS substitution first.

b. Compile and Mapping Procedures

The commands used to compile and map the symbolic elements into absolute elements are shown in Figure 4-17 through 4-30. Prior to any of these runs, the PRIM utility programs should be compiled using the commands shown in Figure 4-31.

4.5 SPECIAL MAINTENANCE PROCEDURES

a. Program File Assignment

The files that will be needed to recreate the PRIM system from a tape backup may be created with the commands shown in Figure 4-32.

b. Data File Assignment

The data files that will be needed can be created using one runstream that creates the files needed by the first processor (Figure 4-33), another that creates the files needed by the next processor (Figure 4-34), and a final runstream that creates the files needed for the Assignment Processor through the Report Processor (Figure 4-35).

4.6 LISTINGS

The program listings, including symbolics, absolutes, and ECL are resident in the MILPERCEN system libraries.

```
:
: * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
:
:  AGGREGATE THE UIC DATA (FROM "HEADER RECORD") TO 3-DIGIT LEVEL
:
:  THE RUNSTREAM FOR CREATING THE EXECUTABLE ELEMENT IS:
:                          08PRIM-MAP.3DIUIC/1-1-1
:
: * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
:
:    - - -   ASSIGN THE 6-DIGIT UIC FILE                    I N P U T
:
@ASG,A        PFUIC6DIG22.
@USE      22.,PFUIC6DIG22.
:
:    - - -   ASSIGN THE 3-DIGIT UIC FILE                  O U T P U T
:
@ASG,A        PFUIC3DIG23.
@ERS          PFUIC3DIG23.
@USE      23.,PFUIC3DIG23.
:
@ASG,A        PFPRIM-ABS.
@FREE         PFPRIM-ABS.
@XQT          PFPRIM-ABS.755ROLUIC/1-1-1
  1
:
@FREE         PFUIC6DIG22.
@FREE         PFUIC3DIG23.
```

Figure 4-1. ROLUIC/1-1-1 Runstream

```
@ .
@ .    FIRST SORT THE FILE ON THE FIRST 4 CHARACTERS OF THE MOS WITHIN
@ .
@ .    CHARACTER 2-4 OF UIC.  I.E.  UIC CHARACTER 2, 3 & 4 IS THE MAJOR
@ .
@ .    SORT AND CHARACTER 1 THROUGH 4 OF MOS IS THE MINOR SORT
@ .
@ASG,A      PFINPUTMOS8.
@USE   8.,PFINPUTMOS8.
@ASG,T      XA.,///1000
@ASG,T      XB.,///11000
@ .
@SORT,ES
FILEIN=8.
FILEOUT=8.
KEY=3,3,S,A
KEY=19,4,S,A
RSZ=132
RECORD=270
@EOF
@ .
@FREE       XA.
@FREE       XB.
@FREE       PFINPUTMOS8.
@ .  - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
@ASG,A      PFERROR13.                  OUTPUT - ERROR MESSAGES
@ERS        PFERROR13.
@ .
@ASG,A      PFPARAMET17.                INPUT  - PARAMETER FILE (MAIN)
@ .
@ASG,A      PFINPUTMOS8.                INPUT  - FIRST INPUT OF MOS DATA
@ .
@ASG,A       PFMOS-ENL-26.            . OUTPUT - USED TO TEMPORARILY HOLD ENLISTED
@ .
@ASG,A       PFMOS-OFF-27.            . OUTPUT - USED TO TEMPORARILY HOLD OFFICER
@ .
@ASG,A       PFMOS-WO-28.             . OUTPUT - USED TO TEMPORARILY HOLD WO
@ .
@ASG,A      PFMOSDATA29.                OUTPUT - ALL MOS DATA ROLLED TO 3-DIGIT
@ .                                              UIC LEVEL - MOS HAS ALSO BEEN
@ .                                              ROLLED AS SPECIFIED BY THE
@ .                                              PARAMETER FILE
@ .
@USE   13.,PFERROR13.
@USE   17.,PFPARAMET17.
@USE    8.,PFINPUTMOS8.
@USE   26.,PFMOS-ENL-26.
@USE   27.,PFMOS-OFF-27.
@USE   28.,PFMOS-WO-28.
@USE   29.,PFMOSDATA29.
@ .
@ .                 ERASE THE ERROR-PRINT FILE AND THE INPUT PARAMETER FILE
@ .
@ERS                      13.
@ERS                      17.
@ .
@ .                 MOVE THE PARAMETER FILE INTO FILE 17 FOR USE NOW
@ .
@ASG,A      PFPRIM-FILES.
@ED         PFPRIM-FILES.PARAMETER,17.
EXIT
@FREE       PFPRIM-FILES.
@ .
@ .
@ .
@ .                 EXECUTE THE PROGRAMS - THIS PROGRAM HAS NO DEBUG FLAG
@ .                 (NOTE:  THIS IS A SET OF TWO SEQUENTIAL PROGRAMS)
@ .                 (       FILE 8 IS INPUT TO "SEPARATE", OUTPUT FILES)
@ .                 (       ARE 26, 27, AND 28.  THESE ARE THE MAIN INPUT)
@ .                 (       TO "ROLMOS" AND FILE 29 IS THE OUTPUT)
@ .
@ASG,A                 PFPRIM-ABS.
@ .
@FREE                  PFPRIM-ABS.
@XQT                   PFPRIM-ABS.755SEPARATE/1-1-2A
@ .
@ED,R                  PFERROR13.
P!
@ .
@XQT                   PFPRIM-ABS.755ROLMOS/1-1-2
@ .
@ .   AFTER MOVING TO MILPERCEN THE FOLLOWING STATEMENTS SHOULD BE
@ .   REPLACED WITH A SYM OF FILE 13 TO THE MILPERCEN PRINTER
@ .
@ED,R   13.
LNP!
EXI
@FREE       PFERROR13.
@FREE       PFPARAMET17.
@FREE       PFINPUTMOS8.
@FREE       PFMOS-ENL-26.
@FREE       PFMOS-OFF-27.
@FREE       PFMOS-WO-28.
@FREE       PFMOSDATA29.
```

Figure 4-2. ROLMOS/1-1-2 Runstream

```
@ .     PRIOR RUNSTREAM:   POLMOS/1-1-2
@ .
@ .      NEXT RUNSTREAM:   AGGMOS/1-3
@ .
@ . - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
@ .
@ .     SORT THE MOS DATA FILE WHICH HAS BEEN ROLLED ON MOS TO 3-DIGIT LEVEL
@ .
@ .     SORT KEY IS THE 3-DIGIT UIC
@ .
@ASG,A      PFMOSDATA29.
@USE   29.,PFMOSDATA29.
@ASG,T       XA.,///1CDC
@ASG,T       XB.,///1CDCD
@SORT,ES
FILEIN=29.
FILEOUT=29.
KEY=2,3,S,A
KEY=13,9,S,A
RSZ=325,CHARACTERS
@ .                    SENSITIVITY TEST FILE IS SMALL - NOT NEEDED  RECORD=9D
@EOF
@FREE       XA.
@FREE       XB.
@FREE       29.
@ . - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
@ .
@ .     ASSIGN THE FILES NEEDED FOR "SETISSUE", PROCESSOR 1-2
@ .
@ . - - - - - - - - - - - - - - - - - -    ERASE THE ERROR PRINT FILE
@ASG,A      PFERROR13.
@ERS        PFERROR13.
@ . - - - - - - - - - - - - - - - - - -  MOVE THE LATEST COPY OF ISSUE
@ .                                        INTO THE FILE FOR USE
@ASG,A      PFISSUE14.
@ERS        PFISSUE14.
@ED       PFPRIM-FILES,ISSUE,PFISSUE14.
EXI
@ . - - - - - - - - - - - - - - - - -   MOVE THE LATEST COPY OF PARAMETER
@ .                                       INTO THE FILE FOR USE
@ASG,A      PFPARAMET17.
@ERS        PFPARAMET17.
@ED       PFPRIM-FILES.PARAMETER,PFPARAMET17.
EXI
@ .                INPUT  -  23 IS THE UIC FILE ROLLED TO 3-DIGIT UIC
@ .                             THIS FILE WAS AN OUTPUT FILE FROM THE
@ .                             RUN  "ROLUIC/1-1-1"
@ASG,A PFUIC3DIG23.
@ .
@ .               OUTPUT  -  24 & 25 - WORK FILES USED FOR UIC WITH
@ .                                    ISSUE ADDED.  EITHER ONE MAY BE
@ .                                    THE FINAL OUTPUT.  THE PROGRAM
@ .                                    IDENTIFIES THE FINAL OUTPUT FILE.
@ASG,A PFUICDATA24.
@ASG,A PFUICDATA25.
@ .
@ .              INPUT - 29 - THE MOS DATA ROLLED TO 3-DIGIT UIC AND TO
@ .                           THE NUMBER OF CHARACTERS OF MOS SPECIFIED
@ .                           IN THE PARAMETER FILE (17)
@ASG,A PFMOSDATA29.
@ .
@ .             OUTPUT  -  30  - THE MOS DATA WITH ISSUE CODE ADDED
@ .                                THIS FILE MUST BE SORTED ON ISSUE AND
@ .                                MOS PRIOR TO RUNNING THE NEXT PROCESSOR
@ASG,A PFMOSDATA30.
@ . - - - - - - - - -  ATTACH LOGICAL UNIT NUMBERS (LUN) TO THE FILES
@USE   13.,PFERROR13.
@USE   14.,PFISSUE14.
@USE   23.,PFUIC3DIG23.
@USE   24.,PFUICDATA24.
@USE   25.,PFUICDATA25.
@USE   29.,PFMOSDATA29.
@USE   30.,PFMOSDATA30.
@ .
@ . - - - - - - - - -    ASSIGN SOME TEMPORARY SCRATCH FILES
@ .
@ASG,T      XA.,///2CD
@ASG,T      XB.,///16DD
@ .
@FREE        PFPRIM-ABS.
@XQT,F    PFPRIM-ABS.755SETISSUE/1-2
 1-0 E B U G   F L A G
@ED,R 13.
ENP*
EXI
@SYM,U    PFERROR13.
@FREE     PFERROR13.
@FREE     PFISSUE14.
@FREE     PFPARAMET17.
@FREE     PFUIC3DIG23.
@FREE     PFUICDATA24.
@FREE     PFUICDATA25.
@FREE     PFMOSDATA29.
@FREE     PFMOSDATA30.
@FREE   XA.
@FREE   XB.
```

Figure 4-3. SETISSUE/1-2 Runstream

```
@ .- - - - - - - - - - - - - - - - SORT THE INPUT MOS FILE BY ISSUE AND MOS
@ .
@ASG,A      PFMOSDATA30.
@ .
@ASG,I  XA.,///1000
@ASG,T  XB.,///11000
@SORT,ES                                  *** FILE MUST BE SORTED! ***
FILEIN=PFMOSDATA30.
FILEOUT=PFMOSDATA30.
KEY=7,4,S,A
KEY=13,9,S,A
@EOF
@ .
@FREE       PFMOSDATA30.
@FREE XA.
@FREE XB.
@
@ .
@ .- - - - - - - - - - - - - - - - ASSIGN THE ERROR PRINT FILE
@ .
@ASG,AX     PFERROR13.
@ERS        PFERROR13.
@ .
@ .- - - - - - - - - - - - - - - - ASSIGN THE PARAMETER FILE
@ .
@ASG,A      PFPARAMET17.
@ED,R       PFPRIM-FILES.PARAMETER,PFPARAMET17.
@ .
@ .- - - - - - - - - - - - - - - - INPUT - MOS DATA AT 3-DIGIT UIC LEVEL
@ .
@ASG,A      PFMOSDATA30.
@ .
@ .- - - - - - - - - - - - - - - - OUTPUT - MOS DATA AGGREGATED TO ISSUE LEVEL
@ .
@ASG,A      PFMOSISSUE31.
@ .
@ .- - - - - - - - - - - - - - - - OUTPUT - ISSUE AGGREGATES & GRADE AGGREGATES
@ .
@ASG,A      PFGRASTREN85.
@ .
@ .- - - - - - - - - - - - - - - - OUTPUT - MOS AGGREGATES BY ISSUE
@ .
@ASG,A      PFMOSSTREN86.
@ .
@USE    13.,PFERROR13.
@USE    17.,PFPARAMET17.
@USE    30.,PFMOSDATA30.
@USE    31.,PFMOSISSUE31.
@USE    85.,PFGRASTREN85.
@USE    86.,PFMOSSTREN86.
@XQT,F      PFPRIM-ABS.755AGGMOS/1-3
 1-D E B U G   F L A G
@ED,R 13.
LNP
EXI
@FREE       PFMOSDATA30.
@FREE       PFERROR-S-13.
@FREE       PFPARAMET17.
@FREE       PFMOSISSUE31.
@FREE       PFGRASTREN85.
@FREE       PFMOSSTREN86.
```

Figure 4-4. AGGMOS/1-3 Runstream

```
@HDG,U  EDIPOL/2-1      E D I T   P O L I C Y   F I L E
@PFPRIM-ABS.TITLE,D   EDIT,POLICY,,2$-$1
@PFPRIM-ABS.TITLE,D   EDIT,POLICY,,2$-$1
@ .                   - - - - - - - ASSIGN THE ERROR PRINT FILE
@ASG,A      PFERROR13.
@ERS        PFERROR13.
@ .                   - - - - - - - ASSIGN THE ISSUE FILE
@ASG,A      PFISSUE14.
@ED         PFPRIM-FILES.ISSUE,PFISSUE14.
EXI
@ .                   - - - - - - - ASSIGN THE PARAMETER FILE
@ASG,A      PFPARAMET17.
@ED         PFPRIM-FILES.PARAMETER,PFPARAMET17.
EXI
@ .                   - - - - - - - ASSIGN THE POLICY FILE
@ASG,A      PFPOLICY91.
@ED         PFPRIM-FILES.POLICY,PFPOLICY91.
EXI
@ .                   - - - - - - - ASSIGN THE OUTPUT POLICY FILE
@ASG,A      PFEDIPOL92.
@USE         13.,PFERROR13.
@USE         14.,PFISSUE14.
@USE         17.,PFPARAMET17.
@USE         91.,PFPOLICY91.
@USE         92.,PFEDIPOL92.
@ .                   - - - - - - - ASSIGN TEMPORARY FILES FOR SORTING POLICY FILE
@ASG,T          XA.,///100
@ASG,T          XB.,///1000
@XQT        PFPRIM-ABS*755FDIPOL
  2-D E B U G   F L A G
@ED,R 13.
LNP!
EXI
@ED,R 92.
LNP!
EXI
@FREE 13.
@FREE 14.
@FREE 17.
@FREE 91.
@FREE 92.
@FREE XA.
@FREE XB.
```

Figure 4-5. EDIT-POLICY/2-1 Runstream

```
@HOG  APPLYPOLICY/2-2-A     APPLY POLICIES TO MOS DATA, PART A
@PFPRIM-ABS.TITLE,D    APPLY,POLICY,,2-2/A
@PFPRIM-ABS.TITLE,D    APPLY,POLICY,,2-2/A
.
.    T H I S   I S   P R O C E S S   2 - 2 - A   (PART A OF APPLY POLICY)
.
.  THIS RUNSTREAM IS FOR PART A OF THE APPLY POLICY PROCESS (2.2/A)
.
.  APPLY POLICY PROCESS WAS BROKEN INTO SEVERAL PARTS TO ALLOW THE
.  USER MORE FLEXIBILITY.  WHEN ERRORS ARE FOUND BY ONE OF THE LATER
.  SUBROUTINES, IT IS NOT NECESSARILY REQUIRED THAT ALL PARTS ARE
.  REPEATED.  WHEN THE POLICY FILE IS CHANGED, AS A RESPONSE TO ERRORS
.  IDENTIFIED AT PART B, THE EDIT POLICY MODULE (PROCESS 2.1) MUST
.  BE RERUN.  THE USER CAN DO SO IMMEDIATELY RATHER THAN WAIT FOR ALL THE
.  REST OF THE PROCESSOR TO RUN TO COMPLETION.
.
. - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
.
. - - - - - - - - - - - - - - - - - - S O R T   M O S   D A T A   O N :
.                                                    (1)  I S S U E
.                                                    (2)  M O S
.
.                                          THIS FILE IS AN OUTPUT FROM 1-3
.
@ASG,A         PFMOSISSUE31.
.
@ASG,T      XA.,///400
@ASG,T      XB.,///2800
.
@SORT,ES
FILEIN=PFMOSISSUE31.
FILEOUT=PFMOSISSUE3I.
KEY=7,4,S,A
KEY=13,9,S,A
@EOF
.
@FREE       PFMOSISSUE31.
@FREE       XA.
@FREE       XB.
.
. - - - - - - - - - - - - - - - - - - ASSIGN THE ERROR OUTPUT PRINT FILE
.
@ASG,A       PFERROR13.
@ERS         PFERROR13.
. - - - - - - - - - - - - - - - - - - ASSIGN THE PARAMETER FILE
@ASG,A       PFPARAMET17.
@ADD         PFPRIM-FILES.PARAMETER,PFPARAMET17.
@EXT
. - - - - - - - - - - - - - - - - - - ASSIGN THE POLICY FILE  - - -  INPUT
@ASG,A       PFEDIPOL92.
.                                          THIS FILE IS AN OUTPUT FROM 2-1
. - - - - - - - - - - - - - - - - - - ASSIGN THE INPUT JOB FILE - - - - INPUT
.                                          THIS FILE IS AN OUTPUT FROM 1-3 AND IS
.                                          SORTED NOW ON ISSUE AND MOS
@ASG,       PFMOSISSUE31.
. - - - - - - - - - - - - - - - - - - ASSIGN THE EXTRA JOB FILE - - - - OUTPUT
@ASG,A       PFISMOEX36.
@ERS         PFISMOEX36.
. - - - - - - - - - - - - - - - - - - ASSIGN THE OUTPUT POLICY FILES
.
@ASG,A       PFAGISPO32.
@ASG,A       PFISSPOL33.
@ASG,A       PFMOSPOL34.
@ASG,A       PFISMOPO35.
@ASG,A       PFAGISIND9O.
@ERS         PFAGISIND9O.
. - - - - - - - - - - - - - - - - - - ASSIGN THE OUTPUT JAV FILE - - - - OUTPUT
@ASG,A       PFJOBASVAL39.
@ERS         PFJOBASVAL39.
.
@USE     17.,PFERROR13.
@USE     17.,PFPARAMET17.
@USE     31.,PFMOSISSUE31.
@USE     32.,PFAGISPO32.
@USE     33.,PFISSPOL33.
@USE     34.,PFMOSPOL34.
@USE     35.,PFISMOPO35.
@USE     36.,PFISMOEX36.
@USE     39.,PFJOBASVAL39.
@USE     9O.,PFAGISIND9O.
@USE     92.,PFEDIPOL92.
. - - - - - - - - - - - - - - - - - - ASSIGN TEMPORARY FILE FOR POLICY SORT
.
@ASG,T      XA.,///2O
@ASG,T      XB.,///14O
.
@XQT,F       PFPRIM-ABS.7SSAPPPOL/2-2-A
@ADD B # G   F L A G
@ERS         PFERROR13.
@EOJ
@FIN*
@EXT.
@FREE       13.
@FREE       17.
@FREE       31.
@FREE       32.
@FREE       33.
@FREE       34.
@FREE       35.
@FREE       36.
@FREE       39.
@FREE       9O.
@FREE       92.
@FREE       XA.
@FREE       XB.
```

Figure 4-6. APPLY-POLICY/2-2-A

```
@HOG    APPLYPOLICY/2-2-B     APPLY POLICIES TO MOS DATA, PART B
@.
@PFPRIM-ABS.TITLE,D APPLY,POLICY,,2-2-B
@PFPRIM-ABS.TITLE,D APPLY,POLICY,,2-2-B
@.    T H I S   I S   P R O C E S S   2 . 2 . B  (PART B OF APPLY POLICY)
@.
@.  THIS RUNSTREAM IS FOR PART B OF THE APPLY POLICY PROCESS (2.2/9)
@.
@.  APPLY POLICY PROCESS WAS BROKEN INTO SEVERAL PARTS TO ALLOW THE
@.  USER MORE FLEXIBILITY.  WHEN ERRORS ARE FOUND BY ONE OF THE LATER
@.  SUBROUTINES, IT IS NOT NECESSARILY REQUIRED THAT ALL PARTS ARE
@.  REPEATED.  WHEN THE POLICY FILE IS CHANGED, AS A RESPONSE TO ERRORS
@.  IDENTIFIED BY PART B, THE EDIT POLICY MODULE (PROCESS 2.1) MUST
@.  BE RERUN.  THEN, AS LONG AS CHANGES ARE NOT MADE TO THE TYPE
@.  OF POLICIES USED IN PART A, THE USER MAY THEN RETURN TO PART B
@.  TO CONTINUE PROCESSING.
@. - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
@.
@. - - - - - - - - - - - - - - - - - S O R T   M O S   D A T A   O N :
@.                                          M O S   O N L Y
@.                                      THIS FILE IS AN OUTPUT FROM 2-2A
@.
@ASG,A          PFISMOEX36.
@.
@ASG,T      XA.,///400
@ASG,T      XB.,///2800
@.
@SORT,ES
FILEIN=PFISMOEX36.
FILEOUT=PFISMOEX36.
KEY=13,9,S,A
@EOF
@.
@FREE       PFISMOEX36.
@FREE       XA.
@FREE       XB.
@. - - - - - - - - - - - - - - - - - - - - ASSIGN THE ERROR OUTPUT PRINT FILE
@.
@ASG,A       PFERROR13.
@ERS         PFERROR13.
@. - - - - - - - - - - - - - - - - - - - - ASSIGN THE PARAMETER FILE
@ASG,A       PFPARAMET17.
@ED          PFPRIM-FILES.PARAMETER,PFPARAMET17.
EXI
@. - - - - - - - - - - - - - - - - - - - - ASSIGN THE MOS POLICY FILE - - -  INPUT
                                           THIS FILE IS AN OUTPUT FROM 2-2A
@ASG,A       PFMOSPOL34.
@.
@. - - - - - - - - - - - - - - - - - - - - ASSIGN THE INPUT JOB FILE - - - - INPUT
                                           THIS FILE IS AN OUTPUT FROM 2-2A
@ASG,A       PFISMOEX36.
@.
@. - - - - - - - - - - - - - - - - - - - - ASSIGN THE EXTRA JOB FILE - - - - OUTPUT
@ASG,A       PFMOSEXT37.
@ERS         PFMOSEXT37.
@.
@. - - - - - - - - - - - - - - - - - - - - ASSIGN THE INPUT JAV FILE - - - - INPUT
                                           THIS FILE IS AN OUTPUT FROM 2-2A
@ASG,A       PFJOBASVAL39.
@.
@. - - - - - - - - - - - - - - - - - - - - ASSIGN THE OUTPUT JAV FILE - - - - OUTPUT
@ASG,A       PFJOBASVAL40.
@ERS         PFJOBASVAL40.
@.
@USE     13.,PFERROR13.
@USE     17.,PFPARAMET17.
@USE     34.,PFMOSPOL34.
@USE     36.,PFISMOEX36.
@USE     37.,PFMOSEXT37.
@USE     39.,PFJOBASVAL39.
@USE     40.,PFJOBASVAL40.
@.
@. - - - - - - - - - - - - - - - - - - - - ASSIGN TEMPORARY FILES FOR POLICY SORT
@.
@ASG,T       XA.,///10
@ASG,T       XB.,///70
@.
@XQT,F       PFPRIM-ABS.755APPPOL/2-2-B
 2-D E B U G   F L A G
@ED,R        PFERROR13.
LVP!
EXI
@FREE       13.
@FREE       17.
@FREE       34.
@FREE       36.
@FREE       37.
@FREE       39.
@FREE       40.
@FREE       XA.
@FREE       XB.
```

Figure 4-7. APPLY-POLICY/2-2-B Runstream

```
@MOG   APPLYPOLICY/2-2-C     APPLY POLICIES TO MOS DATA, PART C
@PFPRIM-ABS.TITLE,O  APPLY,POLICY,,2-2-C
@PFPRIM-ABS.TITLE,D  APPLY,POLICY,,2-2-C
@ .
@ .    T H I S   I S   P R O C E S S   2 . 2 . 6  (PART C OF APPLY POLICY)
@ .
@ .  THIS RUNSTREAM IS FOR PART C OF THE APPLY POLICY PROCESS (2-2-C)
@ .
@ .  APPLY POLICY PROCESS WAS BROKEN INTO SEVERAL PARTS TO ALLOW THE
@ .  USER MORE FLEXIBILITY.  WHEN ERRORS ARE FOUND BY ONE OF THE LATER
@ .  SUBROUTINES, IT IS NOT NECESSARILY REQUIRED THAT ALL PARTS ARE
@ .  REPEATED.  WHEN THE POLICY FILE IS CHANGED, AS A RESPONSE TO ERRORS
@ .  IDENTIFIED BY PART C, THE EDIT POLICY MODULE (PROCESS 2.1) MUST
@ .  BE RERUN.  THEN, AS LONG AS CHANGES ARE NOT MADE TO THE TYPE
@ .  OF POLICIES USED IN PART A OR B, THE USER MAY THEN RETURN TO PART C
@ .  TO CONTINUE PROCESSING.
@ .
@ . - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
@ .
@ . - - - - - - - - - - - - - - - - - S O R T   M O S   D A T A   O N :
@ .                                          I S S U E   O N L Y
@ .                                      THIS FILE IS BN OUTPUT FROM 2-28
@ .
@ASG,A         PFMOSEXT37.
@ .
@ASG,T      XA:,///400
@ASG,T      XB:,///28C0
@ .
@SORT,ES
FILEIN=PFMOSEXT37.
FILEOUT=PFMOSEXT37.
KEY=1,4,S,A
@EOF
@ .
@FREE       PFMOSEXT37.
@FREE       XA.
@FREE       XB.
@ .
@ . - - - - - - - - - - - - - - - - - - - - ASSIGN THE ERROR OUTPUT PRINT FILE
@ .
@ASG,A       PFERROR13.
@ERS         PFERROR13.
@ . - - - - - - - - - - - - - - - - - - - - ASSIGN THE PARAMETER FILE
@ASG,A       PFPARAMET17.
@ED          PFPRIM-FILES.PARAMETER,PFPARAMET17.
@XI
@ . - - - - - - - - - - - - - - - - - - - - ASSIGN THE ISSUE POLICY FILE - - - -  INPUT
@ .                                        THIS FILE IS AN OUTPUT FROM 2-2A
@ASG,A       PFISSPOL33.
@ .
@ . - - - - - - - - - - - - - - - - - - - - ASSIGN THE INPUT JOB FILE - - - - INPUT
@ .                                        THIS FILE IS AN OUTPUT FROM 2-2B
@ .                                        AND SHOULD HAVE BEEN SORTED ON ISSUE
@ .                                        BY THE TIME THIS POINT IS REACHED
@ASG,A       PFMOSEXT37.
@ .
@ . - - - - - - - - - - - - - - - - - - - - ASSIGN THE EXTRA JOB FILE - - - - OUTPUT
@ASG,A       PFISSEXT38.
@ERS         PFISSEXT38.
@ .
@ . - - - - - - - - - - - - - - - - - - - - ASSIGN THE INPUT JAV FILE - - - - INPUT
@ .                                        THIS FILE IS AN OUTPUT FROM 2-2B
@ASG,A       PFJOBASVAL4C.
@ .
@ . - - - - - - - - - - - - - - - - - - - - ASSIGN THE OUTPUT JAV FILE - - - - OUTPUT
@ASG,A       PFJOBASVAL39.
@ERS         PFJOBASVAL39.
@ .
@ . - - - - - - - - - - - - - - - - - - - - ASSIGN THE AGGREGATED INDICATOR FILE
@ .                                        THIS FILE IS AN OUTPUT FROM 2-2A
@ASG,A       PFAGISIND9O.
@ .
@ . - - - - - - - - - - - - - - - - - - - - DEFINE THE FILE NUMBERS
@ .
@USE      17.,PFERROR13.
@USE      17.,PFPARAMET17.
@USE      33.,PFISSPOL33.
@USE      37.,PFMOSEXT37.
@USE      38.,PFISSEXT38.
@USE      39.,PFJOBASVAL39.
@USE      4C.,PFJOBASVAL4C.
@USE      9C.,PFAGISIND9C.
@ .
@ . - - - - - - - - - - - - - - - - - - - - ASSIGN TEMPORARY FILES FOR POLICY SORT
@ .
@ASG,T       XA.,///1C
@ASG,T       XB.,///7O
@ .
@XQT,F       PFPRIM-ABS,755APPPOL/2-2-C
  -D E B U G   F L A G
@ED,R        PFERROR13.
LNP
@XI
@FREE        17.
@FREE        17.
@FREE        33.
@FREE        37.
@FREE        38.
@FREE        39.
@FREE        4C.
@FREE        9C.
@FREE        XA.
@FREE        XB.
```

Figure 4-8. APPLY-POLICY/2-2-C Runstream

```
@HDG    SET-BASEVALU/2-3          SET BASE VALUES IN REST OF MOS DATA
@PFPRIM-ABS.TITLE  SETSBASE,VALUES,,2-3
@PFPRIM-ABS.TITLE  SETSBASE,VALUES,,2-3
@ .
@ .   T H I S   I S   P R O C E S S   2 . 3        (SET BASE VALUES)
@ .
@ .  THIS RUNSTREAM IS FOR SET BASE VALUE MINIMUMS, MAXIMUMS AND VALUES
@ .                P R O C E S S   2 - 3
@ .
@ .  APPLY POLICY PROCESS WAS BROKEN INTO SEVERAL PARTS TO ALLOW THE
@ .  USER MORE FLEXIBILITY.  WHEN ERRORS ARE FOUND BY ONE OF THE LATER
@ .  SUBROUTINES, IT IS NOT NECESSARILY REQUIRED THAT ALL PARTS ARE
@ .  REPEATED.  WHEN ERRORS ARE FOUND ONLY BY THE SET BASE VALUE PROCESSOR
@ .  THE USER MAY CORRECT THE VALUE ELEMENT OF THE PFPRIM-FILES FILE AND RERUN
@ .  ONLY THIS PORTION OF THE APPLY POLICY PROCESSOR.
@ .
@ . - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
@ .
@ . - - - - - - - - - - - - - - - - - S O R T   M O S   D A T A   O N :
@ .                                                 I S S U E   O N L Y
@ .
@ .                                  THIS FILE IS AN OUTPUT FROM 2-2-C
@ .
@ASG,A        PFISSEXT38.
@ .
@ASG,T     XA.,///400
@ASG,T     XB.,///2800
@ .
@SORT,ES
FILEIN=PFISSEXT38.
FILEOUT=PFISSEXT38.
KEY=1,4,S,A
@EOF
@ .
@FREE      PFISSEXT38.
@FREE      XA.
@FREE      XB.
@ .
@ . - - - - - - - - - - - - - - - - - - ASSIGN THE ERROR OUTPUT PRINT FILE
@ .
@ASG,A      PFERROR13.
@ERS        PFERROR13.
@ . - - - - - - - - - - - - - - - - - - ASSIGN THE PARAMETER FILE
@ASG,A      PFPARAMET17.
@ED         PFPRIM-FILES.PARAMETER,PFPARAMET17.
EXI
@ . - - - - - - - - - - - - - - - - - - ASSIGN THE VALUE FILE  - - -  INPUT
@ASG,A      PFVALUE19.
@ED  PFPRIM-FILES.VALUE,PFVALUE19.
@ .
@ . - - - - - - - - - - - - - - - - - - ASSIGN THE INPUT JOB FILE - - - - INPUT
@ .                                    THIS FILE IS AN OUTPUT FROM 2-2-C AND IS
@ .                                    SORTED NOW ON ISSUE
@ASG,A      PFISSEXT38.
@ .
@ . - - - - - - - - - - - - - - - - - - ASSIGN THE UNFILLED JOB FILE - - - - OUTPUT
@ASG,A      PFUNFILLEDSO.
@ .
@ .
@ . - - - - - - - - - - - - - - - - - - ASSIGN THE INPUT JAV FILE - - - -
@ .
@ASG,A      PFJOBASVAL39.
@ .
@ . - - - - - - - - - - - - - - - - - - ASSIGN THE OUTPUT JAV FILE - - - - OUTPUT
@ASG,A      PFJOBASVAL4O.
@ERS        PFJOBASVAL4O.
@ .
@USE     17.,PFERROR13.
@USE     17.,PFPARAMET17.
@USE     19.,PFVALUE19.
@USE     38.,PFISSEXT38.
@USE     39.,PFJOBASVAL39.
@USE     4O.,PFJOBASVAL4O.
@USE     5O.,PFUNFILLEDSO.
@ERS     5O.
@ .
@ . - - - - - - - - - - - - - - - - - - ASSIGN TEMPORARY FILES FOR VALUE FILE SORT
@ .
@ASG,T      XA.,///1O
@ASG,T      XB.,///7O
@ .
@XQT,F      PFPRIM-ABS.755SETBASE/2-3
 1-O E B U G      F L A G
@ED,R       PFERROR13.
LNP*
EXI
@ED,R     5O.
LNP*
@FREE       13.
@FREE       17.
@FREE       19.
@FREE       38.
@FREE       39.
@FREE       4O.
@FREE       51.
@FREE       XA.
@FREE       XB.
```

Figure 4-9. SET-BASEVALU/2-3 Runstream

```
@HOG    ASSIGNMENT/3        ASSIGNMENT PROCESSOR 3 0
@ASG,A          PFPRIM-ABS.
@
@ .
@ASG,A          PFPRIM-FILES.
@ASG,A          PFISSUE14.
@ED,U           PFPRIM-FILES.ISSUE, PFISSUE14.
EXI
@ASG,A          PFNETIN15.
@ASG,A          PFPARAMET17.
@ED             PFPRIM-FILES.PARAMETER,PFPARAMET17.
EXI
@ASG,A          PFNUMBPEOP21.
@ASG,A          PFJOBASVAL4C.
@ASG,A          PFEXCPEOP41,///5000
@ASG,A          PFUNFILLED51,///5000
@ASG,A          PFASSIGNED61,///5000
@ASG,T          PFNETOUT2,///5000
@ASG,T          PFNETWORK3,///5000
@ASG,T          PFNETWORK4,///5000
@ASG,T          PFNETWORK8,///5000
@ASG,T          PFNETWORK9,///5000
@
@ .         - - - - - - - - - - - -       SORT THE INPUT FILES
@ASG,T    XA.,///2000
@ASG,T    XB.,///11000
@SORT,ES
FILEIN=PFNUMBPEOP21.
FILEOUT=PFNUMBPEOP21.            . SORTED BY MOS
KEY=9,9,S,A
@END
@SORT,ES
FILEIN=PFJOBASVAL4C.
FILEOUT=PFJOBASVAL4C.            . SORTED BY MOS
KEY=10,9,S,A
KEY=1,8,S,A                      . SORTED BY DEMAND NAME
@END
@FREE    XA.
@FREE    XB.
@ .         - - - - - - - - - - - - -   ASSIGN REST OF FILES
@USE      2.,PFNETOUT2.
@USE      3.,PFNETWORK3.
@USE      4.,PFNETWORK4.
@USE      9.,PFNETWORK9.
@USE     14.,PFISSUE14.
@USE     15.,PFNETIN15.
@USE     17.,PFPARAMET17.
@USE     21.,PFNUMBPEOP21.
@USE     4C.,PFJOBASVAL4C.
@USE     41.,PFEXCPEOP41.
@ERS     41.
@USE     51.,PFUNFILLED51.
@ERS     51.
@USE     61.,PFASSIGNED61.
@ERS     61.
@XQT              PFPRIM-ABS.755ASSIGN/3
@ED,R 41.
LNP!
@ASG,A  PFUNFILLED50.
@ED,U 51.
ADD PFUNFILLED50.
LNP!
@ED,R 61.
LNP!
OMI
@ .         - - - - - - -                SORT THE OUTPUT FILES
@ASG,T XA.,///1100
@ASG,T XB.,///12000
@SORT,ES
FILEIN=PFEXCPEOP41.
FILEOUT=PFEXCPEOP41.
KEY=1,9,S,A                  . SORTED BY MOS
@END
@ED,U     PFUNFILLED51.
ADD PFUNFILLED50.
EXI
@SORT,ES
FILEIN=PFUNFILLED51.
FILEOUT=PFUNFILLED51.
KEY=10,9,S,A                  . SORTED BY MOS
KEY=1,8,S,A                   . SORTED BY DEMAND NAME
@END
@SORT,ES
FILEIN=PFASSIGNED61.
FILEOUT=PFASSIGNED61.
KEY=1,4,S,A                  . SORTED BY JOB'S FIRST 4 CHARACTERS
@END
@FREE 41.
@FREE 51.
@FREE 61.
```

Figure 4-10. ASSIGNMENTS/3 Runstream

```
@*          - - - - - - - - - - - - ASSIGN NETWORK WORKING FILES
@ASG,T PFNETOUT2.,///1000
@ASG,T PFNETWORK3.
@ASG,T PFNETWORK4.
@ASG,T PFNETWORK8.
@ASG,T PFNETWORK9.
@ASG,A PFNETIN15.,///2000
@*          - - - - - - - - - - - ISSUE FILE
@ASG,A PFISSUE14.
@ED    PFPRIM-FILES.ISSUE,PFISSUE14.
EXI
@*          - - - - - - - - - - - PARAMETER FILE
@ASG,A PFPARAMET17.
@ED    PFPRIM-FILES.PARAMETER,PFPARAMET17.
EXI
@ASG,A PFNUMBPEOP21.
@ASG,A PFJOBASVAL40.
@ASG,A PFEXCPEOP41.
@ASG,A PFEXPEOGRA43.
@*          - - - - - - - - - - - - - - - - ADD UNFILLED JOBS FROM POLICY PROCESSOR
@* ASG,A PFUNFILLED5C.
@ASG,A PFUNFILLED51.
@ASG,A PFUNFILLGR53.
@ASG,A PFASSIGNGR63.
@ERS       PFASSIGNGR63.
@USE        2.,PFNETOUT2.
@USE        3.,PFNETWORK3.
@USE        4.,PFNETWORK4.
@USE        8.,PFNETWORK8.
@USE        9.,PFNETWORK9.
@USE       14.,PFISSUE14.
@USE       15.,PFNETIN15.          . SCRATCH FILE,  INPUT TO MODEL
@USE       17.,PFPARAMET17.        . PARAMETER FILE
@USE       21., PFNUMBPEOP21       . CONTAINS NUMBER OF PEOPLE BY MOS
@USE       4C.,PFJOBASVAL40.       . INPUT JOB FILE
@USE       41.,PFEXCPEOP41.        . INPUT EXCESS PEOPLE FILE BY MOS
@USE       43.,PFEXPEOGRA43.       . OUTPUT EXCESS PEOPLE FILE BY GRADE
@USE       51.,PFUNFILLED51.       . INPUT EXCESS JOB FILE BY JOB
@USE       53., PFUNFILLGR53.      . OUTPUT UNFILLED JOB FILE - GRADE
@USE       63.,PFASSIGNGR63.       . OUTPUT JOB ASSIGNMENT FILE
@XQT    PFPRIM-ABS.755SUBST-GRD/4-1
@FREE  PFNETOUT2.
@FREE  PFNETWORK3.
@FREE  PFNETWORK4.
@FREE  PFNETWORK9.
@ED,R 43.
P!
@ED,R 53.
P!
@ED,R 63.
P!
@MSG,N
@ASG,T  XA.,///2000
@ASG,T  XB.,///10000
@SORT,ES
FILEIN=PFEXPEOGRA43.
FILEOUT=PFEXPEOGRA43.
KEY=1,9,S,A
@END
@SORT,ES
FILEIN=PFUNFILLGR53.
FILEOUT=PFUNFILLGR53.
KEY=10,9,S,A
KEY=1,9,S,A
@END
@SORT,ES
FILEIN=PFASSIGNGR63.
FILEOUT=PFASSIGNGR63.
KEY=1,4,S,A
KEY=10,9,S,A
@END
@FREE   XA.
@FREE   XB.
@FREE 43.
@FREE 53.
@FREE 63.
```

Figure 4-11. SUBST-GRD1ST/4-1 Runstream

```
@.          - - - - - - - - - - - ASSIGN NETWORK WORKING FILES
@ASG,T  PFNETOUT2.,///1000
@ASG,T  PFNETWORK3.
@ASG,T  PFNETWORK4.
@ASG,T  PFNETWORK8.
@ASG,T  PFNETWORK9.
@ASG,T  PFNETIN15.
@.          - - - - - - - - - - - ISSUE FILE
@ASG,A  PFPRIM-FILES.
@ASG,A  PFISSUE14.
@EQ     PFPRIM-FILES.ISSUE,PFISSUE14.
EXI
@.          - - - - - - - - - - - PARAMETER FILE
@ASG,A  PFPARAMET17.
@EQ     PFPRIM-FILES.PARAMETER,PFPARAMET17.
EXI
@FREE   PFPRIM-FILES.
@ASG,A  PFJOBASVAL40.
@ASG,A  PFEXCPEOP41.
@ASG,A  PFEXPEOMOS42.
@ASG,T  PFEXPEOMOS44.
@.          - - - - - - -       ADD UNFILLED JOBS FROM POLICY PROCESSOR
@ASG,A  PFUNFILLED50.
@ASG,A  PFUNFILLED51.
@ASG,A  PFUNFILLMOS2.
@ASG,T  PFUNFILLED54.
@ASG,A  PFASSIGNMO62.
@USE      2.,PFNETOUT2.
@USE      3.,PFNETWORK3.
@USE      4.,PFNETWORK4.
@USE      8.,PFNETWORK8.
@USE      9.,PFNETWORK9.
@USE     14.,PFISSUE14.
@USE     15.,PFNETIN15.          . SCRATCH FILE,  INPUT TO MODEL
@USE     17.,PFPARAMET17.        . PAPAMETER FILE
@USE     21.,  PFPRIM21          . CONTAINS NUMBER OF PEOPLE BY MOS
@USE     40.,PFJOBASVAL40.       . INPUT JOB FILE
@USE     41.,PFEXCPEOP41.        . INPUT EXCESS PEOPLE FILE BY MOS
@USE     42.,PFEXPEOMOS42.       . OUTPUT EXCESS PEOPL FILE BY MOS
@USE     44.,PFEXPEOMOS44.       . SCRATCH EXCESS PEOPL FILE BY MOS
@USE     51.,PFUNFILLED51.       . INPUT EXCESS JOB FILE BY JOB
@USE     52.,  PFUNFILLMO52.     . OUTPUT EXCESS JOB FILE BY JOB
@USE     54.,PFUNFILLED54.       . SCRATCH EXCESS PEOPL FILE BY MOS
@USE     62.,PFASSIGNMO62.       . OUTPUT JOB ASSIGNMENT FILE
@ASG,A  PFPRIM-ABS.
@FREE   PFPRIM-ABS.
@XQT PFPRIM-ABS.755SUBST-MOS/4-2
@FREE   PFNETOUT2.
@FREE   PFNETWORK3.
@FREE   PFNETWORK4.
@FREE   PFNETWORK9.
@ED,R 42.
P!
@ED,R 52.
P!
@ED,R 62.
P!
@ASG,T   XA.,///2000
@ASG,T   XB.,///10000
@SORT,ES
FILEIN=PFEXPEOMOS42.
FILEOUT=PFEXPEOMOS42.
KEY=1,9,S,A
@END
@SORT,ES
FILEIN=PFUNFILLMO52.
FILEOUT=PFUNFILLMO52.
KEY=10,9,S,A
@END
@SORT,ES
FILEIN=PFASSIGNMO62.
FILEOUT=PFASSIGNMO62.
KEY=1,4,S,A
@END
@FREE    XA.
@FREE    XB.
```

Figure 4-12. SUBST-MOS1ST/4-2 Runstream

```
@.          - - - - - - - - - - - ASSIGN NETWORK WORKING FILES
@ASG,A  PFNETOUT2.,///1000
@ASG,T  PFNETWORK3.
@ASG,T  PFNETWORK4.
@ASG,T  PFNETWORK8.
@ASG,T  PFNETWORK9.
@ASG,A  PFNETIN15.
@.          - - - - - - - - - - - ISSUE FILE
@ASG,A  PFISSUE14.
@ED     PFPRIM-FILES.ISSUE,PFISSUE14.
EXI
@.          - - - - - - - - - - - PARAMETER FILE
@ASG,A  PFPARAMET17.
@ED     PFPRIM-FILES.PARAMETER,PFPARAMET17.
EXI
@ASG,A  PFJOBASVAL40.
@ASG,A  PFEXPEOMOS42.          . INPUT EXCESS PEOPLE FILE
@ASG,A  PFEXPEOGRA43.          . OUTPUT EXCESS PEOPLE FILE
@ASG,A  PFUNFILLMO52.          . INPUT UNFILLED JOB FILE
@ASG,A  PFUNFILLGR53.          . OUTPUT UNFILLED JOB FILE
@ASG,A  PFASSIGNGR63.
@USE      2.,PFNETOUT2.
@USE      3.,PFNETWORK3.
@USE      4.,PFNETWORK4.
@USE      8.,PFNETWORK8.
@USE      9.,PFNETWORK9.
@USE     14.,PFISSUE14.
@USE     15.,PFNETIN15.              . SCRATCH FILE,  INPUT TO MODEL
@USE     17.,PFPARAMET17.            . PARAMETER FILE
@USE     21., PFNUMBPEOP21           . CONTAINS NUMBER OF PEOPLE BY MOS
@USE     40.,PFJOBASVAL40.           . INPUT JOB FILE
@USE     42.,PFEXPEOMOS42.           . OUTPUT EXCESS PEOPL FILE BY MOS
@USE     43.,PFEXPEOGRA43.           . OUTPUT EXCESS PEOPL FILE BY GRADE
@USE     52., PFUNFILLMO52.          . OUTPUT EXCESS JOB FILE BY JOB
@USE     53., PFUNFILLGR53.          . OUTPUT UNFILLED JOB FILE - GRADE
@USE     63.,PFASSIGNGR63.           . OUTPUT JOB ASSIGNMENT FILE
@XQT PFNET.755SUBST-GRD/4-1
@FREE   PFNETOUT2.
@FREE   PFNETWORK3.
@FREE   PFNETWORK4.
@FREE   PFNETWORK9.
@ED,R 43.
P;
@ED,R 53.
P;
@ED,R 63.
P;
@ASG,T   XA.,///2000
@ASG,T   XB.,///10000
@SORT,ES
FILEIN=PFEXPEOGRA43.
FILEOUT=PFEXPEOGRA43.
KEY=1,9,S,A
@END
@SORT,ES
FILEIN=PFUNFILLGR53.
FILEOUT=PFUNFILLGR53.
KEY=10,9,S,A
KEY=1,9,S,A
@END
@SORT,ES
FILEIN=PFASSIGNGR63.
FILEOUT=PFASSIGNGR63.
KEY=1,4,S,A
@END
@FREE    XA.
@FREE    XB.
@FREE 43.
@FREE 53.
@FREE 63.
```

Figure 4-13. SUBST-GRD2ND/4-1 Runstream

```
@.          - - - - - - - - - - - ASSIGN NETWORK WORKING FILES
@ASG,A  PFNETOUT2.,///1000
@ASG,T  PFNETWORK3.
@ASG,T  PFNETWORK4.
@ASG,T  PFNETWORK8.
@ASG,T  PFNETWORK9.
@ASG,A  PFNETIN15.
@.          - - - - - - - - - - ISSUE FILE
@ASG,A  PFISSUE14.
@ED     PFPRIM-FILES.ISSUE,PFISSUE14.
EXI
@.          - - - - - - - - - - PARAMETER FILE
@ASG,A  PFPARAMET17.
@ED     PFPRIM-FILES.PARAMETER,PFPARAMET17.
EXI
@ASG,A  PFJOBASVAL40.
@ASG,A  PFEXPEOMOS42.
@ASG,A  PFEXPEOGRA43.
@ASG,T  PFEXPEOMOS44.
@ASG,A  PFUNFILLMO52.
@ASG,A  PFUNFILLGR53.
@ASG,T  PFUNFILLED54.
@ASG,A  PFASSIGNMO62.
@USE      2.,PFNETOUT2.
@USE      3.,PFNETWORK3.
@USE      4.,PFNETWORK4.
@USE      8.,PFNETWORK8.
@USE      9.,PFNETWORK9.
@USE     14.,PFISSUE14.
@USE     15.,PFNETIN15.          . SCRATCH FILE,  INPUT TO MODEL
@USE     17.,PFPARAMET17.        . PARAMETER FILE
@USE     21., PFPRIM21           . CONTAINS NUMBER OF PEOPLE BY MOS
@USE     40.,PFJOBASVAL40.       . INPUT JOB FILE
@USE     42.,PFEXPEOMOS42.       . OUTPUT EXCESS PEOPL FILE BY MOS
@USE     43.,PFEXPEOGRA43.       . OUTPUT EXCESS PEOPL FILE BY GRADE
@USE     44.,PFEXPEOMOS44.       . SCRATCH EXCESS PEOPL FILE BY MOS
@USE     52., PFUNFILLMO52.      . OUTPUT EXCESS JOB FILE BY JOB
@USE     53., PFUNFILLGR53.      . OUTPUT UNFILLED JOB FILE - GRADE
@USE     54.,PFUNFILLED54.       . SCRATCH EXCESS PEOPL FILE BY MOS
@USE     62.,PFASSIGNMO62.       . OUTPUT JOB ASSIGNMENT FILE
@XQT PFNET.755SUBST-MOS/4-2
@FREE   PFNETOUT2.
@FREE   PFNETWORK3.
@FREE   PFNETWORK4.
@FREE   PFNETWORK9.
@ED,R 42.
P!
@ED,R 52.
P!
@ED,R 62.
P!
@ASG,T  XA.,///2000
@ASG,T  XB.,///10000
@SORT,ES
FILEIN=PFEXPEOMOS42.
FILEOUT=PFEXPEOMOS42.
KEY=1,9,S,A
@END
@SORT,ES
FILEIN=PFUNFILLMO52.
FILEOUT=PFUNFILLMO52.
KEY=10,9,S,A
KEY=1,9,S,A
@END
@SORT,ES
FILEIN=PFASSIGNMO62.
FILEOUT=PFASSIGNMO62.
KEY=1,4,S,A
@END
@FREE    XA.
@FREE    XB.
```

Figure 4-14. SUBST-MOS2ND/4-2 Runstream

```
@HDG  READINESS/5                    READINESS PROCESSOR 5 D
@TITLE READINESS,5$-$C,READINESS,PROCESSOR
@TITLE READINESS,5$-$C,READINESS,PROCESSOR
@ .
@ .   THIS RUNSTREAM EXECUTES THE READINESS PROCESSOR
@ .
@ .              - - - - - - - - -   ASSIGN ISSUE FILE
@ASG,A           PFISSUE14.
@ED              PFPRIM-FILES.ISSUE,PFISSUE14.
EXI
@ .              - - - - - - - - - - ASSIGN PARAMETER FILE
@ASG,A           PFPARAMET17.
@ED              PFPRIM-FILES.PARAMETER,PFPARAMET17.
EXI
@ .              - - - - - - - - - - ASSIGN BASE VALUE FILE
@ASG,A           PFVALUE19.
@ . ED              PFPRIM-FILES.VALUE,PFVALUE19.
@ . EXI
@ .              - - - - - - - - - - -  ASSIGN INPUT FILES
@ASG,A           PFMOSISSUE31.
@ASG,A           PFASSIGNED61.
@ASG,A           PFASSIGNMO62.
@ASG,A           PFASSIGNGR63.
@ .              - - - - - - - - - - -  ASSIGN OUTPUT FILES
@ASG,A           PFASSIGNED64.,///10000
@ASG,A           PFASSIGNMO65.,///5000
@ASG,A           PFASSIGNGR66.,///5000
@ASG,A           PFASSIGNED67.,///10000
@ASG,A           PFISSREADI7D.,///5000
@ASG,A           PFISSPERCR72.,///5000
@ASG,A           PFGRAREADI73.,///10000
@ASG,A           PFMOSREADI74.,///10000
@ .              - - - - - - - - - - ATTACH FILE NUMBERS
@USE        14.,PFISSUE14.
@USE        17.,PFPARAMET17.
@USE        19.,PFVALUE19.
@USE        31.,PFMOSISSUE31.
@USE        61.,PFASSIGNED61.
@USE        62.,PFASSIGNMO62.
@USE        63.,PFASSIGNGR63.
@USE        64.,PFASSIGNED64.
@USE        65.,PFASSIGNMO65.
@USE        66.,PFASSIGNGP66.
@USE        67.,PFASSIGNED67.
@USE        7D.,PFISSREADI7D.
@USE        72.,PFISSPERCR72.
@USE        73.,PFGRAREADI73.
@USE        74.,PFMOSREADI74.
@ERS        64.
@ERS        65.
@ERS        66.
@ERS        67.
@ERS        7D.
@ERS        72.
@ERS        73.
@ERS        74.
@XQT        PFPRIM-ABS.755READINESS/5
@ED,R          64.
LNP!
@ED,R          65.
LNP!
@ED,R          66.
LNP!
@ED,R          67.
LNP!
@ED,R          7D.
LNP!
@ED,R          72.
LNP!
@ED,R          73.
LNP!
@ED,R          74.
LNP!
OMI
@FREE  64.
@FREE  65.
@FREE  66.
@FREE  67.
@FREE  7D.
@FREE  72.
@FREE  73.
@FREE  74.
```

Figure 4-15. READINESS Runstream

```
@MDG   P R I M   R E P O R T S
@PFPRIM-ABS.TITLE,D FORMATTED,REPORTS,PROCESS$6,P$R$I$M,REPORTS
@PFPRIM-ABS.TITLE,D FORMATTED,REPORTS,PROCESS$6,P$R$I$M,REPORTS
@ . - - - - - - - - - - - - - -   ASSIGN TEMPORARY FILES FOR INTERNAL SORTS
@ASG,T  XP.,///560
@ASG,T  XA.,///80
@ . - - - - - - - - - - - - - -   ASSIGN THE ERROR AND ISSUE FILES
@ASG,A        PFERROR13.
@USE     13.,PFERROR13.
@ASG,A        PFISSUE14.
@USE     14.,PFISSUE14.
@ . - - - - - - - - - - - - - -   ASSIGN THE EXCESS PEOPLE FILES
@ASG,A        PFEXCPEOP41.
@USE     41.,PFEXCPEOP41.
@ASG,A        PFEXPEOMOS42.
@USE     42.,PFEXPEOMOS42.
@ASG,A        PFEXPEOGRA43.
@USE     43.,PFEXPEOGRA43.
@ . - - - - - - - - - - - - - -   ASSIGN THE UNFILLED JOB FILES
@ASG,A        PFUNFILLED50.
@USE     50.,PFUNFILLED50.
@ASG,A        PFUNFILLED51.
@USE     51.,PFUNFILLED51.
@ASG,A        PFUNFILLMO52.
@USE     52.,PFUNFILLMO52.
@ASG,A        PFUNFILLGR53.
@USE     53.,PFUNFILLGR53.
@ . - - - - - - - - - - - - - -   ASSIGN THE READINESS FILES
@ASG,A        PFASSIGNED64.
@USE     64.,PFASSIGNED64.
@ASG,A        PFASSIGNED67.
@USE     67.,PFASSIGNED67.
@ASG,A        PFISSREADI70.
@USE     70.,PFISSREADI70.
@ASG,A        PFISSPERCR72.
@USE     72.,PFISSPERCR72.
@ASG,A        PFGRAREADI73.
@USE     73.,PFGRAREADI73.
@ASG,A        PFMOSREADI74.
@USE     74.,PFMOSREADI74.
@ . - - - - - - - - - - - - - -   ASSIGN THE REPORT REQUEST FILE
@ASG,A        PFREPORT93.
@USE     93.,PFREPORT93.
@ED           PFPRIM-FILES.REPORT,PFREPORT93.
@ . - - - - - - - - - - - - - -   MAKE SURE THE ABSOLUTE FILE IS AVAILABLE
@ASG,A        PFPRIM-ABS.
@FREE         PFPRIM-ABS.
@XQT   PFPPIM-ABS.755REPORT/6
@ADD   PFPPIM-FILES.PARAMETER
```

Figure 4-16. REPORT/6 Runstream

```
@ .
@ .   C O M P I L E   &   M A P   P R O C E S S O R   1 - 1 - 1
@ . - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
@ .
@ .   THIS RUNSTREAM COMPILES THE PROGRAM "MAKE3DIGUIC", AND MAKES THE
@ .   ABSOLUTE ELEMENT NEEDED FOR RUNNING PPOCESS 1.1.1
@ . - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
@ .
@ASG,A        PFPRIM-ABS.
@ASG,A        PFPRIM-REL.
@ASG,A        PFPRIM-SYMB.
@ .       - - - COMPILE THE MAIN (AND ONLY) PROGRAM
@ .
@FIN,FS   PFPRIM-SYMB.ROLUIC/1-1-1,PFPRIM-REL.ROLUIC/1-1-1
@EOF
@ .       - - - MAKE THE ABSOLUTE ELEMENT (EXECUTABLE PROGRAM)
@ .
@MAP     ,PFPRIM-ABS.755ROLUIC/1-1-1
IN        PFPRIM-REL.ROLUIC/1-1-1
END
@ .
@PACK         PFPRIM-ABS.
@PACK         PFPRIM-REL.
@PACK         PFPRIM-SYMB.
@ .
@ .   THE RUNSTREAM FOR THE ACTUAL RUN IS:
@ .
@ .                       PFPRIM-RUN.MAKE3DIGUIC/1-1-1
@ .
```

Figure 4-17. ROLUIC/1-1-1 Compile amd Map Runstream

```
@ .
@ .    C O M P I L E    A N D    M A P    P R O C E S S O R    1 - 1 - 2
@ .
@ .    THE ABSOLUTE ELEMENT CREATED BY THIS RUNSTREAM WILL AGGREGATE
@ .
@ .    THE MOS DATA FROM THE 6-DIGIT UIC LEVEL TO THE 3-DIGIT UIC LEVEL
@ .
@ . - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
@ .
@ .      - - - ASSIGN THE FILES FOR THE SYMBOLIC (FORTRAN) AND RELATIVE
@ .            VERSIONS OF THE ROLMOS PROGRAM (PROCESSOR 1-1-2)
@ .
@ASG,A    PFPRIM-REL.
@ASG,A    PFPRIM-SYMB.
@ .
@ .    - - - COMPILE THE MAIN PROGRAMS
@ .
@FTN,F     PFPRIM-SYMB.SEPARATE/1-1-2A,PFPRIM-REL.SEPARATE/1-1-2A
@FTN,F     PFPRIM-SYMB.ROLMOS/1-1-2,PFPRIM-REL.ROLMOS/1-1-2
@PACK      PFPRIM-SYMB.
@FREE      PFPRIM-SYMB.
@ .
@ .    - - - ASSIGN THE FILE FOR THE ABSOLUTE ELEMENTS AND THE UTILITY
@ .          PROGRAM FILE
@ .
@ASG,A    PFPRIM-ABS.
@ .
@ .    - - - MAKE THE ABSOLUTE ELEMENTS (MAP THE PROGRAMS)
@ .
@MAP         ,PFPRIM-ABS.755SEPARATE/1-1-2A
IN            PFPRIM-REL.SEPARATE/1-1-2A
IN            PFPRIM-REL.CHKNUL
IN            PFPRIM-REL.WRTERR
END
@ .
@MAP        ,PFPRIM-ABS.755ROLMOS/1-1-2
IN           PFPRIM-REL.ROLMOS/1-1-2
IN           PFPRIM-REL.WRTERR
IN           PFPRIM-REL.ICHINT
IN           PFPRIM-REL.F2FRT
END
@ .
@ .    - - - FREE THE FILES USED AND STOP
@ .
@PACK      PFPRIM-REL.
@FREE      PFPRIM-REL.
@PACK      PFPRIM-ABS.
@FREE      PFPRIM-ABS.
```

Figure 4-18. ROLMOS/1-1-2 Compile and Map Runstream

```
@TOP:HDG    MAP-SETISSUE/1-2          PROCESSOR 1-2        COMPILE AND MAP
@ .
@ASG,A     PFPRIM-ABS.
@ASG,A     PFPRIM-REL.
@ASG,A     PFPRIM-SYMB.
@ .
@ .
@ .      COMPILE PROGRAMS USED ONLY BY PROCESS 1.2, SET ISSUE
@ .
@PDP,F       PFPRIM-SYMB.PPIM-PROCS,PFPRIM-SYMB.MAIN-PROCS
@ .
@FIN,F       PFPRIM-SYMB.DRIVE1/1-2,PFPRIM-REL.DRIVE1/1-2
@FIN,F       PFPRIM-SYMB.SETTPS,PFPRIM-REL.SETTPS
@FIN,F       PFPRIM-SYMB.SETUIC,PFPRIM-REL.SETUIC
@FIN,F       PFPRIM-SYMB.SETASG,PFPRIM-REL.SETASG
@FIN,F       PFPRIM-SYMB.SETMOS,PFPRIM-REL.SETMOS
@FIN,F       PFPRIM-SYMB.AGGUIC,PFPRIM-REL.AGGUIC
@FIN,F       PFPRIM-SYMB.CHKNUL,PFPRIM-REL.CHKNUL
@FIN,F       PFPRIM-SYMB.DOIT12,PFPRIM-REL.DOIT12
@FIN,F       PFPRIM-SYMB.DOITNO,PFPRIM-REL.DOITNO
@FIN,F       PFPRIM-SYMB.ISSFIN,PFPRIM-REL.ISSFIN
@ .
@ .      MAKE THE EXECUTABLE ELEMENT
@ .
@MAP    ,PFPRIM-ABS.755SETISSUE/1-2
IN       PFPRIM-REL.DRIVE1/1-2
IN       PFPRIM-REL.SETTPS
IN       PFPRIM-REL.SETUIC
IN       PFPRIM-REL.SETASG
IN       PFPRIM-REL.SETMOS
IN       PFPRIM-REL.AGGUIC
IN       PFPRIM-REL.CHKNUL
IN       PFPRIM-REL.DOIT12
IN       PFPRIM-REL.DOITNO
IN       PFPRIM-REL.ISSFIN
IN       PFPRIM-REL.WRTERR
IN       PFPRIM-REL.GETDAY
IN       PFPRIM-REL.GETWCT
IN       PFPRIM-REL.ICHINT
END
@ .
@ .      PACK ALL THREE FILES INVOLVED
@ .
@PACK       PFPRIM-SYMB.
@PACK       PFPRIM-ABS.
@PACK       PFPRIM-REL.
```

Figure 4-19. SETISSUE/1-2 Compile and Map Runstream

```
@HDG   MAP-AGGMOS/1-3      PROCESSOR 1-3          COMPILE AND MAP
@ .
@ .   C O M P I L E   &   M A P   P R O C E S S O R   1 - 3
@ .
@ASG,A     PFPRIM-ABS.
@ASG,A     PFPRIM-REL.
@ASG,A     PFPRIM-SYMB.
@ .
@ . - - - - - - - - - - - - - - - - COMPILE THE AGGREGATE MOS PROGRAMS
@ .
@TOP:FTN,F       PFPRIM-SYMB.DRIVE1/1-3,PFPRIM-REL.DRIVE1/1-3
@FTN,F       PFPRIM-SYMB.AGGMOS,PFPRIM-REL.AGGMOS
@ .
@ . - - - - - - - - - - - - - - - - MAP THE AGGMOS PROGRAMS & UTILITIES
@ .
@MAP            ,PFPRIM-ABS.755AGGMOS/1-3
IN           PFPRIM-REL.DRIVE1/1-3
IN           PFPRIM-REL.AGGMOS
IN           PFPRIM-REL.GETDAY
IN           PFPRIM-REL.GETWCT
IN           PFPRIM-REL.WRTERR
IN           PFPRIM-REL.ICHINT
IN           PFPRIM-REL.F2FRT
END
@ .
@PACK      PFPRIM-ABS.
@PACK      PFPRIM-REL.
@PACK      PFPRIM-SYMB.
```

Figure 4-20. AGGMOS/1-3 Compile and Map Runstream

```
@HDG    MAP-EDIT-POLICY/2-1        PROCESSOR 2-1        COMPILE AND MAP
@ .
@ .   C O M P I L E   &   M A P   P R O C E S S O R   2 - 1
@ .
@HDG,U  EDIT-POLICY/2-1  - - -  COMPILE AND MAP EDIT POLICY
@PFPRIM-ABS.TITLE  COMPILE,AND$MAP,EDIPOL,EDIT,POLICY
@PFPRIM-ABS.TITLE  COMPILE,AND$MAP,EDIPOL,EDIT,POLICY
@ .
@ASG,A     PFPRIM-ABS.
@ASG,A     PFPRIM-REL.
@ASG,A     PFPRIM-SYMB.
@TOP:FTN,F       PFPRIM-SYMB.DRIVE2/1,PFPRIM-REL.DRIVE2/1
@FTN,F     PFPRIM-SYMB.EDIPOL/2-1,PFPRIM-REL.EDIPOL/2-1
@MAP       ,PFPRIM-ABS.755EDIPOL
IN              PFPRIM-REL.DRIVE2/1
IN              PFPRIM-REL.EDIPOL/2-1
IN              PFPRIM-REL.GETDAY
IN              PFPRIM-REL.GETWCT
IN              PFPRIM-REL.WRTERR
IN              PFPRIM-REL.F2FRT
IN              PFPRIM-REL.ICHINT
IN              PFPRIM-REL.RCHFLT
END
@ .
@PACK      PFPRIM-ABS.
@PACK      PFPRIM-REL.
@PACK      PFPRIM-SYMB.
```

Figure 4-21. EDIT-POLICY/2-1 Compile and Map Runstream

```
@HDG     MAP-APPLY-POLICY/2-2-A          PROCESSOR 2-2-A          COMPILE
@ .
@ .   C O M P I L E   &   M A P   P R O C E S S O R   2 - 2 / A
@ .
@PFPRIM-ABS.TITLE,D     COMPILE,AND$MAP,APPLY,POLICY-2A
@PFPRIM-ABS.TITLE,D     COMPILE,AND$MAP,APPLY,POLICY-2A
@ .
@ASG,A     PFPRIM-ABS.
@ASG,A     PFPRIM-REL.
@ASG,A     PFPRIM-SYMB.
@ .
@PDP,F       PFPRIM-SYMB.PRIM-PROCS,PFPRIM-REL.PRIM-PROCS
@FTN,F      PFPRIM-SYMB.DRIVE2/2-A,PFPRIM-REL.DRIVE2/2-A
@FTN,F      PFPRIM-SYMB.APP22A/2-2-A,PFPRIM-REL.APP22A/2-2-A
@FTN,F      PFPRIM-SYMB.AGISPO,PFPRIM-REL.AGISPO
@FTN,F      PFPRIM-SYMB.CONPOL,PFPRIM-REL.CONPOL
@FTN,F      PFPRIM-SYMB.DOIT,PFPRIM-REL.DOIT
@FTN,F      PFPRIM-SYMB.FINDAT,PFPRIM-REL.FINDAT
@FTN,F      PFPRIM-SYMB.GETDAY,PFPRIM-REL.GETDAY
@FTN,F      PFPRIM-SYMB.GETJOB,PFPRIM-REL.GETJOB
@FTN,F      PFPRIM-SYMB.GETPAR,PFPRIM-REL.GETPAR
@FTN,F      PFPRIM-SYMB.GETPOL,PFPRIM-REL.GETPOL
@FTN,F      PFPRIM-SYMB.GETWCT,PFPRIM-REL.GETWCT
@FTN,F      PFPRIM-SYMB.ISMO,PFPRIM-REL.ISMO
@FTN,F      PFPRIM-SYMB.MAKAGG,PFPRIM-REL.MAKAGG
@FTN,F      PFPRIM-SYMB.MAKIND,PFPRIM-REL.MAKIND
@FTN,F      PFPRIM-SYMB.MATPOL,PFPRIM-REL.MATPOL
@FTN,F      PFPRIM-SYMB.NXTISS,PFPRIM-REL.NXTISS
@FTN,F      PFPRIM-SYMB.SETPOL,PFPRIM-REL.SETPOL
@FTN,F      PFPRIM-SYMB.SRTPOL,PFPRIM-REL.SRTPOL
@FTN,F      PFPRIM-SYMB.WRTEXT,PFPRIM-REL.WRTEXT
@FTN,F      PFPRIM-SYMB.WRTVAL,PFPRIM-REL.WRTVAL
@FTN,F      PFPRIM-SYMB.ZEROIN,PFPRIM-REL.ZEROIN
@FTN,F      PFPRIM-SYMB.ZILCH,PFPRIM-REL.ZILCH
@MAP          ,PFPRIM-ABS.755APPPOL/2-2-A
IN        PFPRIM-REL.DRIVE2/2-A
IN        PFPRIM-REL.APP22A/2-2-A
IN        PFPRIM-REL.AGISPO
IN        PFPRIM-REL.CONPOL
IN        PFPRIM-REL.DOIT
IN        PFPRIM-REL.FINDAT
IN        PFPRIM-REL.GETDAY
IN        PFPRIM-REL.GETJOB
IN        PFPRIM-REL.GETPAR
IN        PFPRIM-REL.GETPOL
IN        PFPRIM-REL.GETWCT
IN        PFPRIM-REL.ISMO
IN        PFPRIM-REL.MAKAGG
IN        PFPRIM-REL.MAKIND
IN        PFPRIM-REL.MATPOL
IN        PFPRIM-REL.NXTISS
IN        PFPRIM-REL.SETPOL
IN        PFPRIM-REL.SRTPOL
IN        PFPRIM-REL.WRTEXT
IN        PFPRIM-REL.WRTVAL
IN        PFPRIM-REL.ZEROIN
IN        PFPRIM-REL.ZILCH
IN        PFPRIM-REL.WRTERR
IN        PFPRIM-REL.BSERCH
IN        PFPRIM-REL.F2FRT
IN        PFPRIM-REL.ICHINT
IN        PFPRIM-REL.RCHFLT
END
@PACK       PFPRIM-ABS.
@PACK       PFPRIM-REL.
@PACK       PFPRIM-SYMB.
@FREE       PFPRIM-ABS.
@FREE       PFPRIM-REL.
@FREE       PFPRIM-SYMB.
```

Figure 4-22. APPLY-POLICY/2-2-A Compile and Map Runstream

```
@TOP:HDG    MAP-APPLY-POLICY/2-2-B      PROCESSOR 2-2-B     COMPILE
@ .
@ .   C O M P I L E   &   M A P   P R O C E S S O R   2 - 2 / B
@ .
@PFPRIM-ABS.TITLE  COMPILE,AND$MAP,APPPOL22B,APPLY,POLICY-2B
@PFPRIM-ABS.TITLE  COMPILE,AND$MAP,APPPOL22B,APPLY,POLICY-2B
@ .
@ASG,A     PFPRIM-ABS.
@ASG,A     PFPRIM-REL.
@ASG,A     PFPRIM-SYMB.
@ .
@PDP,F          PFPRIM-SYMB.PRIM-PROCS,PFPRIM-REL.PRIM-PROCS
@FTN,F     PFPRIM-SYMB.DRIVE2/2-B,PFPRIM-REL.DRIVE2/2-B
@FTN,F     PFPRIM-SYMB.APP22B/2-2-B,PFPRIM-REL.APP22B/2-2-B
@FTN,F     PFPRIM-SYMB.CONPOL,PFPRIM-REL.CONPOL
@FTN,F     PFPRIM-SYMB.DOIT,PFPRIM-REL.DOIT
@FTN,F     PFPRIM-SYMB.FINDAT,PFPRIM-REL.FINDAT
@FTN,F     PFPRIM-SYMB.FINVAL,PFPRIM-REL.FINVAL
@FTN,F     PFPRIM-SYMB.GETJOB,PFPRIM-REL.GETJOB
@FTN,F     PFPRIM-SYMB.GETPAR,PFPRIM-REL.GETPAR
@FTN,F     PFPRIM-SYMB.GETPOL,PFPRIM-REL.GETPOL
@FTN,F     PFPRIM-SYMB.GETVAL,PFPRIM-REL.GETVAL
@FTN,F     PFPRIM-SYMB.ISMO,PFPRIM-REL.ISMO
@FTN,F     PFPRIM-SYMB.MAKAGG,PFPRIM-REL.MAKAGG
@FTN,F     PFPRIM-SYMB.MAKIND,PFPRIM-REL.MAKIND
@FTN,F     PFPRIM-SYMB.MATPOL,PFPRIM-REL.MATPOL
@FTN,F     PFPRIM-SYMB.NXTISS,PFPRIM-REL.NXTISS
@FTN,F     PFPRIM-SYMB.SETPOL,PFPRIM-REL.SETPOL
@FTN,F     PFPRIM-SYMB.SRTPOL,PFPRIM-REL.SRTPOL
@FTN,F     PFPRIM-SYMB.WRTEXT,PFPRIM-REL.WRTEXT
@FTN,F     PFPRIM-SYMB.WRTVAL,PFPRIM-REL.WRTVAL
@FTN,F     PFPRIM-SYMB.ZEROIN,PFPRIM-REL.ZEROIN
@FTN,F     PFPRIM-SYMB.ZILCH,PFPRIM-REL.ZILCH
@MAP            ,PFPRIM-ABS.755APPPOL/2-2-B
IN         PFPRIM-REL.DRIVE2/2-B
IN         PFPRIM-REL.APP22B/2-2-B
IN         PFPRIM-REL.AGISPO
IN         PFPRIM-REL.CONPOL
IN         PFPRIM-REL.DOIT
IN         PFPRIM-REL.FINDAT
IN         PFPRIM-REL.FINVAL
IN         PFPRIM-REL.GETDAY
IN         PFPRIM-REL.GETJOB
IN         PFPRIM-REL.GETPAR
IN         PFPRIM-REL.GETPOL
IN         PFPRIM-REL.GETWCT
IN         PFPRIM-REL.GETVAL
IN         PFPRIM-REL.ISMO
IN         PFPRIM-REL.MAKAGG
IN         PFPRIM-REL.MAKIND
IN         PFPRIM-REL.MATPOL
IN         PFPRIM-REL.NXTISS
IN         PFPRIM-REL.SETPOL
IN         PFPRIM-REL.SRTPOL
IN         PFPRIM-REL.WRTEXT
IN         PFPRIM-REL.WRTVAL
IN         PFPRIM-REL.ZEROIN
IN         PFPRIM-REL.ZILCH
IN         PFPRIM-REL.WRTERR
IN         PFPRIM-REL.BSERCH
IN         PFPRIM-REL.F2FRT
IN         PFPRIM-REL.ICHINT
IN         PFPRIM-REL.RCHFLT
END
@ .
@PACK      PFPRIM-ABS.
@PACK      PFPRIM-REL.
@PACK      PFPRIM-SYMB.
@FREE      PFPRIM-ABS.
@FREE      PFPRIM-REL.
@FREE      PFPRIM-SYMB.
```

Figure 4-23. APPLY-POLICY/2-2-B Compile and Map Runstream

```
@TOP:HDG    MAP-APPLY-POLICY/2-2-C     PROCESSOR 2-2-C    COMPILE
@ .
@ .   C O M P I L E   &   M A P   P R O C E S S O R   2 - 2 / C
@ .
@PFPRIM-ABS.TITLE  COMPILE,AND$MAP,APPPOL22C,ISSUE,POLICY-2C
@PFPRIM-ABS.TITLE  COMPILE;AND$MAP;APPPOL22C;ISSUE;POLICY-2C
@ .
@ASG,A     PFPRIM-ABS.
@ASG,A     PFPRIM-REL.
@ASG,A     PFPRIM-SYMB.
@ .
@POP,F     PFPRIM-SYMB.PRIM-PROCS,PFPRIM-SYMB.PRIM-PROCS
@FTN,F     PFPRIM-SYMB.DRIVE2/2-C,PFPRIM-REL.DRIVE2/2-C
@FTN,F     PFPRIM-SYMB.APP22C/2-2-C,PFPRIM-REL.APP22C/2-2-C
@FTN,F     PFPRIM-SYMB.CONPOL,PFPRIM-REL.CONPOL
@FTN,F     PFPRIM-SYMB.DOIT,PFPRIM-REL.DOIT
@FTN,F     PFPRIM-SYMB.FINDAT,PFPRIM-REL.FINDAT
@FTN,F     PFPRIM-SYMB.FINVAL,PFPRIM-REL.FINVAL
@FTN,F     PFPRIM-SYMB.GETDAY,PFPRIM-REL.GETDAY
@FTN,F     PFPRIM-SYMB.GETJOB,PFPRIM-REL.GETJOB
@FTN,F     PFPRIM-SYMB.GETPAR,PFPRIM-REL.GETPAR
@FTN,F     PFPRIM-SYMB.GETPOL,PFPRIM-REL.GETPOL
@FTN,F     PFPRIM-SYMB.GETVAL,PFPRIM-REL.GETVAL
@FTN,F     PFPRIM-SYMB.GETWCT,PFPRIM-REL.GETWCT
@FTN,F     PFPRIM-SYMB.ISMO,PFPRIM-REL.ISMO
@FTN,F     PFPRIM-SYMB.MAKAGG,PFPRIM-REL.MAKAGG
@FTN,F     PFPRIM-SYMB.MAKIND,PFPRIM-REL.MAKIND
@FTN,F     PFPRIM-SYMB.MATPOL,PFPRIM-REL.MATPOL
@FTN,F     PFPRIM-SYMB.NXTISS,PFPRIM-REL.NXTISS
@FTN,F     PFPRIM-SYMB.SETPOL,PFPRIM-REL.SETPOL
@FTN,F     PFPRIM-SYMB.SRTPOL,PFPRIM-REL.SRTPOL
@FTN,F     PFPRIM-SYMB.WRTEXT,PFPRIM-REL.WRTEXT
@FTN,F     PFPRIM-SYMB.WRTVAL,PFPRIM-REL.WRTVAL
@FTN,F     PFPRIM-SYMB.ZEROIN,PFPRIM-REL.ZEROIN
@FTN,F     PFPPIM-SYMB.ZILCH,PFPRIM-REL.ZILCH
@MAP            ,PFPRIM-ABS.755APPPOL/2-2-C
IN        PFPRIM-REL.DRIVE2/2-C
IN        PFPRIM-REL.APP22C/2-2-C
IN        PFPRIM-REL.AGISPO
IN        PFPRIM-REL.CONPOL
IN        PFPRIM-REL.DOIT
IN        PFPRIM-REL.FINDAT
IN        PFPRIM-REL.FINVAL
IN        PFPRIM-REL.GETDAY
IN        PFPRIM-REL.GETJOB
IN        PFPRIM-REL.GETPAR
IN        PFPRIM-REL.GETPOL
IN        PFPRIM-REL.GETWCT
IN        PFPRIM-REL.GETVAL
IN        PFPRIM-REL.ISMO
IN        PFPRIM-REL.MAKAGG
IN        PFPRIM-REL.MAKIND
IN        PFPRIM-REL.MATPOL
IN        PFPRIM-REL.NXTISS
IN        PFPRIM-REL.SETPOL
IN        PFPRIM-REL.SRTPOL
IN        PFPRIM-REL.WRTEXT
IN        PFPRIM-REL.WRTVAL
IN        PFPRIM-REL.ZEROIN
IN        PFPRIM-REL.ZILCH
IN        PFPRIM-REL.WRTERR
IN        PFPRIM-REL.BSERCH
IN        PFPRIM-REL.F2FRT
IN        PFPRIM-REL.ICHINT
IN        PFPRIM-REL.PCHFLT
END
@ .
@PACK PFPPIM-ABS.
@PACK PFPPIM-REL.
@PACK PFPRIM-SYMB.
@FREE PFPPIM-ABS.
@FREE PFPPIM-REL.
@FREE PFPPIM-SYMB.
```

Figure 4-24. APPLY-POLICY/2-2-C Compile and Map Runstream

```
@TOP;HDG     MAP-SET-BASEVALU/2-3      PROCESSOR 2-3     COMPILE AND MAP
@PFPRIM-APS.TITLE,D    COMPIL-MAP,,APPLY,POLICY2-3
@PFPRIM-APS.TITLE,D    COMPIL-MAP,,APPLY,POLICY2-3
@FIN,F    PFPRIM-SYMB.DRIVE2/3,PFPRIM-REL.DRIVE2/3
@FIN,F    PFPRIM-SYMB.APP23/2-3,PFPRIM-REL.APP23/2-3
@FIN,F    PFPRIM-SYMB.GETBAS,PFPRIM-REL.GETBAS
@FIN,F    PFPRIM-SYMB.SETBAS,PFPRIM-REL.SETBAS
@FIN,F    PFPRIM-SYMB.WRTUNF,PFPRIM-REL.WRTUNF
@MAP      ,PFPRIM-ABS.755SETBASE/2-3
IN          PFPRIM-REL.DRIVE2/3
IN          PFPRIM-REL.APP23/2-3
IN          PFPRIM-REL.FINVAL
IN          PFPRIM-REL.GETBAS
IN          PFPRIM-REL.GETDAY
IN          PFPRIM-REL.GETJOB
IN          PFPRIM-REL.GETPAR
IN          PFPRIM-REL.GETVAL
IN          PFPRIM-REL.GETWCT
IN          PFPRIM-REL.NXTISS
IN          PFPRIM-REL.SETBAS
IN          PFPRIM-REL.WRTERR
IN          PFPRIM-REL.WRTUNF
IN          PFPRIM-REL.WRTVAL
IN          PFPRIM-REL.ZILCH
IN          PFPRIM-REL.BSERCH
IN          PFPRIM-REL.F2FRT
IN          PFPRIM-REL.ICHINT
IN          PFPRIM-REL.RCHFLT
END
@PACK       PFPRIM-ABS.
@PACK       PFPRIM-REL.
@PACK       PFPRIM-SYMB.
@FREE       PFPRIM-ABS.
@FREE       PFPRIM-REL.
@FREE       PFPRIM-SYMB.
```

Figure 4-25. SET-BASEVALU/2-3 Compile and Map Runstream

```
@TOP;HDG  ASSIGN/3     COMPILE AND MAP ASSIGNMENT PROCESSOR 3 0
@PFPRIM-APS.TITLE,D      ASSIGN-,MENT,PROCESSOR,3$-$0,ASSIGN
@PFPRIM-APS.TITLE,D      ASSIGN-,MENT,PROCESSOR,3$-$0,ASSIGN
@MSG,N
@POP,F          PFPRIM-SYMB.BLOCKDATA,PFPRIM-REL.BLOCKDATA
@END
@FIN,O     PFPRIM-SYMB.NETWORKMAIN,PFPRIM-REL.NETWORKMAIN
@FIN,O     PFPRIM-SYMB.PRENET,PFPRIM-REL.PRENET
@FIN,O     PFPRIM-SYMB.PSTNET,PFPRIM-REL.PSTNET
@FIN,O     PFPRIM-SYMB.FORM,PFPRIM-REL.FORM
@FIN,O     PFPRIM-SYMB.ADDMFR,PFPRIM-REL.ADDMFR
@FIN,O     PFPRIM-SYMB.BUILD,PFPRIM-REL.BUILD
@FIN,O     PFPRIM-SYMB.SNET,PFPRIM-REL.SNET
@FIN,O     PFPRIM-SYMB.RIGHT,PFPRIM-REL.RIGHT
@FIN,O     PFPRIM-SYMB.REPORT,PFPRIM-REL.REPORT
@FIN,O     PFPRIM-SYMB.SORTTG,PFPRIM-REL.SORTTG
@FIN,O     PFPRIM-SYMB.FOUR,PFPRIM-REL.FOUR
@FIN,O     PFPRIM-SYMB.FIVE,PFPRIM-REL.FIVE
@FIN,O     PFPRIM-SYMB.SIX,PFPRIM-REL.SIX
@FIN,O     PFPRIM-SYMB.SEVEN,PFPRIM-REL.SEVEN
@MAP,E       ,PFPRIM-ABS.755ASSIGN/3
IN      PFPRIM-REL.NETWORKMAIN
IN      PFPRIM-REL.F2FRT
IN      PFPRIM-REL.PRENET
IN      PFPRIM-REL.PSTNET
IN      PFPRIM-REL.FORM
IN      PFPRIM-REL.ADDMFR
IN      PFPRIM-REL.BUILD
IN      PFPRIM-REL.SNET
IN      PFPRIM-REL.RIGHT
IN      PFPRIM-REL.REPORT
IN      PFPRIM-REL.SORTTG
IN      PFPRIM-REL.FOUR
IN      PFPRIM-REL.FIVE
IN      PFPRIM-REL.SIX
IN      PFPRIM-REL.SEVEN
END
```

Figure 4-26. ASSIGNMENTS/3 Compile and Map Runstream

```
@ASG,A   PFPRIM-SYMB.
@ASG,A   PFPRIM-REL.
@ASG,A   PFPRIM-ABS.
@PDP,F   PFPRIM-SYMB.BLOCKDATA, PFPRIM-REL.BLOCKDATA
@END
@TOP:FIN,0    PFPRIM-SYMB.GRDMAIN,        PFPRIM-REL.GRDMAIN
@TOP:FIN,0    PFPRIM-SYMB.PREGRD,         PFPRIM-REL.PREGRD
@TOP:FIN,0    PFPRIM-SYMB.PSTGRD,         PFPRIM-REL.PSTGRD
@TOP:FIN,0    PFPRIM-SYMB.FORM,           PFPRIM-REL.FORM
@TOP:FIN,0    PFPRIM-SYMB.ADDMFR,         PFPRIM-REL.ADDMFR
@TOP:FIN,0    PFPRIM-SYMB.BUILD,          PFPRIM-REL.BUILD
@TOP:FIN,0    PFPRIM-SYMB.SNET,           PFPRIM-REL.SNET
@TOP:FIN,0    PFPRIM-SYMB.RIGHT,          PFPRIM-REL.RIGHT
@TOP:FIN,0    PFPRIM-SYMB.REPORT,         PFPRIM-REL.REPORT
@TOP:FIN,0    PFPRIM-SYMB.SORTTG,         PFPRIM-REL.SORTTG
@TOP:FIN,0    PFPRIM-SYMB.FOUR,           PFPRIM-REL.FOUR
@TOP:FIN,0    PFPRIM-SYMB.FIVE,           PFPRIM-REL.FIVE
@TOP:FIN,0    PFPRIM-SYMB.SIX,            PFPRIM-REL.SIX
@TOP:FIN,0    PFPRIM-SYMB.SEVEN,          PFPRIM-REL.SEVEN
@MSG,N
@TOP:MAP,E      ,PFPRIM-ABS.755SUBST-GRD/4-1
IN PFPRIM-REL.GRDMAIN
IN PFPRIM-REL.F2FRT
IN PFPRIM-REL.PREGRD
IN PFPRIM-REL.PSTGRD
IN PFPRIM-REL.FORM
IN PFPRIM-REL.ADDMFR
IN PFPRIM-REL.BUILD
IN PFPRIM-REL.SNET
IN PFPRIM-REL.RIGHT
IN PFPRIM-REL.REPORT
IN PFPRIM-REL.SORTTG
IN PFPRIM-REL.FOUR
IN PFPRIM-REL.FIVE
IN PFPRIM-REL.SIX
IN PFPRIM-REL.SEVEN
END
@PACK PFPRIM-REL.
@PACK PFPPIM-SYMB.
@FREE PFPPIM-SYMB.
@FREE PFPPIM-REL.
@FREE PFPPIM-ABS.
```

Figure 4-27. SUBST-GRD/4-1 Compile and Map Runstream

```
@ .
@ . C O M P I L E   &   M A P   S U B S T I T U T E   M O S   -   4 - 2
@ .
@ASG,A     PFPRIM-ABS.
@ASG,A     PFPRIM-REL.
@ASG,A     PFPRIM-SYMB.
@ .
@PDP,F   PFPRIM-SYMB.BLOCKDATA, PFPRIM-SYMB.BLOCKDATA
@END
@TOP:FIN,0   PFPRIM-SYMB.MOSMAIN,        PFPRIM-REL.MOSMAIN
@TOP:FIN,0   PFPRIM-SYMB.PREMOS,         PFPRIM-REL.PREMOS
@TOP:FIN,0   PFPRIM-SYMB.PSTMOS,          PFPRIM-REL.PSTMOS
@TOP:FIN,0   PFPRIM-SYMB.MOSEND,          PFPRIM-REL.MOSEND
@TOP:FIN,0   PFPRIM-SYMB.FORM,            PFPRIM-REL.FORM
@TOP:FIN,0   PFPRIM-SYMB.ADDMFR,          PFPRIM-REL.ADDMFR
@TOP:FIN,0   PFPRIM-SYMB.BUILD,           PFPRIM-REL.BUILD
@TOP:FIN,0   PFPRIM-SYMB.SNET,            PFPRIM-REL.SNET
@TOP:FIN,0   PFPRIM-SYMB.RIGHT,           PFPRIM-REL.RIGHT
@TOP:FIN,0   PFPRIM-SYMB.REPORT,          PFPRIM-REL.REPORT
@TOP:FIN,0   PFPRIM-SYMB.SORTTG,          PFPRIM-REL.SORTTG
@TOP:FIN,0   PFPRIM-SYMB.FOUR,            PFPRIM-REL.FOUR
@TOP:FIN,0   PFPRIM-SYMB.FIVE,            PFPRIM-REL.FIVE
@TOP:FIN,0   PFPRIM-SYMB.SIX,             PFPRIM-REL.SIX
@TOP:FIN,0   PFPRIM-SYMB.SEVEN,           PFPRIM-REL.SEVEN
@TOP:MAP,E      ,PFPRIM-ABS.755SUBST-MOS/4-2
IN PFPRIM-REL.MOSMAIN
IN PFPRIM-REL.F2FRT
IN PFPRIM-REL.PREMOS
IN PFPRIM-REL.PSTMOS
IN PFPRIM-REL.MOSEND
IN PFPRIM-REL.FORM
IN PFPRIM-REL.ADDMFR
IN PFPRIM-REL.BUILD
IN PFPRIM-REL.SNET
IN PFPRIM-REL.RIGHT
IN PFPRIM-REL.REPORT
IN PFPRIM-REL.SORTTG
IN PFPRIM-REL.FOUR
IN PFPRIM-REL.FIVE
IN PFPRIM-REL.SIX
IN PFPRIM-REL.SEVEN
END
@ .
@PACK PFPRIM-ABS.
@PACK PFPRIM-REL.
@PACK PFPRIM-SYMB.
```

Figure 4-28. SUBST-MOS/4-2 Compile and Map Runstream

```
@ASG,A PFPRIM-SYMB.
@ASG,A PFPRIM-REL.
@ASG,A PFPRIM-ABS.
@  .
@FIN,O     PFPPIM-SYMB.READINESS/5,     PFPRIM-REL.READINESS/5
@MAP,E       ,PFPRIM-ABS.755READINESS/5
IN PFPRIM-REL.F2FRT
IN PFPRIM-REL.READINESS/5
END
@PACK PFPPIM-REL.
@PACK PFPRIM-SYMB.
@FREE PFPRIM-SYMB.
@FREE PFPRIM-REL.
@FREE PFPRIM-ABS.
```

Figure 4-29. READINESS/5 Compile and Map Runstream

```
@  . HDG P R I M   R E P O R T   C O M P I L E   A N D   M A P
PFPRIM-ABS.TITLE,D COMPILE,AND$MAP,REPORTS,PROCESS$6
PFPPIM-ABS.TITLE,D COMPILE,AND$MAP,REPORTS,PROCESS$6
@ASG,A      PFPRIM-REL.
@ASG,A      PFPRIM-ABS.
@ASG,A      PFPRIM-SYMB.
@PDP,F     PFPRIM-SYMB.REPORTPROC,PFPRIM-SYMB.REPORTPROC
@FIN,F       PFPRIM-SYMB.DRIVE6,PFPRIM-REL.DRIVE6
@FIN,F       PFPRIM-SYMB.REPT01,PFPRIM-REL.REPT01
@FIN,F       PFPRIM-SYMB.REPT02,PFPRIM-REL.REPT02
@FIN,F       PFPRIM-SYMB.REPT03,PFPRIM-REL.REPT03
@FIN,F       PFPRIM-SYMB.REPT04,PFPRIM-REL.REPT04
@FIN,F       PFPRIM-SYMB.REPT05,PFPRIM-REL.REPT05
@FIN,F       FFPRIM-SYMB.REPT06,PFPRIM-REL.REPT06
@FIN,F       PFPRIM-SYMB.REPT07,PFPRIM-REL.REPT07
@FIN,F       PFPRIM-SYMB.REPT08,PFPRIM-REL.REPT08
@FIN,F       PFPRIM-SYMB.REPT09,PFPRIM-REL.REPT09
@FIN,F       PFPRIM-SYMB.REPT10,PFPRIM-REL.REPT10
@FIN,F       PFPRIM-SYMB.REPT11,PFPRIM-REL.REPT11
@FIN,F       PFPRIM-SYMB.ZERODV,PFPRIM-REL.ZERODV
@FIN,F       PFPRIM-SYMB.PAR,PFPRIM-REL.PAR
@FIN,F       PFPRIM-SYMB.ISSUNA,PFPRIM-REL.ISSUNA
@MAP      ,PFPRIM-ABS.755REPORT/6
IN       PFPRIM-REL.DRIVE6
IN       PFPRIM-REL.REPT01
IN       PFPRIM-REL.REPT02
IN       PFPRIM-REL.REPT03
IN       PFPRIM-REL.REPT04
IN       PFPRIM-REL.REPT05
IN       PFPRIM-REL.REPT06
IN       PFPRIM-REL.REPT07
IN       PFPRIM-REL.REPT08
IN       PFPRIM-REL.REPT09
IN       PFPRIM-REL.REPT10
IN       PFPRIM-REL.REPT11
IN       PFPRIM-REL.F2FRT
IN       PFPRIM-REL.WRTERR
IN       PFPRIM-REL.PAR
IN       PFPRIM-REL.GETDAY
IN       PFPRIM-REL.ISSUNA
IN       PFPRIM-REL.ICHINT
IN       PFPRIM-REL.ZERODV
END
@PACK       PFPRIM-SYMB.
@PACK       PFPRIM-REL.
@PACK       PFPRIM-ABS.
@FREE       PFPRIM-SYMB.
@FREE       PFPRIM-REL.
@FREE       PFPRIM-ABS.
```

Figure 4-30. REPORT/6 Compile and Map Runstream

```
@ .
@ .      ------------------------------------------------------
@ .        THIS RUNSTREAM COMPILES THE PRIM UTILITY SUBROUTINES
@ .      ------------------------------------------------------
@ .  SINCE THE PRIM UTILITY PROGRAMS ARE A MIXTURE OF PROGRAMS THAT
@ .  WERE PROGRAMMED AT CAA AND PROGRAMS THAT ARE ON THE MILPERCEN
@ .  SYSTEM, BUT NOT A PART OF THE SYSTEM DOCUMENTATION, THE UTILITY
@ .  PROGRAMS ARE ACCESSED IN A VARIETY OF MANNERS THAT ARE DISCUSSED
@ .  BELOW
@ .
@ .  - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
@ .
@ .  APPICA      A UTILITY PROGRAM FURNISHED BY THE SPERRY SYSTEM
@ .              REPRESENTATIVE IN THE RELATIVE VERSION ONLY.  IT
@ .              IS, THEREFORE, NOT COMPILED.
@ .
@ .  F2FRTS      AN ASSEMBLY LANGUAGE PROGRAM WRITTEN TO THE SYSTEM
@ .              SPECIFICATION.  IT CHANGES THE LIMIT OF THE ALLOWABLE
@ .              UNIT NUMBER REFERENCES FROM 30 TO 99.
@ .
@ .  GETDAY      A FORTRAN PROGRAM STORED IN THE UTILITY FILE
@ .
@ .  GETWCT      A FORTRAN PROGRAM STORED IN THE UTILITY FILE
@ .
@ .  ICHINT      A FORTRAN FUNCTION STORED IN THE UTILITY FILE
@ .
@ .  ICOMP2      A FORTRAN FUNCTION STORED IN THE UTILITY FILE
@ .              WITHIN IN A SET OF PROGRAMS CALLED BSERCH
@ .
@ .  RCHFLT      A FORTRAN FUNCTION STORED IN THE UTILITY FILE
@ .
@ .  TITLE       A UTILITY PROGRAM FURNISHED BY THE SPERRY SYSTEM
@ .              REPRESENTATIVE IN THE ABSOLUTE VERSION ONLY.  IT
@ .              IS, THEREFORE, NOT COMPILED.
@ .
@ .  WRTERR      THIS PROGRAM IS A UTILITY BECAUSE IT IS USED BY
@ .              MORE THAN ONE PROCESSOR.  HOWEVER, SINCE IT WAS
@ .              WRITTEN ESPECIALLY FOR PRIM, IT MAY NEED TO BE
@ .              CHANGED AS LATER VERSIONS OF PRIM EVOLVE.  IT IS,
@ .              THEREFORE, STORED IN THE PRIM-SYMBOLIC FILE.
@ .
@ .  - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
@ .
@ .      ---- ASSIGN THE FILES
@ .
@ASG,A          PFPRIM-REL.
@ASG,A          PFPRIM-SYMB.
@ASG,A          PFPRIM-UTIL.
@ .
@ .
@ .      ---- COMPILE THE PROGRAMS
@ .
@TOP:PDP,FS     PFPRIM-SYMB.PRIM-PROCS,PFPRIM-SYMB.MAIN-PROCS
@TOP:FTN,FS     PFPRIM-UTIL.BSERCH,PFPRIM-REL.BSERCH
@TOP:FTN,FS     PFPRIM-UTIL.GETDAY,PFPRIM-REL.GETDAY
@TOP:FTN,FS     PFPRIM-UTIL.GETWCT,PFPRIM-REL.GETWCT
@TOP:FTN,FS     PFPRIM-UTIL.ICHINT,PFPRIM-REL.ICHINT
@TOP:FTN,FS     PFPRIM-UTIL.RCHFLT,PFPRIM-REL.RCHFLT
@TOP:FTN,FS     PFPRIM-SYMB.WRTERR,PFPRIM-REL.WRTERR
@ .
@ .      ---- PACK THE FILES THAT MAY HAVE BEEN MADE BIGGER
@ .
@PACK           PFPRIM-REL.
@PACK           PFPRIM-SYMB.
@PACK           PFPRIM-UTIL.
```

Figure 4-31. Compile PRIM Utility Programs

```
.  - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
.
.  THESE FILES WOULD BE NEEDED TO CREATE THE PRIM SYSTEM FROM A BACKUP
.  FILE , IF ALL PROGRAMS HAD BEEN DELETED FOR AN INACTIVE PERIOD
.
.  THE FILES WHICH WILL BE NEEDED FROM TAPE ARE THE SYMBOLIC
.  VERSIONS OF THE PROGRAMS, THE MAP RUNSTREAMS, THE RUNSTREAMS
.  AND THE UTILITY PROGRAMS
.
.  THE RELATIVE AND ABSOLUTE ELEMENTS CAN BE CREATED USING THE
.  MAP RUNSTREAMS.  THE RUNSTREAMS TO CREATE THE REST OF THE FILES
.  WILL BE FOUND IN THE RUNSTREAM FILE.
.
.  - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
.
@ASG,CP        PFPRIM-ABS.,///1000     .  ABSOLUTE ELEMENTS
@ASG,CP        PFPRIM-FILES.,///1000   .  USER-DEFINED (CONTROL) FILES
@ASG,CP        PFPRIM-MAP.,///1000     .  MAP RUNSTREAMS - CREATE ABSOLUTES
@ASG,CP        PFPRIM-REL.,///1000     .  RELATIVE VERSIONS OF PROGRAMS
@ASG,CP        PFPRIM-RUN.,///1000     .  RUNSTREAMS
@ASG,CP        PFPRIM-SYMB.,///1000    .  SYMBOLIC (FORTRAN) PROGRAMS
@ASG,CP        PFPRIM-UTIL.,///1000    .  PRIM UTILITY PROGRAMS (SYMBOLIC)
```

Figure 4-32. Commands to Create PRIM System From Tape Backup

```
@HDG  PFPPIM-RUN ASSIGNPROC-1 --- CREATE PREPROCESSOR FILES
.
.  THIS RUNSTREAM CREATES THE FILES NEEDED BY THE PRIM PREPROCESSOR
.
@ASG,CP        PFINPUTMOS8.,///10000      .  THIS IS ORIGINAL MOS INPUT
.                                         .  AND MUST CONTAIN DATA
.
@ASG,CP        PFERROR13.,///1000
.
@ASG,CP        PFISSUE14.,///100          .  ISSUE FILE SHOULD BE UPDATED
.                                         .  IN PFPRIM-FILES
.
@ASG,CP        PFPARAMET17.,///100        .  PARAMETER FILE SHOULD BE UPDATED
.                                         .  IN PFPRIM-FILES
.
@ASG,CP        PFVALUE19.,///100          .  VALUE FILE SHOULD BE UPDATED
.                                         .  IN PFPRIM-FILES
.
@ASG,CP        PFUIC6DIG22.,///1000       .  THIS IS ORIGINAL UIC INPUT
.
.                    THE FOLLOWING FILES MAY BE BLANK
.
@ASG,CP        PFUIC3DIG23.,///1000
@ASG,CP        PFUICDATA24.,///1000
@ASG,CP        PFUICDATA25.,///1000
@ASG,CP        PFMOS-ENL-26.,///10000
@ASG,CP        PFMOS-OFF-27.,///10000
@ASG,CP        PFMOS-WO-28.,///1000
@ASG,CP        PFMOSDATA29.,///10000
@ASG,CP        PFMOSDATA30.,///10000
@ASG,CP        PFMOSISSUE31.,///10000
@ASG,CP        PFGRASTREN85.,///100
@ASG,CP        PFMOSSTREN86.,///10000
```

Figure 4-33. Create Preprocessor Files

```
@HDG  PFPRIM-RUN ASSIGN-PPOC2 --- CREATE POLICY PROCESSOR FILES
@ .
@ .  THIS RUNSTREAM CREATES THE FILES NEEDED BY THE PRIM POLICY PROCESSOR
@ .
@ .   - - - - - - - - - - - - - - - - - - ALL FILES MAY BE BLANK
@ .                                       (UPDATE POLICY IN PFPRIM-FILES)
@ .
@ASG,CP        PFAGISPO32.,///1000                  . AGGREGATED ISSUE POLICY
@ASG,CP        PFISSPOL33.,///1000                  . ISSUE POLICY
@ASG,CP        PFMOSPOL34.,///1000                  . MOS POLICY
@ASG,CP        PFISMOPO35.,///1000                  . COMBINED POLICY
@ASG,CP        PFISMOEX36.,///10000                 . COMBINED EXTRA JOB
@ASG,CP        PFMOSEXT37.,///10000                 . MOS EXTRA JOB
@ASG,CP        PFISSEXT38.,///10000                 . ISSUE EXTRA JOB
@ASG,CP        PFJOBASVAL39.,///10000               . JAV SCRATCH
@ASG,CP        PFJOBASVAL40.,///10000               . JOB ASSIGNMENT VALUE
@ASG,CP        PFUNFILLED50.,///1000                . UNFILLED JOBS
@ASG,CP        PFAGISIND90.                         . AGGREGATED INDICATORS
@ .
@ASG,CP        PFPOLICY91.          . POLICY FILE -- MUST BE UPDATED
@ .                                     IN PFPRIM-FILES
@ .
@ASG,CP        PFEDIPOL92.          . EDITED POLICY
```

Figure 4-34. Create Policy Processor Files

```
@HDG  PFPRIM-RUN ASSIGN-PROC3 --- CREATE ASSIGNMENT PROCESSOR FILES
@ .
@ .  THIS RUNSTREAM CREATES THE FILES NEEDED BY THE PRIM ASSIGNMENT PROCESSOR
@ .
@ .  THE PRIM SUBSTITUTE ASSIGNMENT PROCESSOR, AND THE READINESS PROCESSOR
@ .
@ .  ALL OF THESE FILES MAY BE BLANK - WILL BE WRITTEN TO BY PROGRAMS
@ .
@ASG,CP        PFNETIN15.,///10000
@ASG,CP        PFEXCPEOP41.,///500
@ASG,CP        PFEXPEOMOS42.,///500
@ASG,CP        PFEXPEOGRA43.,///500
@ASG,CP        PFEXPEOMOS44.,///500
@ASG,CP         PFUNFILLED51.,///5000
@ASG,CP        PFUNFILLMO52.,///5000
@ASG,CP        PFUNFILLMO53.,///5000
@ASG,CP        PFUNFILLGR53.,///5000
@ASG,CP        PFUNFILLED54.,///5000
@ASG,CP        PFASSIGNED61.,///5000
@ASG,CP        PFASSIGNMO62.,///5000
@ASG,CP        PFASSIGNGR63.,///5000
@ASG,CP        PFASSIGNED67.,///5000
@ASG,CP        PFISSREADI70.,///5000
@ASG,CP        PFISSPERCR72.,///5000
@ASG,CP        PFGRAREADI73.,///5000
@ASG,CP        PFMOSREADI74.,///5000
@ASG,CP        PFREPORT93.
@FREE          PFNETIN15.
@FREE          PFEXCPEOP41.
@FREE          PFEXPEOMOS42.
@FREE          PFEXPEOGRA43.
@FREE          PFEXPEOMOS44
@FREE           PFUNFILLED51.
@FREE          PFUNFILLMO52.
@FREE          PFUNFILLGR53.
@FREE          PFUNFILLED54.
@FREE          PFASSIGNED61.
@FREE          PFASSIGNMO62.
@FREE          PFASSIGNGR63.
@FREE          PFASSIGNED67.
@FREE          PFISSREADI70.
@FREE          PFISSPERCR72.
@FREE          PFGRAREADI73.
@FREE          PFMOSREADI74.
@FREE          PFREPORT93.
```

Figure 4-35. Create Rest of PRIM Files

APPENDIX A

DATA DICTIONARY

Name	File number or subroutine	Format	Description
AGASGR	73	I6	Aggregate assigned over all MOS for each grade
AGASMO	74	I6	Aggregate assigned over all grades for each MOS
AGASOK	70	I6	Correct MOS assigned aggregate
AGASSE	70	I6	Senior grade assigned aggregate
AGAUGR	73 85	I6 I5	Aggregate authorized over all MOS for each grade
AGAUOK	70	I6	Correct MOS authorized aggregate
AGAUMO	74 86	I6 I12	Aggregate authorized over all grades for each MOS
AGAUSE	70	I6	Aggregate authorized senior grades
AGAVGR	73	I6	Aggregate available over all MOS for each grade
AGAVMO	74	I6	Aggregate available over all grades for each MOS
AGAVOK	70	I6	Correct MOS available aggregate
AGAVSE	70	I6	Aggregate available senior grades

Name	File number or subroutine	Format	Description
AGG	32 33 34 35 91 92	A3	Indicator of need to aggregate grades or not (yes/no)
AGGASS	70	I6	Aggregate assigned strength
AGGAUT	70 85	I6 I9	Aggregate authorized strength
AGGAVA	70	I6	Aggregate available strength
AGGFIL	39 40	I6	Desired aggregate fill
AGGMAX	39 40	I6	Minimum aggregate fill
AGGMIN	39 40	I6	Minimum aggregate fill
AGGREQ	70 85	I6 I9	Required aggregate strength
AGGVAL	39 40	I6	Value of filling to this aggregate
AGIND	90	I2	Grade aggregation indicators
AGPER	90	F5.3	Aggregate percentages
AGREGR	73 85	I6 I5	Required aggregate over all MOS for each grade
AGREMO	74 86	I6 I12	Required aggregate over all grades for each MOS
AGREOK	70	I6	Correct MOS required aggregate
AGRESE	70	I6	Senior grade required aggregate

Name	File number or subroutines	Format	Description
ALPHA	ICHINT RCHFLT	A	Alphanumeric representation of integer or real number
AMOSC	74	I2	C-rating based on authorized
ASDAT	17 21	I6	Input as of date
ASGMT	14 22 23 24 25	A2	MACOM or organization
ASI	26 27 28 29 30 31 36 37 38	A2	Additional skill identifier
APAUOK	72	F7.1	Correct MOS assigned aggregate divided by aggregate authorized strength--multiplied by 100
APREOK	72	F7.1	Correct MOS available aggregate divided by aggregate required strength--multiplied by 100
ASPEAU	64 67	F5.1	Assigned strength divided by authorized strength--multiplied by 100
ASSEPE	72	F7.1	Senior grade required aggregate divided by senior grade authorized strength--multiplied by 100

Name	File number or subroutines	Format	Description
ASSSTR	61 62 63 64 65 66 67	I7	Assigned strength
AUASC	72	I2	C-rating based on AVASPE (aggregate)
AUASPE	72	F7.1	Aggregated assigned strength divided by aggregated authorized strength--multiplied by 100
AUPERE	61 62 63 64 65 66	F5.1	Available strength divided by required strength--multiplied by 100
AVPEAU	61 62 63 64 65 66	F5.1	Available strength divided by authorized strength--multiplied by 100
AUSAGE	22 23 24 25	I4	Aggregated authorized strengths
AUSCIV	22 23 24 25	I4	Authorized strength, civilian
AUSENL	22 23 24 25	I4	Authorized strength, enlisted

Name	File number or subroutine	Format	Description
AUSOFF	22 23 24 25	I4	Authorized strength, officers
AUSWOF	22 23 24 25	I4	Authorized strength, warrant officers
AUTSTK	8 26 27 28 29 30 31 36 37 38	I3 I3 I3 I3 I4 I4 I6 I6 I6 I6	Authorized strength
AVSEC	72	I2	C-rating based on AVSEPE (senior grade)
AVSEPE	72	F7.1	Senior grade available aggregate divided by senior grade required aggregate--multiplied by 100
BEGIN	ICHINT RCHFLT	I	Number of the beginning character for conversion
C1VALA	17	A6	C1/C2 break for aggregates
C1VALM	17	A6	C1/C2 break for MOS & high 5
C2VALA	17	A6	C2/C3 break for aggregates
C2VALM	17	A6	C2/C3 break for MOS & high 5
C3VALA	17	A6	C3/C4 break for aggregates
C3VALM	17	A6	C3/C4 break for MOS & high 5

Name	File number or subroutine	Format	Description
CHKFIE	CHKNUM	A	Field to be checked
CMF	8 21 26 27 28 29 30 31 36 37 38 39 40 86	A2	Career management field
COMPO	22 23 24 25	I1	Component
COMPUT	90	I1	Indicator for individual grade policy
DEMAN	36 37 38 39 40 50 51 52 53 54 55 56 57 61 62 63 64 65 66	A9	Name used to label demand node. The first four characters are the issue code.

Name	File number or subroutine	Format	Description
DESCRI	17	A66	Description
EH	GETWCT	I	Elapsed time - hours
EM	GETWCT	I	Elapsed time - minutes
ES	GETWCT	I	Elapsed time - seconds
ENLSUB	17	A3	Enlisted grade level submission (yes/no)
FLAG	ICHINT RCHFLT	I	Flag indicating successful conversion from character
GRADE	8 26 27 28 29 30 31 36 37 38 61 62 63 64 65 66 73 85	A2	Grade
GRAFIL	39 40	I6	Desired fill level for grade
GRAMAX	39 40	I6	Maximum fill level for grade
GRAMIN	39 40	I6	Minimum fill level for grade
GRAVAL	39 40	I6	Value for filling grade level

Name	File number or subroutine	Format	Description
GRDACT	61 62 63	A2	Actual grade filling demand
GRDFRM	17	A2	Grade from which to make substitution
GRDGRP	26 27 28 29 30 31 36 37 38	A16	Grade group - included are GRADE ASI LIC IDENT REQSTR AUSTR AUSSTR
GRDREQ	61 62 63 64	A2	Grade required for demand
GRDTO	17	A2	Grade to which substitution is to be made
GRVAGR	39 40	4I6	Grade value group Includes GRAFIL, GRAMIN, GRAMAX, GRAVAL
HIGRAD	32 33 34 35 39 40 91 92	A2	Highest grade to which policy applies
H5ASAG	70	I6	Hign 5 assigned aggregate
H5ASMO	74	I6	High 5 assigned in MOS
H5AUAG	70	I6	High 5 authorized aggregate
H5AUMO	74	I6	High 5 authorized in MOS

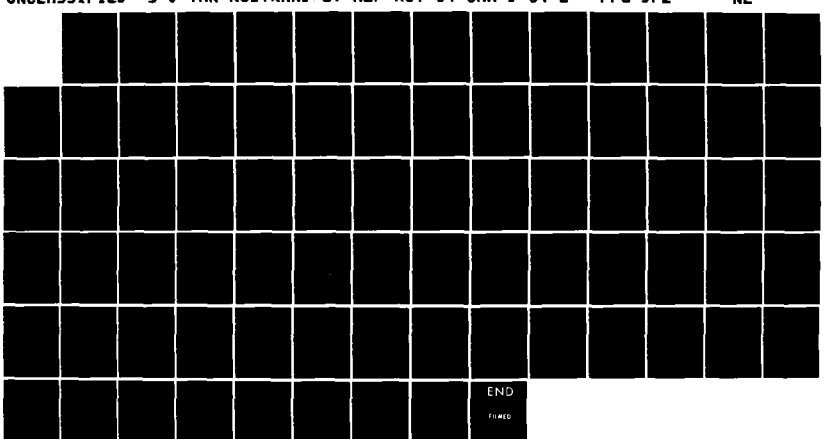
Name	File number or subroutine	Format	Description
H5AUPE	72	F7.1	High 5 assigned aggregate divided by high 5 authorized aggregate
H5AVAG	70	I6	High 5 available aggregate
H5AVMO	74	I6	High 5 available in MOS
H5REAG	70	I6	High 5 required aggregate
H5REMO	74	I6	High 5 required in MOS
H5REPE	72	F7.1	High 5 available aggregate divided by high 5 required aggregate
IDEBUG	SETTPS SETUIC SETASG SETMDS AGGCHK DOITNO DOIT12	I	Flag for debug print
IDENT	8 26 27 28 29 30 31 36 37 38	A1	Type of person for position. Male, female, either, and whether ENL, WO, or OFF
IDIFF1	2	Binary	Resource category pointer to MOS in (NAMES) array
IDIFF2	2	Binary	Activity pointer; points to grade in (IACT)
IDIFF3	2	Binary	Demand points to job in NAMES array

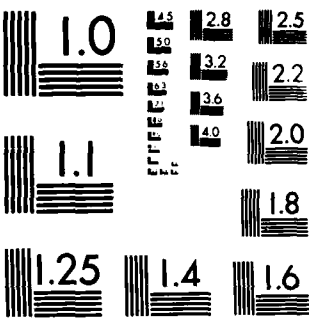
Name	File number or subroutine	Format	Description
IDIFF4	2	Binary	Number of people assigned to job
ID1	14 ISSFIN SETASG SETTPS SETUIC	A6	First method of identifying ISSUE
ID1VAL	14 SETASG SETTPS SETUIC	A6	Value associated with first method of identifying ISSUE
ID2	14 ISSFIN SETASG SETTPS SETUIC	A6	Second method of identifying ISSUE
ID2VAL	14 SETASG SETTPS SETUIC	A6	Value associated with second method of identifying ISSUE
IE	ISSFIN	I	Count of errors found in ISSFIN
IEOF	GETPOL	I	If set to 1, EOF found while reading file
IFILE	REPT01 REPT05 REPT06 REPT07 REPT09 REPT10 REPT11	I	Logical unit number of input file

AD-A149 505 PERSONNEL READINESS INDICATOR MODEL (PRIM) DOCUMENTATION PROGRAM MAINTENANCE MANUAL(U) ARMY CONCEPTS ANALYSIS AGENCY BETHESDA MD

4/4

UNCLASSIFIED S J VAN NOSTRAND ET AL. NOV 84 CAA-D-84-2 F/G 9/2 NL





Name	File number or subroutine	Format	Description
IOK	GETBAS GETJOB WRTVAL SETPOL MATPOL SETTPS SETUIC SETASG SETMOS AGGCHK AGISPO GETPAR SRTPOL ISMO WRTEXT FINVAL GETPOL DOIT WRTUNF GETVAL	I	When IOK not equal to zero program cannot continue
ISCRAT	72	I2	Overall C-rating based on lowest of REAVC, REASC, AVSEC, and OKMOSC
ISH	GETWCT	I	Hour of start time for this computation
ISM	GETWCT	I	Minute of start time for this computation
ISS	GETWCT	I	Second of start time for this computation
ISSEOF	ISSFIN	I	ISSUE file end of file flag
ISSNAM	14	A13	Name of ISSUE for report processor

Name	File number or subroutine	Format	Description
ISSUE	14 19 24 25 30 31 32 33 34 35 36 37 38 39 40 50 51 52 53 61 62 63 64 65 66 67 70 72 73 74 85 86 90 91 92	A4	ISSUE identification
JMOS	39 40	A9	Job MOS
LENG	CHKNUM	I	Length of CHKFIE
LENGTH	ICHINT RCHFLT	I	Numbers of characters to be converted to integer or real number

Name	File number or subroutine	Format	Description
LIC	26 27 28 29 30 31 36 37 38	A2	Language identifier code
LOCCO	14 22 23 24 25	A3	Location code
LOGRAD	32 33 34 35 39 40 91 92	A2	Lowest grade to which policy applies
LUNIN	DOITNO DOIT12	I	Logical unit for input
LUNISS	ISSFIN	I	Logical unit number of ISSUE file
LUNOUT	DOITNO DOIT12	I	Logical unit for output
LUVIN	FINVAL	I	Logical unit for reading JAV records
LUVOUT	FINVAL	I	Logical unit for reading JAV records
MISGOA	70	I6	Missed goal - authorized strength

Name	File number or subroutine	Format	Description
MISGOR	70	I6	Missed goal - required strength
MOS	8 21 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 50 51 52 53 54 64 65 66 74 91 92	A9	MOS or specialty code
MOSACT	61 62 63	A9	Actual MOS filling demand
MOSFRM	17	A9	MOS from which to make substitutions
MOSREQ	61 62 63	A9	MOS required for demand

Name	File number or subroutine	Format	Description
MOSSUB	17	A3	MOS substitution (yes/no)
MOSTO	17	A9	MOS to which substitution is to be made
MPERAU	74	F7.1	AGASMO divided by AGAVMO multiplied by 100
MPERRE	74	F7.1	AGAVMO divided by AGREMO multiplied by 100
NCCHAR	17	A6	Number of characters in MOS
NCCHWO	17	A6	Number of characters in warrant officer MOS
NENUS	21	I7	Total number of enlisted personnel grades E1 through E9
NERR	WRTERR	I	Error number found in Preprocessor, Policy Processor, or Report Processor
NEWPOL	GETPOL	A80	Area into which a policy is read and stored until needed
NE1	21	I7	Number of E1 personnel
NE2	21	I7	Number of E2 personnel
NE3	21	I7	As of September 1983, the sum of E1, E2, and E3 personnel
NE4	21	I7	Number of E4 personnel
NE5	21	I7	Number of E5 personnel
NE6	21	I7	Number of E6 personnel
NE7	21	I1	Number of E7 personnel
NE8	21	I7	Number of E8 personnel

Name	File number or subroutine	Format	Description
NE9	21	I7	Number of E9 personnel
NGRADE	17	I2	Number of separate grades
NOFF	21	I7	Number of officers
NO1	21	I7	Number of officers in grade 01
NO2	21	I7	Total number of officers in grades 01 and 02
NO3	21	I7	Number of officers in grade 03
NO4	21	I7	Number of officers in grade 04
NO5	21	I7	Number of officers in grade 05
NO6	21	I7	Number of officers in grade 06
NUMEXC	41 42 43 44	16I4	Number of unassigned personnel by grade
NUMPER	70	I6	MINPER times aggregate assigned strength
NUMUNF	50 51 52 53 54	I5	Number of unfilled jobs by job
NWO	21	I7	Number of warrant officers
OFFSUB	17	A3	Officer level substitution (yes/no)
OKMOSC	72	I2	C-rating based on APREOK (correct MOS)

Name	File number or subroutine	Format	Description
PARAM	17	A6	Alphanumeric of 1 to 6 characters
PEAUGR	73	F7.1	AGASGR divided by AGAUER--multiplied by 100
PEREGR	73	F7.1	AVAVGR divided by AGREGR--multiplied by 100
PERIND	90	F5.3	Percentages for each grade
PERMAX	19 32 33 34 35 70 91 92	F5.3	Maximum percent fill
PERMIN	19 32 33 34 35 70 91 92	F5.3	Minimum percent fill
POLTYP	32 33 34 35 91 92	A5	Policy type - either ISSUE or MOSSC
REASC	72	I2	C-rating based on REASEP (aggregate)
REASPE	72	F7.1	Aggregated assigned strength divided by aggregated required strength

Name	File number or subroutine	Format	Description
REAVC	72	I2	C-rating based on REAVPE (aggregate)
REAVPE	72	F7.1	Aggregated available strength divided by aggregated required strength
REPNAM	93	A61	Report name
REPNUM	93	A2	Report number
REQSTR	8 26 27 28 29 30 31 36 37 38	I3 I3 I3 I3 I4 I4 I6 I6 I6 I6	Required strength
RMOSC	74	I2	C-rating based on required
RUNDAY	GETDAY	A	Rundate
SCSUB	17	A3	Specialty code substitution (yes/no)
STACO	14 22 23 24 25	A5	Station code
STNNM	22 23 24 25	A9	Station name
STSAGG	22 23 24 25	I4	Required strength - aggregate

Name	File number or subroutine	Format	Description
STSCIV	22 23 24 25	I4	Required strength - civilian
STSENL	22 23 24 25	I4	Required strength - enlisted
STSOFF	22 23 24 25	I4	Required strength - officer
STSWOF	22 23 24 25	I4	Required strength - warrant officers
TPSN	14 22 23 24 25	A5	Troop program sequence number
TYPCO	22 23 24 25	A1	Unit type
TYPSTR	17	A3	Type strength (REQ/AUT)
UIC	8 14 22 23 24 25	A6	UIC code
UIC26	26 27 28 29 30	A5	Characters 2-6 of UIC

Name	File number or subroutine	Format	Description
UNTDS	22 23 24 25	A21	Unit description
USEASI	17	A3	Use additional skill (yes/no)
USEGRA	17	A3	Use pay grade (yes/no)
USEGRD	90	I1	Indicator of inclusion
USEIDE	17	A3	Use person (sex/grade) IDENT (yes/no)
USELIC	17	A3	Use language code (yes/no)
USEMOS	17	A3	Use MOS or SC (yes/no)
VALGRD	90	I3	Value of assignment at this grade
VALUE	19 32 33 34 35 91 92	I3	Fill value of each job up to minimum percentage
VALUE2	19	I3	Value of filling job above minimum percentage up to maximum percentage
VALGRD	90	I3	Value of assignment to this grade
WHICH	93	A12	Which report subtype
YES/NO	93	A3	"Yes" or "No" for print of report numbered REPNUM

APPENDIX B

COMMON DOCUMENTATION

POLICY PROC

Stored in: PFPRIM-SYMB.PRIM-PROCS
Obtained by: INCLUDE POLICY (FORTRAN statement)
Common Name: POLCOM
Used by: Policy Processor Only

Variable	Dim	Fmt	Length	Description
ADDON	29	A	1	All characters of the alphabet plus blank and the numbers 1 and 2. Used for creating unique demand node name
AGGFIL	1	I		Desired aggregate fill
AGGMAX	1	I		Desired aggregate maximum fill
AGGMIN	1	I		Desired aggregate minimum fill
AGGVAL	1	I		Value of making each assignment up to AGGMAX
AGIND	9,3	I		Policy indicators for aggregate records. The first dimension is one greater than the maximum possible number of different policy statements that could be input about aggregates within one ISSUE, MOS, or combination (I,1) The location of the lowest grade to which this policy applies (I,2) The location of the highest grade to which this policy applies (I,3) The value of each assignment made under this policy
AGPER	8,2	R		Percentages converted to decimal from the Policy file for the aggregate policies. The 8 is the maximum possible number of different policy statements that could be input about aggregates (I,1) The minimum percentage of desired fill that may be assigned (I,2) The maximum percentage of desired fill that may be assigned

Variable	Dim	Fmt	Length	Description
AGRADE	16	A	2	The alpha representation of valid grades--obtained from the Parameter file
ASDATE	1	A	6	The as of date from the Parameter file
BASVAL	1	A	80	The area a value record (file 19) is first read into
BLNK1	1	A	1	An alpha filled with blanks for length = 1
BLNK4	1	A	4	An alpha filled with blanks for length = 4
BLNK5	1	A	5	An alpha filled with blanks for length = 5
BLNK9	1	A	9	An alpha filled with blanks for length = 9
BLNK10	1	A	10	An alpha filled with blanks for length = 10
CMF	1	A	2	Career management field
COMPUT	16	I		An indicator, one for each grade, of whether a policy has been previously input for this grade
DEMAN	1	A	10	Demand node name. Created from the ISSUE code (first four characters) and a unique combination of alpha characters in locations 5, 6, and 7, plus alpha or 1 or 2 in location 8. The method of creating DEMAN limits the number of MOS per issue to 676.
DIDISS	1	A	4	Set to the ISSUE which policy indicators were last set for
DIDMOS	1	A	9	Set to the MOS which policy indicators were last set for
E22ISS	1	A	4	The ISSUE code for which error 22 was last output. This is used to keep from generating the same error more than once.

Variable	Dim	Fmt	Length	Description
E26ISS	1	A	4	The ISSUE code for which error 26 was last output. This is used to keep from generating the same error more than once.
E46ISS	1	A	4	The ISSUE code for which error 46 was last output. This is used to keep from generating the same error more than once.
GRAFIL	16	I		The desired fill for each grade
GRAMAX	16	I		The maximum fill for each grade
GRAMIN	16	I		The minimum fill for each grade
GRAVAL	16	I		The value of each assignment for each grade. GRAVAL is set from VALGRD and is the value output to the Job Assignment Value (JAV) file.
ICOMP	1	I		The number of characters of ISSUE on which the ISSUE should be compared. For the aggregated-ISSUE indicators, only the first character is significant. Dashes (--) represent place holders instead of exact characters
IEOF	1	I		End-of-file (on Policy file) indicator
IDEBUG	1	I		A flag controlling the amount of debug printout that will be produced
IGOTIT	1	I		A flag indicating that memory has "got" a job record that cannot be matched with a policy record. The job record should be written to the Extra Job Data file
IGOTP	1	I		A flag indicating that memory has already "got" a policy and is waiting for a job record for comparison

Variable	Dim	Fmt	Length	Description
IFLAG	1	I		A flag controlling the path used by the subroutine FINDAT If IFLAG = 1, the policy is an ISSUE policy If IFLAG = 2, the policy is a combined ISSUE and MOS policy If IFLAG = 3, the policy is an MOS policy
INDGRD	16	I		An indicator, one for each grade, that the data for this grade in this job record has already been written to the JAV file
INJOB	1	I		The number of jobs that were matched with a policy
INPOL	1	I		The number of policy records read in
INVAL	1	I		The number of JAV records read in
ISSFIR	1	I		The number of the first characters of ISSUE that should be used to compare
ISTREN	16,2	I		The authorized and required strength for each grade from the job record
IUSE	1	I		The flag that indicates which strength (from ISTREN) will be the desired fill level
IVAL	1	I		The number of new JAV records written during this portion of the policy processor
JEOF	1	I		The indicator of end-of-file (on the job record file)
JMOS	1	A	9	The MOS from the job record
KMOS	1	I		The location of the first character checked after a place holder in an MOS policy
KNTAGG	1	I		The number of aggregate policy records for each ISSUE or combined ISSUE and MOS

Variable	Dim	Fmt	Length	Description
KNTEXT	1	I		The count of the number of records written to the extra job file
KNTJOB	1	I		The count of the number of job records read
KNTUNF	1	I		Count of the number of unfilled job records written
KNTVAL	1	I		Count of the total number of JAV records written (includes records written in previous modules as well as this)
LASTAG	1	I		Number of different aggregation policies read for one aggregated issue in MATPOL
LISSUE	1	A	4	The ISSUE code from the last policy record used
LMOS	1	A	9	The MOS from the last policy record processed
LUNAGG	1	I		The logical unit number for the aggregated-ISSUE indicators
LUNBAS	1	I		The logocal unit number of the Value file (19)
LUNEXT	1	I		The logical unit number of the Extra Job file for output (35, 36, 39, or 38)
LUNJOB	1	I		The logical unit number of the input job file. The LUNEXT from one module becomes the LUNJOB of the next module
LUNPOL	1	I		The logical unit number of the input Policy file (92, 32, 33, 34, or 35)
LUNUNF	1	I		The logical unit number of the output unfilled job file (50)
LUNVAL	1	I		The logical unit number of the output JAV file (39 or 40)
MAXSUM	1	I		The sum of the maximums for individual grades

Variable	Dim	Fmt	Length	Description
MINSUM	1	I		The sum of the minimums for individual grades
MNEMON	1	A	6	An alpha for reading mnemonics from the Parameter file
NCCHAR	1	I		The number of characters of MOS that will be used for matching purposes. This is from the Parameter file.
NEEDAG	1	I		A flag that signals whether the policy is of a type that aggregated-ISSUE indicators should be looked for
NEEDP	1	I		A flag that signals whether there is a policy in memory that needs to go through SETPOL
NGRADE	1	I		The number of valid grade codes in input data. This is from the Parameter file
NEWPOL	1	A	80	The area into which a new policy is read
NMOS	1	I		The number of characters checked prior to a place holder in an MOS Policy. If a place holder was not used, NMOS equals NCCHAR from the Parameter file
NOMAT	1	I		A flag that no match was found between the present policy and the aggregated ISSUE indicators
NREC	1	I		The number of aggregated records written for each input record
NXT5	1	I		A flag for which letter of the alphabet (stored in ADDON) was last put into character 5
NXT6	1	I		A flag for which letter of the alphabet (stored in ADDON) was last put into character 6
NXT7	1	I		A flag for which letter of the alphabet (stored in ADDON) was last put into character 7

Variable	Dim	Fmt	Length	Description
NXT8	1	I		A flag for which letter of the alphabet (stored in ADDON) was last put into character 8
NXTAGG	1	I		A counter of which set of aggregate indicators is now being used
NXTFLG	1	I		A flag that indicates whether the extra characters needed for a DEMAN (demand node name) are for the first output record for an input, and, if not, which output record it is
PERIND	16,2	R		The percentages of desired fill for each grade (I,1) = the minimum percentage converted to decimal (I,2) = the maximum percentage converted to decimal
PISSUE	1	A	4	The ISSUE from the policy record not yet processed
PMOS	1	A	9	The MOS from the policy record not yet processed
USEGRD	16	I		A flag that the grade has already been included in an aggregation, one for each grade.
VALGRD	16	I		The value associated with each grade set in SETPOL
YES	1	A	3	The value "YES" for checking for aggregates
Z4	1	A	4	An alpha containing the letter Z repeated four times
ZERO2	1	A	2	An alpha containing two alpha zeros ('00')

SETCOM

Stored in: PFPRIM-SYMB.PRIM-PROCS
Obtained by: INCLUDE SETPRO (FORTRAN statement)
Common Name: SETCOM
Used by: SET ISSUE, PROCESS 1-2 only

Variable	Dim	Fmt	Length	Description
AUSAGG	1	I		Authorized aggregate strength
AUSCIV	1	I		Authorized civilian strength
AUSENL	1	I		Authorized enlisted strength
AUSOFF	1	I		Authorized officer strength
AUSWOF	1	I		Authorized warrant officer strength
BLNK1	1	A	1	An alpha filled with blanks for length = 1
BLNK2	1	A	2	An alpha filled with blanks for length = 2
BLNK3	1	A	3	An alpha filled with blanks for length = 3
BLNK4	1	A	4	An alpha filled with blanks for length = 4
BLNK5	1	A	5	An alpha filled with blanks for length = 5
BLNK6	1	A	6	An alpha filled with blanks for length = 6
IDASGM	1	A	6	Alpha = "ASGMT " for checking ID1 and ID2 from ISSUE Definition file
IDLOCC	1	A	6	Alpha = "LOCCO " for checking ID1 and ID2 from ISSUE Definition file
IDSTAG	1	A	6	Alpha = "STACCO" for checking ID1 and ID2 from ISSUE Definition file
IDTPSN	1	A	6	Alpha = "TPSN " for checking ID1 and ID2 from ISSUE Definition file

Variable	Dim	Fmt	Length	Description
ISSNAM	1	A	12	ISSUE name from ISSUE Definition file
LUNIN	1	I		Logical unit number to use for next input
LUNOUT	1	I		Logical unit number to use for next output of the UIC-data file
LUNISS	1	I		Logical unit number to use for ISSUE Definition file (14)
NUIC	1	I		Number of UIC records read
NUMASG	1	I		Number of ISSUEs set on the basis of ASGMT1 or ASGMT
NUMTPS	1	I		Number of ISSUEs set on the basis of TPSN
NUMUIC	1	I		Number of ISSUEs set on the basis of UIC
NUMSET	1	I		Total number of ISSUEs set
TYPCO	1	A	1	Unit type
UNTDS	1	A	21	Unit description

AGGCOM

Stored in: PFPRIM-SYMB.PRIM-PROCS
Obtained by: INCLUDE AGGREG (FORTRAN statement)
Common Name: AGGCOM
Used by: SET ISSUE, Process 1-2 only

Variable	Dim	Fmt	Length	Description
ASI	16	A	2	ASI code from MOS record
ASSSTR	16	I		Assigned strength
AUTSTR	16	I		Authorized strength from the MOS-data file
GRAD16	16	A	2	Alpha representation of each grade from Parameter file

Variable	Dim	Fmt	Length	Description
IDENT	16	I		Person identity code, one for each grade
LIC	16	I		Language identification code, one for each grade
MOSIN	1	I		Unit number for reading the input MOS-data file
MOSOUT	1	I		Unit number for writing the output MOS data
REQSTR	16	I		Required or structure strength from the MOS-data file
UIC24	1	A	3	Characters 2 through 4 of the UIC from the UIC-data file
UICMOS	1	A	6	The UIC from the MOS-data file record
UMOS24	1	A	3	Characters 2 through 4 of the UIC from the MOS-data file record

ERROR PROC

Stored in: PFPRIM-SYMB.PRIM-PROCS
Obtained by: INCLUDE ERROR (FORTRAN statement)
Common name: ERRCOM
Used by: Preprocessor, Policy Processor, and Report Processor to pass information to the write error subroutine (WRTERR)

Variable	Dim	Fmt	Length	Description
AGG	1	A	2	Whether policy is aggregated or not, YE = YES, NO = NO
ASGMT	1	A	2	A code representing the organization to which the unit belongs. May be, but not necessarily, the major command
GRADE	1	A	2	An alphanumeric code for the pay grade - from E1 through O6
GRDFRM	1	A	2	Pay grade code that the grade substitution is from

Variable	Dim	Fmt	Length	Description
GRDTO	1	A	2	Pay grade code that the grade substutition is to
HIGRAD	1	A	2	Highest pay grade on policy record
ID1	1	A	6	ISSUE identification method one (TPSN, STACO, etc.)
ID1VAL	1	A	6	Value of ISSUE identification method one
ID2	1	A	6	ISSUE identification method two (STACO, ASGMT, etc.)
ID2VAL	1	A	6	Value of ISSUE identification method two
INTEG1	1	I		Variable for passing an integer to WRTERR
INTEG2	1	I		Variable for passing a second integer to WRTERR
ISSUE	1	A	4	An alphanumeric code for the ISSUE assigned by the Preprocessor using specifications in the ISSUE Definition file. The first character represents the highest (most) aggregation, characters two and three represent the second aggregation level
JISSUE	1	A	4	The ISSUE code from the job file; needed when two separate files contain the ISSUE code
KERROR	1	I		Number of errors written by WRTERR - separate count for each
KWARN	1	I		Number of warnings written by WRTERR
LOCCO	1	A	3	Unit location code - from the UIC-data file
LOGRAD	1	A	2	Lowest grade on policy record
LUNIT	1	I		Logical unit number to which the message applies
MSCFRM	1	A	9	The MOS the MOS substitution is from

Variable	Dim	Fmt	Length	Description
MSCTO	1	A	9	The MOS the MOS substitution is to
MOS	1	A	9	Military occupational specialty. In PRIM, the officer specialty code is also called MOS
NERROR	1	I		Total number of errors plus warning messages
NFATAL	1	I		Meant to be the number of fatal errors - at implementation time, no errors had been identified for which it were possible to continue present processing but not continue to the next module. All errors either made it impossible to continue this module, or could, under some circumstance, not be considered fatal
NTOTAL	1	I		If NFATAL is greater than zero, then NTOTAL would be NERROR plus NFATAL - the sum of all error and warning messages

REPORT PROC

Stored in: PFPRIM-SYMB.PRIM-PROCS
Obtained by: INCLUDE REPORT (FORTRAN statement)
Common name: Unnamed
Used by: Report Processor Only

Variable	Dim	Fmt	Length	Description
ASDATE	1	A	11	Alpha representation of date as DDMMYY
GRAD16	16	A	2	Alpha representation of valid grades
NGRADE	1	I		Number of valid grades
RUNDAY	1	A	9	Alphanumeric representation of date as YYMMDD, converted to DD MMM YY

APPENDIX C

USER DEFINED FILES

C.1 ISSUE DEFINITION FILE

a. Purpose. The ISSUE Definition file specifies the unit aggregation level and the method PRIM will use to locate the correct units. Two additional aggregation levels are specified by the combinations of characters used in the ISSUE code.

b. ISSUE Code Development. The ISSUE code is always four characters as shown in the example ISSUE Definition file, Figure C-1. The first character of the ISSUE code represents the highest aggregation level that is less than the total Army; many of the reports provide totals for every set of ISSUEs that have the same first character. Readiness reports containing "Aggregated ISSUE" in the title will sum all data with the same first character and report only the totals. The second and third characters are used to specify a second aggregation level. When ISSUEs share the same first three characters of the ISSUE code, they will be subtotaled on output reports. The fourth character of individual ISSUEs that do not belong to other ISSUEs at the second level should be the special character of "-", dash (also called hyphen or minus). A graphic representation of this hierarchy is shown in Figure C-2. In the example ISSUE Definition file, CONUS divisions are the lowest aggregation level. Since CONUS divisions are in specific geographic locations (in the example location is the second level), the fourth character of the code is alphabetic, not dash. The units that are colocated are specified with the same characters 1, 2, and 3; the one with a geographic name has a dash (-) for the fourth character, and that name will be used as a title for the subtotals. Included in that ISSUE will be nondivisional units and TDAs that are located there. The third (highest) aggregation level (in the example MACOM) is designated by the first character. The relationships between the aggregation levels of the example are shown in Figure C-3. The network assignment program and the report programs need a name for the third aggregation level. This is entered in the file by placing XOO-(where X stands for any aggregation desired; the example uses MACOMs or CONUS) in the ISSUE code space and the desired name in the correct space, with other fields unspecified. To allow programs to proceed, the programs assign an ISSUE code of "----" to any units not otherwise assigned an ISSUE. Therefore, a dummy entry for both the XOO-and the "----" has been included in the example.

c. Conditions and Requirements

(1) An individual ISSUE may contain one or many units.

(2) Different numbers of units may be in each ISSUE.

(3) A PRIM policy will be applied to ISSUEs, not to individual units. If ISSUEs are defined as individual units, the policy will be indirectly applied to individual units by being applied to an ISSUE which is an aggregate of one unit only.

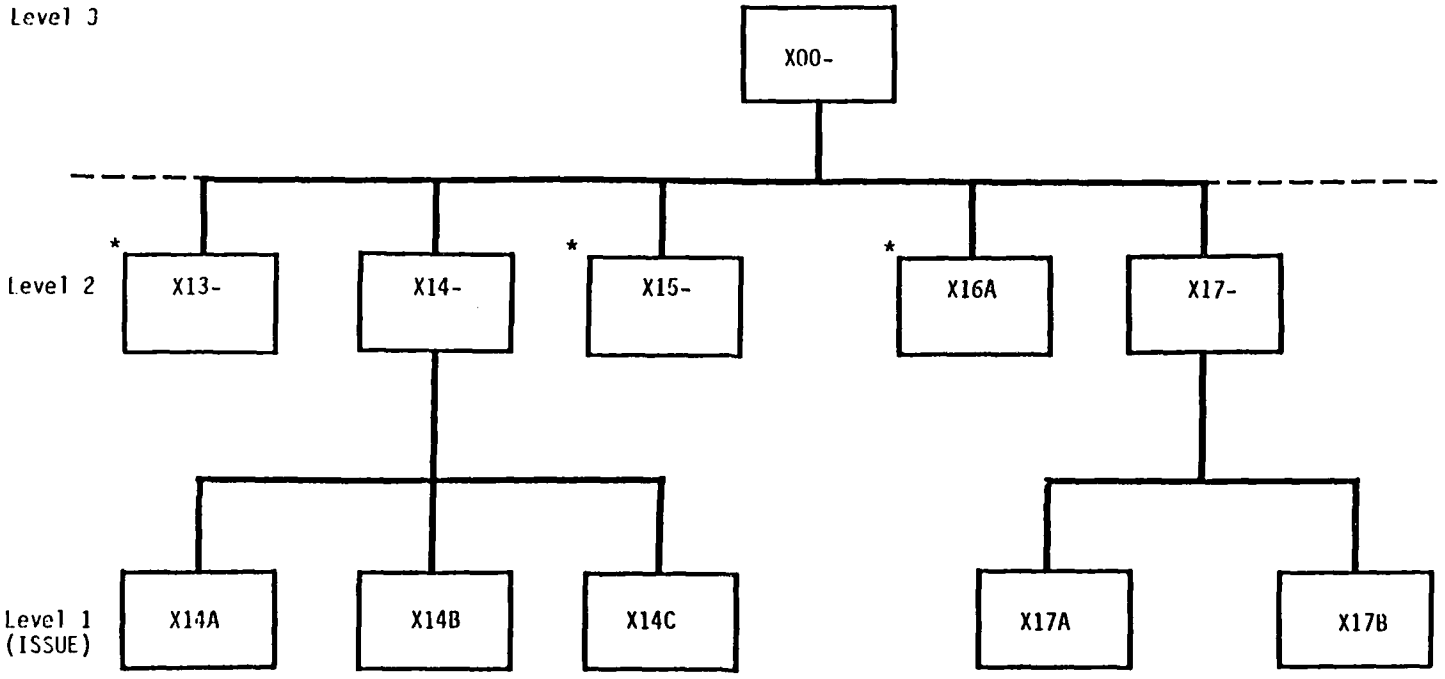
```
                    P R I M   I S S U E   F I L E

ISSUE=-00-                                          NAME=NO UIC ERROR
ISSUE=----ID1=UIC3  =      ID2=      =            NAME=BLANK-UIC
ISSUE=    ID1=UIC3  =      ID2=      =            NAME=BLANK-UIC

ISSUE=C00-                                          NAME=CONUS (OTHER)
ISSUE=C01-ID1=ASGMT =SA    ID2=      =            NAME=ARMY SECRETARY
ISSUE=C02-ID1=ASGMT =SF    ID2=      =            NAME=ARMY STAFF
ISSUE=C03-ID1=ASGMT =RC    ID2=      =            NAME=USAREC
ISSUE=C04-ID1=ASGMT =MA    ID2=      =            NAME=USMA
ISSUE=C05-ID1=ASGMT =DF    ID2=      =            NAME=DEFENSE ACTIV
ISSUE=C06-ID1=ASGMT =JA    ID2=      =            NAME=JOINT ACTIVITY
ISSUE=C07-ID1=ASGMT =PC    ID2=      =            NAME=MEPCOM
ISSUE=C08-ID1=ASGMT =CM    ID2=      =            NAME=COMPUTER SYST
ISSUE=C09-ID1=ASGMT =SC    ID2=      =            NAME=BALLISTIC MIS
ISSUE=C10-ID1=ASGMT =X     ID2=      =            NAME=DARCOM
ISSUE=C11-ID1=ASGMT =MW    ID2=      =            NAME=MDW
ISSUE=C12-ID1=ASGMT =HS    ID2=      =            NAME=HEALTH SERVICES
ISSUE=C13-ID1=ASGMT =CC    ID2=      =            NAME=ACC
ISSUE=C14-ID1=ASGMT =AS    ID2=      =            NAME=INSCOM
ISSUE=C15-ID1=UIC3  =A4A   ID2=      =            NAME=OLD GUARD
ISSUE=C16-ID1=UIC3  =35T   ID2=      =            NAME=SITE R
ISSUE=C17-ID1=UIC3  =CUM   ID2=      =            NAME=MDW MP CO
ISSUE=C18-ID1=ASGMT =CB    ID2=      =            NAME=CID
ISSUE=C19-ID1=ASGMT =AG    ID2=      =            NAME=ADJUTANT GEN
ISSUE=C20-ID1=ASGMT =CE    ID2=      =            NAME=CHIEF OF ENGR
ISSUE=C21-ID1=ASGMT =CS    ID2=      =            NAME=CHIEF OF STAFF
ISSUE=C22-ID1=ASGMT =ME    ID2=      =            NAME=SURGEON GEN
ISSUE=C23-ID1=ASGMT =MP    ID2=      =            NAME=MILPERCEN
ISSUE=C24-ID1=ASGMT =MT    ID2=      =            NAME=MTMC
ISSUE=C25-ID1=UIC3  =CYL   ID2=      =            NAME=SPEC MP CO
ISSUE=C25-ID1=UIC3  =CWB   ID2=      =            NAME=SPEC MP CO
ISSUE=C25-ID1=UIC3  =C4L   ID2=      =            NAME=SPEC MP CO
ISSUE=C25-ID1=UIC3  =FH4   ID2=      =            NAME=SPEC MP CO
ISSUE=C26-ID1=ASGMT =AU    ID2=      =            NAME=ARMY AUDIT AG
ISSUE=C27-ID1=ASGMT =GB    ID2=      =            NAME=NATL GUARD BU

ISSUE=E00-                                          NAME=USAREUR
ISSUE=E01-ID1=TPSN  =02001 ID2=      =            NAME=1ST ARM DIV
ISSUE=E02-ID1=TPSN  =02003 ID2=      =            NAME=3RD ARM DIV
ISSUE=E03-ID1=TPSN  =04008 ID2=      =            NAME=8TH INF DIV
ISSUE=E04-ID1=TPSN  =04003 ID2=      =            NAME=3RD INF DIV
ISSUE=E05-ID1=ASGMT =EH    ID2=      =            NAME=5TH FA BDE
ISSUE=E99-ID1=ASGMT =E     ID2=      =            NAME=OTHER USAREUR
ISSUE=E99AID1=TPSN  =04004 ID2=ASGMT =E5          NAME=4TH MECH-EUR
ISSUE=E99BID1=TPSN  =02002 ID2=ASGMT =EP          NAME=2ND ARM-EUR
ISSUE=E99CID1=TPSN  =04001 ID2=LOCCC =GE          NAME=1ST MECH-EUR
```

Figure C-1. ISSUE File Example



*When there is no need for both level 1 and level 2, the ISSUE may be coded with either a dash or an alphabetic character in the fourth position.

Figure C-2. Aggregation Hierarchy

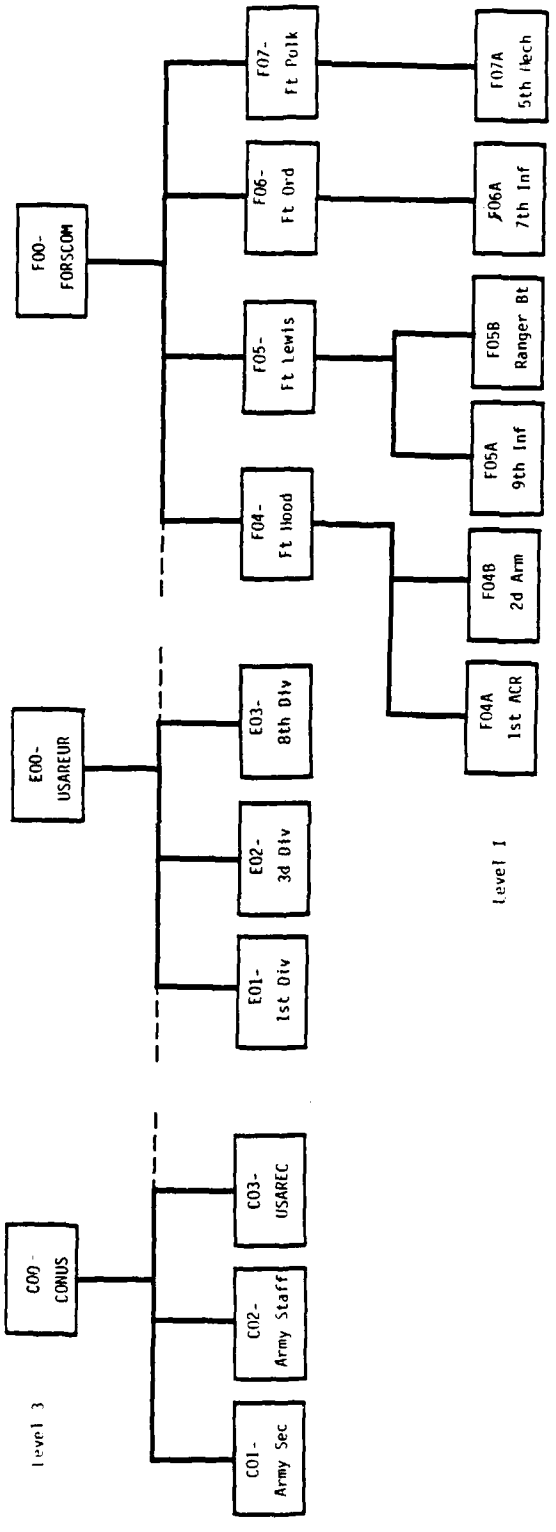


Figure C-3. Hierachy of Example

(4) The maximum number of ISSUEs that may be defined for one run is 300.

(5) A PRIM policy may apply to more than one ISSUE (See Section C.3 Policy file)

(6) If necessary, multiple combinations of identification methods may be used for each ISSUE, but each method should uniquely identify a subset of units.

(7) Units may not be included in more than one ISSUE.

(8) All units should be included in an ISSUE.

d. Data Entries. The format is described in the file description for file 14, ISSUE Definition file. The unnamed fields such as "ISSUE=" must be on each definition record; all records without these fields will be ignored. ID1, ID1VAL, ID2, ID2VAL, and ISSNAM must be left justified. All fields except ID2 and ID2VAL are required as described below. Valid combinations of ID1 and ID2 are shown in Table C-1. The sequence is the processing sequence, and the reset variable indicates whether an ISSUE will be reset during that process if it had been set during a previous process.

Field	Needed for
ISSUE Code	Data aggregated to the desired level is labeled with the ISSUE code.
ID1	At least one method of identification must be provided for each ISSUE code. See Table C-1 for the valid combinations of ID1 and ID2.
ID1 Value	In addition to the method, one or more specific values which are unique to the desired units must be specified.
ID2	A second identification method is only required when it is impossible to uniquely identify the units using ID1. If it is not needed, this field may be blank. See Table C-1 for valid ID2 variables.
ID2 Value	When a second identification method is specified, the associated value(s) must also be specified. If ID2 is blank, ID2 value should also be blank.
ISSUE Name	The report programs use this name for print purposes.

Table C-1. ISSUE Identification Methods

Sequence	ID1	ID2	Reset
1	TPSN	ASGMT	N/A
2	TPSN	LOCCO	Yes
3	TPSN	STACO	Yes
4	TPSN	-	No
5	UIC3	-	Yes
6	ASGMT	LOCCO	No
7	ASGMT	STACCO	No
8	ASGMT	-	No
9	ASGMT1	-	No

C.2 PARAMETER FILE

a. Purpose. The Parameter file provides the user with maximum flexibility. Much of the program logic such as number of characters desired in an MOS, the amount of assignment substitution allowed and the cut-points for readiness ratings is controlled by entries in the Parameter file.

b. Conditions and Requirements

(1) All variables described below must be provided.

(2) Parameter file records must be in the exact format described in the file description for file number 17, Parameter file. An example of the Parameter file is shown in Figure C-4. Blank records or comment records may be added; they must not mimic the format of another required record.

(3) Except for the relationship between NGRADE and the valid grade names, the Parameter file records may be in any order.

c. Data Entries. This section provides an expanded description of the data required for record type 1, and of the relationships between record type 1 and record type 2 or 3. See the file description for the Parameter file (file 17) for the format of record types 2 and 3. All entries for record type 1 and for record type 2 are required; record type 3 is required only when assignment substitution is requested.

```
                    P R I M   P A R A M E T E R   F I L E

ASOF  =860930 AS OF DATE TO SELECT FROM AUTHORIZATIONS
              SELECTION LOGIC IS:  TDATE < ASOF <= EDATE
NGRADE=13:    NUMBER OF DISTINCT GRADES ON INVENTORY DATA

    E1=00:    LOCATION OF E1 DATA IN GRADE-DATA
    E2=00:    LOCATION OF E2 DATA IN GRADE-DATA
    E3=01:    LOCATION OF E3 DATA IN GRADE-DATA
    E4=02:    LOCATION OF E4 DATA IN GRADE-DATA
    E5=03:    LOCATION OF E5 DATA IN GRADE-DATA
    E6=04:    LOCATION OF E6 DATA IN GRADE-DATA
    E7=05:    LOCATION OF E7 DATA IN GRADE-DATA
    E8=06:    LOCATION OF E8 DATA IN GRADE-DATA
    E9=07:    LOCATION OF E9 DATA IN GRADE-DATA
    W0=08:    LOCATION OF W0 DATA IN GRADE-DATA
    01=00:    LOCATION OF 01 DATA IN GRADE-DATA
    02=09:    LOCATION OF 02 DATA IN GRADE-DATA
    03=10:    LOCATION OF 03 DATA IN GRADE-DATA
    04=11:    LOCATION OF 04 DATA IN GRADE-DATA
    05=12:    LOCATION OF 05 DATA IN GRADE-DATA
    06=13:    LOCATION OF 06 DATA IN GRADE-DATA
NCHENL=3:     NUMBER OF MOS CHARACTERS TO AGGREGATE - ENLISTED ONLY
NCHWOF=4:     NUMBER OF MOS CHARACTERS TO AGGREGATE - WARRANT OFFICERS ONLY
NCHOFF=2:     NUMBER OF MOS CHARACTERS TO AGGREGATE - OFFICERS ONLY
NCCHAR=4:     NUMBER OF MOS CHARACTERS FOR MATCHING (MAX=9) - USE LARGEST
MOSPEO=41     INPUT EXCESS PEOPLE FILE FOR MOS SUBSTITION
MOSJOB=51     INPUT UNFILLED JOB FILE FOR MOS SUBSTITUTION
GRDPEO=41     INPUT EXCESS PEOPLE FILE FOR GRADE SUBSTITUTION
GRDJOB=51     INPUT UNFILLED JOB FILE FOR GRADE SUBSTITUTION
NONAVA=.05    NONAVAILABILITY RATE FOR READINESS PROCESSOR
MOSSUB=YES    MOS SUB ALLOWED - FORMAT(7X,A5,A9,6X,A3,A9)
C
        FROM=11B            TO=1008

SCDSUB=NO :   SPECIALTY CODE SUBSTITUTION ALLOWED
ENLSUB=YES    ENLISTED GRADE SUB ALLOWED

        FROM=E3             TO=E4
        FROM=E4             TO=E5
        FROM=E5             TO=E6
        FROM=E6             TO=E7
        FROM=E7             TO=E8
        FROM=E8             TO=E9
OFFSUB=YES:   OFFICER GRADE SUB ALLOWED     FORMAT(7X,A5,A2,13X,A3,A2)

        FROM=02             TO=03
        FROM=03             TO=04
        FROM=04             TO=05
        FROM=05             TO=06
TYPSTR=AUT:   USE AUTHORIZED (AUT) OR REQUIRED (REQ) STRENGTH?
C1VALA=0.900:  BREAK BETWEEN C2 AND C1 RATINGS - USED FOR AGGREGATES
C2VALA=0.800:  BREAK BETWEEN C3 AND C2 RATINGS - USED FOR AGGREGATES
C3VALA=0.700:  BREAK BETWEEN C4 AND C3 RATINGS - USED FOR AGGREGATES
C1VALM=0.850:  BREAK BETWEEN C2 AND C1 RATINGS - USED FOR MOS & HIGH 5
C2VALM=0.750:  BREAK BETWEEN C3 AND C2 RATINGS - USED FOR MOS & HIGH 5
C3VALM=0.650:  BREAK BETWEEN C4 AND C3 RATINGS - USED FOR MOS & HIGH 5
```

Figure C-4. Parameter File Example

Columns	Description
1 8 ASOF =YYMMDD	As of date of the inventory and authorization data, this date is printed on the formatted reports.
1 8 NGRADE=nn	Two-digit integer representing the number of separately identified grades in the data. The largest valid integer is 16. This record must be followed by one blank or comment record which must be followed by at least nn type 2 records.
1 8 NCHENL=n	The number of characters of the MOS to use for enlisted data; maximum value is 9.
1 8 NCHOFF=n	The number of characters of the specialty code to use for officer data; maximum value is 9.
1 8 NCHWOF=n	The number of characters of the MOS to use for warrant officer data; maximum value is 9.
1 8 NCCHAR=n	Number of characters PRIM will use to match MOS of people against MOS of job. The maximum value is 9; the normal setting is the largest of NCHENL, NCHOFF, and NCHWOF. However, if desired, this may be smaller than any of the three other values. In the Assignment Processor this number of characters will be used for all grades; preprocessing sets all characters beyond NCHENL and NCHWOF to blank.
1 8 MOSSUB=YES or MOSSUB=NO	Whether MOS substitution will be allowed on this run. If the value is "YES", type 3 records must follow. If the value is "NO," all of the type 3 records following this will be ignored.
1 8 SCDSUB=YES or SCDSUB=NO	Whether specialty code substitution will be allowed. If the value is "YES", type 3 records must follow. If the value is "NO," all of the following type 3 records will be ignored.
1 8 ENLSUB=YES or ENLSUB=NO	Whether enlisted grade substitution will be allowed. As for MOSSUB and SCDSUB, type 3 records must follow a value of "YES" and will be ignored following a "NO" value.

Columns	Description
1 8 OFFSUB=YES or OFFSUB=NO	Whether officers grade substitution will be allowed, Type 3 records must follow a value of "YES" and will be ignored following a "NO" value.
1 8 TYPSTR=AUT or TYPSTR=REQ	A value of "AUT" means the authorized strength will be used for developing the fill levels; a value of "REQ" means the required (structure) strength will be used.
1 8 MOSPEO=nn	The file number for input of excess people for MOS substitution; the number may be 41 or 43. Normally, 41 is used when MOS substitution is run immediately after regular assignment, and 43 is used when grade substitution is run first.
1 8 MOSJOB=nn	The file number for input of unfilled jobs for MOS substitution; the number may be 51 or 53. When MOSPEO is 41, MOSJOB should be 51; when MOSPEO is 43, MOSJOB should be 53.
1 8 GRDPEO=nn	The file number for input of excess people for grade substitution; the number may be 41 or 42. Normally, 41 is used when grade substitution is run immediately after regular assignment, and 42 is used when MOS substitution is run prior to grade substitution.
1 8 GRDJOB=nn	The file number for input of unfilled jobs for grade substitution; the number may be 51 or 52.
1 8 NONAVA=.nn	The percentage of assigned personnel not available for duty converted to a decimal and subtracted from 1.00; could be considered as the percentage of assigned personnel who will be available for duty converted to a decimal rate.
1 8 C1VALA=.XXX	The break between C-2 and C-1 rating for aggregate available strengths (page 3-11, AR 220-1, Unit Status Reporting, dated 1 June 81).

Columns	Description
1 8 C2VALA=.XXX	The break between C-2 and C-3 rating for aggregate available strengths (page 3-11, AR 220-1, Unit Status Reporting, dated 1 June 81).
1 8 C3VALA=.XXX	The break between C-3 and C-4 rating for aggregate available strengths (page 3-11, AR 220-1, Unit Status Reporting, dated 1 June 81).
1 8 C1VALM=.XXX	The break between C-2 and C-1 ratings for available MOS or senior grade percentage (page 3-12, AR-220-1, Unit Status Reporting, dated 1 June 1981).
1 8 C2VALM=.XXX	The break between C-3 and C-2 ratings for available MOS or senior grade percentage (page 3-12, AR-220-1, Unit Status Reporting, dated 1 June 1981).
1 8 C3VALM=.XXX	The break between C-4 and C-3 ratings for available MOS or senior grade percentage (page 3-12, AR-220-1, Unit Status Reporting, dated 1 June 1981).

C.3 POLICY FILE

a. Purpose. The Policy file specifies the minimum and maximum fill levels and the associated assignment value for specific grades or aggregations of grades in specific ISSUEs, specific MOS or SC, or in a specific combination of ISSUE and MOS. All policies which are related to specific MOS or SC must be entered in this file; ISSUE policies that apply to all grades individually may be entered here or in the Value file; policies specified in terms of aggregations should be entered in the Policy file; policies that do not apply to all grades must be specified here (see Table C-2).

b. Conditions and Requirements

(1) All ISSUE codes must be defined in the ISSUE Definition file.

(2) The same grade cannot be used on more than one individual grade record or in more than one aggregated grade record for an ISSUE, an MOS or SC, or a specific combination.

Table C-2. Choice of Policy or Value

Related to	Pay grade	File
ISSUE	Less than all	Policy
ISSUE	All, aggregated	Policy
ISSUE	All, individually	Policy or Value
MOS	Any number, both aggregated and not	Policy
ISSUE & MOS	Any number, both aggregated and not	Policy

(3) If an ISSUE-specific policy is inconsistant with a policy stated for the including aggragated-ISSUE, the ISSUE-specific policy will be ignored; an error message will be given. *The only way to correct this type* of error is to delete the ISSUE-specific policy or to delete the aggregated ISSUE policy and replace it with one record for each ISSUE included in the aggregation, except the one that would otherwise cause the error.

(4) A warning message is given for every fill percentage greater than 200 percent (2.000 in decimal).

(5) Although the Value file is not compared with the Policy file, the user should consider both files when setting the values in each. In general, Policy file values should probably be less than the Value file minimum fill values; it may be desirable to set the Policy file values to smaller values than the Value file minimum values and larger than the Value file maximum values.

(6) When a policy record applies to fewer grades than are in the data, only those grades specified are written to the Job Assignment Value file; the data for other grades is saved for the application of other policies or the Value file.

(7) The minimum percentage must be less than or equal to the maximum percentage.

(8) The low grade must be less than or equal to the high grade.

(9) The value of each assignment between the minimum and the maximum will, by default, be the same as the value for each assignment up to the minimum.

c. Data Entries. All entries are required except "ISSUE=XXXX" for MOS policies and "MOS=XXXXXXXXX" for ISSUE policies. For correct column numbers, see the file description for file 91, Policy file. In the following data descriptions, the description on the right applies to the X value on the left. An example of a Policy file will be found in Figure C-5.

Field	Description
ISSUE=XXXX or MOSSC=bbbb	For ISSUE policies, this is a valid, four character code from the ISSUE file; the ISSUE code is blank for MOS policies.
LO=XX	The lowest grade to which this policy applies.
HI=XX	The highest grade to which this policy applies.
AG=XXX	If the policy applies only to the aggregate of the low grade through the high grade, XXX should be YES. Another record(s) should be input whch specifies the policy for the individual grades. On those records, XXX=NOb.
VALUE=XXX	The value or worth of meeting this policy. In general, the magnitude of the policy values will be larger than the maximum fill value in the Value file. It may be advantageous to set the minimum fill value in the Value file larger than this policy value. The job with the largest value will be filled first.
MIN=X.XXX	The smallest percentage (converted to a decimal) of assignments that should be made for this grade(s), ISSUE, and/or MOS.
MAX=X.XXX	The largest percentage of assignments that may be made for this grade(s), ISSUE, and/or MOS (converted to a decimal).
MOS=XXXXXXXXX	This field is blank for ISSUE policies; otherwise, it is required. Although the MOS may be up to nine characters in length, the number of characters should not be larger than the number of characters specified for the grade in the Parameter file.

```
                    P R I M   P O L I C Y   F I L E

ISSUE=C01-LO=02HI=02AG=NO  VALUE=  5   MIN= .70 MAX=1.90 MOS=          ARMY SECR
ISSUE=C01-LO=03HI=03AG=NO  VALUE=  5   MIN=1.00 MAX=1.10 MOS=          ARMY SECR
ISSUE=C01-LO=04HI=06AG=NO  VALUE=  5   MIN= .70 MAX=1.90 MOS=          ARMY SECR
ISSUE=C01-LO=WOHI=WOAG=NO  VALUE=  5   MIN=1.00 MAX=1.10 MOS=          ARMY SECR
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
ISSUE=C02-LO=02HI=02AG=NO  VALUE=  5   MIN= .70 MAX=1.90 MOS=          ARMY STAF
ISSUE=C02-LO=03HI=03AG=NO  VALUE=  5   MIN=1.00 MAX=1.10 MOS=          ARMY STAF
ISSUE=C02-LO=04HI=06AG=NO  VALUE=  5   MIN= .70 MAX=1.90 MOS=          ARMY STAF
ISSUE=C02-LO=WOHI=WOAG=NO  VALUE=  5   MIN=1.00 MAX=1.10 MOS=          ARMY STAF
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
ISSUE=C03-LO=E3HI=E9AG=YESVALUE=  5   MIN=1.00 MAX=1.10 MOS=          USAREC
ISSUE=C03-LO=02HI=02AG=NO  VALUE=  5   MIN= .70 MAX=1.90 MOS=          USAREC
ISSUE=C03-LO=03HI=03AG=NO  VALUE=  5   MIN=1.00 MAX=1.10 MOS=          USAREC
ISSUE=C03-LO=04HI=06AG=NO  VALUE=  5   MIN= .70 MAX=1.90 MOS=          USAREC
ISSUE=C03-LO=WOHI=WOAG=NO  VALUE=  5   MIN=1.00 MAX=1.10 MOS=          USAREC
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
ISSUE=C04-LO=WOHI=06AG=NO  VALUE=  5   MIN=1.00 MAX=1.10 MOS=          USMA
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
ISSUE=C05-LO=WOHI=06AG=NO  VALUE=  5   MIN=1.00 MAX=1.10 MOS=          DEFENSE A
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
ISSUE=C06-LO=WOHI=06AG=NO  VALUE=  5   MIN=1.00 MAX=1.10 MOS=          JOINT ACT
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
ISSUE=C07-LO=02HI=02AG=NO  VALUE=  5   MIN= .70 MAX=1.90 MOS=          MEPCOM
ISSUE=C07-LO=03HI=03AG=NO  VALUE=  5   MIN=1.00 MAX=1.10 MOS=          MEPCOM
ISSUE=C07-LO=04HI=06AG=NO  VALUE=  5   MIN= .70 MAX=1.90 MOS=          MEPCOM
ISSUE=C07-LO=WOHI=WOAG=NO  VALUE=  5   MIN=1.00 MAX=1.10 MOS=          MEPCOM
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
ISSUE=C08-LO=02HI=02AG=NO  VALUE=  5   MIN= .70 MAX=1.90 MOS=          CMP SYS C
ISSUE=C08-LO=02HI=06AG=NO  VALUE=  5   MIN=1.00 MAX=1.10 S C=53       CMP SYS/S
ISSUE=C08-LO=03HI=03AG=NO  VALUE=  5   MIN=1.00 MAX=1.10 MOS=         CMP SYS C
ISSUE=C08-LO=04HI=06AG=NO  VALUE=  5   MIN= .70 MAX=1.90 MOS=         CMP SYS C
ISSUE=C08-LO=WOHI=WOAG=NO  VALUE=  5   MIN=1.00 MAX=1.10 MOS=741A     CMP SYS/7
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
ISSUE=C09-LO=WOHI=06AG=NO  VALUE=  5   MIN=1.00 MAX=1.10 MOS=          BALLIS MI
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
ISSUE=C10-LO=02HI=06AG=NO  VALUE=  5   MIN=1.00 MAX=1.10 S C=97       DARCOM/SC
ISSUE=C10-LO=02HI=06AG=NO  VALUE=  5   MIN=1.00 MAX=1.10 S C=51       DARCOM/SC
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
ISSUE=C15-LO=E3HI=E9AG=YESVALUE=  5   MIN= .97 MAX=1.10 MOS=          OLD GUARD
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
ISSUE=C16-LO=E3HI=E9AG=YESVALUE=  5   MIN=1.00 MAX=1.10 MOS=          SITE "R"
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
ISSUE=C17-LO=E3HI=E9AG=YESVALUE=  5   MIN= .97 MAX=1.10 MOS=          MDW MP CO
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

Figure C-5. Policy File Example

C.4 VALUE FILE

a. Purpose. The Value file specifies, by ISSUE, the minimum and maximum percentage of authorized or required spaces that the Assignment Processor should attempt to fill. In addition to the value for filling up to minimum fill level, a value for filling from the minimum up to the maximum is specified. An example of a Value file is in Figure C-6.

b. Conditions and Requirements

(1) There should be one record for every ISSUE, but no more than one per ISSUE.

(2) In general, the value of filling from zero to the minimum should be greater than the value of filling from the minimum to the maximum.

(3) The minimum fill value may be equal to the maximum fill value.

(4) The Value file is not applied to data for which a valid policy had been present. In other words, the Policy file must contain the largest desired percentage for all entries; no data element will have both a Policy file record and a Value file record applied.

c. Data Entries. All entries in this file are required. For the correct columns, see the file description for file 19, Value file. In the following data descriptions, the description on the right applies to the XX field on the left.

Entry	Description
ISSUE=XXXX	A valid, four-character ISSUE code from the ISSUE file.
MINIMUM=X.XXX	Minimum percentage fill converted to decimal. The records that are created from this data will show the minimum and maximum fill levels both equal to this rate times the desired fill for each grade level; the aggregate fill levels will be the sum of the grade fills.
MINVAL=XXX	This value will be used on the records described above for MINIMUM. This is normally larger than MAXVAL and may be larger than the policy values.

```
ISSUE=----  MINIMUM=  .60  MINVAL=  3   MAXIMUM=   .9  MAXVAL=  1
ISSUE=      MINIMUM=  .60  MINVAL=  3   MAXIMUM=   .9  MAXVAL=  1
ISSUE=C01-  MINIMUM=  .90  MINVAL=  4   MAXIMUM=  1.0  MAXVAL=  2
ISSUE=C02-  MINIMUM=  .70  MINVAL=  4   MAXIMUM=  1.0  MAXVAL=  2
ISSUE=C03-  MINIMUM=  .70  MINVAL=  4   MAXIMUM=  1.0  MAXVAL=  2
ISSUE=C04-  MINIMUM=  .70  MINVAL=  4   MAXIMUM=  1.0  MAXVAL=  2
ISSUE=C05-  MINIMUM=  .90  MINVAL=  4   MAXIMUM=  1.0  MAXVAL=  2
ISSUE=C06-  MINIMUM=  .90  MINVAL=  4   MAXIMUM=  1.0  MAXVAL=  2
ISSUE=C07-  MINIMUM=  .70  MINVAL=  4   MAXIMUM=  1.0  MAXVAL=  2
ISSUE=C08-  MINIMUM=  .70  MINVAL=  4   MAXIMUM=  1.0  MAXVAL=  2
ISSUE=C09-  MINIMUM=  .70  MINVAL=  4   MAXIMUM=  1.0  MAXVAL=  2
ISSUE=C10-  MINIMUM=  .70  MINVAL=  4   MAXIMUM=  1.0  MAXVAL=  2
ISSUE=C11-  MINIMUM=  .70  MINVAL=  4   MAXIMUM=  1.0  MAXVAL=  2
ISSUE=C12-  MINIMUM=  .70  MINVAL=  4   MAXIMUM=  1.0  MAXVAL=  2
ISSUE=C13-  MINIMUM=  .70  MINVAL=  4   MAXIMUM=  1.0  MAXVAL=  2
ISSUE=C14-  MINIMUM=  .70  MINVAL=  4   MAXIMUM=  1.0  MAXVAL=  2
ISSUE=C15-  MINIMUM=  .70  MINVAL=  4   MAXIMUM=  1.0  MAXVAL=  2
ISSUE=C16-  MINIMUM=  .70  MINVAL=  4   MAXIMUM=  1.0  MAXVAL=  2
ISSUE=C18-  MINIMUM=  .70  MINVAL=  4   MAXIMUM=  1.0  MAXVAL=  2
ISSUE=C19-  MINIMUM=  .70  MINVAL=  4   MAXIMUM=  1.0  MAXVAL=  2
ISSUE=C20-  MINIMUM=  .70  MINVAL=  4   MAXIMUM=  1.0  MAXVAL=  2
ISSUE=C21-  MINIMUM=  .90  MINVAL=  4   MAXIMUM=  1.0  MAXVAL=  2
ISSUE=C22-  MINIMUM=  .70  MINVAL=  4   MAXIMUM=  1.0  MAXVAL=  2
ISSUE=C23-  MINIMUM=  .90  MINVAL=  4   MAXIMUM=  1.0  MAXVAL=  2
ISSUE=C24-  MINIMUM=  .70  MINVAL=  4   MAXIMUM=  1.0  MAXVAL=  2
ISSUE=C25-  MINIMUM=  .70  MINVAL=  4   MAXIMUM=  1.0  MAXVAL=  2
ISSUE=C27-  MINIMUM=  .70  MINVAL=  4   MAXIMUM=  1.0  MAXVAL=  2
ISSUE=E01-  MINIMUM=  .98  MINVAL=  5   MAXIMUM=  1.0  MAXVAL=  2
ISSUE=E02-  MINIMUM=  .98  MINVAL=  5   MAXIMUM=  1.0  MAXVAL=  2
ISSUE=E03-  MINIMUM=  .98  MINVAL=  5   MAXIMUM=  1.0  MAXVAL=  2
ISSUE=E04-  MINIMUM=  .98  MINVAL=  5   MAXIMUM=  1.0  MAXVAL=  2
ISSUE=E05-  MINIMUM=  .98  MINVAL=  5   MAXIMUM=  1.0  MAXVAL=  2
ISSUE=E99-  MINIMUM=  .98  MINVAL=  5   MAXIMUM=  1.0  MAXVAL=  2
ISSUE=E99B  MINIMUM=  .98  MINVAL=  5   MAXIMUM=  1.0  MAXVAL=  2
ISSUE=E99C  MINIMUM=  .98  MINVAL=  5   MAXIMUM=  1.0  MAXVAL=  2
ISSUE=F01-  MINIMUM=  .70  MINVAL=  4   MAXIMUM=  2.0  MAXVAL=  2
ISSUE=F01A  MINIMUM=  .70  MINVAL=  4   MAXIMUM=  2.0  MAXVAL=  2
ISSUE=F02-  MINIMUM=  .70  MINVAL=  4   MAXIMUM=  2.0  MAXVAL=  2
ISSUE=F02A  MINIMUM=  .70  MINVAL=  4   MAXIMUM=  2.0  MAXVAL=  2
ISSUE=F03-  MINIMUM=  .70  MINVAL=  4   MAXIMUM=  2.0  MAXVAL=  2
ISSUE=F04-  MINIMUM=  .70  MINVAL=  4   MAXIMUM=  2.0  MAXVAL=  2
ISSUE=F04A  MINIMUM=  .70  MINVAL=  4   MAXIMUM=  2.0  MAXVAL=  2
ISSUE=F04B  MINIMUM=  .70  MINVAL=  4   MAXIMUM=  2.0  MAXVAL=  2
ISSUE=F05-  MINIMUM=  .70  MINVAL=  4   MAXIMUM=  2.0  MAXVAL=  2
ISSUE=F05A  MINIMUM=  .70  MINVAL=  4   MAXIMUM=  2.0  MAXVAL=  2
ISSUE=F05B  MINIMUM=  .70  MINVAL=  4   MAXIMUM=  2.0  MAXVAL=  2
ISSUE=F06-  MINIMUM=  .70  MINVAL=  4   MAXIMUM=  2.0  MAXVAL=  2
ISSUE=F06A  MINIMUM=  .70  MINVAL=  4   MAXIMUM=  2.0  MAXVAL=  2
ISSUE=F07-  MINIMUM=  .70  MINVAL=  4   MAXIMUM=  2.0  MAXVAL=  2
ISSUE=F07A  MINIMUM=  .70  MINVAL=  4   MAXIMUM=  2.0  MAXVAL=  2
ISSUE=F08-  MINIMUM=  .70  MINVAL=  4   MAXIMUM=  2.0  MAXVAL=  2
ISSUE=F08A  MINIMUM=  .70  MINVAL=  4   MAXIMUM=  2.0  MAXVAL=  2
ISSUE=F09-  MINIMUM=  .70  MINVAL=  4   MAXIMUM=  2.0  MAXVAL=  2
ISSUE=F09A  MINIMUM=  .70  MINVAL=  4   MAXIMUM=  2.0  MAXVAL=  2
ISSUE=F09B  MINIMUM=  .70  MINVAL=  4   MAXIMUM=  2.0  MAXVAL=  2
ISSUE=F99-  MINIMUM=  .70  MINVAL=  4   MAXIMUM=  2.0  MAXVAL=  2
ISSUE=F99A  MINIMUM=  .70  MINVAL=  4   MAXIMUM=  2.0  MAXVAL=  2
ISSUE=J01-  MINIMUM=  .98  MINVAL=  5   MAXIMUM=  1.0  MAXVAL=  2
ISSUE=K01-  MINIMUM=  .98  MINVAL=  5   MAXIMUM=  1.0  MAXVAL=  2
ISSUE=K99-  MINIMUM=  .98  MINVAL=  5   MAXIMUM=  1.0  MAXVAL=  2
ISSUE=O01-  MINIMUM=  .98  MINVAL=  5   MAXIMUM=  1.0  MAXVAL=  2
ISSUE=T01-  MINIMUM=  .70  MINVAL=  4   MAXIMUM=  1.5  MAXVAL=  2
ISSUE=T02-  MINIMUM=  .70  MINVAL=  4   MAXIMUM=  1.5  MAXVAL=  2
ISSUE=T03-  MINIMUM=  .70  MINVAL=  4   MAXIMUM=  1.5  MAXVAL=  2
ISSUE=T04-  MINIMUM=  .70  MINVAL=  4   MAXIMUM=  1.5  MAXVAL=  2
ISSUE=T05-  MINIMUM=  .70  MINVAL=  4   MAXIMUM=  1.5  MAXVAL=  2
ISSUE=T06-  MINIMUM=  .70  MINVAL=  4   MAXIMUM=  1.5  MAXVAL=  2
ISSUE=T07-  MINIMUM=  .70  MINVAL=  4   MAXIMUM=  1.5  MAXVAL=  2
ISSUE=T08-  MINIMUM=  .70  MINVAL=  4   MAXIMUM=  1.5  MAXVAL=  2
ISSUE=T09-  MINIMUM=  .70  MINVAL=  4   MAXIMUM=  1.5  MAXVAL=  2
ISSUE=T10-  MINIMUM=  .70  MINVAL=  4   MAXIMUM=  1.5  MAXVAL=  2
ISSUE=T11-  MINIMUM=  .70  MINVAL=  4   MAXIMUM=  1.5  MAXVAL=  2
ISSUE=T12-  MINIMUM=  .70  MINVAL=  4   MAXIMUM=  1.5  MAXVAL=  2
ISSUE=T13-  MINIMUM=  .70  MINVAL=  4   MAXIMUM=  1.5  MAXVAL=  2
ISSUE=T14-  MINIMUM=  .70  MINVAL=  4   MAXIMUM=  1.5  MAXVAL=  2
ISSUE=T99-  MINIMUM=  .70  MINVAL=  4   MAXIMUM=  1.5  MAXVAL=  2
ISSUE=W01-  MINIMUM=  .98  MINVAL=  5   MAXIMUM=  1.0  MAXVAL=  2
ISSUE=W99-  MINIMUM=  .98  MINVAL=  5   MAXIMUM=  1.0  MAXVAL=  2
```

Figure C-6. Value File Example

MAXIMUM=X.XXX — Maximum percentage fill converted to decimal. For each record created above, if MAXIMUM is greater than MINIMUM, another record is created with a minimum fill of zero for all grade levels and for the minimum aggreate fill. The desired maximum fill level is the desired (i.e., authorized or required) fill times MAXIMUM. Since the minimum fill should be assigned in response to the first record, the number entered as the maximum in this record is the minimum fill level subtracted from the desired maximum fill level.

MAXVAL=XXX — The value for each assignment above the minimum fill level, up to the maximum fill level. This value is normally smaller than the values used in the Policy file and the MINVAL in this file.

C.5 REPORT REQUEST FILE

a. Purpose. The Report Request file specifies the formatted readiness reports to be produced by the Report Processor. The different readiness report types are specified in Table C-3; the column numbers can be found in the description Job file 93.

Table C-3. Readiness Report Types

Report type	Report subtype	Report name
1		ISSUE Summary Report
2		Specialty Summary by Aggregate ISSUE
3	ALL	Specialty Summary by ISSUE - All Specialties
	MOS code	Specialty Summary by ISSUE - Specified Specialty
4		Specialty Summary by Grade
5		Grade Summary by Aggregate ISSUE
6		Grade Summary by ISSUE
7		High Five Summary
8	ALL	ISSUE Listing - ALL ISSUEs
	ISSUE code	ISSUE Listing - Specific ISSUE
9		C-rating
10	ALL	Excess Personnel - ALL Types
	ORIGINAL	Excess Personnel from Regular Assignment
	MOS SUB	Excess Personnel from MOS Substitution
	GRADE SUB	Excess Personnel from Grade Substitution
11	ALL	Unfilled Jobs - All Types
	ORIGINAL	Unfilled Jobs from Regular Assignment
	MOS SUB	Unfilled Job from MOS Substitution
	GRADE SUB	Unfilled Jobs from Grade Substitution

b. Conditions and Requirements. All formatted readiness report types within the range of 01 to 11.

c. Data Entries. This file is not variable except for the YES/NO and WHICH fields. All report entries should remain in the file, change only the YES/NO field and MOS or ISSUE in the WHICH field.

Field	Description
REPNUM=XX	A valid two character readiness report number
YES/NO	Either "YES" or "NO." When this report type is to be produced, set YES/NO to "YES." When this report type is not to be produced, set YES/NO to "NO."
WHICH	Report type must be "MOS," "ALL," "ORIGINAL," "MOS SUB," "GRADE SUB," MOS code, or ISSUE code when report number is 03, 08, 10, or 11; must be blank when report number 01, 02, 04, 05, 06, 07, or 09.
REPNAM	Report name

APPENDIX D

PERSONNEL SCHEDULING PROGRAM

USER MANUAL*

The operations of the personnel scheduling program are reviewed in this report from the standpoint of a user of the system.

Schematically, an overall flowchart of the system is shown in Figure 1. There are three independent subroutines: FORM, in which the network and the informational data base are established; SOLVE, in which the network is optimized; and REPORT, in which the report writing functions are contained. Broad input categories and their respective position in the program logic are also depicted in Figure 1.

The underlying network model is portrayed in Figure 2 and contains two general categories of data, people-related and job-related. In this model, people are designated as resources which are allocated to jobs and other work-related tasks. Flow in the network is specified by full time equivalents (1 man year).

The FORM routine accepts people and job data, and generates the network arrays as well as a series of linked-list pointer arrays. These linked-lists are used in identifying and categorizing subsets of people and jobs. For example, a reasonable subset descriptor might be job rank, since the user might be interested in viewing the officers' schedules. This is accomplished by providing one linked-list for all personnel at the officer position.

---

*Source: Mulvey, John M., Personnel Models With Multiple Objectives, Research Report EES-84-3, Princeton University, January, 1984. Submitted as a final report for Scientific Services Agreement, DAAG29-81-D-0100, Delivery Order Number 0789(TCN 83-403).

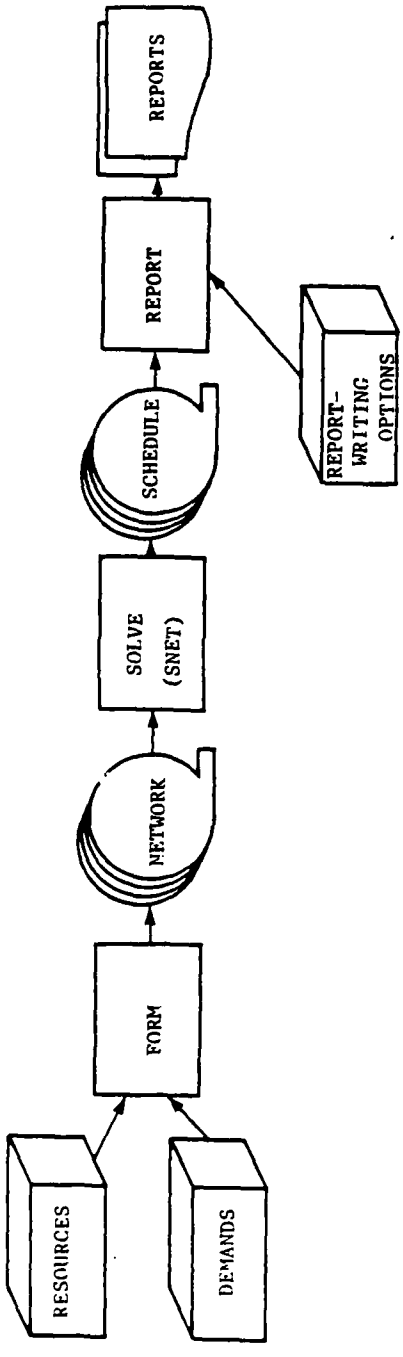


Figure 1

System Components

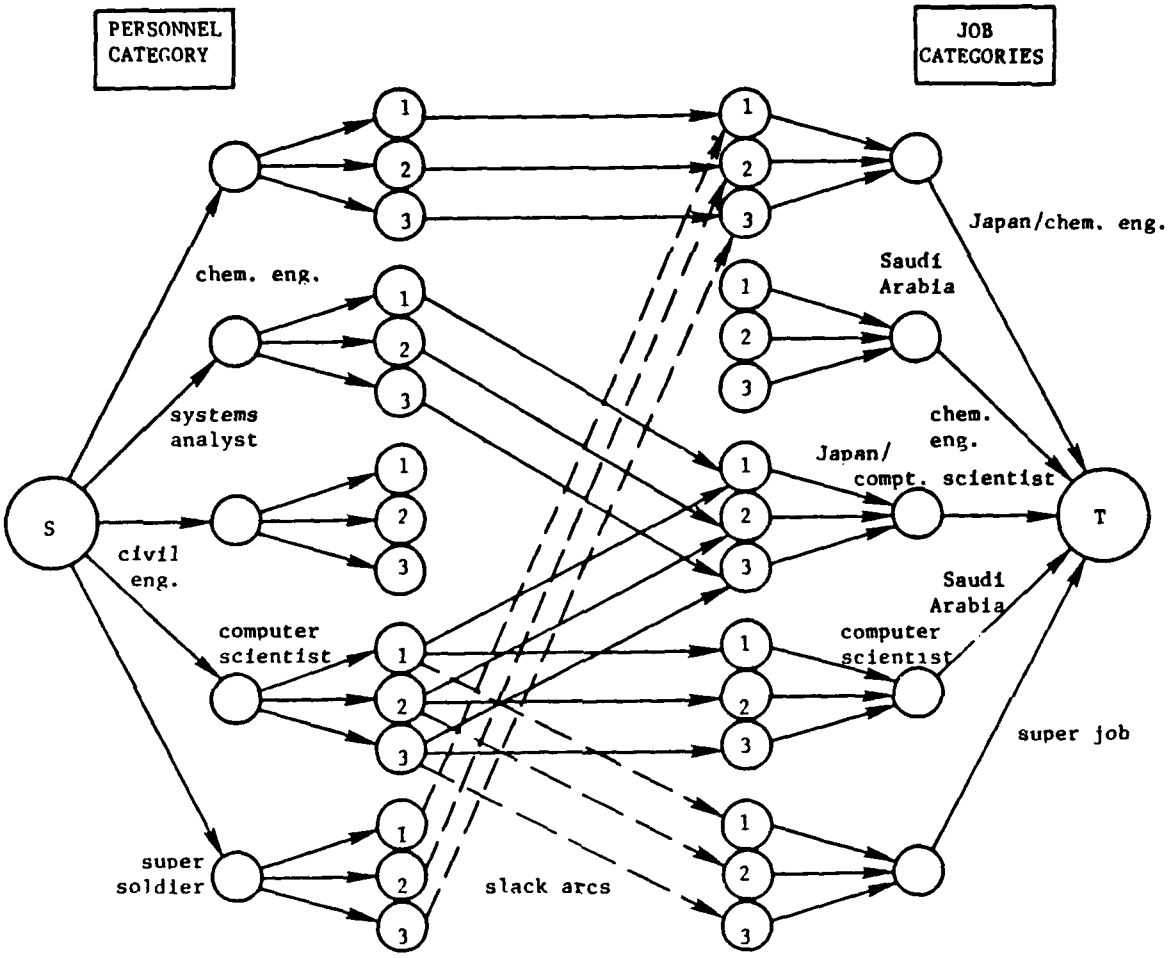


Figure 2

Personnel Planning Model

The structure of the linked-lists is shown in Figure 3. For each descriptor subset (subgrouping), there is an array which points to the first element in the list. This array is called IRES for the resource subsets, and IDEM for the demand subset. The main link-list vector is called MFTHRD. To save high-speed memory, the actual resources and demand names are not stored in the full linked-list but reside in a separate array (NAMES). This scheme saves high-speed memory but requires an extra computer reference. The maximum length of the main link-list array is fifteen hundred entries. Thus, each personnel and demand can reside in three subsets on the average because there are a maximum of 500 entries in the NAMES array. Obviously, the average number of subsets can be increased if there are less than five hundred resources/ demands. For further details of linked-list structures, see Klingman and Mulvey [5].

The SOLVE routine identifies the optimal assignment of resources to jobs based on the preference data (cost coefficients) and resource/demand restrictions (lower and upper bounds on flow). At present, there are no user inputs for SOLVE, and the only printed output is the value of the objective function at the optimal solution or an infeasibility message, whichever is appropriate.

SOLVE employs a modification of the primal-dual network algorithm SUPERK that is described in Barr, Glover, and Klingman [1]. This algorithm can easily restart from any point (feasible or infeasible) provided a circulation condition is observed. The subroutine requires nine arc-length and four node-length arrays during execution. Unfortunately, all of these arrays must remain in high speed memory; hence, an efficient external version of the system seems unlikely. In the current implementation, the program has a limitation of 1000 nodes and 3500 arcs resulting in 144K byte-length arrays and a 244K byte-length total code which is quite manageable on most mainframe computers.

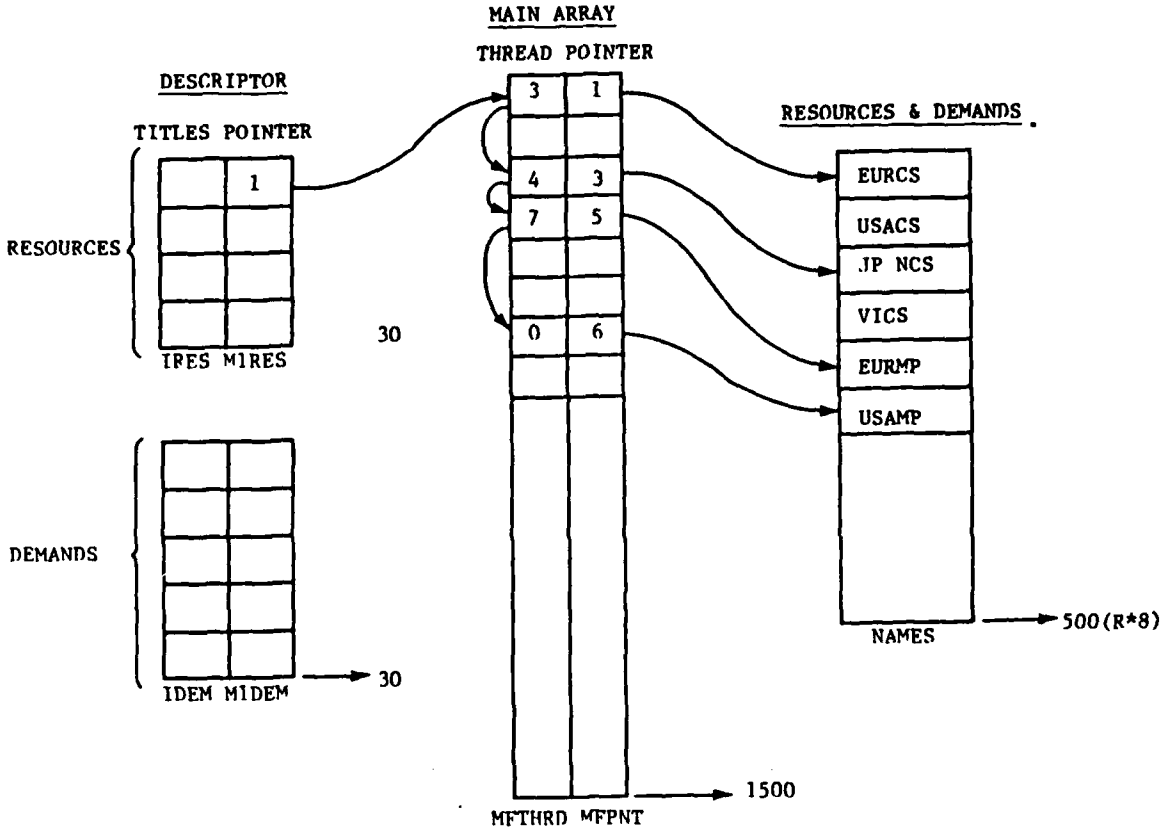


Figure 3

Linked List Structure

The third major subroutine, REPORT, takes the optimal network and the informational data base and develops the personnel schedule for output printing according to user input settings. Multiple printing options are available (see section 2). Furthermore, REPORT is designed as a multi-pass routine.

Several basic files make up the personnel scheduling system. The major components of the input data file are displayed in Figure 4. The initial elements consist of the namelist parameters. This data is followed by demand-related and then resource-related information. The report writing data occupies the last elements in the input data deck.

We have subdivided the remaining description of this users manual into 3 subsections. In the first subsection, the inputs to FORM are examined in the context of an example problem. Next, the inputs to REPORT are detailed, including the subset printing feature. Here the four principal types of printing functions are described in terms of an example. Lastly, the error detection and warning devices are prescribed. We also provide a brief discussion of important input data and its effect on the final schedule.

1. Inputs to the FORM Subroutine

The procedure for describing user inputs is the following. First, a listing of the data is provided which includes a description of type of data and the various options. An example is shown in which the exact input formats are described. Finally the delimiters, if any, are listed.

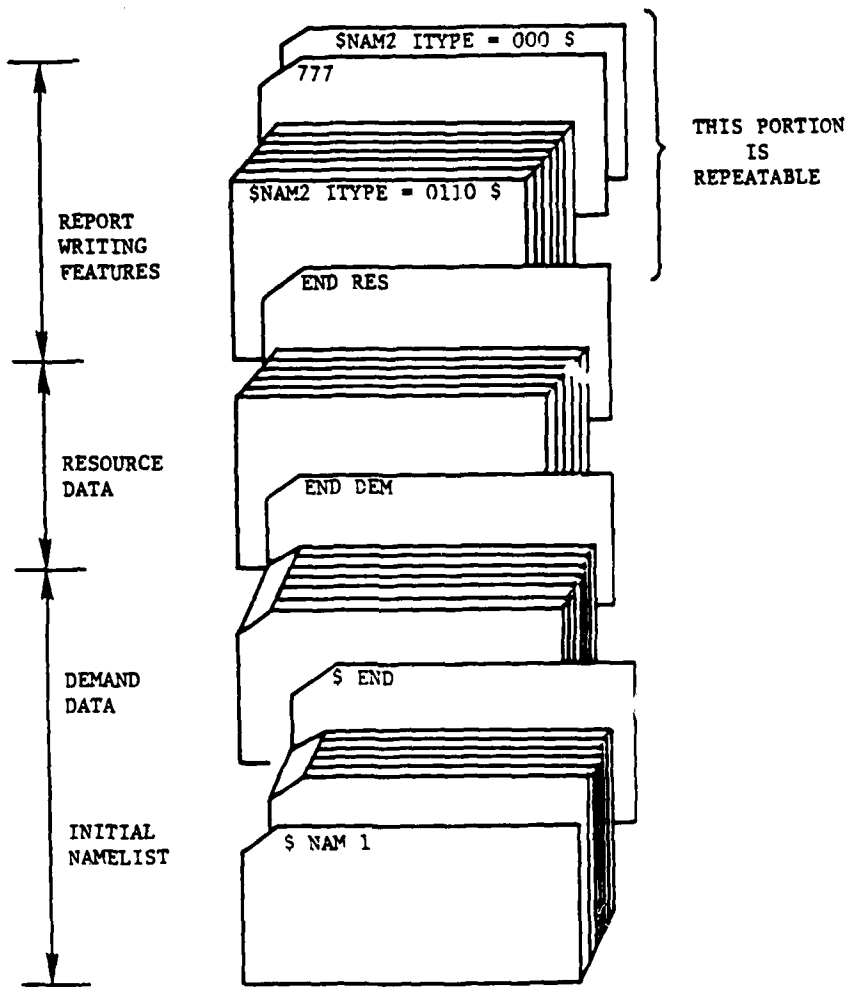


Figure 4

Basic Input Job Deck

1.1 $NAM1 - NAMELIST

The first data category is the namelist NAM1, which is used to alter basic network data -- in particular the names (titles) of the linked list arrays which identify the subsets -- and to engage the debug options (flags). It should be mentioned that namelist is a free format method for inputing data, and variables are separated by blanks or commas. The following parameters are provided in this namelist:

IRES : identifies resource subsets

default values:

```
DATA  IRES              /3HALL,5HCMPTR,5HMOTOR,6HHEALTH,
X 5HOTHER ,25*4H    /
```

NUMRS: number of consecutive non-blank entries in IRES vector

default value: 23

IDEM: identifies demand subsets

default values:

```
DATA  IDEM              /3HALL,5HCMPTR,5HMOTOR,6HHEALTH,
X5HOTHER,3HEUR,3HUSA,3HJPN,2HV1,3HEN1,3HEN2,3HEN3,18*4H
```

NUMDM: number of non-blank entries in IDEM vector

default value: 23

IACT(20): identifies appropriate names for subsets of the resources and demands (called activities)

default values:

DATA IACT /3HEN1,3HEN2,3HEN3,7*4H /

NUMACT: number of activities

default values: 8

FL: logical debug flags (except for FL(15) which is a printing option)

default values:

DATA FL /15*.FALSE./

The sole restriction regarding the construction of subset descriptions is that the universal subset must be the first entry in the IRES and IDEM arrays.

It should also be mentioned that if the fifteenth element of the FL vector is set to true, the program skips the scheduling mechanism, SOLVE, and output consisted exclusively of data from the updating capabilities. A sample namelist is shown here

&NAM1 IDEM(13)=3HOTHER

column 2 delimiter

In this example, all default parameters will be used except for the 12th element in the IDEM vector which is changed from the heading VACANT to the heading OTHER. It is important to note that a namelist card must be included in input even if it is totally blank.

1.2 Demand Related Inputs

The second partition consists of demand-related information including the following:

1. A complete list of all jobs to be offered
2. For each job
   a. Number of offerings
      1) total (min, max)
      2) per activity (min, max)
   b. Relevant subset descriptors (e.g., Europe)
3. Super Jobs
   a. One that can be assigned to any person
   b. Others that can be assigned to any person in some particular group (e.g., computer specialist)

Job names should appear in alphabetical order or in some other suitable scheme since output will occur in the sequence that they are input. Also, since names are unique identifiers, they cannot be duplicated within one run. The lower and upper bound restrictions (min, max) are checked for consistency as described in Section 3.

An example of job-related data with 12 real jobs and 9 super jobs is listed in Figure 5 . Consisting of three parts, the input specifications for the real jobs are shown in Figure 6 . The first part (card #1) defines the name of the job -- EURCS -- and the number of subsets that belong to this job--2. The next part (card #2) lists the names of the subsets -- EUR, and CMPTR. The third part (cards #3 and 4) provide information regarding the arc bounds: card #3 indicates the lower and upper bounds for the total flow,--12,12--,whereas card #4 indicates lower and upper bounds for flow on each activity arc--3,3,4,4,5,5. Additional cards may be needed when the number of activities is increased.

A critical input category is the super job. These are distinguished by the name SINK in the first column and have the characteristic that all people within the appropriate subset are able to be assigned to this job. Figure 7 provides an example of the input specifications. Again, three parts are defined. The first part (card #1) indicates (1) that the job is a super job (SINK), (2) that there are x subsets -- 3 -- to follow on the next card, and (3) the name of the super job -- CSEN1$. The second part (card #2) simply lists the subsets for this particular job--CMPTR, EN1,OTHER. Card 3 defines the resource subset that will connect to this super job--CMPTR--, upper bound for all arcs pointing into this node -- 45 --, preference cost for inward pointing arcs --$-6 --, relevant activity -- EN1 --, preference cost for outward pointing arc -- 0 --, and upper bound for outward pointing arc -- 45. It is assumed that all lower bounds for these arcs are equal to zero.

The final delimiter for the demand-related data is the job name "END DEM".

```
EURCS    2
EUR       CHPTR
         12          12
          3           3          4          4          5          5
USACS    2
USA       CHPTR
         18          18
          4           4          5          5          9          9
JPNCS    2
JPN       CHPTR
          5           5
          0           0          2          2          3          3
VICS     2
VI        CHPTR
          3           3
          0           0          1          1          2          2
EURNP    2
EUR       NOTCH
          8           8
          5           5          2          2          1          1
USANP    2
USA       NOTCH
         10          10
          6           6          3          3          1          1
JPNNP    2
JPN       NOTCH
          2           2
          1           1          0          0          1          1
VINP     2
VI        NOTCH
          2           2
          0           0          1          1          1          1
EURHOS   2
EUR       HEALTH                                                              12
          4           4
          1           1          3          3          0          0
USAHOS   2
USA       HEALTH
          8           8
          3           3          4          4          1          1
JPNHOS   2
JPN       HEALTH
          1           1
          0           0          1          1          0          0
VIHOS    2
VI        HEALTH
          0           0
          0           0          0          0          0          0
SINK     3            CSEN1S
CHPTR     EN1         CTHEA
 CHPTR   45          -6           EN1        0          45
 SINK     3           CSEN2S
 CHPTR     EN2       OTHER
 CHPTR   45          -6           EN2        0          45
 SINK     3           CSEN3S
 CHPTR     EN3       OTHEA
 CHPTR   45          -6           EN3        0          45
 SINK     3           RPEN1S
 NOTCH     EN1       CTHEA
 NOTCH   45          -7           EN1        0          45
 SINK     3           RPEN2S
 NOTCH     EN2       OTHEA
 NOTCH   45          -7           EN2        0          45
 SINK     3           RPEN3S
 NOTCH     EN3       OTHEA
 NOTCH   45          -7           EN3        0          45
 SINK     3           HOSEN1S
 HEALTH    EN1       CTHEA
 HEALTH  45          -7           EN1        0          45
 SINK     3           HCSEA2S
 HEALTH    EN2       OTHER
 HEALTH  45          -7           EN2        0          45
 SINK     3           HCSEN3S
 HEALTH    EN3       OTHEA
 HEALTH  45          -7           EN3        0          45
 END CHN
```

Figure 5

Total Demand Related Inputs

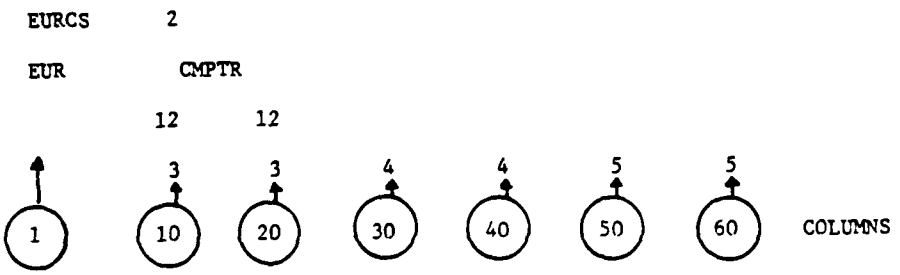


Figure 6

Input Specifications for Real Jobs

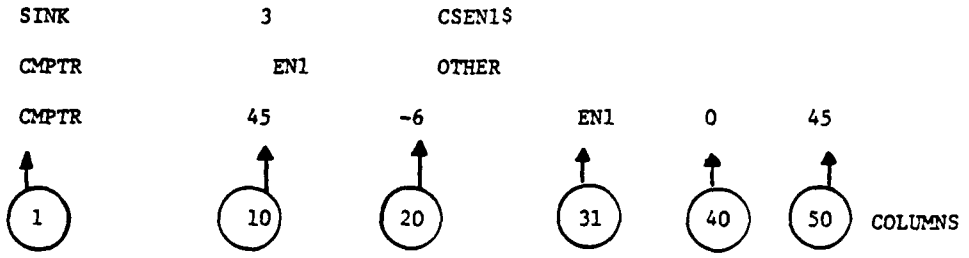


Figure 7

Input Specifications for Super Jobs

1.3 Resource Related Inputs

The following parameters are relevant for resource-related inputs:

1. Complete list of all resource categories to be assigned
2. For each category
   a. A list of the jobs that are eligible to be assigned
      1. From historical records
      2. From other inputs, including requests from the relevant administrations
   b. Preferences for assignment for each job on some numerical scale (e.g., -2 to +2)
   c. Number of people of each job that the resource category is willing to be assigned.
   d. Supply limitations
      1. total (min, max)
      2. per activity (min, max)
   e. Relevant subset groupings (e.g., European command, computer specialists)

Similar restrictions apply to the resource subset descriptions and to the demand subgroups. That is, any subset can be used, provided that the name is preset in IRES and the universal subset is the first entry in IRES.

The resource data is input in a manner similar to the demand-related data, with the addition of a menu of jobs and related preferences. An example of 6 resource categories and 3 super categories is detailed in Figure 8 .

```
CONSCI   1          4
CHPTR
             9          9
             6          6          3          3          3          3
EUBCS       +3          3         +3          3         +3          3
USACS       +4          4         +4          4         +4          4
JPNCS       -1          2         -1          2         -1          2
VICS         0          1          0          1          0          1
SYSANST      1          4
CHPTR
            15         15
             0          0          5          5         10         10
EUBCS       +4          5         +4          5         +4          5
USACS       +5          9         +5          9         +5          9
JPNCS        0          3          0          3          0          3
VICS        -1          2         -1          2         -1          2
TRRRPR       1          4
NOTCH
            20         20
            10         10          7          7          3          3
EUBHP       +2          4         +2          4         +2          4
USAHP       +5          6         +5          6         +5          6
JPNHP        0          1          0          1          0          1
VIHP         0          1          0          1          0          1
AIRRPR       1          4
NOTCH
            10         10
             5          5          3          3          2          2
EUBHP       +3          5         +3          5         +3          5
USAHP       +5          5         +5          5         +5          5
JPNHP        0          1          0          1          0          1
VIHP         0          1          0          1          0          1
DOC          1          4
HEALTH
             5          5
             1          1          2          2          2          2
EURHOS       0          2          0          2          0          2
USAHOS      +5          2         +5          2         +5          2
JPNHOS      -1          1         -1          1         -1          1
VIHOS       -1          1         -1          1         -1          1
BASE         1          4
HEALTH
             7          7
             3          3          2          2          2          2
EURHCS       0          2          0          2          0          2
USAHOS      +5          5         +5          5         +5          5
JPNHOS      -1          1         -1          1         -1          1
VIHCS       -1          1         -1          1         -1          1
CHPTR        2          4
CHPTR      OTHER
             0         30
             0          5          0         10          0         15
EUBCS       -3         15         -3         15         -3         15
USACS       -3         15         -3         15         -3         15
JPNCS       -3         15         -3         15         -3         15
VICS        -3         15         -3         15         -3         15
MOTOR        2          4
MOTOR      OTHER
             0         30
             0          5          0         10          0         15
EURHP       -2         15         -2         15         -2         15
USAHP       -2         15         -2         15         -2         15
JPNHP       -2         15         -2         15         -2         15
VIHP        -2         15         -2         15         -2         15
HEALTH       2          4
HEALTH     OTHER
             0         30
             0          5          0         10          0         15
EURHCS      -5         15         -5         15         -5         15
USAHOS      -5         15         -5         15         -5         15
JPNHCS      -5         15         -5         15         -5         15
VIHOS       -5         15         -5         15         -5         15
END RES
```

Figure 8

Total Resource Related Inputs

The input data is subdivided into 4 parts for each resource category. In part 1 (card #1), the name of the category is specified -- COMSCI*, followed by the number of subsets -- 1, and followed by the number of relevant jobs -- 5. Part 2 (card #2) contains the names of the subsets -- CMPTR. Part 3 (cards #3 and #4) indicates the supply limitations for the total category (card #3 -- 9,9), and the supply limitations for each activity (card #4), -- 6,6,3,3. Finally, part 4 lists the relevant jobs with the accompanying preference cost and upper bound (for each activity). For example, the third job for "COMSCI" is "JPNCS" with cost = $-1, bound = 2.

It should be stressed that the relevant jobs can be defined with regard to subsets. Therefore, the definition of a super-person is simply a matter of specifying the appropriate subset. The input format is identical to that of the real resource categories.

The final delimiter for the resource-related data is the title "END RES" in columns 1-7.

2. Report Writing Features

The final component REPORT provides the following four basic types of output:

1. a listing by resource category showing the total assignments schedules by activity, (ITYPE=1000)
2. a listing by demand category showing the people assigned to it by activity (ITYPE=0100)

* First resource category in Figure 8

3. for each activity, a listing by resource category showing the assignments, and (ITYPE=0010)
4. for each activity a listing by job showing the resource categories assigned to it. (ITYPE=0001)

Any or all of these four types of outputs can be obtained for any specified subset of the resources and of the demands -- provided the subset has been previously defined in the FORM routine. The report writer continues printing as long as data exist in the input file. Each pass contains the following elements:

```
                $NAM2       IRES = ?      IDEM = ?      FL = ?
                            NUMDM = ?     NUMRS = ?     JRUN = ?
                            IACT = ?      ITITLE = ?    ITYPE = ?
COLUMN     2                NUMACT = ?    MLINES = ?    JTITLE = ?
                            $END
-------------------------------------------------------------------
                RESOURCE    VAR1          VAR2          VAR3    ...
                DEMANDS     VAR4          VAR5          VAR6    ...
                ACTIVITY    VAR7          VAR8          VAR9    ...

COLUMN     1                10            20            30
```

The 12 parameters in the NAM2 namelist define various control settings for the report writer: ITYPE defines the type of run and its characteristics are shown above (any combination of runs is possible by adding ITYPE values, for example, a type 1 and type 2 run is output by setting ITYPE=1100); MLINES defines the number of lines per pages, depending upon the printer; and ITITLE defines the title for each page in the output.

The default values: for these namelist parameters are shown below:

```
      DATA NUMRS /23/,NUMDM/23/,NUMACT/ 8/,JRUN/1/
C
      DATA IDLNK /8H        /,MLINES/60/
C
      DATA ITYPE/1111/
C
C
      DATA  ITITLE /'PERSONNE','L       ','SCHEDULL',' FOR    ',
     X' 1983-84',3*1H /
C


      DATA  JTITLE            /  '  ACT1  ','  ACT2  ','  ACT3  ',
     X'  ACT4  ','  ACT5  ','  ACT6  ','  ACT7  ','  ACT8  ','  ACT9  ',
     X'  ACT10 ','  ACT11 ','  ACT12 ','  ACT13 ','  ACT14 ','  ACT15 ',
     X'  ACT16 ',
     X'  ACT1  ','  ACT2  ','  ACT3  ','  ACT4  ','  ACT5  ','  ACT6  ',
     X'  ACT7  ','  ACT8  ','  ACT9  ','  ACT10 ','  ACT11 ','  ACT12 ',
     X'  ACT13 ','  ACT14 ','  ACT15 ','  ACT16 ',16*'  JOBS  ',
     X    16*'  RES  ' /
```

After this namelist has been input, the resource categories to be output are identified on a special card called a RESOURCE card. Several possibilities exist: any individual resource category can be listed as well as any subset of the resources (as identified by the linked lists) and any combination thereof. Following this data, the jobs are specified on a similar card which is called a DEMANDS card. Finally, the activities can also be identified on an ACTIVITY card. A 777 card is a delimiter for this data group.

The following example demonstrates some of the input characteristics of the input writer:

```
$NAM2          ITYPE=1111,MLINES=60          $END
RESOURCE       ALL
DEMANDS        ALL
ACTIVITY       EN1          EN2       EN3
777
COLUMNS→ 1          10          20          30
```

and the respective sample output for the example appear in Figures 9 through 12 corresponding to type 1, 2, 3, and 4 output (since ITYPE=1111).

PERSONNEL SCHEDULE FOR 1983-84

PAGES = 1

ACT1	ACT2	ACT3	ACT4	ACT5	ACT6	ACT7	ACT8
********** COMSCI **********							
EURCS	EURCS						
2							
	2						
USACS	USACS						
4							
	1						
********** SYSANSC **********							
	EURCS	EURCS					
	1						
		1					
	USACS	USACS					
	4						
		9					
********** TEKRPS **********							
USAMP	USAMP	USAMP					
6							
	3						
		1					
JPNMP	VIMP	JPNMP					
1							
	1						
		1					
JPEN1S	JPEN2S	VIMP					
3							
	3						
		1					
********** AIRPS **********							
EURMP	EURMP	EURMP					
5							
	2						
		1					

Figure 9

Type 1 Print

```
                PERSONNEL        SCHEDULE FOR      1983-84
                                 PAGES =     2

  ACT1     ACT2     ACT3     ACT4     ACT5     ACT6     ACT7     ACT8


***********    EURCS       **********
  COMSCI   COMSCI  SYSANST

      2
               2
                        1
  CMPTR    SYSANST CMPTR


      1
               1
                        4
        .   CMPTR
               1


***********    USACS       **********
  COMSCI   COMSCI  SYSANST

      4
               1
                        9
           SYSANST
               4


***********    JPNCS       **********
             CMPTR    CMPTR

               2
                        3


***********    VICS        **********
             CMPTR    CMPTR

               1
                        2
```

Figure 10

Type 2 Print

```
PERSONNEL        SCHEDULE FOR      1983-84
                 PAGES =     5
                 EN1



***********   COMSCI      ***********
  EURCS   USACS

      2
                4

***********   TRKAPR      ***********
  USAMP   JENMP   MPEN15

      6
                1
                          3
***********   AIRRPR      ***********
  EURMP

      5

***********   DOC         ***********
  USAHOS

      1

***********   NRSE        ***********
  EURHOS  USAHOS

      1
                2

***********   CHPTR       ***********
  EURCS
```

Figure 11

Type 3 Print (EN1 Activity)

EN 1

```
***********    EURCS        **********
  COMSCI  CMPTR

      2
                1

***********    USACS        **********
  COMSCI

      4

***********    EURMP        **********
  AIRBPR

      5

***********    USAMP        **********
  TRKBPR

      0

***********    JPNMP        **********
  IRKBPR

      1

***********    EURHCS       **********
  NRSE

      1

***********    USARCS       **********
  DCC       NRSE

      1
                2

***********    MPRNT        **********
  IRKBPR

      3
```

Figure 12

Type 4 Print (EN1 Activity)

3. Error Detection Analysis

Various conditions of the inputs are checked for consistency and errors are identified. Warning messages indicate non-fatal conditions; however, these should be checked by the user. Several types of inconsistencies are fatal. For example, a minimum flow of 5 and a maximum flow of 4 on the same arc cannot be processed by SOLVE and is deemed as a fatal error.

Any time that an infeasible network occurs, the final printed results will always be in error since the SOLVE subroutine immediately terminates upon detection of an infeasible arc. In this situation, the user is encouraged to search the output for inconsistency messages or other error or warning comments. If this search does not indicate the source of the error, the network is probably infeasible because the super people and/or the super jobs do not have enough capacity or do not link with the appropriate resource categories. The easiest method to remedy this situation is using a single super person (ALL) and a single super job (ALL) with a very large capacity, although this scheme is inefficient and may cause boundary overflows.

Another potential problem involves the array dimensions. The following limits cannot be exceeded:

NAMES: 500 people + jobs	
number of resource subsets:	30
number of job subsets:	30
number of periods:	16
number of nodes in the network:	1000
number of arcs in the network:	3500

unless a programming change is made to the FORTRAN source file.

Several debug flags are provided to assist a knowledgeable user in tracing difficulties, but these should be used only as a last resort:

FLAG	APPLICABLE PORTION OF THE CODE & COMMENTS
FL(1)	FORM - duplicates network inputs
2	not used
3	more details in FORM (network arcs)
4	arrays for SOLVE (all input network arrays)
5	
6	results from SOLVE (final flow array)
7	report writing inputs (duplicates reporting writing inputs)
8	
9	detailed report writing arrays
10	lists arrays for SOLVE
11	not used
12	not used
13	not used
14	not used
15	see section 1

We next turn to a brief discussion of procedural concerns. The worth of the final schedule depends upon a variety of factors. One of the most important is the menu of job for each personnel category. This listing must represent all of those jobs which the scheduler is willing to place the resource category in, since preference cannot override an arc that does not occur. By this we mean that a resource category will never be assigned to any job which are not in his list. Hence the importance of a proper menu. Therefore, the scheduler should carefully review the list of relevant jobs for each resource category.

Priority schemes can easily be incorporated into this system by biasing the preference weights for certain individuals, such as officers (although this procedure is not directly used in most environments). Likewise, important jobs which are not being assigned can also be biased through the weights. For instance, the European assignments can be allowed a preference weight of -99

to +99, while the other jobs are allowed a preference range of -10 to +10. In this situation, the European assignments overrides the other preferences since the objective function is maximizing the sum of the preferences. In effect, the European jobs are scheduled ahead of all other jobs with this scheme. See references [2,3,6,7,8] for descriptions of multiple objective methods.

4. Advanced Feature

In addition to the aforementioned printed outputs, the report writer generates a compact version of the final assignments. This file is saved on disk (or tape) unit #2. Four vectors are included:

IDIFF1(·): resource (personnel category)

IDIFF2(·): demand (job category)

IDIFF3(·): activity

IDIFF4(·): arc flow

Each element in these arrays corresponds to a particular assignment arc in the network. Thus, the j^{th} position will provide a (resource-demand-activity-flow) combination. To save storage, we employ pointers to the NAMES array for the first three of these vectors, i.e., IDIFF1, IDIFF2, and IDIFF3. The fourth vector -- IDIFF4 -- consists of numbers, indicating arc flow.

APPENDIX E

ERRORS, WARNINGS, AND OTHER MESSAGES

ERROR 1 THE ISSUE FILE IS EMPTY.

Conditions: No valid ISSUE definitions were found in the ISSUE file.

Result: File must be corrected and the Set ISSUE program must be rerun.

ERROR 2 ROLMOS/PROCESS 1.1.2 SHOULD BE REPROGRAMED. THE STRENGTHS AT THE 3-DIGIT UIC LEVEL ARE LARGER THAN ALLOWED FOR BY THE FORMAT.

Conditions: The number of jobs is larger than can be written with the format I6.

Result: The computer system will set the number of jobs to asterisks (*) when the number is greater than 999,999.

ERROR 3 THE MINIMUM PERCENTAGE IN THE VALUE FILE MULTIPLIED BY THE NUMBER OF JOBS CREATED A RECORD OF ALL ZEROES.

Condition: When the number of jobs to be filled is very small, the result of the multiplication may round down to zero. For example, if there is only one job to be filled, a percentage of less than 50 percent will round down to zero.

Result: The job record is written to the Unfilled Job file instead of to the Job Assignment Value file.

ERROR 4 GRADE IN JOB DATA IS NOT IN PARAMETER FILE.

Condition: The Parameter file did not specify the grade code found in the MOS data as a valid grade.

Result: The data with the invalid grade is not written to the output file.

ERROR 5 ISSUE DATA NOT COMPLETE. ISSUE NOT FOUND FOR UIC.

Condition: If the message is from process 1.2.3, the UIC record specified does not belong to a valid ISSUE, i.e., the ISSUE definitions were not complete. The specified unit was not included in any ISSUE definitions. If the message is from process 1.2.5, there was an MOS record which did not get the ISSUE set; the UIC record may have been missing from the input file (file 22).

Result: The unit for which the ISSUE was not defined will have the ISSUE set to "----", the "invalid" designation.

ERROR 6 PROGRAMING MUST BE ADDED TO PROCESS 1.2 TO SET ISSUES FOR ID1 and ID2.

Condition: ISSUE definitions of ID1 and ID2 were not one of the valid combinations from Table B-1, ISSUE Identification Methods.

Result: No units will have the ISSUE set according to this definition.

ERROR 7 NO UIC-DATA RECORD FOUND FOR ISSUE.

Condition: An ISSUE definition did not match any data in the UIC-data file.

Result: The definition is otherwise ignored.

ERROR 9 ISSUE ON JOB FILE NOT ON CURRENT ISSUE FILE.

Condition: During the writing of the formatted readiness reports, the ISSUE codes found on the readiness files (which come from the Job file) are matched with the ISSUE Definition file to get the ISSUE name. If an ISSUE code is not found, this message is written, and the subroutine ISSUNA provides an error indication to the report writing program.

Result: The report processor stops for the user to provide the correct ISSUE Definition file.

ERROR 10 ISSUE ON POLICY FILE NOT DEFINED IN ISSUE FILE.

Condition: The ISSUE and Policy files are incompatible. All ISSUEs on the Policy file must be defined on the ISSUE file.

Result: The Policy record is ignored.

ERROR 11 LOW GRADE IN POLICY FILE IS BLANK.

Condition: The low grade in the Policy file is blank.

Result: The Policy record is ignored.

ERROR 12 HIGH GRADE IN POLICY FILE IS BLANK.

Condition: The high grade in the Policy file is blank.

Result: The Policy record is ignored.

ERROR 13 LOW GRADE IN POLICY FILE IS INVALID.

Condition: The low grade in the Policy file must be E1 through E9, WO, or O1 through O6 and must be specified in the Parameter file.

Result: The Policy record is ignored.

ERROR 14 HIGH GRADE IN POLICY FILE IS INVALID.

Condition: The high grade in the Policy file must be E1 through E9, WO, or O1 through O6 and must be specified in the Parameter file.

Result: The Policy record is ignored.

ERROR 15 HIGH GRADE IN POLICY FILE IS LESS THAN LOW GRADE.

Condition: The high grade in the Policy file must be the same as or larger than the low grade in the Policy file.

Result: The Policy record is ignored.

ERROR 16 MINIMUM IN POLICY FILE IS GREATER THAN 2.0. MINIMUM USUALLY 1.0.

Condition: The minimum percentage is greater than 200 percent; the usual minimum percentage is less than 100 percent.

Result: The Policy record is applied using the rate that was input.

ERROR 17 MAXIMUM IN POLICY FILE IS GREATER THAN 2.0.

Condition: The maximum percentage in the Policy file is greater than 200 percent; the usual maximum percentage is near 100 percent.

Result: The Policy record is applied using the rate that was input.

ERROR 18 MINIMUM IN POLICY FILE GREATER THAN MAXIMUM ON POLICY FILE.

Condition: The minimum in the Policy file is greater than the maximum.

Result: The Policy record is ignored.

ERROR 19 VALUE IN POLICY FILE LESS THAN ZERO.

Condition: A negative value was found in the Policy file.

Result: The negative value is applied to the MOS data.

ERROR 20 CHECK VALIDITY OF MINIMUM AND MAXIMUM PERCENTAGES ON POLICY FILE FOR ISSUE.

Condition: The minimum percentage and/or the maximum percentage on the Policy file is invalid for the specified ISSUE.

Result: The Policy record is ignored.

ERROR 21 AGGREGATE VALUE SHOULD NOT BE BLANK, VALID VALUES ARE YE (FOR YES) OR NO.

Condition: AG= on the Policy file must be followed by NO for a non-aggregated policy or by YES for an aggregated policy.

Result: The Policy record is ignored.

ERROR 22 INVALID AGGREGATE. THERE IS AN INVALID OVERLAP OF GRADES USING MULTIPLE AGGREGATES FOR ONE ISSUE.

Condition: The same grade may not be included in more than one specification for the same ISSUE, MOS, or combined ISSUE/MOS.

Result: All but the first specification will be ignored. Aggregate-ISSUE level specifications are considered first. A later specification for the same grade(s) for a specific ISSUE will be ignored.

ERROR 23 UNABLE TO SET MINIMUM AND MAXIMUM FILL LEVELS FOR ISSUE.

Condition: Either the minimum percentage or the maximum percentage in the Value file is invalid.

Result: Record is not written to the JAV file. Correct the Value file and rerun.

ERROR 24 THE JOB-ASSIGNMENT-VALUE FILE IS EMPTY.

Condition: No valid entries found in Value file, File 19.

Result: No additional records were written to the Job Assignment Value file (40).

ERROR 25 INCORRECT FORMAT FOUND IN VALUE FILE FOR ISSUE.

Condition: Data in Value file is not in the specified format.

Result: No additional records for the specified ISSUE were written to the JAV file.

ERROR 26 UNABLE TO SET ASSIGNMENT VALUES FOR ISSUE.

Condition: When written from Apply Policy, the number of MOS per ISSUE was greater than 676; when written from Set Base Values, there was not a valid Value record for this ISSUE.

Result: Record is not written to the JAV file. Check output from Set ISSUE, 1.2. If more than 676 MOS are in one ISSUE, delete or correct errors in the MOS file. Make sure ISSUE codes match (all four characters) in the ISSUE and Value file.

ERROR 27 POLICY FILE IS EMPTY.

Condition: No valid entries were found in the Policy file.

Result: All JAV records will be based on the data from the Value file.

ERROR 30 INVALID GRADE ON PARAMETER FILE.

Condition: The grade location in the Parameter file was not a 2-digit integer with a value of 16 or less.

Result: The location is not included as a valid location.

ERROR 31 INVALID LOW GRADE VALUE ON PARAMETER FILE. VALUE SHOULD BE INTEGER.

Condition: Low grade must be an integer value greater than zero and less than 16.

Result: The program cannot correctly set the subscript values for storing the data by grades. The error must be corrected.

ERROR 32 INVALID HIGH GRADE ON PARAMETER FILE. VALUE SHOULD BE INTEGER AND EQUAL TO NGRADE.

Condition: High grade must be an integer value greater than zero and less than or equal to 16.

Result: The program cannot correctly set the subscript values for storing the data by grades. The error must be corrected.

ERROR 34 GRADE IN POLICY FILE IS NOT IN PARAMETER FILE

Condition: One of the grades on the specified policy record does not match the valid grade codes entered in the Parameter file. A policy cannot be applied to grades that do not exist in the input data, and the data will contain only grades specified in the Parameter file.

ERROR 35 INVALID GRADE OVERLAP. ONE OF THE INCLUDED GRADES HAS BEEN USED ON A PREVIOUS POLICY FOR AGGREGATED GRADES.

Condition: One or more grades has been used in more than one policy that is of the type that is aggregated across grades. No more than one policy can be applied.

Result: The entire record containing the policy that is specified in the error message is ignored.

ERROR 36 INVALID GRADE OVERLAP. ONE OF THE INCLUDED GRADES HAS BEEN USED ON A PREVIOUS POLICY FOR NONAGGREGATED GRADES.

Condition: One or more grades has been used in more than one policy that is of the type that is not aggregated across grades. No more than one policy can be applied.

Result: The entire record containing the policy that is specified in the error message is ignored.

ERROR 37 AS OF DATE IS MISSING FROM THE PARAMETER FILE.

Condition: As stated, there is no as of date in the Parameter file or the variable name is not spelled correctly in the Parameter file.

Result: The report programs will show a blank as of date.

ERROR 39 NGRADE IS MISSING FROM PARAMETER FILE.

Condition: NGRADE is not specified correctly. Either it was not spelled correctly, not placed in the correct columns, or the integer did not convert correctly to an integer of 16 or less.

Result: The program will not correctly process any further data.

ERROR 40 NCCHAR IS MISSING FROM PARAMETER FILE.

Condition: Either NCCHAR is not in the Parameter file, it is misspelled in the Parameter file, or it is not in the right columns in the Parameter file.

Result: The number of characters cannot be compared in the Assignment program. All programs using NCCHAR must be rerun.

ERROR 41 TYPSTR IS MISSING FROM PARAMETER FILE - USER MUST SPECIFY WHETHER STRENGTH SHOULD BE AUT OR REQ.

Condition: TYPSTR is either not in the Parameter file, is misspelled in the Parameter file, is not in the right columns in the Parameter file, or the AUT and/or REQ is missing.

Result: The Policy programs cannot continue. Correct the Parameter file and rerun.

ERROR 42 NCCHWO IS MISSING FROM PARAMETER FILE.

Condition: Either NCCHWO is missing, misspelled, or in the wrong columns in the Parameter file.

Result: ROLMOS program cannot work correctly. Correct the Parameter file and rerun the ROLMOS program.

ERROR 43 THERE WERE NO ISSUES DEFINED BY UIC ON THE ISSUE DEFINITION FILE (warning message).

Condition: There was an ISSUE definition in File 14 that was not used by any of the data in the files.

Result: The ISSUE definition was ignored. If it should have been used, look at the ISSUE in File 14, find the error, correct, and rerun the Set ISSUE program.

ERROR 44 INVALID MOS SPECIFICATION ON POLICY FILE. NUMBER OF CHARACTERS SPECIFIED IS LARGER THAN NUMBER OF CHARACTERS SPECIFIED ON PARAMETER FILE (warning message).

Condition: The Policy stated in the Policy file contained a larger number of characters in the MOS than had been specified in the Parameter file.

Result: This Policy will be ignored unless the Policy file is corrected and the Apply Policy programs rerun.

ERROR 45 THE SUM OF THE MAXIMUM FILL OF GRADES IS LESS THAN THE MINIMUM AGGREGATE FILL.

Condition: At least two policy records were taking effect at the same time. One would have been for the aggregate of several grades, and the other would have been for the grades within that aggregate having a different minimum percentage fill. When the sum of the maximum fill of the individual grades is less than the minimum required fill for the aggregate of those grades, the Assignment programs will give an infeasible arc. This error must be corrected on the Policy file and the Policy programs rerun.

Result: The record is ignored.

ERROR 46 INVALID GRADE OVERLAP ON POLICY FILE. THERE IS MORE THAN ONE SPECIFICATION FOR GRADE.

Condition: More than one policy included this grade.

Result: The second policy is ignored.

ERROR 61 THE MAXIMUM IS LESS THAN THE MINIMUM ON THE VALUE FILE.

Condition: The maximum percentage fill was less than the minimum percentage fill on the Value file (File 19). This will be impossible to fill to more than the minimum stated. One record for the minimum percentage fill will be written to the Job Assignment Value file.

Result: Only the minimum is used.

ERROR 62 MAXIMUM VALUE ON VALUE FILE IS LARGER THAN THE MINIMUM (warning message).

Condition: The value for the maximum percentage would normally be smaller than the value for the minimum percentage. When it is not, this error is given as a warning.

Result: The records have been written to the Job Assignment Value file. If this was not desired, the Value file should be corrected and the Set Base Value program rerun.

ERROR 70 THE INPUT MOS FILE IS NOT SORTED CORRECTLY. ISSUES ARE NOT IN ORDER.

Condition: Each ISSUE code is checked against the previous ISSUE code. The last code must always be greater than or equal to the previous code. If not, the MOS-data file aggregation will not be performed correctly.

Result: The program will halt with the error message. The user must sort the MOS-data file on MOS within ISSUE code and rerun process 1-3, Aggregate MOS-data.

ERROR 71 THE INPUT MOS FILE IS NOT SORTED CORRECTLY. MOS NOT IN ORDER.

Condition: Each MOS is checked against the previous MOS. The last MOS must always be greater than or equal to the previous MOS. If not, the MOS-data file aggregation will not be performed correctly.

Result: The program will halt with the error message. The user must sort the MOS-data file on MOS within ISSUE code and rerun process 1-3, Aggregate MOS-data.

ERROR 96 MORE THAN 300 ISSUES ARE DEFINED ON THE ISSUE FILE.

Condition: No more than 300 ISSUE codes may be defined. The Edit Policy program found more than 300 unique ISSUE codes.

Result: Policies for all ISSUEs numbered 301 and up will be ignored.

ERROR 97 UNABLE TO WRITE UNIQUE DEMAND NODE NAME IN POLICY PROCESSOR FOR THE PRESENT DEMAND NODE IDENTIFIER.

Condition: More than 676 different MOS records were read for one ISSUE and all records beyond 676 will have the same demand node name.

Result: Results are unpredictable.

MESSAGE 98 END OF DATA FOR PROCESS.

Condition: During running of the Apply Policy modules, different sets of data are used at different points. Whenever the end of the particular set of data is reached, MESSAGE 98 is output to signal the user that the end of file was needed.

Result: No effect.

MESSAGE 99 READ OR WRITE ERROR ON UNIT.

Condition: A read or write error on the unit specified has occurred. The reason is probably that the file has disappeared, is not available for use, or the file name is misspelled on the assign and/or use statement in the runstream.

Result: The programs cannot continue. The error must be corrected and the programs rerun.

ERROR 103 MULTIPLE DEFINITION OF ISSUE FOR THIS UNIT. RESETTING ISSUE (warning message).

Condition: The unit having the ISSUE set at the moment has already been included in an ISSUE definition, and the ISSUE is being reset from the previous ISSUE to this ISSUE. If that is not desired, the ISSUE file must be corrected, and the Set ISSUE program rerun.

Result: ISSUE has been changed to reflect the new ISSUE definition. This message is a warning only.

ERROR 104 THERE IS NO SUMMARY LINE TITLE FOR ISSUES (warning message).

Condition: The Report processor was unable to find the "00" level record in the ISSUE file.

Result: This report cannot be produced unless the ISSUE file is corrected.

ERROR 113 MINIMUM ON POLICY FILE IS GREATER THAN 1.0 (warning message).

Condition: Merely a warning that the minimum percentage from the Policy file is greater than 100 percent. Normally, the minimum would be less than 1.0, but it is allowed to be more than 100 percent.

Result: The minimum is used as stated.

ERROR 125 JOB-DATA RECORD CONTAINS NO JOBS FOR ISSUE (warning message).

Condition: If this message is output from ROLMOS, it means that there were no authorized or required jobs for this ISSUE and MOS combination. In other words, both numbers were zero. If output from the Apply Policy program, it means that there were no jobs of the specific type of data requested in the Parameter file (authorized or required). There may have been required jobs, but the request was for authorized, which was zero. It also can be output after the multiplication of the Policy or Base Value minimum or maximum times the authorized or required jobs. Rounded to the nearest integer, it might turn out to be zero. In other words, fairly small numbers of people required for a job multiplied times a very small percentage may round to a zero.

Result: The data is dropped from the data base. If it should not be dropped, then the applicable file should be corrected and the programs rerun.

ERROR 126 JOB-DATA RECORD CONTAINS NO JOBS FOR UIC (warning message).

Condition: Similar to ERROR 125, the Job-data record contains a zero in the number of people authorized and required. The program that outputs this message is ROLMOS.

Result: The data is dropped from the data base by the program ROLMOS.

END

FILMED

5-85

DTIC