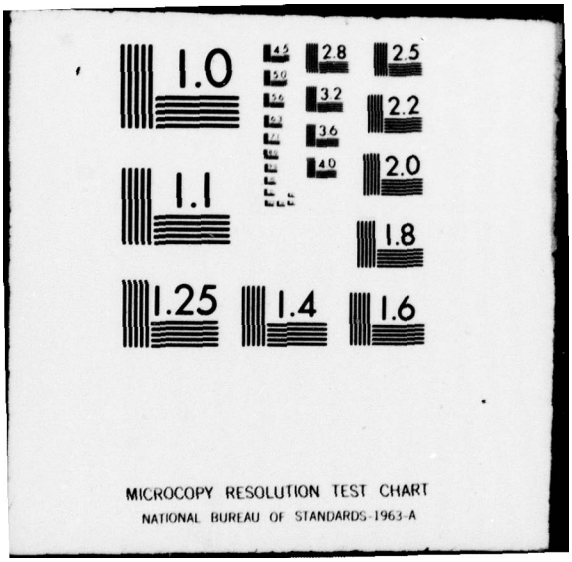
AD-A063 433 COMMAND AND CONTROL TECHNICAL CENTER WASHINGTON D C F/G 9/2
NMCS INFORMATION PROCESSING SYSTEM 360 FORMATTED FILE SYSTEM (N--ETC(U)
SEP 78 C K HILL
UNCLASSIFIED CCTC-CSM-UM-15-78-V8-REV SBIE-AD-E100 133 NL

1 OF 2
AD
A063433



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

AD A063433

DDC FILE COPY

CCTC

(12)

AD-E100 133

COMPUTER SYSTEM MANUAL
CSM UM 15-78
VOLUME VIII (REVISED)
1 SEPTEMBER 1978

LEVEL III

COMMAND & CONTROL TECHNICAL CENTER

DDC
RECEIVED
JAN 18 1979
B

NMCS INFORMATION PROCESSING SYSTEM 360 FORMATTED FILE SYSTEM (NIPS 360 FFS)

DEFENSE COMMUNICATIONS AGENCY

VOLUME VIII (REVISED) JOB PREPARATION

USERS MANUAL

THIS DOCUMENT HAS BEEN APPROVED FOR PUBLIC RELEASE AND SALE; ITS DISTRIBUTION IS UNLIMITED.

78 12 08 016

(12) 188p.

COMMAND AND CONTROL TECHNICAL CENTER

(9) Computer System Manual, CSM UM 15-78

(11) 1 Sept[redacted]78

AD63 433

(6) NMCS INFORMATION PROCESSING SYSTEM
360 FORMATTED FILE SYSTEM (NIPS 360 FFS).

User's Manual.

Volume VIII (Revised) - Job Preparation.

[redacted]

DDC
RECEIVED
JAN 18 1979
B

(14) CCTC-CSM-UM-15-78-V8-REV

(18) SBIE (19) AD-E100 133

SUBMITTED BY:

SK Hill

(10) CRAIG K. HILL
Captain, USA
CCTC Project Officer

APPROVED BY:

Frederic K. Graf, Jr.

FREDERIC K. GRAF, JR.
Captain, U.S. NAVY
Deputy Director
NMCS ADP

Copies of this document may be obtained from the Defense Documentation Center, Cameron Station, Alexandria, Virginia 22314

This document has been approved for public release and sale; its distribution is unlimited.

409658

78 12 08 016

ACKNOWLEDGMENT

This manual was prepared under the direction of the Chief for Programming with general technical support provided by the International Business Machines Corporation under contracts DCA 100-67-C-0062, DCA 100-69-C-0029, DCA 100-70-C-0031, DCA 100-70-C-0080, DCA 100-71-C-0047 and DCA 100-77-C-0065.

ACCESSION for
NTIS White Section ☑
DDC Buff Section ☐
UNANNOUNCED ☐
JUSTIFICATION
BY
DISTRIBUTION/AVAILABILITY CODES
Dist. AVAIL. and/or SPECIAL
A

CONTENTS

Section		Page
	ACKNOWLEDGMENT	ii
	ABSTRACT	vii
1	INTRODUCTION	1
2	FEATURES OF NIPS 360 FFS PROCEDURES	2
2.1	Symbolic Parameters	2
2.2	File Name Conventions	3
2.3	File Block Size Conventions	3
2.3.1	Default File Block Size	4
2.3.2	User Specified File Block Size	4
2.4	Example of File Name Conventions and Symbolic Parameters	4
2.4.1	Single Data Base	4
2.4.2	Multiple Data Bases	6
2.4.3	File Libraries	8
2.5	Catalog Requirements	8
2.6	Checkpoint/Restart	9
2.6.1	Checklist for Using Checkpoint/Restart	11
2.6.2	Sample Job Setup	11
2.7	Segmented Files	13
2.8	Source Language Storage	13
3	COMPONENT PROCEDURE DESCRIPTIONS	14
3.1	FM Procedure (XFM)	14
3.1.1	Sample Job Setup	18
3.2	FR Procedure (XFR)	27
3.2.1	Sample Job Setup	27
3.2.2	File Structure, Revision, and Maintenance	28
3.3	FS Procedure (XFS)	28
3.3.1	Sample Job Setup	28
3.3.2	Non-NIPS FS	29
3.4	ISAM to SAM Procedure (XISTOS)	30
3.4.1	Sample Job Setup	30
3.5	OP Procedures (XOP, XOPSD)	30
3.5.1	XOPSD Sample Job for Single File Processing	32
3.5.2	XOPSD Sample Job for Merge File Processing	32
3.5.3	XOP Sample Job	33
3.5.4	Segmented File Sample	34
3.5.5	Output Data Set Overrides	34
3.5.6	Additional DD Card for Tape Output	35
3.6	QRT/QDF Utility Procedure (XQRTQDF)	36
3.6.1	Sample Job Setup	36
3.7	QUIP Procedures (XQUIP,XQUIPSD)	37
3.7.1	Sample Job Setup	37

Section Page

3.7.2 Interfile Output (IFO)..... 39
3.7.3 Non-NIPS Files..... 39
3.8 RASP Procedure (XRASP)..... 40
3.8.1 Sample Job Setup..... 41
3.9 SAM to ISAM Procedure (XSTOIS)..... 44
3.9.1 Sample Job Setup..... 45
3.10 Subroutine Loader Procedure (XSUBLDR)..... 45
3.10.1 Sample Job Setup..... 45
3.11 Table Generation Procedure (XTABGEN)..... 46
3.11.1 Sample Job Setup..... 46
3.12 Source Language Library Procedure (XUTSOURC) 46
3.12.1 Sample Job Setup..... 46
3.13 1410 to 360 Conversion (X360CON)..... 47
3.13.1 Sample Job Setup..... 47
3.14 360 to 1410 Conversion (X1410CON)..... 48
3.14.1 Sample Job Setup..... 48
3.15 List Logic Statement/Report Names Procedure (XDMPLIB)..... 48
3.15.1 Sample Job Setup..... 49
3.16 Subroutine Check Procedure (XSUBCHK)..... 49
3.16.1 Sample Job Setup..... 49
3.17 Classification Change Procedure (XCLASS)..... 49
3.17.1 Sample Job Setup..... 50
3.18 Data Field Scan Procedure (XUTFSCAN)..... 51
3.18.1 Sample Job Setup..... 51
3.19 Index Specification Procedure (XSP)..... 52
3.19.1 Sample Job Setup..... 52
3.20 Unload Index Data Set Procedure (XTRDISK).... 53
3.20.1 Sample Job Setup..... 53
3.21 Load Index Data Set Procedure (XTRTAPE)..... 54
3.21.1 Sample Job Setup..... 54
3.22 Keyword Analysis (XKA)..... 54
3.22.1 Sample Job Setup..... 54
3.23 Dictionary Maintenance Utility (XKM)..... 55
3.23.1 Sample Job Setup..... 55
3.24 Format Definition Translator Procedure (XUTODE)..... 55
3.24.1 Sample Job Setup..... 55

4 SUPPLEMENTARY PROCEDURES DESCRIPTIONS..... 57
4.1 Restore Answers Procedure (XRESTANS)..... 57
4.1.1 Sample Job Setup..... 57
4.2 Restore Library Procedure (XRESTLIB)..... 58
4.2.1 Sample Job Setup..... 58
4.3 Save Answers Procedure(XSAVEANS)..... 59
4.3.1 Sample Job Setup..... 59
4.4 Save Library Procedure (XSAVELIB)..... 59
4.4.1 Sample Job Setup..... 60

Section Page

5 PRODUCTION PROCEDURES DESCRIPTION..... 61
5.1 FM Production Procedure (XFMEX)..... 61
5.2 OP Production Procedure (XOPEX, XOPSDEX)..... 61
5.3 Compression and Compaction of Data Records... 62
5.4 RASP Production Procedure (XRASPEX)..... 63

6 S/370 VSAM CONSIDERATIONS 64
6.1 VSAM Service Routine IDCAMS 64
6.1.1 Creating a NIPS User Catalog 65
6.1.2 Defining a Cluster for a VSAM File 65
6.1.3 Converting an ISAM File to VSAM 66
6.1.4 Deleting a Cluster 67
6.2 File Maintenance (FM)..... 67
6.2.1 VSAM Generate 68
6.2.2 VSAM Update 68
6.2.3 VSAM Transactions 69
6.2.4 Processing SAM Files with a VSAM FFT/LS 69
6.2.5 Other FM VSAM Considerations 70
6.3 File Revision (FR)..... 71
6.4 File Structure (FS)..... 72
6.5 VSAM to SAM Procedure (XISTOS)..... 72
6.6 OP 73
6.7 QUIP 74
6.8 RASP 75
6.9 SAM to VSAM Procedure (XSTOIS)..... 76
6.10 XDMPLIB..... 76
6.11 XCLASS 77
6.12 XUTFSCAN 77
6.13 XSP 77
6.14 XTRDISK 78
6.15 XKA 78

APPENDIX

A SYMBOLIC PARAMETER DEFINITIONS..... 80

B INPUT SOURCE DDNAME..... 89

C PROCEDURES DESCRIPTIONS..... 90

D NIPS PROCEDURE RETURN CODES..... 93

E NIPS PROCEDURE DD STATEMENT USAGE..... 95

F PROCEDURE LISTINGS..... 128

XCLASS
XDMPLIB
XFM
XFMEX

XFR
XFS
XISTOS
XKA
XKM
XOP
XOPEX
XOPSD
XOPSDEX
XQRTQDF
XQUIP
XQUIPSD
XRASP
XRASPEX
XRESTANS
XRESTLIB
XSAVEANS
XSAVELIB
XSP
XSTOIS
XSUBCHK
XSUBLDR
XTABGEN
XTP
XTRDISK
XTRTAPE
XUTFSCAN
XUTODE
XUTSOURC
X1410CON
X360CON

DISTRIBUTION..... 176
DD Form 1473..... 180

ABSTRACT

This document familiarizes the user with the features available using NIPS 360 FFS Job Preparation procedures. It describes symbolic parameters and file naming conventions, illustrates general approaches to running jobs using single or multiple data bases and file libraries. Job Control Language (JCL) examples for each system component are given.

This document supersedes CSM UM 15-~~78~~74, Volume VIII.

CSM UM 15-78, Volume VIII, is part of the following additional NIPS 360 FFS documentation.

CSM UM 15-78	Vol I	- Introduction to File Concepts
CSM UM 15-78	Vol II	- File Structuring (FS)
CSM UM 15-77	Vol III	- File Maintenance (FM)
CSM UM 15-78	Vol IV	- Retrieval and Sort Processor (RASP)
CSM UM 15-78	Vol V	- Output Processor (OP)
CSM UM 15-78	Vol VI	- Terminal Processing (TP)
CSM UM 15-78	Vol VII	- Utility Support (UT)
CSM UM 15-78	Vol IX	- Error Codes
TR 54-78		- Installation of NIPS 360 FFS
CSM GD 15-78		- General Description

Section 1

INTRODUCTION

This volume is intended to familiarize the user with the features available using NIPS 360 PPS Job Preparation procedures and the manner in which they are used.

Section 2 describes symbolic parameters and file naming conventions. It also illustrates general approaches to running jobs using single or multiple data bases and file libraries.

Section 3 contains JCL examples for each component and additional information pertinent to each procedure.

Section 4 contains JCL examples for S/360 Utilities which are used to dump and restore RASP answers and the User File Library from disk to tape and vice versa.

Section 5 describes procedures which are used in File Maintenance, Output Processing and the Retrieval and Sort Processor component.

Section 6 describes the procedures and considerations necessary to create, maintain and process NIPS data bases using the S/370 Virtual Storage Access Method (VSAM).

Section 2

FEATURES OF NIPS 360 FFS PROCEDURES

NIPS 360 FFS has single-step cataloged procedures using symbolic parameters. This feature allows a user to run all FFS jobs with no DD statement overrides unless one of the following conditions exists:

a. RASP OP, or QUIP run with more than three data bases and/or more than three file libraries

b. FM run with transaction source from tape or disk

c. FM run with auxiliary output on tape or disk

d. QUIP run storing a query into a user library.

e. OP run with record output.

The procedures provide a convention for naming user data files and file libraries which will be discussed in subsequent paragraphs.

2.1 Symbolic Parameters

Symbolic parameters simplify the overriding of DD statements in the execution of jobs. Basically, they allow the user to equate names, units, and volumes to symbolic parameters in the EXEC card without concern for the step name or the order of DD cards within a procedure. The use of symbolic parameters does not preclude the overriding of DD cards. A DD card override takes precedence over symbolic parameters. Each procedure has default options for every symbolic parameter not referenced by the user in his run. See examples in section 2.3.

2.2 File Name Conventions

Names of data files and file libraries must begin with an alphabetic character and contain only alphabetic and numeric characters. ISAM data file names are seven characters or less, not ending in L, S, or X. The SAM form of a data file is named by suffixing the ISAM name with S. A file Index Data Set is named by adding the suffix X to the ISAM name form, and a file library by suffixing the ISAM name with L. "L," "X," and "S" are concatenated to the ISAM form of the file name by the procedures to obtain the full library, Index Data Set, and SAM file names. Thus TEST360S, TEST360X, and TEST360L are the names of the SAM file, Index Data Set, and library for the TEST360 file.

Qualified data set names are valid as names of NIPS data files, libraries and index data sets. The file name used on control cards should appear as the last segment of a qualified name. Thus JULY.VERSION3.TEST360 would be a qualified data set name for a version of the TEST360 file. Because of the imbedded special characters in a qualified data set name the value must be enclosed in apostrophes when referenced as a symbolic parameter in a procedure:

```
// EXEC XRASP,ISAM='JULY.VERSION3.TEST360'
```

The transaction data set dynamically output by the File Analysis Statistics capability expects the data set to have the name of the data file concatenated with a 'T' suffix. The presence of an entry in the catalog for this name on the same volume as the data file will cause the transactions to be generated and output.

2.3 File Block Size Conventions

All NIPS 360 FFS cataloged procedures, components and utilities are designed to either generate or process default block size files or files with a user specified block size. The default block size is 1,004 bytes. A user specified block size can be 1,004 or greater, up to the files storage device limitation.

2.3.1 Default File Block Size

No action is required on the part of the user to generate or process standard block size files.

2.3.2 User Specified File Block Size

User block size specifications are accomplished by use of BSZFILE and BSZNEWF symbolic parameters in the applicable cataloged procedures. BSZFILE is used to indicate the block size of an input file, and it is required only when the input file resides on unlabeled magnetic tape and its block size is greater than 1,004 bytes. BSZNEWF is used to generate a file with a block size greater than 1,004 bytes or to change the block size of an existing file. The block size of an existing file can be changed only when a new copy is produced; you cannot change the block size of an ISAM file during execution of XFM in the update mode, because it is updated in place and it is still the same physical data set.

Once a user specified block size has been established for a file and that file resides on a direct access storage device or on labeled magnetic tape, the user is never required to provide the block size with the BSZFILE symbolic parameter.

2.4 Example of File Name Conventions and Symbolic Parameters

Naming conventions and symbolic parameters have a great impact upon the JCL required for running NIPS jobs. The amount of JCL the user becomes concerned with depends principally on whether he is using single or multiple data bases and single or multiple file libraries. A general discussion of job setups under each environment is included in the following subsections. Although RASP is used in the examples, the same concepts apply when using the other procedures.

2.4.1 Single Data Base

A <u>portion</u> of the XRASP procedure is shown below to illustrate how symbolics and naming conventions help the user in setting up various RASP jobs for a single data base:

JOB PREPARATION

```
//XRASP        PROC ISAM='DUMMY.FILE',SAM='DUMMY.FILE'.        X
//                  VISAM=,UISAM='(2314,P)',BSZFILE=,           X
//                  VSAM=,USAM='(2400,,DEFER)',LAB=SL
//RASP         EXEC PGM=RSEXEC
//DATAFILE     DD   DSNAME=&ISAM,DISP=SHR,UNIT=&UISAM,VOLUME=&VISAM
//SAMFILE           DD DSNAME=&SAM.S,DISP=SHR,UNIT=&USAM,       X
//                VOLUME=&VSAM,LABEL=(,&LAB)
                 .
                 .
                 .
```

a. When the user wants to run RASP against a <u>cataloged</u> ISAM data base named TESTER, he writes

```
//          EXEC XRASP,ISAM=TESTER
```

which causes OS to mount TESTER on a 2314. Note that the volume parameter for DATAFILE defaults to a NULL parameter.

b. To execute a RASP run against an <u>uncataloged</u> ISAM data base named TEST360 residing on a 2311 labeled TSTLAB, he would write

```
//   EXEC XRASP,ISAM=TEST360,UISAM=2311,VISAM='SER=TSTLAB'
```

c. Similarly, to run RASP against a <u>cataloged</u> SAM version of the TEST360 data base named TEST360S (note the 'S' suffix), he would write

```
//   EXEC     XRASP,SAM=TEST360
```

which causes OS to defer mounting the (first) TEST360S tape when the program calls for it.

d. To run RASP against an <u>uncataloged</u>, nonlabeled tape data base named TEST360S with a volume serial of MYTAPE and a block size of 7000 bytes, he would write

```
//    EXEC     XRASP,SAM=TEST360,VSAM='SER=MYTAPE',
//             LAB=NL,BSZFILE=7000
```

which would cause OS to defer mounting the TEST360S tape.

e. To run RASP against either an ISAM or SAM data base, cataloged or not, whose associated Index Data Set is cataloged, he would write

```
//  EXEC XRASP,ISAM=TEST360,INDEX=TEST360
```

f. To run RASP against a data base whose associated Index Data Set is <u>not</u> cataloged, a user would use the following statements (assume the ISAM form of the data base is cataloged)

```
//  EXEC XRASP,ISAM=TEST360,XUNIT=2314,
//       XVOL='SER=ND0026',INDEX=TEST360
```

2.4.2 Multiple Data Bases

RASP, OP, and QUIP are the only FFS components that support multiple data base capabilities. This is accomplished by having multiple DATAFILx and SAMFILEx DD cards in the procedures. Multiple Index Data Sets are provided by having XINDEXx DD cards. Although up to 10 data bases could be used, the following portion of the XRASP procedure is shown to illustrate the concepts behind job setups for a two-file environment.

JOB PREPARATION

```
//XRASP  PROC ISAM='DUMMY.FILE',ISAM1='DUMMY.FILE',          X
//            SAM='DUMMY.FILE',SAM1='DUMMY.FILE',            X
//            LAB=SL,VISAM=,VISAM1=,UISAM='(2314,P)',        X
//            UISAM1='(2314,P)',VSAM=,VSAM1=,                X
//            USAM='(2400,,DEFER)',USAM1='(2400,,DEFER)',    X
//       JOBLIB='FFS.JOBLIB'
//RASP   EXEC PGM=RSEXEC
//STEPLIB       DD DSN=&JOBLIB,DISP=SHR
//DATAFILE  DD DSNAME=&ISAM,DISP=SHR,UNIT=&UISAM,            X
//            VOLUME=&VISAM
//DATAFIL1  DD DSNAME=&ISAM1,DISP=SHR,UNIT=&UISAM1,          X
//            VOLUME=&VISAM1
//SAMFILE   DD DSNAME=&SAM.S,DISP=SHR,UNIT=&USAM,            X
//            VOLUME=&VSAM,LABEL=(,&LAB)
//SAMFIL1       DD DSNAME=&SAM1.S,DISP=SHR,UNIT=&USAM1,      X
//            VOLUME=&VSAM1,LABEL=(,&LAB)
            .
            .
            .
```

a. To run RASP against two cataloged ISAM data bases named TRAINER and TEST360, he writes

```
//   EXEC XRASP,ISAM=TRAINER,ISAM1=TEST360
```

b. To run RASP against two cataloged SAM versions of TRAINER and TEST360 named TRAINERS and TEST360S, he writes

```
//   EXEC  XRASP,SAM=TRAINER,SAM1=TEST360
```

c. To run RASP against a cataloged ISAM data base named TRAINER and a cataloged SAM version of TEST360 named TEST360S, he writes

```
//   EXEC XRASP,ISAM=TRAINER,SAM=TEST360
```

The single and multiple data base runs use the same procedure, XRASP; therefore, examples using uncataloged data bases would follow the same rules shown in the discussion of single data bases.

2.4.3 File Libraries

It is anticipated that many files containing RITs, retrievals, subroutines, and tables will have a library associated with them. The procedure is set up to form user library names by suffixing the ISAM data base name with an 'L'. This should be taken into consideration when cataloging user libraries. If the ISAM symbolic parameter specifies a qualified data set name, the user library name will be formed by suffixing an 'L' to only the last segment of the DSNAME.

Private user libraries are specified by using the LIB, LIB1, and LIB2 symbolic parameters. These libraries have a disposition of SHR which means they are read-only. The only user library that should be written into is the library defined by the LIB symbolic parameter. Therefore, to store permanent RITs or retrievals on a user library, define this library with the LIB symbolic parameter, and change the disposition symbolic parameter to LIBDISP=OLD.

Caution should be exercised in specifying multiple user libraries in that a search for information from these libraries is in a LIB, LIB1, LIB2 sequence. If a subroutine, for example, exists on LIB and LIB2 under the same name, the subroutine will always be fetched from LIB (the first library on which it was found).

An additional library named DUMMY.FILEL is referenced in the procedures. This can be used by those installations which desire to maintain an installation library of common subroutines and tables. The name of this library may be changed if desired.

2.5 Catalog Requirements

Aside from the usual catalog requirements for user's data bases, job libraries, and file libraries, procedures imply that the following data sets must be cataloged to run FFS jobs:

a. DUMMY.FILE - This data set must be cataloged but need not actually exist. It should be assigned to a system residence volume to avoid taking up an extra tape drive.

b. DUMMY.FILEL - This data set must be cataloged and physically exist as a partitioned data set, usually put on the same pack as FFS.JOBLIB.

c. DUMMY.FILES - This data set must be cataloged but need not actually exist.

d. DUMMY.FILEX - This data set must be cataloged but need not actually exist.

Caution should be observed when file libraries, Index Data Sets, and SAM files are cataloged since the procedure suffixes an 'X', 'S' or 'L' on the ISAM data base name to form the final name.

2.6 Checkpoint/Restart

The checkpoint/restart capability installed in FM, RASP, and OP is designed to serve as an aid in efficient use of the efficient computer in that processing functions already completed need not be duplicated if an incomplete job is returned to the user.

FM, RASP, and OP allow the user the option to periodically take checkpoints (time or end-of-volume) as the job progresses. A detailed description of the OS 360 checkpoint/restart capability (which is utilized in NIPS) is available to the interested user in IBM Systems Reference Library, Number C28-6708. Checkpoints are taken in the execution phases only, not during edit code generation, library action, etc.

The execute-only procedures (XFMEX, XRASPEX, XOPEX, and XOPSDEX) are the only ones designed for use with the checkpoint capability.

The coding of the following keyword values on the EXEC card will initiate the checkpoint/restart capability:

CHKID	A user-assigned name up to seven characters long which is used to assign DSNAMEs to

all normally temporary data sets (the procedure will concatenate suffixes to make the names unique) and the checkpoint data set defined on the CHECKDD DD card.

PARM='CHKPT= EOV ' / mm — The value EOV or a time interval in minutes (mm) to designate the type of checkpoint desired.

CHKDSP — The conditional disposition for the work data sets and the CHECKDD data set to be used in case of an ABEND. The value in the procedures is 'DELETE'; the user should override it to 'KEEP'.

NRMDSP — The normal disposition for the work data sets and the CHECKDD data set to be used for a normal run (one which does not ABEND). The value in the procedures is 'DELETE'; the user should override this parameter only if he needs to keep his temporary data sets from a successful run.

CHKSP — The space required for the checkpoint data set in cylinders. Approximately one cylinder per checkpoint is required for a 100K job. This substitution is mandatory for a new data set.

VCHK — The volume onto which the checkpoint data set is to be allocated.

UCHK — The unit onto which the checkpoint data set is to be allocated.

Note: VCHK defaults to a null value and UCHK defaults to NIPW. The above default values will cause one of the work packs to be used for the checkpoint data set, but it will be difficult to ascertain the volume serial number of the work pack used if a system error produces no listing. To avoid the problem, one should provide values for UCHK and VCHK to

reference a known volume. Example: UCHK=2314, VCHK='SER=ND000'.

For recurring jobs using checkpoint/restart, a previously allocated data set for the CHECKDD DD card will save time.

2.6.1 Checklist for Using Checkpoint/Restart

After the user establishes the need for checkpoint/restart insurance, the following list may serve as a guide to assist him in setting up the deck correctly:

a. All routines to be used (logic statements, queries, RITs) should be prestored.

b. Invoke the appropriate procedure and provide the CHKID, CHKPT, CHKDSP, CHKSP, VCHK, and UCHK substitutions.

c. If a restart is necessary, the changes required are the inclusion in the run deck of a SYSCHK DD card immediately preceding the first execute card and the coding of a RESTART parameter on the JOB card.

d. The deferred restart <u>must</u> be done prior to the scratching of the work packs. This requires that the analyst in charge of the critical job be available to resubmit the job within some reasonable time period after the job was originally run unsuccessfully.

2.6.2 Sample Job Setup

The following example shows initial and restart submissions of an FM SAM update run using the TESTER file and a multireel tape transaction file.

Only the front end of the deck is shown since the only changes to be made for restart are in the JCL. The end-of-volume checkpoint option is used.

The JOB card on the restart deck shows the RESTART parameter. The 'A.FM' represents the mandatory stepname on the EXEC card of the step to be restarted and the stepname

in the XFMEX procedure. The 'C0000002' represents the number of the checkpoint (obtained from the console listing) to be used for the restart.

INITIAL SUBMISSION

```
//JOBNAME    JOB (normal job card parameters)
//A    EXEC    XFMEX,CHKID=REQUEST,PARM='CHKPT=EOV',
//     CHKDSP=KEEP,CHKSP=5,UCHK=2314,
//     VCHK='SER=ND0000',SAM=TESTER,SAMOUT=,
//     VSAM='SER=INPUT',VSMOUT='SER=OUTPUT'
```

RESTART SUBMISSION

```
//JOB   JOB (normal job card parameters) RESTART=(A.FM,C0000002)
//SYSCHK  DD    DSNAME=REQUEST,DISP=(OLD,KEEP,KEEP),
//    VOLUME=SER=ND0000,UNIT=2314
//A   EXEC  XFMEX,CHKID=REQUEST,PARM='CHKPT=EOV',
//    CHKDSP=KEEP,CHKSP=5,UCHK=2314,
//    VCHK='SER=ND0000',SAM=TESTER,SAMOUT=,
//    VSAM='SER=INPUT',VSMOUT='SER=OUTPUT'
```

In the following example, the OP step theoretically abended with an SB37 ABEND after the eighth checkpoint had been taken. When the deferred restart is attempted, the batch mix of jobs will hopefully use less disk space. The restart procedure is the same as in the first example.

INITIAL SUBMISSION

```
//JOBNAME  JOB (normal job card parameters)
//AB   EXEC   XRASPEX,ISAM=TESTER,QDF=QDFILE,
//     QRT=QRTFILE,UQDF=2314,VQDF='SER=ND0000',
//     VQRT='SER=ND0000',CHKID=RASP,PARM='CHKPT=15',
//     CHKDSP=KEEP,LIB=TESTER,QDISP=KEEP
//AC   EXEC   XOPEX,QDF=QDFILE,LIB=TESTER,
//     QRT=QRTFILE,UQDF=2314,VQDF'SER=ND0000',
//     VQRT='SER=ND0000',CHKID=OP,PARM='CHKPT=10',
//     CHKDSP=KEEP,QDISP=KEEP,UCHK=2314,
//     VCHK='SER=ND0000'
```

RESTART SUBMISSION

```
//JOBNAME  JOB (normal job card parameters) RESTART=(AC.OP,C0000008)
//SYSCHK DD  DSNAME=OP,DISP=(OLD,KEEP,KEEP),
//  VOLUME=SER=ND0000,UNIT=2314
//AB  EXEC   XRASPEX,ISAM=TESTER,QDF=QDFILE,
//   QRT=QRTFILE,UQDF=2314,VQDF='SER=ND0000',
//   VQRT='SER=ND0000',CHKID=RASP,PARM='CHKPT=15',
//   CHKDSP=KEEP,LIB=TESTER,QDISP=KEEP
//AC       EXEC   XOP,QDF=QDFILE,LIB=TESTER,
//   QRT=QRTFILE,UQDF=2314,VQDF'SER=ND0000',
//   VQRT='SER=ND0000',CHKID=OP,PARM='CHKPT=10,'
//   CHKDSP=KEEP,QDISP=KEEP,UCHK=2314,
//   VCHK='SER=ND0000'
```

2.7 Segmented Files

The FM component has the capability to generate and maintain segmented data files. This capability allows the user to segment his large chronological SAM file into segments which may be updated individually, thereby reducing processing time. Each segment, when generated, will contain records with record keys within a specified range.

The OP, RASP, and QUIP components can retrieve and output from the segments singly or as a group making up one complete data file. No additional parameters are needed to process one segment. However, if more than one segment is to be processed, the OS S/360 concatenation capability must be used (see S/360 SRL C28-6539). Examples of this capability will be included in the Sample Job Setups for FM, RASP, OP, and QUIP.

2.8 Source Language Storage

All NIPS batch components procedures which perform language compilation or structuring will also allow source decks to be added, replaced, or deleted from a source library. The FS, FM, RASP, OP, and QUIP procedures contain a SOURCLIB DD statement which is used to define the source library. When a DSNAME other than 'DUMMY.FILE' is provided and source control statements are included in the input stream, source library maintenance will be performed.

Section 3

COMPONENT PROCEDURE DESCRIPTIONS

This section examines each procedure, pointing out specific ways of setting up jobs for each component.

The sample jobs will omit the JOB card which is always required and standard for each installation.

Note: In all JCL examples, the 'X' signifying a continuation card is usually punched in card column 72, even though it is no longer required by the Operating System.

3.1 FM Procedure (XFM)

The procedure XFM is used to accomplish all FM functions. The file name must be specified using the ISAM or SAM symbolic parameter. If an ISAM file is being generated, INDEX, PRIME, and OVFLOW should be used to specify the amount of disk space (in cylinders) required for each of these areas. If an assembler listing of logic statements being compiled is desired, code PARM=DEBUG or PARM=LIST on the EXEC card.

When adding new DD cards to the run that do not appear in the cataloged procedure, the DD card(s) should be placed immediately before the SYSIN DD card. If the POOL AFR capability is used, the DD name referenced in the AFR operator forms the DD name for a new JCL card by affixing it to the characters 'FM.' Hence, if AFR referenced the DD name 'AFRDD1', the user supplied JCL card for the FM run would have a DD name of 'FM.AFRDD1'.

If a file is being generated and the output file block size is to be different than the input FFT block size, use symbolic parameter BSZNEWF to specify the output block size.

If a SAM file is being updated and the output file block size is to be changed, use BSZNEWF to specify the new block size.

JOB PREPARATION

If the input is a SAM file on unlabeled magnetic tape and the file block size is not 1,004, use symbolic parameter BSZFILE to indicate the input file's block size.

In some cases, the user may need to provide JCL information that is not included in the procedure. Overriding DD cards must have a DD statement name which includes the stepname (FM) and the name of the DD statement to which the override applies. All override DD cards must be placed in the order that they appear in the procedure.

If the transaction source is a single tape or disk file, the user must include an override DD statement named FM.TRANS which specifies the DSNAME, DISP, UNIT, and VOLUME parameters plus the DCB parameters if the transaction file is an unlabeled tape. When multiple transaction sources are to be used in an FM execution, the user must provide a DD statement for each source. The DD statement must be named PSTRANxx for sequential tape or disk transactions and ISTRANxx for ISAM NIPS files. The XX may be a unique user specified ID for each DD statement. Parameters to be specified in each of the added transaction DD statements are the same as those required in the FM.TRANS DD statement. The FM.TRANS override DD must <u>not</u> be used when utilizing the multiple transaction source capability as it is used for describing single source transactions.

The following DD statements would be required only if the user is producing auxiliary output files by using the WRT,WT2,WT3, WT4 or WT5 instructions in his logic statement. These statements must include the DSNAME parameter and should include UNIT, VOLUME and DISP parameters. If the user specifies a direct access device for output, he must also include a space parameter. DCB parameters should be specified if other than standard parameters are desired.

A DD statement named FM.AUX1, is used to identify the name given to the first auxiliary output file. This file is produced by the POOL instruction WRT.

A DD statement, FM.AUX2, is used to identify the name given to the second auxiliary output file. This file is produced by the POOL instruction WT2.

JOB PREPARATION

A DD statement, FM.AUX3, is used to identify the name given to the third auxiliary output file. This file is produced by the POOL instruction WT3.

A DD statement, FM.AUX4, is used to identify the name given to the fourth auxiliary output file. This file is produced by the POOL instruction WT4.

A DD statement FM.AUX5, is used to identify the name given to the fifth auxiliary output file. This file is produced by the POOL instruction WT5.

A DD statement must always be used to identify the user's input deck. The format of this card is (//FM.SYSIN DD *). This is followed by the user's input deck, followed by a /* card to indicate the end of the input deck.

The FM component is designed to use a disk sort provided enough disk sort work space (DD cards SORTWK01-06) is available. The FM component required this space to sort the transactions prior to their being added to the data file. As these transactions are processed by FM prior to sorting, FM calculates the amount of disk space required for the sort, and if this space is not available it will default to a tape sort. When this occurs the following DD cards must be added to the standard procedure:

```
//TAPEWK01 DD parameters defining a 9-track tape unit.
//TAPEWK02 DD parameters defining a 9-track tape unit.
//TAPEWK03 DD parameters defining a 9-track tape unit.
//TAPEWK04 DD parameters defining a 9-track tape unit.
```

In addition, if the number of transactions necessitates that SORTIN and SORTOUT will contain more than one tape volume, the following JCL changes are needed. Otherwise, only the transactions on the last volume specified on your VOLUME parameter will be used to update the file.

```
//FM.SORTIN DD UNIT=(2400,,DEFER),VOL=SER=(.....,....)
//FM.SORTOUT DD UNIT=AFF=SORTIN,VOL=SER=(some tapes as SORTIN)
```

The above DD cards are not distributed with the standard procedure. The space defined for the disk sort work area is adequate for most applications.

JOB PREPARATION

When generating an ISAM file, space allocation in terms of INDEX, PRIME, and OVFLOW areas must be specified for the new file. This is done utilizing the INDEX, PRIME, and OVFLOW symbolic parameters. These parameters default to 1, 5, and 1 cylinders respectively. For example, if 1 cylinder of index, 20 cylinders of prime area and 5 cylinders of independent overflow area are desired when generating the TEST360 file, the following EXEC card would be coded:

```
//FMGEN    EXEC    XFM,GEN=,ISAM=TEST360,PRIME=20,OVFLOW=5
```

The XFM procedure will also maintain SAM data files. The use of the SAM or ISAM parameter specifies the type of input file to File Maintenance. The type of file must always agree with that specified on the FM control card, with one exception. A SAM file <u>may be generated</u> from an ISAM FFT. For this one exception, the ISAM symbolic parameter would be used to specify the input FFT and the FM control card would specify 'TAPE'.

When updating a SAM file, the updated file will normally be written on the data set (tape) specified by the FMSAMOUT DD statement. However, if any record controls are being changed by use of the MCT or MCW POOL instructions, the updated file will be written on the data set (tape) specified by the FMNDATA DD statement. These data sets are normally written on a 9-track tape. A message is written on the console stating the file name and which tape is to be saved at the completion of FM.

The XFM procedure is generalized since it has a variety of applications: SAM or ISAM data files in either the generate or update mode. To avoid possible run terminations due to insufficient space for the NEWFILE DD statement, special provisions have been made to DUMMY this statement in the XFM procedure. Also, this statement must be DUMMY when updating a multivolume ISAM data file. No space will be allocated for this statement unless GEN= is coded on the EXEC card. Coding of GEN= on the EXEC card is applicable for an ISAM generate run only. Special consideration has been directed to the FMSAMOUT and FMNDATA DD statements. These DD statements have been put to DUMMY. They should remain in the DUMMY status for all ISAM runs. For all SAM runs, the user must code SAMOUT= on the execute card to

allow allocation of the FMSAMOUT and FMNDATA data sets (tapes).

During COM and LIB mode runs, if the FPT and logic statements are on a sequential file, no library action will be performed. However, the XFM procedure may be used for debugging a logic statement. If the FPT is on an indexed sequential file, library action may be performed and logic statements may be added to the library.

In FM the default for the processing block size is 16,000 bytes. The user may override this size by using the PARM='PBSIZE=nK' or PARM='PB=nK' parameter on the EXEC card, where n may be any integer from 1 to 99 inclusive.

3.1.1 Sample Job Setup

The following examples generally use the ISAM symbolic parameters. Except where noted, these same examples can be used for SAM files by using the SAM, VSAM and USAM parameters in place of the respective ISAM parameters. The FM control card must also be changed to TAPE instead of DISK file update. For SAM files, SAMOUT= must be coded on the EXEC card.

a. <u>Compile Logic Statements Only</u> - The following FM run deck would be used to compile logic statements for the TEST360 file. The purpose of this run is to produce listings of the logic statement source cards for use in debugging new logic statements. This type of run would be used as the first step in setting up a new file. The file is not cataloged but resides on a 2314 disk pack.

A typical run deck follows:

```
//    EXEC      XFM,ISAM=TEST360,VISAM='SER=MYPACK'
//FM.SYSIN DD      *
$FMS/COM,TEST360
.                                                    .
.                                                    .
.   LOGIC STATEMENT LIBRARY UPDATE DECK              .
.                                                    .
.                                                    .
/*
```

b. Update Logic Statement Library - The following run deck could be used to update the Logic Statement Library for the TEST360 file which is cataloged.

```
//  EXEC      XFM,ISAM=TEST360
//FM.SYSIN     DD        *
$FMS/LIB,TEST360
.                                    .
.    LOGIC STATEMENT LIBRARY UPDATE  .
.                                    .
/*
```

If the file is a SAM file, logic statements can only be added during a GEN or UPD run.

c. Update a File (Card Transactions) - The following deck could be used to update the TEST360 file using card transactions. The first transaction report name is ONE. The second report name is TWO. The user's subroutine library is named TEST360L and is cataloged. The processing block size will be increased from 16,000 bytes to 20,000 bytes.

```
//            EXEC XFM,ISAM=TEST360,LIB=TEST360,
//            PARM='PBSIZE=20K'
//FM.SYSIN     DD        *
$FMS/UPD,TEST360,ONE
.
.   TRANSACTIONS FROM REPORT ONE
.
.   NEW REPORT TWO
.
.   TRANSACTIONS FROM REPORT TWO
.
/*
```

d. Update a File (Tape Transactions) - The following deck could be used to update the TEST360 file with tape transactions. The transaction report name is XYZ. The transaction file label is TAPXACT and it resides on tape with the volume label 1234. The installation has 2400 tape drives, and the transaction tape has standard labels.

```
//  EXEC      XFM,ISAM=TEST360,LIB=TEST360
//FM.TRANS  DD   DSNAME=TAPXACT,UNIT=2400,      X
//            VOLUME=SER=1234,DISP=OLD
//FM.SYSIN  DD   *
$FMS/UPD,TEST360,XYZ,,DISK,TAPE
/*
```

e. Compile Logic Statements, Update a File (Tape Transactions), and Produce Two Tape Auxiliary Outputs - The following run deck could be used to compile logic statements and update the file TEST360 with a tape transaction file labeled TAPEX. The transaction tape volume label is 2345. The transaction report name is ABC; 2400 series tape drives are available for mounting tapes. Two auxiliary output tapes are produced, named AUDITA and AUDITB. These are to be produced on tapes with serial numbers TAPEA and TAPEB.

```
//    EXEC     XFM,ISAM=TEST360,LIB=TEST360
//FM.TRANS DD DSNAME=TAPEX,UNIT=2400,                X
//            VOLUME=SER=2345,DISP=OLD
//FM.AUX1     DD DSNAME=AUDITA,UNIT=2400,            X
//            VOLUME=SER=TAPEA,DISP=(,KEEP)
//FM.AUX2     DD DSNAME=AUDITB,UNIT=2400,            X
//            VOLUME=SER=TAPEB,DISP=(,KEEP)
//FM.SYSIN      DD      *
$FMS/UPD,TEST360,ABC,LS,DISK,TAPE
.                                                  .
.                                                  .
.  LOGIC STATEMENT LIBRARY UPDATE DECK             .
.                                                  .
.                                                  .
/*
```

f. Compile Logic Statement and Generate a File (Card Transactions) - The following run deck could be used to compile logic statements for a TEST360 file and to generate the TEST360 file using card transactions. The transaction report names are A and B. The users subroutine library is named TEST360L. The index, prime and overflow areas for the generated file are 1, 25 and 10 cylinders, respectively.

JOB PREPARATION

```
//  EXEC      XFM,ISAM=TEST360,LIB=TEST360,      X
//            PRIME=25,OVFLOW=10,GEN=,INDEX=1
//FM.SYSIN    DD        *
$FMS/GEN,TEST360,,LS,DISK
.                                                .
.    LOGIC STATEMENT LIBRARY UPDATE DECK         .
.                                                .
.   NEW REPORT A                                 .
.                                                .
.   REPORT A TRANSACTIONS                        .
.                                                .
.   NEW REPORT B                                 .
.                                                .
.   REPORT B TRANSACTIONS                        .
.                                                .
/*
```

If a SAM file were being generated, the PRIME, INDEX and OVFLOW parameters would not be used.

g. <u>Generate an ISAM File (Card Transactions) Using the FPT and Logic Statement Library of Another File</u> - The following run deck could be used to generate a file having the same FPT and Logic Statement Library as an existing file. (The data base of the new file must still be supplied via transaction inputs.) It is assumed that the input data file, TEST360, is cataloged and that the required logic statements have been added to the Logic Statement Library. The new file will be named MYFILE, and will be generated with card transactions. At the end of the FM run, both TEST360 and MYFILE will be cataloged data sets. The transaction report name is REPORT. The user's subroutine library is named SUBSL and is cataloged.

```
//     EXEC  XFM,ISAM=TEST360,LIB=SUBS,GEN=
//FM.SYSIN    DD    *
$FMS/GEN,MYFILE,REPORT
.        .
.        .
. TRANSACTIONS FOR REPORT
.        .
.        .
/*
```

h. <u>Generate a SAM File (ISAM FFT)</u> - For GEN mode sequential processing, the input file containing the FFT (and logic statements, if desired) may be either a sequential or an indexed sequential file. The following is an example of using card transactions to generate a SAM file from an ISAM file consisting of FFT and logic statements. The TEST360 file is cataloged. The sequential data file will be cataloged at the end of the processing with the name specified on the FMS control card padded with a suffix character of 'S'.

```
//  EXEC  XFM,ISAM=TEST360,SAMOUT=
//FM.SYSIN    DD    *
$FMS/GEN,TEST360,RPT,,TAPE,CARD
.        .
.        .
.   TRANSACTIONS FOR REPORT RPT
.        .
/*
```

Note that GEN= is not coded on the EXEC card for a SAM generate run. It only applies to an ISAM generate run. Note also that SAMOUT= must be coded on the EXEC card for any FM run that produces a SAM output file. The Indexed sequential data file will be cataloged at the end of the processing. If a qualified data set name was used, and the last segment of the name matches the name on the FMS control card, the fully qualified name will be used as the catalog entry.

i. <u>Update a SAM File and Add Logic Statements</u> - For UPD mode sequential processing, the input file must be a sequential file and SAMOUT= must be coded on the EXEC card.

JOB PREPARATION

```
//  EXEC  XFM,SAM=TEST360,SAMOUT=
//FM.SYSIN   DD    *
$FMS/UPD,TEST360,RPT,LS,TAPE,CARD
.        .
. LOGIC STATEMENTS
. TRANSACTIONS FOR REPORT RPT
.        .
.        .
/*
```

Note: Although the sequential file is cataloged with a suffix character 'S', it should be specified without the suffix on the FMS control card and the execute card.

j. <u>Generate a First Segment From an FFT</u> - The following JCL and input control cards would be used to generate an initial segment from an ISAM file consisting of FFT and logic statements.

```
//     EXEC   XFM,ISAM=TEST360,SAMOUT=,
//            LAB=SL
//FM.SYSIN DD   *
$FMS/GEN,TEST360,RPT,,TAPE,CARD,SEG
$SEG   M00001   M00999
.        .
.        .
.   CARD TRANSACTION FOR REPORT RPT
.        .
.        .
/*
```

k. <u>Generate a Segment From Another Segment</u> - The following JCL and input control cards would be used to generate a segment. (The first segment was created using the JCL and control cards in sample j above.)

JOB PREPARATION

```
//   EXEC    XFM,SAM=TEST360,SAMOUT=,
//                LAB=SL
//FM.SYSIN DD *
$FMS/GEN,TEST360,RPT,,TAPE,CARD
$SEG    N00001    N00999
.         .
.         .
.    CARD TRANSACTIONS FOR REPORT RPT
.         .
.         .
/*
```

1. <u>Generate a New Segment From a Segment and Add a New Segment Record</u> - The following JCL and input control cards would be used to generate a new segment from an existing segment and add new segment control records.

```
//    EXEC  SAM=TEST360,SAMOUT=,LAB=SL
//FM.SYSIN     DD      *
$FMS/GEN,TEST360,RPT,,TAPE,CARD
$ADD         J00001    J00999        N36098
$SEG         W00001    W00999
         .
         .
    CARD TRANSACTIONS
         .
         .
/*
```

Note: After generation of a segment using the FFT and logic statements from the previous segment, the new segment will contain the FFT and logic statements and old segment control records from the previous segments. A new segment control record for the new segment and the new data records generated by the transactions will also be on the new segment.

m. Updating a Segment With No Segment Processing - The following JCL and input control cards would be used to update a segment with no segment processing.

```
//          EXEC XFM,SAM=TEST360,SAMOUT=,LAB=SL
//FM.SYSIN  DD          *
$FMS/UPD,TEST360,RPT,,TAPE,CARD,NOSEG
          .
          .
          CARD TRANSACTIONS
          .
          .
/*
```

Note: When updating a segment with no segment processing to be performed, the volume serial number field in the segment control record will be updated by the system.

n. Generating a Segment Using an ISAM FFT to Maintain Segment Information - The following JCL and input control cards would be used to generate a segment using an ISAM FFT and maintaining segment information and logic statements on the ISAM file.

```
//        EXEC  XFM,ISAM=ISAMFFT,VISAM='SER=XXXXXX'
//                      SAMOUT=,LABEL=SL
//FM.SYSIN  DD  *
$FMS/GEN,TEST360,RPT,LS,TAPE,CARD,SEG
          .
          .
       LOGIC STATEMENT
       STOP
       CARD TRANSACTION
/*
```

After completion of the JOB, ISAMFFT will contain the new segment record, any old segment records, and the new logic statement.

o. Updating a Segment Using an ISAM FFT to Maintain Segment Information - The following JCL and input control cards would be used to update a segment using the ISAM FFT to maintain segment information.

JOB PREPARATION

```
//        EXEC XFM,SAM=TEST360,VSAM=SER=XXXXXX,
//             SAMOUT=,LABEL=SL
//FM.ISAMWORK  DD  DSNAME=ISAMWORK,UNIT=2314,VOL=SER=XXXXXX,
                 DISP=OLD
//FM.SYSIN     DD  *
$FMS/UPD,TEST360,RPT,,TAPE,CARD
        .
        .
      CARD TRANSACTIONS
        .
        .
/*
```

p. Updating an ISAM File with Transactions from Multiple Sources - The following JCL and input control cards would be used to update a file from four transaction sources of varying data attributes in a single execution. Particular attention should be paid to the added DD cards and the parameters of the $FMS control card. Note: CARD must be specified as transaction source for multiple transaction source capability.

```
// EXEC XFM,ISAM=TEST360,VISAM=SER=XXXXXX,LIB=TEST360
//PSTRAN1 DD DSN=LTRANS,UNIT=(2400,,DEFER),VOL=SER=AAAAAA,    X
//          DISP=(OLD,KEEP),                                  X
//         DCB=(RECFM=FB,LRECL=480,BLKSIZE=2400)
//PSTRAN2 DD DSN=ATRANS,UNIT=AFF=PSTRAN1,VOL=SER=BBBBBB,      X
//          DISP=(OLD,KEEP),                                  X
//         DCB=(RECFM=F,LRECL=80,BLKSIZE=80)
//PSTRAN3 DD DSN=FTRANS,UNIT=2314,VOL=SER=YYYYYY,             X
//          DISP=(OLD,KEEP),                                  X
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=160)
//FM.SYSIN DD *
$FMS/UPD,TEST360,RPTM,,DISK,CARD
J00008-----------------------M
J00009-----------------------M
J00004-----------------------M   CARD Transactions for
J00007-----------------------M    Report RPTM
  .                        .
  .                        .
  .                        .
J00003-----------------------M

        NEW REPORT RPTL,PSTRAN1 TAPE TRANS FOR REPORT L
```

JOB PREPARATION

```
        NEW REPORT RPTA,PSTRAN2 TAPE TRANS FOR REPORT A
        NEW REPORT RPTF,PSTRAN3 DISK TRANS FOR REPORT F
/*
```

3.2 FR Procedure (XFR)

This procedure is used to revise the format of a NIPS 360 FFS data base. The procedure invokes FR to analyze the old and new FFTs and to produce logic statements which are then used by FM to revise the file format. The old data file may be SAM or ISAM; the new FFT must be ISAM; revised file is always SAM.

The block size of the new file will be the same as the old file block size unless a new block size is specified with the BSZNEWF symbolic parameter.

If the old file is a SAM file on unlabeled magnetic tape and its block size is not 1,004 bytes, the block size must be indicated with the BSZFILE symbolic parameter.

3.2.1 Sample Job Setup

The following JCL could be used to revise the TEST360 file to a new format. The old file is cataloged on disk. The new FFT is not. The new TEST360S SAM file will be a standard label tape on a 2400 unit. The new TEST360 FFT is on a 2314 disk pack labeled OURUSE. As the file being revised; i.e., the old TEST360, resides on tape as a SAM data set, the parameter TRANTYP must equal SAM.

```
//REV     EXEC    XFR,SAM=TEST360,NEWFFT=TEST360,          X
//                VFFT='SER=OURUSE',LAB=SL,TRANTYP=SAM
//FR.SYSIN   DD   *

                  FR CONTROL CARDS

/*
```

If the old data file is a cataloged ISAM file instead of SAM, simply replace the SAM= parameter with ISAM= in the above example. TRANTYP defaults to ISAM and need not be included in an ISAM File Revision.

3.2.2 File Structure, Revision, and Maintenance

```
//FRJOB   JOB  (Standard Parameters)
//JOBLIB  DD   (Installation JOBLIB Parameters)
//        EXEC XFR ISAM=TEST360,NEWFFT=FRTEST
//FR.SYSIN DD   *
      FILE = TEST360
      NEWFILE = FRTEST
      PRINT = ALL
      CNAM = CNAME
      LYN = STLOC
      PLEAC = STPLAN
/*
```

File Structuring of the new FFT has been completed and the FFT is a cataloged ISAM data set named 'FRTEST'. The file to be revised is a cataloged ISAM data set named 'TEST360'. The user has specified that all logic statements are to be printed. Three field name changes are specified. Other nonmatching field names found in the two FFTs will be treated as deletions when not found in the new FFT and as additions when not found in the old FFT. The new sequential file produced by the FM generate step may be loaded to disk using the SAM to ISAM utility procedure XSTOIS. The sequential file may be retained as backup.

3.3 FS Procedure (XFS)

This procedure is used to structure FFTs. If a user library is desired, use the LIB parameter, remembering that the procedure will suffix the name with an 'L'. The disposition of the new data file (FFT) can be specified using the NDISP parameter. NDISP defaults to KEEP. The file name must be specified by the ISAM symbolic parameter. If a block size greater then 1,004 bytes is desired, use symbolic parameter BSZNEWF to indicate the size.

3.3.1 Sample Job Setup

The following control cards could be used to structure an FFT for debugging purposes. A file library called TEST360L is used but is not cataloged. It resides on a 2314 pack with a volume serial number of MYPACK.

JOB PREPARATION

```
//FSSTR        EXEC       XFS,ISAM=TEST360,
//                        LIB=TEST360,VLIB='SER=MYPACK'
//FS.SYSIN     DD                *

             SOURCE FFT DECK

/*
```

The following JCL could be used to structure and to catalog the FFT for TEST360 using the library TEST360L which has been cataloged and to specify a file block size of 7,000 bytes.

```
//FSSTR   XFS,ISAM=TEST360,LIB=TEST360,                X
//        VISAM='SER=MYPACK',NDISP=CATLG,
//             BSZNEWF=7000
//FS.SYSIN     DD   *

             SOURCE FFT DECK

/*
```

Explanation of the EXEC XFS statement:

EXEC XFS Symbolic parameter values are as follows: ISAM= where TEST360 is the name of the created FFT data set, NDISP=CATLG where the user wishes to catalog the FFT. When this parameter is not provided, the FFT is not saved. VISAM= where MYPACK is the actual device label name for the new FFT data set, LIB= where TEST360 is the name of the user's cataloged file or general library. BSZNEWF=7000 specifies a file block size of 7,000 bytes. When BSZNEWF is not used, the file block size defaults to 1,004 bytes.

3.3.2 Non-NIPS FS

The standard FS procedure (XFS) can be used to structure FFTs for non-NIPS without modifications.

3.4 ISAM TO SAM Procedure (XISTOS)

This procedure is used to unload an ISAM data file from disk-to-tape (SAM) or copy a SAM data file from tape-to-tape (SAM). The disposition of the old file defaults to KEEP. A new block size can be specified for the output file by using symbolic parameter BSZNEWF.

If the input is a SAM file on unlabeled magnetic tape and its block size is not 1,004 bytes, its block size must be specified with the BSZFILE symbolic parameter.

3.4.1 Sample Job Setup

In the first example, the JCL could be used to unload the TEST360 file to a 9-track tape with standard labels and volume serial of TSTVOL. It is desired to delete the ISAM file from the disk pack.

```
//ISTS    EXEC  XISTOS,ISAM=TEST360,VISAM='SER=MYPACK',      X
//              LAB=SL,ODISP=DELETE,
//              SAM=TEST360,VSAM='SER=TSTVOL'
/*
```

In the second example, the JCL could be used to copy the TEST360 file onto a 9-track tape with standard labels and volume serial of TSTVOL. The old SAM file is kept and is on the volume OLDVOL. The block size of the new SAM file is changed to 10,004 bytes.

```
//CSTS    EXEC  XISTOS,SAM=TEST360,VSAM='SER=TSTVOL',        X
//              BSZNEWF=10004,OLDSAM=TEST360,                 X
//              OLDVSAM='SER=OLDVOL'
/*
```

3.5 OP Procedures (XOP, XOPSD)

There are two standard procedures for executing OP.

a. XOP should be used when publishing the results of a RASP retrieval (source retrieval mode). This procedure is normally executed immediately following the execution of the retrieval component. If it is executed at any other time, the user must insure that the retrieval answer sets (QRT and QDF)

have been kept and are properly defined in the XOP procedure. The XOP procedure can be used when structuring single or merge file RITs. The XOP procedure must be used when publishing merge file RITs.

b. XOPSD should be used when structuring and/or publishing directly from a data base (source direct mode). Merge file RITs may be structured but not executed using XOPSD.

Either procedure, XOP or XOPSD, may be used when structuring and/or publishing an RIT that makes use of the Interfile Output (IFO) capability. Although these RITs are written as one would write a merged file RIT, they are considered to be a single primary file with a maximum of nine secondary ISAM files. Only one of the secondary files can have the same name as the primary file. Secondary file(s) are named in the symbolic parameters, ISAM1 and ISAM2 or in additional DATAFILE3 through nine DD statements.

When using the XOPSD (source direct) procedure, a data base name must be assigned to the ISAM or SAM symbolic parameter. A source retrieval (XOP) run does not require a data base name unless a RIT is being structured in the run and the FFTs required for structuring are not on the QDF (either because PARM='NOFL' was specified on the preceding RASP execute statement or the file(s) required for structuring are not the same files being retrieved and published). If the input is a SAM file on unlabeled magnetic tape and its block size is not 1,004 bytes, its block size must be specified with the BSZFILE symbolic parameter.

To structure permanent RITs using either procedure, a user library must be specified using the LIB parameter and by indicating LIBDISP=OLD on the EXEC card. User (private) libraries must be specified using the LIB, LIB1, LIB2, etc., parameters if a RIT uses subroutines and/or tables. Use care in assigning the LIB parameter since the procedure will suffix this file name with an 'L' to form the final library name. This point should also be considered when cataloging user libraries.

When tape output is desired, the user must include an additional DD statement named OP.OPRECORD which specifies the DSNAME, UNIT, VOLUME, LABEL, and DISP parameters. If the DSNAME OPTAPE is used, then OP will replace it with the name specified on the FORMAT card of the RIT, if any.

Inclusion of the keyword and operand, PARM=LIST, on the EXEC card will cause creation of the generated code assembly listing during the RIT structuring process. This option should not normally be used, but is included as a diagnostic aid in case of system malfunction. The system will automatically generate this assembly listing if an error occurs during the code generation process and is sensed by the assembler.

3.5.1 XOPSD Sample Job for Single File Processing

To structure a RIT for a single file using XOPSD, only the data file name and library name need be specified. If the RIT being structured is to be stored permanently on the library, the library disposition must be specified.

Example:

```
//OP  EXEC  XOPSD,ISAM=TEST360,LIB=TEST360,LIBDISP=OLD
//OP.SYSIN DD *
    (OP Run Deck)
/*
```

The example given would allow RITs to be structured and/or published for the indexed sequential data file TEST360. User library TEST360L would be used and any permanently structured RITs would be stored on that library.

3.5.2 XOPSD Sample Job for Merge File Processing

When structuring for a merge file using XOPSD, all data files and libraries must be named. The data files may be indexed sequential or sequential; they do not have to be the same type.

If a library is specified for symbolic LIB, any RITs structured to be stored permanently will be stored on that library. If there is no library specified for LIB and there is a library specified for LIB1, then that library will

receive any permanent RITs structured during the run. If neither LIB or LIB1 is specified, the RITs will not be permanently stored. Also, a library disposition must be specified as OLD to store permanent RITs.

Example:

```
//OP      EXEC  XOPSD,ISAM=TEST360,SAM=TRAINER,
//          LIB=TEST360,LIB1=TRAINER,LIBDISP=OLD
//OP.SYSIN DD *
      (OP Run Deck)
/*
```

The example given would allow a merge file RIT to be structured for the indexed sequential data file TEST360 and the sequential data file TRAINERS using libraries TEST360L and TRAINERL. Any permanent RITs would be stored on TEST360L. The use of the XOPSD procedure for merge file report is limited to structuring the RITS. Publishing merge file reports requires a retrieval step and the use of the XOP procedure.

3.5.3 XOP Sample Job

If the execution is to publish retrieval answer sets, the following job could be used.

Example:

```
//R EXEC   XRASP,ISAM=TEST360,SAM=TRAINER,LIB=TEST360,
//  LIB1=TRAINER
//RASP.SYSIN   DD   *
               (RASP run deck)
/*
//OP      EXEC    XOP,LIB=TEST360,LIB1=TRAINER,LIBDISP=OLD
//OP.SYSIN       DD   *
               (OP run deck)
/*
```

Note that the RIT could be structured and the reports published in one execution of the XOP procedure. Any user subroutine required by the RITs would be found on the library TEST360L, which would also be used for storing any permanent RITs structured during the run. Note also that in the OP job step no data files are named. Although file

names could have been specified it is never necessary to do so when publishing an answer set. When structuring a RIT using the XOP procedure, it is only necessary to specify file names when PARM='NOFL' has been specified on the preceding RASP EXEC statement or when the RIT being structured requires a file or files other than the files retrieved by RASP. When the XOP procedure is used to structure RITs, the QRT and QDF are used to locate the FFT in the absence of a designated file name. It is possible in one XOP run to structure one RIT using the QRT/QDF and to structure additional RITS specifying data file names.

3.5.4 Segmented File Sample

The following example will publish a report from three segments of a segmented data file. The first segment is specified through the symbolic parameters. The additional DD cards must provide all necessary DD parameters.

```
//      EXEC    DD    XOP,SAM=TEST360,LIB=TEST360
//OP.SAMFILE    DD
//              DD        DSNAME=TEST360S,VOL=SER=XXXXXX,
//                        DISP=OLD,UNIT=2400
//              DD        DSNAME=TEST360S,VOL=SER=YYYYYY,
//                        DISP=OLD,UNIT=2400
//OP.SYSIN        DD        *
              RIT DECK
/*
```

3.5.5 Output Data Set Overrides

The OP component and the OP procedures (XOP and XOPSD) are designed to allow the user certain override options on his output data sets. The DDNAMEs for the data sets which may be overridden follow, together with a brief description of the data set:

OPLINE - This is the output data set for printer output. It can be overridden and put out on any SAM data set, provided the following parameters are provided by the user.

BLKSIZE - Block size
LRECL - Record length

RECFM - Record form
DSNAME - Data set name

OPPUNCH - This is the output data set for card (punch) output. It can be overridden and put out on any SAM data set. The same parameters used when overriding OPLINE must be furnished by the user.

Examples:

To override the printer output and create a tape to be printed elsewhere (by the utility IEBPTPCH) -

```
//OP.OPLINE  DD  DSNAME=OPPLINE,DCB=(BLKSIZE=133,        X
//        LRECL=133,RECFM=F,DEN=2)
```

To override the punch and create a tape to be punched later -

```
//OP.OPPUNCH  DD  DSNAME=PUNCHIT,DCB=(BLKSIZE=80,        X
//        LRECL=80,RECFM=F,DEN=2)
```

3.5.6 Additional DD Card for Tape Output

If the RIT being executed contains a FORMAT TAPE statement, an additional DD card must be added to the input job stream with a DDNAME of OP.OPRECORD. This is the data set for tape record output. DISP, LABEL, DSNAME, and VOLUME are the required parameters. The remaining parameters are normally taken from the RIT but may be overridden by the user at run time. The following are parameters which may be overridden by the user on the DD card:

BLKSIZE - Block size
LRECL - Record length
RECFM - Record form
DEN - Tape density
TRTCH - Recording technique

Example:

To override the standard blocking and record form on a record output and generate fixed length unblocked records -

```
//OP.OPRECORD  DD  DCB=(BLKSIZE=150,LRECL=150,RECFM=F),
//  DISP=(NEW,KEEP),LABEL=(,SL),UNIT=2400,VOL=SER=MYTAPE,
//  DSNAME=OPTAPE
```

Note that OP will replace the DSNAME with the name specified on the format card of the RIT, if any.

3.6 QRT/QDF Utility Procedure (XQRTQDF)

This procedure is used to create a NIPS 360 FFS data file from the answer file (QRT and QDF) produced by RASP. The output from the utility is a SAM file. A block size greater than 1,004 bytes can be specified for the SAM file with the BSZNEWF symbolic parameter.

3.6.1 Sample Job Setup

The following statements illustrate the job setup used to execute the XQRTQDF procedure. The retrieval is from the TEST360 file and uses the file library TEST360L; both are cataloged. The new data file will be named TEST361.

```
//RET    EXEC   XRASP,ISAM=TEST360,LIB=TEST360
//RASP.SYSIN    DD  *
                    RETRIEVAL DECK
/*
//NEWDF  EXEC   XQRTQDF,SAM=TEST361,LAB=SL
//QRTQDF.SYSIN  DD *
                    CONTROL CARD
/*
```

Using this job setup, the user obtains his new data file on a standard labeled, 9-track tape written according to the installation assigned density. The user may override the symbolic parameters in the procedure to use input and output data sets differing from the procedure defaults.

The XQRTQDF procedure may be executed in a separate job from the RASP if the QRT and QDF are saved. In this case,

the QRT and QDF symbolic parameters would have to be specified on the EXEC card with the appropriate values.

3.7 QUIP Procedures (XQUIP, XQUIPSD)

The XQUIP and XQUIPSD procedures are used to execute QUIP in the batch mode. No DD card overrides are necessary, but additional DD statements are required when processing more than one segment of a segmented file, when using more than two secondary files with Interfile Output (IFO) and when using the TRACE operator. A DD statement named SYSIN must always be included to identify the user's input. When using the XQUIPSD procedure, a file name must be specified using either the ISAM or SAM symbolic parameter. For IFO, the symbolic parameters ISAM and SAM identify the primary file.

If the input is a SAM file on unlabeled magnetic tape and its block size is not 1,004 bytes, its block size must be specified with the BSZFILE symbolic parameter.

In addition, when executing QUIP with the XQUIPSD procedure, the File Indexing mode of retrieval will be automatically invoked whenever practical if the input file is indexed and the user's input statements include either a FIND or a retrieval type IF statement that is compatible with Secondary Indexing or a KEYWORD statement is present in the query. The user may override the Secondary Indexing mode of retrieval by using the PARM='INDEX=NO' parameter on the EXEC card. However, the presence of a KEYWORD statement negates the PARM override, so the indexing will be invoked as long as it is feasible.

In QUIP, the processing blocksize is computed using the file statistics records. If the records are not present in the file, the size defaults to 10,000 bytes. The user may override this size by using the PARM='PBSIZE=nk' or PARM='PB=nk' parameter on the EXEC card, where n is any integer from 1 to 99 inclusive. For IFO, only the size of the processing block for the primary file may be overridden.

3.7.1 Sample Job Setup

The following JCL may be used to retrieve, sort, or output from the TEST360 file using the file library

TEST360L. Both files are cataloged. Since the TEST360 file itself is used, the XQUIPSD procedure is executed. The processing block size will be reduced from 10,000 to 6,000 bytes, and the Secondary Indexing mode of retrieval will be bypassed.

```
//SAMPLE1     EXEC    XQUIPSD,ISAM=TEST360,LIB=TEST360,
//                    PARM='PB=6K,INDEX=NO'
//QUIP.SYSIN DD       *
              .
              .
              .
              .
              QUIP SOURCE STATEMENTS
              .
              .
              .
              .
/*
```

The XQUIPSD procedure may be used to query and output from a segmented data file using either one segment or a concatenated group of segments. If more than one segment is to be processed, additional DD cards must be included for the required segments. The first segment can be described by the symbolic parameters in the procedure, with the other segments being defined by the additional DD cards.

```
//SAMPLE2    EXEC  XQUIPSD,SAM=TEST360,LIB=TEST360
//QUIP.SAMFILE   DD
//                DD DSNAME=TEST360S,VOL=SER=XXXXXX,
//                   UNIT=2400,DISP=OLD
//              DD  DSNAME=TEST360,VOL=SER=YYYYYY,
//                  UNIT=2400,DISP=OLD
//QUIP.SYSIN DD  *
```

In addition to using an ISAM or SAM data base, QUIP can also use a RASP QRT and QDF. To use a QRT/QDF, the XQUIP procedure is executed. The QRT and QDF symbolic parameters should be overridden to specify the QRT and QDF names unless QUIP is being run in the same job as RASP. The following JCL can be used to retrieve, sort, and output from a QRT named TESTQRT and a QDF named TESTQF both located on a 2314 whose volume serial is MYPACK. The user subroutine library is a cataloged data set named TESTERL.

```
//SAMPLE3   EXEC   XQUIP,QRT=TESTQRT,QDF=TESTQDF,  X
//                 VQRT='SER=MYPACK',              X
//              VQDF='SER=MYPACK',LIB=TESTER
//QUIP.SYSIN        DD     *
                  .
                  .
                  .
              QUIP SOURCE STATEMENTS
                  .
                  .
                  .
/*
```

The XQUIPSD procedure may be used to output statistical data on the subfiles in a subfile library employing the TRACE operator. A SUBFILE DD statement defining the library must be included.

```
//SAMPLE4 EXEC XQUIPSD,ISAM=TEST360
//QUIP.SUBFILE DD DSN=NMCSSC1.S132205,UNIT=2314,
//      VOL=SER=MYPACK,DISP=OLD
//QUIP.SYSIN DD *
FILE TEST360 CLASS UNCLASSIFIED
TRACE NMCSSC1.S132205 FORMAT
/*
```

3.7.2 Interfile Output (IFO)

Either the XQUIP or XQUIPSD procedures may be used with IFO. When XQUIPSD is used, the primary file may be SAM or ISAM and is specified by the SAM or ISAM symbolic parameter. When XQUIP is used, the primary file is the RASP answer set and is specified in the same manner as for a single file query.

With either procedure, up to nine secondary files may be referenced. All secondary files must be ISAM. The symbolic parameters, ISAM1 and ISAM2, are provided for two secondary files.

3.7.3 Non-NIPS Files

The XQUIPSD procedure must be used when querying a non-NIPS file. The non-NIPS file may be either a SAM file or an ISAM file and the name is specified by either the SAM or

ISAM parameter. Also the FFT to be used for the non-NIPS file must be specified by the FFT symbolic parameter.

Non-NIPS files can be used with IFO, either as the primary file or as one or more secondary files. When a non-NIPS is the primary, the JCL specifications are as stated above. When a non-NIPS file is used as a secondary file, the file name is specified using either ISAM1 or ISAM2 symbolic parameter or by the addition of the appropriate DATAFILx DD statement (DATABIL3 through DATAFIL9). Also when a non-NIPS file is used as a secondary file, the FFT for the file must be specified by either the FFT1 or FFT2 symbolic parameter or by the addition of the appropriate FFTx DD statement (FFT3 through FFT9). The 'x' suffix number on the FFTx DD statement must be the same as the DATAFILx DD statement. All secondary non-NIPS files must be ISAM.

NOTE: It is possible to intermix NIPS and non-NIPS files during IFO.

3.8 RASP Procedure (XRASP)

This procedure may be used for either single-file or merge-file retrievals. If the retrieval is to be added to the file library identified by the LIB parameter, LIBDISP=OLD must be specified on the EXEC card. Use care when specifying file libraries using LIB, LIB1, LIB2 parameters since the procedure will suffix the specified name with an 'L'. Indexed files will be suffixed with an 'X' when the parameters XINDEX, INDEX1, and INDEX2 are specified, while SAM files will be suffixed with an 'S' when using the SAM, SAM1, and SAM2 parameters.

If an assembler list of the retrieval is desired, code PARM='LIST' on the EXEC card.

PARM='NOFL' on the EXEC card causes the FFT and logic statements not to be copied to the QDF. This option should be used only when the QDF is input to OP.

If the file being queried is indexed, the File Indexing mode of retrieval will automatically be invoked if the retrieval logic allows. The user may override the Secondary

Indexing mode of retrieval by using the PARM='INDEX=NO' parameter on the EXEC card. However, the presence of a KEYWORD statement in the retrieval negates the PARM override so that indexing will be involved as long as it is feasible.

PARM=SORTX on the EXEC card will cause RASP to limit the contents of the QDF to a minimum of a few required control records. This technique will significantly reduce the processing time for large retrievals; however, all file data required for later processing must be placed in the QRT via the SORT statement. The truncated QRT/QDF can be used as input to OP to publish a previously structured RIT or to any program or system which is able to process the QRT exclusive of the QDF. However, the truncated QRT/QDF cannot be used to structure an RIT nor as input to the XQRTQDF utility.

3.8.1 Sample Job Setup

The following JCL could be used to retrieve from TEST360 using the file library TEST360L. Both files are cataloged.

```
//RET          EXEC      XRASP,ISAM=TEST360,LIB=TEST360
//RASP.SYSIN   DD   *

          RETRIEVAL DECK

/*
```

The following JCL could be used if two libraries are required and TEST360 is on tape (9-track). The TEST360S data base and both libraries (TEST360L and TRAINERL) are cataloged. The retrieval is to be added to the TEST360L library.

```
//RET      EXEC      XRASP,SAM=TEST360,LIB=TEST360,      X
//                   LIB1=TRAINER,LIBDISP=OLD
//RASP.SYSIN  DD  *

          RETRIEVAL DECK

/*
```

The XRASP procedure can be used to query a segmented data file. If only one segment is to be queried, the deck setup is the same as for a SAM FILE execution. If more than

one segment is to be queried, additional DD cards must be provided to define the additional segments. The first segment may be defined through the symbolic parameters, but the additional DD statements must completely define the data set.

```
//      EXEC  XRASP,SAM=TEST360,LIB=TEST360
//RASP.SAMFILE     DD
//                DD  DSNAME=TEST360S,DISP=OLD,
//                    UNIT=2400,VOL=SER=XXXXXX
//                DD  DSN=TEST360S,DISP=OLD,
//                    UNIT=2400,VOL=SER=YYYYYY
//RASP.SYSIN DD *

      RETRIEVAL DECK

/*
```

The following JCL could be used to query an index data file. The Index Data Set name is designated by the symbolic parameter XINDEX and its volume by the symbolic parameter XVOL. For this example, the TEST360 file, the TEST360L library, and the TEST360X Index Data Set are not cataloged. Note that the 'X' suffix on the Index Data Set name is affixed by the procedure as is the 'L' suffix for the library.

```
//RETSX EXEC XRASP,ISAM=TEST360,LIB=TEST360,
//          VISAM='SER=MYPACK',VLIB='SER=MYLIBR',
//          XINDEX=TEST360,XVOL='SER=SXPACK'
//RASP.SYSIN DD *
    RETRIEVAL DECK
/*
```

The following JCL could be used to keep the retrieval answer data sets (QRTFILE and QDFILE) on the user's disk pack for later use. This is done by specifying non-temporary DSNAMES, the pack volume serial numbers, units, and the dispositions (KEEP). These same parameters (with the exception of disposition) will have to be specified when executing OP against these answer data sets. The TEST360 disk datafile and TEST360L library are used for the example. If a possibility exists that the retrieval answer set might occupy more than one volume, the UQDF parameter should be

coded as UQDF='(unit type,2)' or as many volumes as necessary up to 59.

```
//RASPV EXEC  XRASP,ISAM=TEST360,LIB=TEST360,                X
//            UQDF=2314,UQRT=2314,VQRT='SER=MYPACK',         X
//            VQDF='SER=MYPACK',QDF=TSTQDF,                  X
//            QRT=TSTQRT,QDISP=KEEP
//RASP.SYSIN DD *

        RETRIEVAL DECK

/*
```

The standard XRASP procedure provides for retrievals against up to three data files with a merge of the records which qualify. This can be expanded to up to 10 data files but will require a change to the procedures XRASP and XOP.

When doing a merge-file retrieval, all files, Index Data Set and libraries must be named. The symbolic parameters for naming the files are ISAM, ISAM1, and ISAM2 for indexed sequential data files and SAM, SAM1, and SAM2 for sequential data files. In a merged-file run both ISAM and SAM data files can be named.

The symbolic parameters for naming the Index Data Sets associated with the retrievals in a merged-file file are XINDEX, XINDEX2, and XINDEX2.

The symbolic parameters for naming the libraries are LIB, LIB1, and LIB2. Library designations do not necessarily correspond to the data file designations; i.e., LIB may not be the library associated with ISAM or LIB1 with ISAM1. Also it is not necessary to specify a library for each data file. However, when library actions are to be performed, the library parameters used take on special significance. Library action will be performed against the library specified by the LIB or LIB1 symbolic parameters. If a library name was specified for LIB, the action will be against that library. If LIB was not specified and a name was specified for LIB1, the action will be for that library. If neither was specified or the library disposition was not specified, no library action will take place.

JOB PREPARATION

In the following example, index sequential data file TEST360 and sequential data file TRAINERS will be queried using libraries TEST360L and TRAINERL. Library action would be performed with library TEST360L. TRAINERS is an indexed file.

```
//RET1 EXEC XRASP,ISAM=TEST360,SAM=TRAINER,LIB=TEST360,   X
//                LIB1=TRAINER,LIBDISP=OLD,XINDEX=TRAINER
//RASP.SYSIN  DD   *
    (RASP CONTROL AND SOURCE LANGUAGE STATEMENTS)
/*
```

3.9 SAM TO ISAM Procedure (XSTOIS)

This procedure is used to load a tape data file (SAM) to disk (ISAM). Space is allocated separately for INDEX, PRIME, and OVFLOW areas. The procedure defaults these areas to 1, 40, and 5 cylinders, respectively. The disposition of the new disk file defaults to KEEP.

If the input is SAM file on unlabeled magnetic tape and its block size is not 1,004 bytes, its block size must be specified with the BSZFILE symbolic parameter.

The BSZNEWF symbolic parameter is used to specify a new block size for the ISAM file. If it is not used, the ISAM block size will be the same as the input SAM block size.

The procedure also can be used to create an ISAM file in the compression/compaction form or to recreate a standard form ISAM file. The PARM parameter can be used for specifying the option. The default value is the form of the input file. The options are:

PARM=COMPRESS	for compression only
PARM=COMPACT	for compaction only
PARM='COMPRESS,COMPACT'	for both compression and compaction
PARM=EXPAND	for reversing the compression/compaction process to produce standard form data records

All unused PRIME space will be filled with system generated PAD records. The PARM option NOPAD may be used if this feature is not desired, e.g.:

```
PARM=NOPAD
```

3.9.1 Sample Job Setup

The following JCL could be used to load the TEST360S file from tape-to-disk. It is desired to increase the PRIME area to 50 cylinders and catalog the new data set. The SAM file is not cataloged.

```
//STOIS   EXEC   XSTOIS,SAM=TEST360,PRIME=50,          X
//               VSAM='SER=MYTAPE',ISAM=TEST360,        X
//               VISAM='SER=MYPACK',NDISP=CATLG
/*
```

3.10 Subroutine Loader Procedure (XSUBLDR)

This procedure is used to place user subroutines on the file library. The two inputs are the user subroutine in load module form on a temporary or permanent data set and a control card. The MODLIB parameter is used to specify the name of the data set containing the input subroutine. It defaults to &TEMP which is the name of the data set containing the subroutine if the ASMFCL procedure has been used in a prior step of the job to assemble and link edit the subroutine.

3.10.1 Sample Job Setup

The following JCL could be used to assemble, link-edit, and add a subroutine to the TEST360L library which is cataloged. The subroutine is named DTGIS.

Note: The MODLIB DSNAME is not specified on the EXEC card since it defaults to the DSNAME of the assembler job step output (&TEMP).

```
//ASML   EXEC  ASMFCL
//ASM.SYSIN   DD  *

                SUBROUTINE SOURCE DECK (DTGIS)

/*
```

JOB PREPARATION

```
//LKED.SYSLMOD DD DSNAME=&TEMP(DTGIS),DISP=(,PASS)
/*
//        EXEC XSUBLDR,LIB=TEST360
//SUB.SYSIN  DD   *
                    CONTROL CARD
/*
```

3.11 Table Generation Procedure (XTABGEN)

This procedure is used to generate one or more table(s). The input is a card deck containing the argument/function pairs and the generated table(s) is placed on the user's File Library.

3.11.1 Sample Job Setup

The following JCL could be used to place one or more table(s) on the TEST360L library which is not cataloged and resides on a 2314 with a volume serial number of MYPACK.

```
//GEN    EXEC  XTABGEN,LIB=TEST360,                          X
//            VLIB='SER=MYPACK'
//TAB.SYSIN  DD   *

      INPUT CARD DECK DEFINING ONE OR MORE TABLES.

/*
```

3.12 Source Language Library Procedure (XUTSOURC)

This procedure may be used to add, replace, delete, or list members from a source library. The library is a direct access partitioned data set which may be the file library or a special library for source material.

3.12.1 Sample Job Setup

The following JCL could be used to list a member named MYRITS from the TEST360L library.

```
//LIST EXEC XUTSOURC,SOURCL=TEST360,NAME=MYRITS
/*
```

JOB PREPARATION

The following JCL could be used to add, replace, or delete source members from the TEST360L library.

```
//UPD EXEC XUTSOURC,SOURCL=TEST360
//SYSIN DD *
./ ADD NAME=MYRITS OP

                    Source Cards for MYRIT

./ REPL NAME=RETROIS   RASP

                    Source Cards for RETROIS

./ DELETE NAME=RETRO7S
/*
```

3.13 1410 to 360 Conversion (X360CON)

This procedure is used to convert 1410 NIPS data bases to the NIPS 360 FFS format. The NIPS 360 FFT must be structured on disk, the input 1410 file is on tape and the new NIPS 360 file is also on tape.

3.13.1 Sample Job Setup

The following JCL could be used to convert the TESTA 1410 file to the NIPS 360 FFS TEST360 file. TESTA is on a seven-track tape at 556 BPI with a volume serial number of XX1410. The TEST360 FFT is on disk, not cataloged, and will be deleted after conversion (specified by the ODISP parameter). The new TEST360S file will be cataloged on a standard label tape whose volume serial number is '123456'.

```
//CON    EXEC  X360CON,ISAM=TEST360,LAB=SL,              X
//            ODISP=DELETE,NDISP=CATLG,                  X
//            V1410='SER=XX1410',                        X
//            VISAM='SER=MYPACK',SAM=TEST360,            X
//            VSAM='SER=123456'
//GO.SYSIN DD *

                    1410 FFT OBJECT DECK

/*
```

3.14 360 to 1410 Conversion (X1410CON)

This procedure is used to convert NIPS 360 FFS data bases to the 1410 NIPS format. The NIPS 360 FFS input is an ISAM data base on disk; the output is tape.

Note: S/360 input for conversion must be an ISAM data base.

When specifying the file name of the new 1410 file, using the SAM symbolic parameter, adhere to 1410 file name rules. The name must be five characters long, beginning with an alphabetic character and ending in 'A'.

The tape used for the new 1410 file must have had at least one block or a tape mark written on it in the same density and track format as is desired for the 1410 file. This should be done with an OS utility (IEBGENER) prior to the conversion step. If more than one volume is expected as output, the number must be specified using the VOLCOUNT field of the volume JCL parameter.

3.14.1 Sample Job Setup

The following JCL could be used to convert the TEST360 file to the 1410 file called TESTA. The 1410 file is created on a 7-track tape, 556 BPI density. TEST360 is cataloged.

```
//CON    EXEC  X1410CON,ISAM=TEST360,SAM=TESTA
//GO.SYSIN  DD   *

                    1410 TESTA FFT OBJECT DECK

/*
```

If two volumes are expected for the new 1410 file, add the following parameter to the EXEC card:

```
VSAM='(,,,2,SER=FL1410)'
```

3.15 List Logic Statement/Report Names Procedure (XDMPLIB)

This procedure is used to enable the user to list report names and/or logic statement names from his file. The

procedure allows the data file to reside on either SAM or ISAM.

3.15.1 Sample Job Setup

The following JCL would list all report names and logic statement names residing on an uncataloged ISAM file named TESTER. The unit is a 2314 disk pack with the volume serial number of MYPACK.

```
//   EXEC  XDMPLIB,ISAM=TEST360,VISAM='SER=MYPACK'
//UTDMP.SYSIN  DD  *
PRINT,REPORT,ALL
/*
```

3.16 Subroutine Check Procedure (XSUBCHK)

This procedure is used to check out a user-written subroutine. The LIB symbolic parameter identifies the user library on which the subroutine resides. The ULIB and VLIB symbolic parameters must be coded if the library is not cataloged.

Input to the XSUBCHK procedure is a card deck containing a control card and test data cards. The control card identifies the subroutine.

3.16.1 Sample Job Setup

The following JCL could be used to test a subroutine on the TEST360L library. The TEST360L library is cataloged.

```
//   EXEC   XSUBCHK,LIB=TEST360
//SUBCHK.SYSIN DD  *

                    CONTROL CARD
                    DATA CARDS

/*
```

3.17 Classification Change Procedure (XCLASS)

This procedure is used to change the classification of a NIPS 360 PFS data file. If the input is a SAM file on unlabeled magnetic tape and its block size is not 1,004

bytes, its block size must be specified with the BSZFILE symbolic parameter.

Use symbolic parameter BSZNEWF to change the block size of the file.

3.17.1 Sample Job Setup

The following JCL could be used to change the classification of an ISAM data file. It is an uncataloged data set that resides on a 2314 pack with a volume serial number of MYLIB.

```
//   EXEC   XCLASS,ISAM=TEST360,VISAM='SER=MYLIB'
//CLASS.SYSIN  DD *
                    CLASSIFICATION CARD
/*
```

The following JCL could be used to change the classification of a SAM file on tape. The input is a cataloged data set that resides on a 9-track tape. The output is a 9-track scratch tape with the volume serial number of TSTVOL.

```
//  EXEC  XCLASS,SAM=TEST360,          X
// VSMOUT='SER=TSTVOL',SAMOUT=

//CLASS.SYSIN DD *
                    CLASSIFICATION CARD
/*
```

Note that SAMOUT= must be coded on the EXEC card for any CLASS run that changes the classification on a SAM tape file.

The following JCL could be used to change the classification of a SAM file on disk. It is an uncataloged data set that resides on a 2314 pack with a volume serial number of MYLIB.

JOB PREPARATION

```
//    EXEC XCLASS,SAM=TEST360,USAM=2314,
//         VSAM='MYLIB'
//CLASS.SYSIN DD *
                    CLASSIFICATION CARD
/*
```

3.18 Data Field Scan Procedure (XUTFSCAN)

This procedure is used to scan all of a user's source component statements to count the number of data field references in each source, to provide a total count of references for each component and to provide transactions suitable for updating and maintaining a NIPS data file. If the input is a SAM file on unlabeled magnetic tape and its block size is not 1,004 bytes, its block size must be specified with the BSZFILE symbolic parameter.

3.18.1 Sample Job Setup

The following JCL would count the data field references in each component source statement for the ISAM file named TEST360. Some of the source statements are members of the library TEST360L and some are cards in the input stream. Transaction records will also be output.

```
// EXEC XUTFSCAN,ISAM=TEST360,UISAM='SER=MYPACK'
//      LIB=TEST360,VLIB='SER=MYPACK',TRANS=TEST360T,
//      VTRANS='SER=MYPACK'
//SYSIN DD *
./ SOURCE COMP=FM,NAME='TESTO A',MEMBER=TESTA
./ SOURCE COMP=RASP,NAME=TEST01
    :
    :
RASP QUERY
/. SOURCE COMP=OP,NAME=TESTFFF
    :
    :
OP RIT
    :
    :
./ SOURCE COMP=QUIP,NAME=QTEST,MEMBER=QTESTA
/*
```

3.19 Index Specification Procedure (XSP)

This procedure is used to specify Keyword Indexing information for a data file without running a File Structure or File Maintenance job. The disk-resident Index Data Set may be either generated or updated from either an ISAM or SAM data file.

Both SUB/TAB and INDEX statements are accepted and must be specified through the SYSIN device. Both additions and deletions of indexed fields may be made in the same run.

If, for any reason, the Index Data Set is destroyed but the file remains operational, the user has the option to recreate the Index Data Set to have it correspond with the existing file by specifying PARM='GEN' and XDISP=NEW. If the input is a SAM file on unlabeled magnetic tape and its block size is not 1,004 bytes, its block size must be specified with the BSZFILE symbolic parameter.

3.19.1 Sample Job Setup

The first example is used to generate a new Index Data Set from an existing ISAM data file.

```
//GENXSP EXEC XSP,XINDEX=TEST360,XDISP=NEW,
//            XVOL='SER=MYPACK',LIB=TEST360,
//            VLIB='SER=MYPACK',ISAM=TEST360,
//            VISAM='SER=MYPACK',
//XSP.SYSIN DD
   INDEX STATEMENTS
/*
```

The second example is used to update an existing Index Data Set from an existing SAM data file.

```
//UPDXSP  EXEC XSP,XINDEX=TEST360,XVOL='SER=MYPACK',
//            LIB=TEST360,VLIB='SER=MYPACK',
//            SAM=TEST360,VSAM='SER=MYTAPE',
//            LAB=SL,DEN=2,
//            SAMOUT=,VSMOUT='SER=NEWTAP',
//XSP.SYSIN DD
   INDEX STATEMENTS
/*
```

The third example is used to recreate an Index Data Set from an existing SAM file containing Index Descriptor records. In this example no changes are desired to the indexed fields. However, the Index Descriptor records may be updated by including the proper Index Statements.

```
//RECXSP EXEC XSP,PARM='GEN',XINDEX=TEST360,
//             XVOL='SER=MYPACK',LIB=TEST360,
//             VLIB='SER=MYPACK',SAM=TEST360,
//            VSAM='SER=MYTAPE',LAB=SL,
//             DEN=2,SAMOUT=,VSMOUNT='SER=NEWTAP',
//            XDISP=NEW
//XSP.SYSIN DD
/*
```

3.20 Unload Index Data Set Procedure (XTRDISK)

This procedure is used to copy a disk-resident, direct-access organization, Index Data Set to a tape-resident, sequential-access organization data set. The XFNAME and XFVOL symbolic parameters must be supplied for the Input Index Data Set. The XTNAME and XTVOL symbolic parameters must be supplied for the unloaded data set residing on tape. No control statements are required.

If statistics on the content of the index data set are desired, STAT=YES should be designated. When this option is chosen the corresponding data file name and volume must also be specified. The parameters are ISAM and VISAM for an ISAM data base and SAM and VSAM for a SAM data base.

3.20.1 Sample Job Setup

The following JCL would unload the TEST360X Index Data Set from a 2314 disk pack to a 9-track labeled tape.

```
//STEPNAME EXEC XTRDISK,
//             XTNAME=INDEXSAM,XTVOL='SER=MYTAPE',
//             XFNAME=TEST360X,XFVOL='SER=MYPACK
/*
```

The following JCL would unload and print the contents of the TEST360X index data set.

```
//STEPNAME EXEC XTRDISK,STAT=YES
//              XTRNAME=INDEXSAM,XTVOL='SER=MYTAPE',
//               XFNAME=TEST360X,XFVOL='SER=MYPACK',
//               ISAM=TEST360
/*
```

3.21 Load Index Data Set Procedure (XTRTAPE)

This procedure is used to reconstruct a disk-resident direct access organization Index Data Set from a previously unloaded sequential access organization data set. The XFNAME and XFVOL symbolic parameters must be supplied for the input tape. The XTNAME and XTVOL symbolic parameters must be supplied for the Output Index Data Set on disk.

3.21.1 Sample Job Setup

The following JCL would load the TEST360X Index Data Set to a 2314 disk pack from a 9-track labeled tape.

```
//STEPNAME EXEC XTRTAPE,
//              XTNAME=TEST360X,XTVOL='SER=MYPACK',
//              XFNAME=INDEXSAM,XFVOL='SER=MYTAPE'
/*
```

3.22 Keyword Analysis (XKA)

This procedure is used to obtain printed listing of text, nonkeywords, and keywords that occur in a data base without updating the index data set. No DD card overrides are necessary, but a DD statement named SYSIN must always be included to identify the user's input. A file name must be specified using either the ISAM or SAM symbolic parameter. If a user scan subroutine, stop word table, or dictionary is required, the library name must be specified with the LIB symbolic parameter.

3.22.1 Sample Job Setup

The following JCL world be used to analyze keyword fields in the TEST360 ISAM data base:

```
//STEPNAME  EXEC  XKA,ISAM=TEST360,LIB=TEST360,
//                VISAM='SER=MYPACK',VLIB='SER=MYPACK'
//XKA.SYSIN  DD  *
          (user supplied control statements)
/*
```

3.23 Dictionary Maintenance Utility (XKM)

This procedure is used to generate or update keywords, stop word tables and dictionaries. More than one stop word table or dictionary may be created or updated in one run provided that all are members in one library (specified by the LIB parameter), stop word table and dictionary name specifications, maintenance commands, and data are accepted through the SYSIN device.

3.23.1 Sample Job Setup

The following JCL would be used to maintain any or all stop word tables and dictionaries in the TEST360 library.

```
// EXEC XKM,LIB=TEST360
//SKM.SYSIN DD *
     Table or Dictionary Names
     Maintenance Commands
     Data (keyword specs)
/*
```

3.24 Format Definition Translator Procedure (XUTODE)

This procedure is used to place format definitions on the user library. The input definition statements may be in punched cards or in card image records stored in a library. More than one format definition may be added to the library at a time.

3.24.1 Sample Job Setup

The following JCL would be used to add a format definition in punched cards to the user library, TEST360.

JOB PREPARATION

```
// EXEC XUTODE,LIB=TEST360,VLIB='SER=MYPACK'
//SYSIN DD *
     Definition source statement cards.

/*
```

To add a format to the TEST360 library when the definition source statements reside in a library (MYLIB), the following JCL would be used.

```
// EXEC XUTODE,LIB=TEST360,VLIB='SER=MYPACK'
//SYSIN DD DSN=MYLIB(FORMAT1),VOL=SER=MYPACK2,
//  DISP=(SHR,KEEP),UNIT=2314,
//  DCB=(RECFM=FB,LRECL=80,BLKSIZE=800
/*
```

Note that the definition source statements may be on any library, including TEST360. The only restriction is that the source member name is not the same as the name of the format definition.

Section 4

SUPPLEMENTARY PROCEDURES DESCRIPTIONS

This section will examine the four procedures used by NIPS 360 FFS that do not execute a NIPS component or utility. These procedures are made available for users to perform the backup and restore functions of RASP generated QRT and QDF answer sets and user libraries. The utilities IEBGENER and IEHMOVE are used to perform the backup and restore functions. These utility programs are documented in the S/360 Systems Reference Library, Utilities manual. Use of these procedures requires knowledge of these programs and their JCL requirements.

The sample jobs will omit the JOB card that is always required for S/360 OS jobs. These supplementary procedures do not execute NIPS components or utilities and therefore do not require a STEPLIB card.

Note: In all JCL examples, the "X" signifying a continuation card is always punched in card column 72, even though it is no longer required by the operating system.

4.1 Restore Answers Procedure (XRESTANS)

This procedure is used to load QRT and QDF answer data sets from tape to disk. The procedure is set up to default the space allocation for the QRT and the QDF if not specified at execution time. The XRESTANS procedure assumes that the QRT and QDF was copied to tape by the XSAVEANS procedure.

4.1.1 Sample Job Setup

The following JCL could load the QRT and QDF answer data sets from tape to disk. It is desired to decrease the space allocation for both the QRT and the QDF. The QRT and QDF answer data sets will be used by subsequent job steps but will not be kept at job termination.

```
//RESTORE  EXEC   XRESTANS,VTAPE='SER=MYTAPE',          X
//                QRTSP='(5,1)',QDFSP='(10,5)'
/*
```

The following JCL could load a QRT named MYQRT and a QDF named MYQDF from tape to disk. It is desired to save the QRT and QDF answer data sets on disk.

```
//RESTORE  EXEC   XRESTANS,VTAPE='SER=MYTAPE',          X
//                QRT=MYQRT,QDF=MYQDF,QDISP=KEEP
/*
```

4.2 Restore Library Procedure (XRESTLIB)

This procedure uses the IEHMOVE utility program to load a user library from tape to disk. The XRESTLIB procedure requires that a volume on which a work data set needed for the IEHMOVE program can be placed be identified with the VWORK symbolic parameter. The LIB and VTAPE symbolic parameters must be supplied for the unloaded user library residing on tape. The ULIB and VLIB symbolic parameters must be supplied for the disk volume that is to receive the user library. The control statement required by the IEHMOVE utility program must be provided.

4.2.1 Sample Job Setup

The following JCL could load the TEST360L library from tape to a 2314 disk pack with the volume serial number MYPACK. The TEST360L library presently resides on a standard labeled, 800 BPI tape labeled MYTAPE.

```
//RESTLIB  EXEC   XRESTLIB,LIB=TEST360,VLIB='SER=MYPACK',     X
//                VTAPE='SER=MYTAPE',VWORK='SER=WORK06',      X
//                LAB=SL
//RESTLIB.SYSIN   DD  *
    COPY  PDS=TEST360,FROM=2400=MYTAPE,TO=2314=MYPACK,
/*
```

4.3 Save Answers Procedure (XSAVEANS)

This procedure is used to unload the QRT and QDF answer data sets from disk to tape.

4.3.1 Sample Job Setup

The following JCL could unload the QRT/QDF answer set from disk to tape. The QRT is named MYQRT and the QDF is named MYQDF. The QRT/QDF answer set resides on a 2314 disk pack with the volume serial number MYPACK.

```
//SAVE  EXEC  XSAVEANS,QRT=MYQRT,QDF=MYQDF,                X
//             UQRT=2314,VQRT='SER=MYPACK',                X
//             UQDF=2314,VQDF='SER=MYPACK',                X
//             VTAPE='SER=QRTQDF'
```

The following JCL could unload a QRT/QDF answer set from disk to tape. This example assumes that RASP was executed in a prior step and that the data set names for the QRT/QDF will be the temporary defaults established in the XRASP procedure.

```
//SAVETEMP  EXEC  XSAVEANS,VTAPE='SER=QRTQDF'
/*
```

4.4 Save Library Procedure (XSAVELIB)

This procedure is used to unload a user library from disk to tape. The XSAVELIB procedure requires that a volume on which a work data set needed for the IEHMOVE program can be placed be identified with the VWORK symbolic parameter. The LIB and VTAPE symbolic parameters must be supplied for the unloaded user library. The ULIB and VLIB symbolic parameters must be supplied for the disk volume on which the library resides. The control statement required by the IEHMOVE utility program must be supplied.

JOB PREPARATION

4.4.1 Sample Job Setup

The following JCL could unload the TEST360L library from a 2314 disk pack with the volume serial number MYPACK to a tape with the volume serial number MYTAPE.

```
//SAVELIB  EXEC  XSAVELIB,LIB=TEST360,VLIB='SER=MYPACK',    X
//               VTAPE='SER-MYTAPE',VWORK='SER=WORK06'
//SAVELIB.SYSIN  DD   *
    COPY  PDS=TEST360L,FROM=2314=MYPACK,TO=2400=MYTAPE,     X
          TODD=TAPE
/*
```

Section 5

PRODUCTION PROCEDURES DESCRIPTION

Production procedures are available for the three major NIPS components: FM, RASP, and OP. These procedures (FMEX, RASPEX, OPEX, and OPSDEX) were developed for the production environment in which execution only is required of previously compiled RITs, retrievals and logic statements. Data sets not required during execution have been eliminated to reduce direct access storage requirements and allocation/deallocation overhead. The reduction in the number of data definition (DD) statements permits more job steps per job. Specifications of symbolic parameters and overide DD statements for the production procedures are the same as for the normal component procedures.

5.1 FM Production Procedure (XFMEX)

The procedure XFMEX is used to update a SAM or ISAM file using previously compiled logic statements. It may not be used if logic statement compilation is necessary, nor if a file is to be generated. The intended use of the XFMEX procedure is for production update runs only.

Specifications of symbolic parameters and override DD statements for the production procedure XFMEX are the same as for the normal component procedure XFM.

5.2 OP Production Procedure (XOPEX, XOPSDEX)

The procedures XOPEX and XOPSDEX are used to publish output using previously structured RIT's. Procedure XOPEX is used when publishing the results of a RASP retrieval run (SOURCE RETRIEVAL); procedure XOPSDEX is used when publishing directly from a data file (SOURCE DIRECT). These procedures may not be used if a RIT is to be structured. The intended use of procedures XOPEX and XOPSDEX is for production runs only.

Specifications of symbolic parameters and override DD statements for the production procedures XOPEX and XOPSDEX

are the same as for the normal component procedures XOP and XOPSD.

5.3 Compression and Compaction of Data Records

Compression and compaction provide a means for the reduction of intermediate storage requirements for data without altering the integrity of the data. This data reduction scheme is particularly suited to data files that contain strings of identical characters or a large quantity of alphabetic data.

A string of identical characters is compressed by translating it to two bytes. The first byte is a control byte which indicate that compression has been applied and gives a count of the number of identical consecutive bytes that were in the original string. The second byte is identical to those in the original string.

A string of alphabetic characters is compacted by translating it to a control byte followed by a string of coded characters. The control byte indicates that compaction has been applied and gives a count of the coded characters. Each coded character represents a combination of two adjacent alphabetic characters.

Compression and compaction can be applied to data files by specifying COMPRESS or COMPACT or 'COMPRESS/COMPACT', if both features are desired, as PARM values in the EXEC statement that calls procedures for UTBLDSAM, UTBLDISM and FM (file generation mode only). If both compression and compaction are specified, compression is applied to a record first and data within the record that cannot be compressed is processed by the compaction routine.

The compression and compaction process can be reversed by specifying EXPAND as the PARM value on the EXEC statement that calls the procedures for UTBLDSAM and UTBLDISM. EXPAND does not apply to FM processing.

5.4 RASP Production Procedure (XRASPEX)

The procedure XRASPEX is used to execute previously compiled single-file or merge-file retrievals. It may not be used to compile a retrieval or to add a retrieval to a library. The intended use of procedure XRASPEX is for production retrievals only.

Specifications for symbolic parameters and override DD statements for the production procedure XRASPEX are the same.

Section 6

S/370 VSAM CONSIDERATIONS

This section addresses the procedures and considerations under which the S/370 user of NIPS FPS may create and process a NIPS file using the Virtual Storage Access Method (VSAM). Each NIPS procedure which might be used to process a VSAM file is examined and the VSAM service routine IDCAMS is examined for use with NIPS VSAM files.

The sample jobs will omit the job card which is always required and standard for each installation.

NIPS PROCEDURES ARE DISTRIBUTED WITHOUT STEPCAT DD CARDS. VSAM USERS SHOULD ADD A STEPCAT TO EACH DISTRIBUTED PROCEDURE.

NOTE: NIPS FPS will not currently process a VSAM NIPS file in the terminal environment. Also NIPS FPS will not process VSAM files which are not NIPS files.

6.1 VSAM Service Routine IDCAMS

VSAM processing requires that VSAM files be defined before they can be accessed by any program or system, including NIPS. The VSAM service routine IDCAMS is used for this purpose. By making use of the various options available, the user can create user catalogs, define clusters (allocate and catalog data sets), copy data sets, move data sets from one operating system to another, and recover from certain types of data damage.

Four of the IDCAMS options are of interest to the NIPS user.

- Creating a NIPS user catalog
- Defining a cluster for a VSAM file
- Converting an ISAM FILE to VSAM
- Deleting an unneeded cluster

JOB PREPARATION

6.1.1 Creating a NIPS User Catalog

All VSAM data sets must be cataloged on a user or a master catalog. In normal applications, the user's VSAM file will be cataloged on the NIPS user catalog. This provides for user control of his data sets. The following JCL could be used to create a NIPS user catalog:

```
//DCAT     EXEC  PGM=IDCAMS
//DDMAST   DD    DSN=AMASTCAT,DISP=SHR
//DDNIP    DD    UNIT=3330,VOL=SER=xxxxxx,DISP=OLD
//SYSPRINT DD    SYSOUT=A
//SYSIN    DD    *
     DEFINE  USERCATALOG -
          (NAME    (NIPSM)  VOLUME  (xxxxxx) -
              FILE (DDNIP)  TRACKS  (133,8)) -
          DATA  (RECORDS  (23,8))
/*
```

In the preceding example the DDMAST statement points to the master catalog, needed to receive a catalog entry for the NIPS user catalog; and the DDNIP statement points to the disk pack which will contain the NIPS user catalog. The NIPS user catalog was named NIPSM. A complete discussion concerning the creation of user catalogs is contained in the OS/VS Access Methods Services manual (GC26-3836).

6.1.2 Defining a Cluster for a VSAM File

Prior to running a NIPS job which creates a new VSAM file, the user must first define the cluster that represents that VSAM file via IDCAMS. In defining the cluster, the user specifies the name of the file, the volume that contains the file, the key size and location, etc. The following JCL could be used to define a cluster for the VSAM file VSAM.SECOND:

```
//DCLUS    EXEC PGM=IDCAMS
//STEPCAT DD    DSN=NIPSM,DISP=SHR
//DDNIP    DD    UNIT=3330,VOL=SER=xxxxxx,DISP=OLD
//SYSPRINT  DD SYSOUT=A
//SYSIN    DD    *
     DEFINE  CLUSTER -
          (  NAME  (VSAM.SECOND)  VOL(xxxxxx) -
             KEYS (21,1)  BUFFERSPACE(13312) -
```

```
                    FREESPACE  (20 20)  CYLINDERS (2)) -
                    DATA  (RECORDSIZE  (100,1000) -
                      CONTROLINTERVALSIZE (4096)) -
                    INDEX  (  CONTROLINTERVALSIZE (1024) -
                      REPLICATE IMED) -
                    CAT  (NIPSM)
/*
```

In the preceding example the STEPCAT statement points to the NIPS user catalog which will contain the catalog entry for the VSAM file. The DDNIP statement points to the disk pack which will contain the VSAM file.

For NIPS VSAM files the KEYS parameter is specified as follows:

```
KEYS  (KS,1)
```

where KS is the keysize. The key is a minimum of 15 bytes and consists of the following:

a. Two bytes for system control

b. The user defined major control group

c. The largest user or system defined set control group, a minimum of four bytes.

A complete discussion concerning the definition of a cluster is contained in the OS/VS Access Method Services Manual.

6.1.3 Converting an ISAM File to VSAM

The user can use the IDCAMS service routine to convert an existing ISAM file to a new VSAM file. The following JCL could be used to convert the ISAM file, ISAM.SECOND, to the VSAM file, VSAM.SECOND:

```
//REP      EXEC PGM=IDCAMS
//STEPCAT DD   DSN=NIPSM,DISP=SHR
//DDVSAM   DD   DSN=VSAM.SECOND,DISP=OLD
//DDISAM   DD   DSN=ISAM.SECOND,UNIT=3330,
//         VOL=SER=xxxxxx,DISP=SHR,
//         DCB=DSORG=IS
```

```
//SYSPRINT DD  SYSOUT=A
//SYSIN    DD  *
     REPRO   INFILE(DDISAM) -
             OUTFILE (DDVSAM)
/*
```

In the preceding example the STEPCAT statement specifies the NIPS user catalog containing the catalog entry for the VSAM file. The DDVSAM statement specifies the VSAM file and the DDISAM statement specifies the input ISAM file.

6.1.4 Deleting a Cluster

If it becomes necessary to delete a VSAM file, the user must use IDCAMS to delete the file. IDCAMS will scratch the file from the user's disk pack and will delete the file catalog entry in the NIPS user catalog. The following JCL could be used to delete the VSAM file, VSAM.SECOND:

```
//DCAT     EXEC PGM=IDCAMS
//STEPCAT DD   DSN=NIPSM,DISP=SHR
//DDNIP    DD   UNIT=3330,VOL=SER=xxxxxx,DISP=OLD
//SYSPRINT DD  SYSOUT=A
//SYSIN    DD  *
     DELETE  VSAM.SECOND CATALOG (NIPSM) -
             FILE (DDNIP) PURGE
/*
```

In the preceding example the STEPCAT statement specifies the user NIPS catalog containing the catalog entry for the VSAM file to be deleted and the DDNIP statement points to the disk pack containing the VSAM file. A complete discussion of the delete procedure within IDCAMS is contained in the OS/VS Access Service Methods Manual.

6.2 File Maintenance (FM)

When File Maintenance (FM) is used to process a VSAM file, as either a data base, transaction source or AFR file, the user must specify the NIPS user catalog that contains the catalog entry for the VSAM file. This is done by including the VSCAT parameter on the EXEC statement. The following is an example of a completed VSCAT parameter:

```
//FMS  EXEC  XFM,VSCAT=NIPSM
```

JOB PREPARATION

6.2.1 VSAM Generate

When running a VSAM generate operation, the user must specify the name of the VSAM file containing the FFT via the VSDSN parameter, and the name of the newly generated VSAM file via the NEWVSM parameter. These two file names (DSNAMEs) cannot be the same, as they refer to two different VSAM files. The file specified by the NEWVSM parameter will be the newly generated file and must have been previously defined via the VSAM service routine IDCAMS (section 6.1.2). The user must not include any of the parameters associated with an ISAM generate. The following job could be used to generate a VSAM file:

```
//GEN      EXEC XFM,VSCAT=NIPSM,LIB=PRIME,
//         VSDSN='VSAM.FFT.PRIME',
//         NEWVSM='VSAM.PRIME'
//FM.SYSIN  DD  *
$FMS/GEN,PRIME,XXX,,DISK,CARD
.
.    GENERATE TRANSACTIONS
.
/*
```

In the above example, the term DISK on the $FMS control card denotes that the data base will be either ISAM or VSAM.

6.2.2 VSAM Update

When running a VSAM update, the user specifies the name of the VSAM file with the VSDSN parameter. The user must not include any of the symbolic parameters associated with an ISAM update. The following job could be used to update a VSAM FILE:

```
//UPD      EXEC XFM,VSCAT=NIPSM,
//         VSDSN='VSAM.PRIME',
//         LIB=PRIME
//FM.SYSIN  DD  *
$FMS/UPD,PRIME,XXX,,DISK,CARD
.
.    UPDATE TRANSACTIONS
.
/*
```

In the above example, the term DISK on the $FMS control card denotes that the data base is either ISAM or VSAM.

6.2.3 VSAM Transactions

When an FM generate or update is to be performed using a VSAM file as the transaction source, the user must code the FM.TRANS DD overide statement which specifies the DSNAME, DISP and AMP parameters. The AMP parameter must be coded as follows:

```
AMP='AMORG'
```

Additional subparameters may be specified as set forth in the OS/VS VSAM Programmer's Guide (GC26-3838).

If more than one VSAM file is to be used as the transaction source, each file must be identified by a separate DD statement. The DD statements must be named ISTRANxx, where xx is a unique identifier for each DD statement. The same parameters specified for a single VSAM file on the FM.TRANS DD statement must be specified for each VSAM file on the corresponding ISTRANxx DD statement. The $FMS/xxx control card would specify the transaction source as ISAM. The following job could be used to update a VSAM file using a VSAM file as input transactions.

```
//UPD       EXEC XFM,VSCAT=NIPSM,
//          VSDSN='VSAM.PRIME',
//          LIB=PRIME
//FM.TRANS  DD  DSN=VSAM.SECOND,DISP=SHR,
//              AMP='AMORG'
//FM.SYSIN  DD   *
$FMS/UPD,PRIME,ZZZ,,DISK,ISAM
/*
```

6.2.4 Processing SAM Files with a VSAM FFT/LS

It is possible for the user to elect to perform a SAM file generate using the FFT and logic statements from a VSAM file. This can be accomplished by specifying the VSCAT, VSDSN, and SAMOUT parameters along with the other parameters needed to perform a SAM run. The following job could be used to generate a SAM file using a VSAM FFT:

JOB PREPARATION

```
//SGEN     EXEC XFM,VSCAT=NIPSM,
//          VSDSN='VSAM.PRIME',
//          SAMOUT=,VSMOUT='SER=xxxxxx'
//FM.SYSIN  DD  *
$FMS/GEN,PRIME,YYY,,TAPE,CARD
.
.    GENERATE TRANSACTIONS
.
/*
```

The user can also perform a SAM update while using the FFT and logic statements from an existing VSAM file. This is accomplished by preparing the EXEC statement for a normal SAM update while including the VSCAT parameter and overriding the FM.ISAMWORK DD statement to point to the VSAM file. The user must provide the DSNAME, DISP, and AMP parameters on the ISAMWORK DD statement. The following job could be used to update a SAM file using the FFT and logic statements from a VSAM file:

```
//SUPD     EXEC XFM,VSCAT=NIPSM,
//          SAM=PRIME,LIB=PRIME,
//          VSAM='SER=xxxxxx',
//          VSMOUT='SER=xxxxxx',
//          SAMOUT=
//FM.ISAMWORK  DD  DSN=VSAM.PRIME,DISP=SHR,
//               AMP='AMORG'
//FM.SYSIN  DD  *
$FMS/UPD,PRIME,XYZ,,TAPE,CARD
.
.    UPDATE TRANSACTIONS
.
/*
```

6.2.5 Other FM VSAM Considerations

If the user employs the AFR operator in POOL to address a VSAM file, the DD statement for the referenced VSAM file must specify the DSNAME, DISP and AMP parameters. Also the user must include the VSCAT parameter on the EXEC statement. The following DD statement could be used for an AFR referenced VSAM file:

```
//AFRVSM  DD   DSN=VSAM.PRIME,DISP=SHR,
//             AMP='AMORG'
```

The BSZNEWF parameter used to specify output block size does not apply when generating a VSAM file, as its block size was established when the file cluster was defined via IDCAMS.

When the user is performing a SAM generate with a VSAM FFT, BSZNEWF must be used to specify the block size of the new SAM file if a block size other than 1004 is desired.

6.3 File Revision (FR)

File Revision (FR) will revise the format of a NIPS VSAM file as well as a SAM or ISAM file. The new FFT can be either an ISAM FFT or a VSAM FFT. When running FR against a VSAM file or using a new VSAM FFT, the user must include the VSCAT symbolic parameter on the EXEC statement to specify the NIPS user catalog.

When the user is revising an old VSAM file, the EXEC statement must include the VSOLDF parameter (specifying the old file name) and the TRANTYP parameter (equal to VSAM). Additionally, the following JCL override DD statement must be included:

```
//FR.VSAM  DD  AMP='AMORG'
```

When the new FFT is a VSAM file, the user must include the VSDSN parameter (specifying the FFT name) on the EXEC statement.

The following JCL could be used to revise an old VSAM file using a new VSAM FFT:

```
//REV      EXEC XFR,VSCAT=NIPSM,
//         VSDSN='NEW.PRIME',
//         VSOLDF='VSAM.PRIME',
//         TRANTYP=VSAM,
//         VSMOUT='SER=xxxxxx'
//FR.VSAM DD AMP='AMORG'
//FR.SYSIN  DD  *
           FR CONTROL CARDS
/*
```

The block size of the new SAM file will be 1004 when the file being revised is a VSAM file, unless a different block size is specified in the BSZNEWF parameter on the EXEC statement.

6.4 File Structure (FS)

File Structure (FS) is used to structure the FFT for a new file. This new file can be either an ISAM file or a VSAM file. When the new file is VSAM, the user must have previously defined the file using the VSAM service routine IDCAMS (section 6.1.2). To structure a VSAM file FFT, the user must include the VSCAT parameter (specifying the name of the NIPS user catalog), the VSDSN parameter (specifying the name of the VSAM file), and the NONVSM parameter (must equal 'DUMMY,'). The following JCL could be used to structure a VSAM FFT:

```
//FSSTR    EXEC XFS,VSCAT=NIPSM,
//         VSDSN='VSAM.FFT.PRIME',
//         NONVSM='DUMMY,',
//         LIB=PRIME
//FS.SYSIN  DD  *
     SOURCE FFT DECK
/*
```

The BSZNEWF parameter, which is normally used to specify a block size other than 1004, has no application when structuring a VSAM FFT. Block size was specified when the VSAM file was defined with the IDCAMS utility.

If the structure of a VSAM FFT fails, the file will not be automatically deleted. The user must delete the VSAM file using IDCAMS (section 6.1.4) and then redefine the file with IDCAMS prior to rerunning the FS job.

6.5 VSAM to SAM Procedure (XISTOS)

The procedure which is used to unload an ISAM file to SAM (XISTOS) can also be used to unload a VSAM file. To unload a VSAM file, the user must include the VSCAT parameter (specifying the NIPS catalog) and the VSDSN parameter (specifying the VSAM file name) on the EXEC statement. ISAM parameters are not permitted. The blocksize of the output SAM file will be 1004 unless

overridden by use of the BSZNEWF parameter on the EXEC statement. The following JCL could be used to unload a VSAM file:

```
//VSTS     EXEC XISTOS,VSCAT=NIPSM,
//          VSDSN='VSAM.PRIME',
//          SAM=PRIME,VSAM='SER=xxxxxx'
/*
```

6.6 OP

The user of a VSAM file can use the OP component of NIPS to produce reports as if the file was an ISAM file. To process a VSAM file, the user specifies the name of the VSAM file using the ISAM parameter and the NIPS user catalog with the VSCAT parameter. The user must also include the following JCL override statement:

```
//OP.DATAFILE  DD  AMP='AMORG'
```

The following JCL could be used to run a source direct OP against the VSAM.PRIME file:

```
//SD       EXEC XOPSD,VSCAT=NIPSM,
//          ISAM='VSAM.PRIME',LIB=PRIME
//OP.DATAFILE  DD  AMP='AMORG'
//OP.SYSIN  DD  *
          OP CONTROL CARDS
/*
```

To perform merged file processing against VSAM files, the user must specify the name of the NIPS user catalog with the VSCAT parameter and the names of the VSAM files using the ISAM, ISAM1, and ISAM2 parameters. For each VSAM file used, the user must include a JCL override DD statement to specify AMP='AMORG'. The DD statements are DATAFILE for ISAM, DATAFIL1 for ISAM1 and DATAFIL2 for ISAM2. Additional secondary files may be specified by including a DATAFILx DD statement for each additional file, when x is a 1-digit number from 3 to 9. Each DD statement must contain the file DSNAME, DISP=SHR and AMP='AMORG'. The following JCL could be used to run a merged file source direct OP against VSAM files:

JOB PREPARATION

```
//SD3      EXEC XOPSD,VSCAT=NIPSM,
//    ISAM='VSAM.PRIME',ISAM1='VSAM.SECOND',
//    ISAM2='VSAM.THIRD',LIB=PRIME
//OP.DATAFILE  DD   AMP='AMORG'
//OP.DATAFIL1  DD   AMP='AMORG'
//OP.DATAFIL2  DD   AMP='AMORG'
//OP.DATAFIL3  DD   AMP='AMORG',DSN=VSAM.FORTH,
//         DISP=SHR
//OP.SYSIN  DD  *
         OP CONTROL CARDS
/*
```

The only operand which must be coded for the AMP parameter is 'AMORG'. Additional operands can be supplied and their descriptions are contained in the OS/VS VSAM Programmer's Guide (GC26-3838).

6.7 QUIP

Information can be retrieved and output from VSAM files using QUIP, but only in the batch mode. To process a VSAM file using QUIP the user must specify the NIPS user catalog with the VSCAT parameter and the name of the file in the ISAM parameter. The user must also override the DATAFILE DD statement to specify AMP='AMORG'. The following JCL could be used to run a source direct QUIP against a VSAM file:

```
//QPVSM   EXEC XQUIPSD,VSCAT=NIPSM,
//    ISAM='VSAM.PRIME',LIB=PRIME
//QUIP.DATAFILE DD AMP='AMORG'
//QUIP.SYSIN  DD  *
     QUIP SOURCE STATEMENT
/*
```

When using IFO, the first two secondary files are specified by using the ISAM1 and ISAM2 parameters on the EXEC statement. The user must also override the DATAFIL1 and DATAFIL2 DD statements to include the AMP='AMORG' parameter. If more than two secondary files are required, each additional file must be specified by including a DATAFILx DD statement, where x is a unique 1-digit number. The DD statement must include the DSNAME, DISP=SHR and AMP='AMORG'. A total of nine secondary files may be specified. The following JCL could be used to run a source direct QUIP with IFO against VSAM files.

JOB PREPARATION

```
//QPVSM    EXEC XQUIPSD,VSCAT=NIPSM,
//         ISAM='VSAM.PRIME',ISAM1='VSAM.SECOND',
//         ISAM2='VSAM.THIRD',LIB=PRIME
//QUIP.DATAFILE     DD   AMP='AMORG'
//QUIP.DATAFIL1     DD   AMP='AMORG'
//QUIP.DATAFIL2     DD   AMP='AMORG'
//QUIP.DATAFIL3     DD   DSN=VSAM.FORTH,DISP=SHR,AMP='AMORG'
//QUIP.SYSIN   DD   *
     QUIP SOURCE STATEMENTS
/*
```

The only operand which must be coded for the AMP parameter is 'AMORG'. Additional operands can be supplied as described in the OS/VS VSAM Programmer's Guide (QC26-3838).

6.8 RASP

RASP may be used to retrieve data from a VSAM file for output by OP or QUIP. To retrieve data from a VSAM file the user must specify the NIPS user catalog with the VSCAT parameter and the file name with the ISAM parameter. The user must also override the DATAFILE DD statement to include AMP='AMORG'.

When doing a merged file retrieval against VSAM files, the user must specify the names of the additional files via the ISAM1 and ISAM2 parameters and must provide overrides for the DATAFIL1 and DATAFIL2 DD statements to include AMP='AMORG'. When more than three files are being processed, the user must add a DATAFILx DD statement for each file, where x is a unique one digit number. Each DD statement must have a DSNAME, DISP=SHR and AMP='AMORG'. The following JCL could be used to run a RASP against VSAM files:

```
//RSP      EXEC XRASP,VSCAT=NIPSM,
//    ISAM='VSAM.PRIME',
//    ISAM1='VSAM.SECOND',
//    LIB=PRIME
//RASP.DATAFILE DD AMP='AMORG'
//RASP.DATAFIL1  DD  AMP='AMORG'
//RASP.SYSIN  DD  *
     RASP SOURCE STATEMENTS
/*
```

JOB PREPARATION

The only operand which must be coded for the AMP parameter is 'AMORG'. Additional operands can be supplied as described in the OS/VS VSAM Programmer's Guide (GC26-3838).

6.9 SAM to VSAM Procedure (XSTOIS)

The procedure which is used to load an ISAM file from a SAM file (XSTOIS) can also be used to load a VSAM FILE. To load a VSAM file, the user must first define the file using the VSAM service routine IDCAMS (section 6.1.2). To use the XSTOIS procedure for VSAM, the user must specify the NIPS user catalog with the VSCAT parameter, specify the file name with the VSDSN parameter and void the ISAM files by specifying NONVSM='DUMMY,'. The following JCL could be used to load a VSAM file:

```
//STOVS    EXEC XSTOIS,VSCAT=NIPSM,
//         VSDSN='VSAM.PRIME',
//         NONVSM='DUMMY,',
//         SAM=PRIME,VSAM='ser=xxxxxx'
/*
```

6.10 XDMPLIB

To run an XDMPLIB job against a VSAM file, the user must specify the NIPS user catalog with the VSCAT parameter and the file name with the ISAM parameter. The user must also override the DATAFILE DD statement to include AMP='AMORG'. The following JCL could be used to run an XDMPLIB job against a VSAM file:

```
//DMP      EXEC XDMPLIB,VSCAT=NIPSM,
//    ISAM='VSAM.PRIME'
//UTDMP.DATAFILE  DD  AMP='AMORG'
//UTDMP.SYSIN  DD  *
     XDMPLIB CONTROL CARDS
/*
```

Additional AMP operands may be specified as described in the OS/VS VSAM Programmer's Guide (GC26-3838).

6.11 XCLASS

To change the classification of a VSAM file, the XCLASS procedure would be used. The user must specify the NIPS user catalog with the VSCAT parameter and the file name with the ISAM parameter. The DATAFILE DD statement must be overriden to include AMP='AMORG'. The following JCL could be used to change the classification of a VSAM file:

```
//CLS      EXEC XCLASS,VSCAT=NIPSM,ISAM='VSAM.PRIME'
//CLASS.DATAFILE  DD  AMP='AMORG'
//CLASS.SYSIN  DD  *
     CLASSIFICATION CARD
/*
```

Additional AMP operands may be specified as described in the OS/VS VSAM Programmer's Guide (GC26-3838).

6.12 XUTFSCAN

To use a VSAM file as the input for a field scan job, the user specifies the NIPS user catalog with the VSCAT parameter and the file name with the ISAM parameter. The DATAFILE DD statement must be overriden to include AMP='AMORG'. The following JCL could be used to run an XUTFSCAN involving a VSAM file:

```
//    EXEC XUTFSCAN,VSCAT=NIPSM,ISAM='VSAM.PRIME',
//      LIB=PRIME,TRANS=PRIMET
//UTFLDSCN.DATAFILE  DD  AMP='AMORG'
//UTFLDSCN.SYSIN DD *
     SCAN CARDS
/*
```

Additional AMP operands may be specified as described in the OS/VS VSAM Programmer's Guide (GC26-3838).

6.13 XSP

The XSP procedure can be used to perform index specification on a VSAM file. The user must specify the NIPS user catalog with the VSCAT parameter and the file name with the ISAM procedure. The NEWFILE DD statement must be overridden to include AMP='AMORG'. The following JCL could be used to index a VSAM file.

JOB PREPARATION

```
//XSPV     EXEC XSP,VSCAT=NIPSM,
//    ISAM='VSAM.PRIME',XINDEX=PRIME,
//    XVOL='SER=xxxxxx',LIB=PRIME
//XSP.NEWFILE  DD  AMP='AMORG'
//XSP.SYSIN  DD  *
      INDEX CARDS
/*
```

Additional AMP operands may be specified as described in the OS/VS VSAM Programmer's Guide (GC26-3838).

6.14 XTRDISK

When the user is unloading an Index Data Set for a VSAM file and the option STAT=YES is specified, the NIPS user catalog must be specified with the VSCAT parameter and the file name with the ISAM parameter. The DATAFILE DD statement must be overridden to include AMP='AMORG'. The following JCL could be used to unload the index of a VSAM file:

```
//STPS     EXEC XTRDISK,VSCAT=NIPSM,STAT=YES,
//    ISAM='VSAM.PRIME',XFNAME=PRIMEX
//XTR.DATAFILE  DD  AMP='AMORG'
/*
```

Additional AMP operands may be specified as described in the OS/VS VSAM Programmer's Guide (GC26-3838).

6.15 XKA

The XKA procedure can be used to perform keyword analysis on a VSAM file. The user must specify the NIPS user catalog with the VSCAT parameter and the file name with the ISAM parameter. The DATAFILE DD statement must be overriden to include AMP='AMORG'. The following JCL could be used to perform keyword analysis on a VSAM file:

JOB PREPARATION

```
//KEY EXEC XKA,VSCAT=NIPSM,
//   ISAM='VSAM.PRIME',LIB=PRIME
//XKA.DATAFILE  DD  AMP='AMORG'
//XKA.SYSIN  DD  *
     XKA CONTROL CARDS
/*
```

Additional AMP operands may be specified as described in the OS/VS VSAM Programmer's Guide (GC26-3838).

Appendix A

SYMBOLIC PARAMETER DEFINITIONS

PARAMETER	DEFINITION
AUXBUFN	This defines the number of buffers used by the auxiliary output data sets in FM.
AUXSP	This defines the space allocation for temporary storage of auxiliary output and card transactions in FM.
BLK	This defines the blocksize for the library created by XTABGEN or XSUBLDR. It defaults to 7294.
BLKSIZE	This defines the blocksize for the Index Data Set.
BSZFILE	This is used to indicate the block size of an input SAM file on unlabeled magnetic tape. Its use is required only when the block size is not 1,004 bytes.
BSZNEWF	This is used to change the block size of a file, or to specify a block size greater than 1,004 bytes when creating a file.
CHKDSP	This parameter defines the conditional disposition for the work data sets and the CHECKDD data set utilized in check-point/restart in case of an ABEND.
CHKID	This parameter defines the DSNAMEs for the work data sets and the CHECKDD DD statement in the 'execute only' procedure.

PARAMETER	DEFINITION
CHKSP	This parameter defines the space allocated for the data set (CHECKDD DD card) utilized in checkpoint/restart for the 'execute only' procedures.
CYLOFL	This specifies amount of cylinder overflow for a data base generated by FM or loaded to disk from tape.
DEN	This specifies the tape density.
DNSMOUT	This defines the name of the SAM file generated by this run. If this parameter is not used, the data set name in the FMS control card is used.
EROPT	This specifies the error option to be selected when an I/O error occurs while reading SAM transactions in FM.
FFT FFT1 FFT2	These define the FFTs for use with non-NIPS files.
GEN	This is used to control the allocation of the NEWFILE data set in FM.

Please note that all references to secondary indexing functions include keyword indexing functions. Keyword stop word tables and dictionaries are stored as members in either private user libraries or in the common DUMMY.FILEL library. To obtain keyword functions, use secondary indexing procedures and specify symbolic parameters to include the libraries that contain keyword data (see File Libraries, section 2.3.3).

INDEX	This defines the number of cylinders allocated to the index of a new ISAM data file.
ISAM ISAM1 ISAM2	These define ISAM or VSAM data file names.
JOBLIB	This defines the partitioned data set containing program load modules.

PARAMETER	DEFINITION
JOBMAC	This defines the partitioned data set containing generative code macros.
LAB	This defines label types for sequential files.
LABIN	This defines the label type for the SAM data file for the XTRDISK procedure when requesting the statistics option.
LIB LIB1 LIB2	These define user-library file names.
LIBDISP	This defines the disposition of user library(s).
NAME	This specifies the name of the member to be listed from a source library.
NBRBLK	This defines the number of blocks required to hold the Index Data Set.
MCTSP	This defines the space allocation for temporary storage of new data records and data records whose key or length has changed during an FM update.
MODLIB	This defines the name of the user's subroutine library for the subroutine loader.
NDISP	This defines the disposition of the ISAM data file in XFS and XSTOIS procedures, and the disposition of the new file in 360 to 1410 conversion.
NEWFFT	This defines the name of the new FFT for FR.
NEWVSM	This defines the name of the newly generated VSAM file in FM.
NRMDSP	This parameter defines the normal disposition for the work data sets and the CHECKDD data set utilized in checkpoint/restart in case of an ABEND.

PARAMETER	DEFINITION
NONVSM	This is used to control allocation of ISAM data files when processing VSAM files in FS and ISTOS.
ODISP	This defines the disposition of the old data file in the X360CON and the XISTOS procedures.
OLDLAB	This defines the label parameter of the 1410 data file during data conversion.
OLDSAM	This defines the name of the old SAM file in the XISTOS procedure.
OLDVSAM	This defines the volume for the old SAM file in the XISTOS procedure.
OSDISP	This defines the disposition of the old SAM file in the XISTOS procedure.
OVFLOW	This defines the number of cylinders allocated to the overflow area for a new ISAM data file.
PRIME	This defines the number of cylinders allocated to the prime area for a new ISAM data file.
PTFJOBL	This defines the data set name of a partitioned data set containing the PTF load modules.
PTFJOBM	This defines the data set name of the partitioned data set containing PTF generative code macros.
QDF	This defines the name of the RASP qualified data file if other than a temporary name is required.
QDFSP	This defines the space allocation for the RASP qualified data file.
QDISP	This defines the disposition of the RASP qualified data file and qualified record table.
QRT	This defines the name of the RASP qualified record table.

PARAMETER	DEFINITION
QRTSP	This defines the space allocation for the RASP qualified record table.
SAM SAM1 SAM2	These define SAM data file names.
SEQNO	This parameter defines the data set's position with respect to other data sets on the volume.
SDISP	This defines the disposition of the user's source library.
SLAB	This defines the label type for sort work tapes.
SAMOUT	This is used to control the allocation of the FMSAMOUT and FMNDATA data sets in FM.
SORTSP	This defines the cylinder allocation for disk sort work areas.
STAT	This requests the statistics option for for XTRDISK index utility.
STG	This defines the unit type for sort work areas.
SOURCL	This defines the user's source library names.
TBLK	This defines the blocksize for the File Analysis Statistics transaction data set.
TDISP	This defines the disposition for the File Analysis Statistics transaction data set.
TRANS	This defines the File Analysis Statistics transaction data file name.
TRANSP	This defines the space allocation for temporary storage of update transactions with the sort key and logic statement name appended.

PARAMETER	DEFINITION
TRANTYP	This defines to the FM component the data set containing records to be used as transactions during execution of FR. Default value is ISAM. The value for this parameter is either ISAM, SAM or VSAM and denotes the access method associated with the file being revised.
TRBUFNO	This defines the number of buffers used by some of the transaction temporary data sets.
TRCH	This specifies the TRTCH DCB parameter for 7-track tapes.
UCHK	This parameter defines the unit type for the CHECKDD DD statement in the 'execute only' procedures.
UFFT	This defines the unit type of the new FFT for FR or the unit type of the FTT used with a primary non-NIPS file in QUIP.
UFFT1 UFFT2	These define the unit type of the FFTs used with secondary non-NIPS files in QUIP.
UISAM	This defines the unit type for all ISAM files.
ULIB ULIB1 ULIB2	These define the unit type for all user libraries.
UQDF	This defines the unit type for the RASP qualified data file.
UQRT	This defines the unit type for the RASP-qualified record table.
USAM	This defines the unit type for all SAM files.
USMMCT	This defines the unit type of the tape on which the new file will be written if major control fields are changed during an FM SAM or FR run.

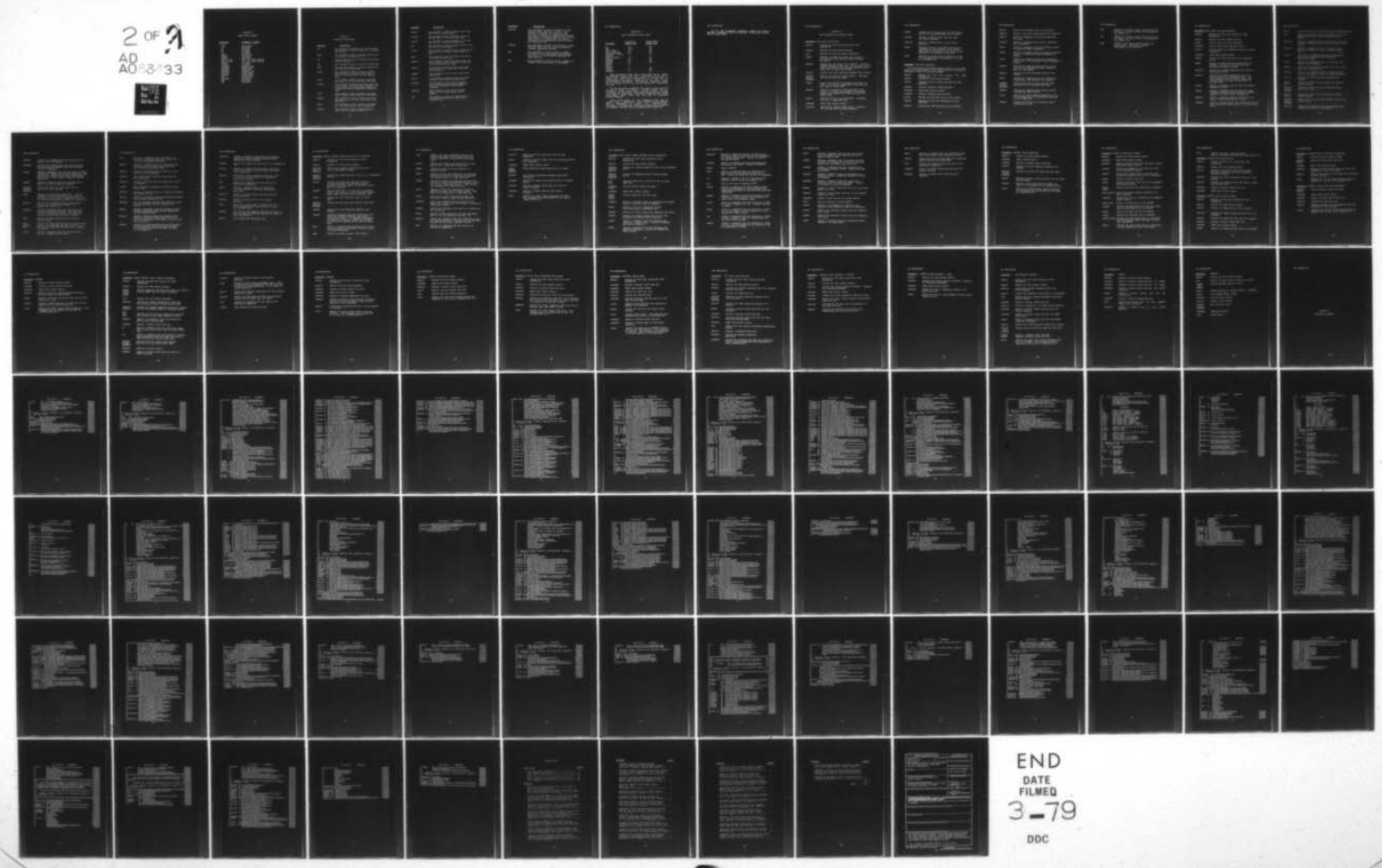
PARAMETER	DEFINITION
USMOUT	This defines the unit type of the tape on which the new file will be written if no major control fields are changed during an FM SAM or FR run.
USOURCL	This defines the unit type for the user source library.
U1410	This defines the unit for a 1410 data base.
VCHK	This parameter defines the volume for the CHECKDD DD statement in the 'execute only' procedures.
VFFT	This defines the volume of the new FFT for FR or the volume of the FFT used with a primary non-NIPS file in QUIP.
VFFT1 VFFT2	These define the volume of the FFTs used with secondary non-NIPS files in QUIP.
VISAM VISAM1 VISAM2	These define the volume for all ISAM files.
VLIB VLIB1 VLIB2	These define the volume for user libraries.
VINDEX VOVFLOW VPRIME	These define the volumes on which the respective portions (INDEX, OVFLOW, PRIME) of the new ISAM data set will be allocated during an FM generate run.
VSOURCL	This defines the volume of the user's source library.
VQDF	This defines the volume for the RASP-qualified data file.
VQRT	This defines the volume for the RASP-qualified record table.
VSAM VSAM1 VSAM2	These define the volume for all SAM files.

PARAMETER	DEFINITION
VSCAT	This defines the NIPS user catalog for VSAM processing.
VSDSN	This defines the VSAM file being processed in FM, FS, ISTOS, and STOIS and the new VSAM FFT in FR.
VSMMCT	This defines the volume on which the new file will be written if major control fields are changed during an FM SAM or FR run.
VSMOUT	This defines the volume on which the new file will be written if no major control fields are changed during an FM SAM or FR run.
VSOLDF	This defines the VSAM file being revised in FR.
V1410	This defines the volume for a 1410 data base.
XDISP	This defines the disposition of the Index Data Set.
XFDISP	This parameter describes the status of the "FROM" Index Data Set, and indicates what is to be done with it after termination of UTNDXTFR.
XFNAME	This parameter defines the name given to the "FROM" Index Data Set for UTNDXTFR.
XFUNIT	This UTNDXTFR parameter is used to specify information about the input unit used by the "FROM" Index Data Set.
XFVOL	This UTNDXTFR parameter provides information about the volume on which the "FROM" Index Data Set resides.
XINDEX	This defines the name of the Index Data Set.
XINDEX1 XINDEX2	This defines additional Index Data Sets to be used in a merge file retrieval.

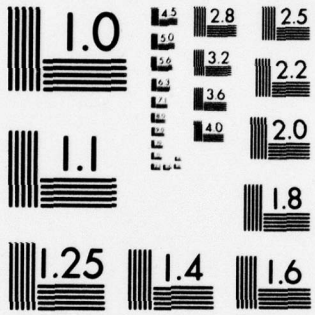
PARAMETER	DEFINITION
XTDISP	This parameter describes the status of the "TO" Index Data Set, and indicates what is to be done with it after termination of UTNDXTFR.
XTNAME	This parameter defines the name given to the "TO" Index Data Set for UTNDXTFR.
XTVOL	This UTNDXTFR parameter provides information about the volume on which the "TO" Index Data Set resides.
XUNIT	This is the unit type for the Index Data Set.
XVOL XVOL1 XVOL2	These define the volume for Index Data Sets.

AD-A063 433 COMMAND AND CONTROL TECHNICAL CENTER WASHINGTON D C F/G 9/2
NMCS INFORMATION PROCESSING SYSTEM 360 FORMATTED FILE SYSTEM (N--ETC(U)
SEP 78 C K HILL
UNCLASSIFIED CCTC-CSM-UM-15-78-V8-REV SBIE-AD-E100 133 NL

2 OF 2
AD
A063433



END
DATE
FILMED
3-79
DDC



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

Appendix B

INPUT SOURCE DDNAME

COMPONENT	STEPNAME . DDNAME
RASP	RASP.SYSIN
OP	OP.SYSIN
FM	FM.SYSIN
FS	FS.SYSIN
FR	FR.SYSIN
QUIP	QUIP.SYSIN
TABGEN	TAB.SYSIN
SUBLDR	SUB.SYSIN
SAM TO ISAM	STP1 (No Input Source)
ISAM TO SAM	STP1 (No Input Source)
360CON	GO.SYSIN
1410CON	GO.SYSIN
UTQRTQDF	QRTQDF.SYSIN
UTDMPLIB	UTDMP.SYSIN
UTFLDSCN	UTF.SYSIN
UTSUBCHK	SUBCHK.SYSIN
UTCLASS	CLASS.SYSIN
UTSOURC	SOURC.SYSIN
XSP	UTXSP.SYSIN
UTNDXKAN	XKA.SYSIN
XKM	XKM.SYSIN
XKA	XKA.SYSIN

Appendix C

PROCEDURES DESCRIPTIONS

PROCEDURE	DESCRIPTION
XOP	This procedure is normally only used to publish the results of a RASP retrieval run or to structure an RIT.
XOPSD	This procedure is used to publish directly from a data base or to structure a RIT.
XFM	This procedure is used to perform all file maintenance functions.
XFS	This procedure is used to structure a data base.
XRASP	This procedure is used to retrieve and sort data from one or more data files.
XISTOS	This procedure is used to convert an ISAM or VSAM data file to a SAM tape data file or copy a SAM tape data file to a SAM tape data file.
XSTOIS	This procedure is used to convert a SAM tape data base to an ISAM or VSAM disk data base.
XSUBLDR	This procedure is used to link-edit a user subroutine (which has already been assembled) into a file library with the necessary control information to load the subroutine at execution time.
XFMEX	This procedure is used to perform file maintenance updates using stored logic statements.
XRASPEX	This procedures is used to retrieve and sort data from one or more data files using stored retrievals.
XOPEX	This procedure is used to publish the results of a RASP retrieval run using a stored RIT.
XOPSDEX	This procedure is used to publish directly from a data file using a stored RIT.

PROCEDURE	DESCRIPTION
XTABGEN	This procedure is used to generate tables and place them in a user library.
X1410CON	This procedure is used to convert a NIPS 360 FFS data base to a 1410 NIPS data base.
X360CON	This procedure is used to convert a 1410 NIPS data base to a NIPS 360 FFS data base.
XFR	This procedure is used to revise a NIPS 360 FFS data base to a new format.
XQRTQDF	This procedure is used to create a NIPS 360 FFS data file from the answer file produced by RASP.
XQUIP	This procedure is used to publish the results from a RASP retrieval run.
XQUIPSD	This procedure is used to retrieve, sort, and output directly from a NIPS 360 FFS data file or a S/360 non-NIPS data file.
XDMPLIB	This procedure is used to print logic statements and/or report names from a NIPS 360 FFS data file.
XSUBCHK	This procedure is used to test user-written subroutines.
XCLASS	This procedure is used to change the classification on any NIPS 360 FFS data file.
XUTFSCAN	This procedure is used to scan component source statements and count the number of data field references in each source statement.
XUTSOURC	This procedure is used to add, replace, delete, or list source members from a library.
XSP	This procedure is used to either generate or update an Index Data Set based on an ISAM data file.

PROCEDURE	DESCRIPTION
XTRDISK	This procedure is used to transfer a disk-resident Index Data Set to tape. This operation condenses the Index Data Set. The tape so created is a sequential data set consisting of variable-length blocked records that contain the source data and control information for subsequently reconstructing the Index Data Set.
XTRTAPE	This procedure is used to reconstruct a disk-resident Index Data Set from a previously unloaded sequential version.
XKA	This procedure is used to obtain printed listings of text, nonkeywords, and keywords that occur in a file without updating the index data set.
XKM	This procedure is used to create, update, or display a stop word table or dictionary.

Appendix D

NIPS PROCEDURE RETURN CODES

PROCEDURE	SUCCESSFUL RETURN CODE	UNSUCCESSFUL RETURN CODE
XFS	0,1	111
XFR	0,1,2	121
XFM, XFMEX	0,1	131
XRASP, XRASPEX	0,1	141
XOP,XOPSD, XOPEX, XOPSDEX	0,1	151
XQUIP, XQUIPSD	0,1	151
XTABGEN	0,1	171
XSUBLDR	0,1	171
XQRTQDF	0,1	181
XISTOS	0	
XSTOIS	0	
XSUBCHK	0	191
XCLASS	0,1	211
XUTSOURC	0	221

These procedures issue return codes which may be tested by subsequent steps of the job. A successful return code is issued upon satisfactory completion of the job step; an unsuccessful return code is issued if the job step is not satisfactorily completed. Satisfactory completion means that the results of the job step are usable for a succeeding job step.

The XOP and XOPSD procedures will issue a return code of 151 when no output is published, unless output was not requested; in this case a return code of 151 will be issued if no RITs are structured. The other procedures will issue unsuccessful return codes when the step is terminated without ABENDing.

Return codes issued by all preceding steps may be tested; e.g., COND=(3,GE); those issued for any specific step or combination of steps; i.e., COND=((131,EQ),(181,EQ)), to bypass this job step if either the FM step or the UTQRTQDF job step was not satisfactorily completed.

JOB PREPARATION

See the IBM System/360 Operating System Job Control Language Reference, Form GC28-6704 for a description of the use of return codes.

Appendix E

NIPS PROCEDURE DD STATEMENT USAGE

PROCEDURE: XFS (File Structuring)

STEPCAT	Defines the NIPS user catalog for VSAM processing.
STEPLIB	Defines the NIPS system library.
SLIB	Defines the user and system libraries.
NEWVSM	Defines the VSAM FFT output by FS phase 2. The file must have been previously defined by IDCAMS.
NEWFILE	Defines the FFT output of FS phase 2. It is an indexed sequential data set. If the FFT is to be saved, the ISAM, VISAM, and DISP symbolic parameters must be coded accordingly.
SORTLIB	Defines the S/360 Operating System's sort library.
SORTWK01 through SORTWK06	Defines work data set used by SORT. They must all be the same unit type.
SORTIN	Defines the output of FS phase 1 and input to SORT. Contains the temporary FFT entries, one per FIELD or GROUP card.
SORTOUT	Defines the output of SORT and input to FS phase 2. Contains the temporary FFT entries sorted into alphabetical order on FIELD and GROUP names.
SYSOUT	Defines the error list from SORT. Currently set up as a dummy data set.
SYSUDUMP	ABEND dump printer output.
SYSPRINT	Defines the printed output of FS. Contains FFT listings and any error messages.

JOB PREPARATION

DELNEW	Defines the FFT output that is scratched if the FS run was unsuccessful due to errors.
SOURCPRT	Defines a printer output data set used for source listings.
SOURCLIB	Defines a library used to store source material.
SYSUT1	Temporary data set for INDEX Specification statements to be processed by the INDEX Specification processor, IXSP, which builds descriptor (D) records for the file.
SYSIN	Defines the input source statements for FS. This statement must be supplied by the user as //FS.SYSIN DD *.

PROCEDURE: XFR (File Revision)

The following DD statements appear in the FR and FRTST steps of this procedure. For a description of all other DD statements, refer to the XFM procedure.

STEPCAT	Defines the NIPS user catalog for VSAM processing.
STEPLIB	Defines the NIPS system library.
SYSOUT	Defines a printer output data set for sort messages.
SYSPRINT	Defines a printer output data set.
SYSUDUMP	ABEND dump printer output.
SYSPUNCH	Defines a punch output data set.
OLDVSM	Defines the VSAM data file to be revised.
VSMFILE	Defines the data set containing the new VSAM FFT.
OLDFILE	Defines the ISAM data file to be revised.

JOB PREPARATION

SAMFILE	Defines the SAM data file to be revised.
DATAFILE	Defines a data set containing the new ISAM FFT.
TDDRCDS	Defines a temporary data set for generated TDD cards.
POOLRCDS	Defines a temporary data set for generated POOL statements.
FRVSNA	Defines a temporary data set containing first 24 generated logic statements.
FRVSNB	Defines a temporary data set containing second group of 24 generated logic statements if needed.
FRVSNC	Defines the temporary data set containing a third group of 24 generated logic statements if needed.
FRGENCD	Defines the temporary data set FMS control card for file generation steps; passed to GENT and GEND steps.
SORTLIB	Defines the S/360 Operating System's Sort library.
SORTIN	Defines the temporary data set containing intermediate work records to be sorted.
SORTWK01 through SORTWK06	Defines the work data sets used by SORT. They must all be the same unit type.
SORTOUT	Defines the temporary data set for sorted intermediate work records.
SYSIN	Defines the input source statements for FR. This statement must be supplied by the user as //FR.SYSIN DD *.
TRANTYP	Defines the DD card to be used for input transaction to FM.

JOB PREPARATION

ISAM	Defines to FM where update transactions may be found. Entries in this DD correspond to the OLDFILE DD.
SAM	Defines to FM where update transactions may be found. Entries in this DD correspond to the SAMFILE DD.
VSAM	Defines to FM where update transactions may be found. Entries in this DD correspond to the OLDVSM DD.

JOB PREPARATION

PROCEDURE: XFM, XFMEX (File Maintenance)

STEPCAT	Defines the NIPS user catalog for VSAM processing.
STEPLIB	Defines the NIPS system library.
SYSOUT	Printer output for sort messages.
SYSPRINT	Printer output for remainder of FM.
SYSLIST	Printer output for assembler listing of logic statements.
SYSUDUMP	ABEND dump printer output.
VSMFILE	Defines the VSAM data file to be processed by FM.
NEWVSM	Defines the VSAM data file generated by an FMS/GEN. This data file must have been previously defined by IDCAMS.
DATAFILE	Defines the indexed sequential data file to be processed by FM.
NEWFILE	Defines the indexed sequential data file created during an FM generation run. The user normally codes the INDEX, PRIME, and OVFLOW symbolic parameters accordingly on the EXEC card.
FMCOMM	Defines a temporary data set used for section communication data.
FMFLUD	Defines a temporary data set used for logic statement compilation and contains formatted images of the user's input statements.
FMTRANS	Defines a temporary data set containing the images of card transactions, if any.
FMLABELS	Defines a temporary data set containing lists of POOL instruction labels used in the logic statements.

JOB PREPARATION

FMSTAT	Printer output for statistics gathered during the run.
FMCM	Defines a temporary data set containing an intermediate form of the logic statement control records.
FMCMFILE	Defines a temporary data set containing the logic statement control member records in their final format.
FMLITFIL	Defines a temporary data set containing the literals produced for logic statement compilation.
SYSLIB	Identifies the library containing those macros necessary to generate logic statements (Gen Code macros).
FMLEIN	Defines the temporary data set containing the output of the compiler.
SYSLMOD	Defines the data set containing the logic statement CSECT load modules.
TEMLSREC	Defines the temporary data set containing the list of the names of temporary logic statements.
SORTLIB	Defines the S/360 Operating System's sort library.
SORTIN TAPEIN	Defines the data sets containing the unsorted update records.
SORTWK01 through SORTWK06	Defines the work data sets residing on disk and used by sort.
SORTOUT TAPEOUT	Defines the data sets containing the sorted update records.
TAPEWK01 through TAPEWK04	Defines the work data sets residing on tape and used by sort.
FMLEFILE	Defines the temporary data set containing the linkage editor control statements.

JOB PREPARATION

FMMACRO	Defines the temporary data set containing the input to the FM compiler.
FMOUTMAC	Defines the temporary data set containing the Gen Code macro statements that will be input to the FM compiler.
RECSIN	Defines a temporary data set containing new data records and subsets to be used as input to a data record sort. These records will eventually be merged into the data file to form the new SAM data file.
RECSOUT	Defines a temporary data set containing the sorted new data records and subsets.
RECSWK01 through RECSWK06	Defines the disk sort work areas for the sort of the data records.
TRANS	Defines the transaction input file. This DD statement must be overridden by the user to identify a single disk or transaction data set. For multiple transaction sources see section 3.1.
FMSETTBL	Defines the temporary data set containing a list of the periodic set numbers to be accessed by the FM run.
FMAUXOP	Defines a temporary data set containing all the auxiliary output records. All auxiliary records, regardless of device they are to be written to, are first written to FMAUXOP.
PUNCH	Defines the punch data set for the user's punched output.
AUX1 through AUX5	Defines the data sets for tape or disk auxiliary output. The user must override these DD statements to describe the data sets he wishes to produce.
FMTAUX	Defines a temporary data set containing output for a second printer output.

JOB PREPARATION

SLIB	Defines a temporary partitioned data set containing temporary logic statements, and the user and system libraries.
OMMACS	Defines a temporary data set containing the compiler input of those macros generated for Ordinary Maintenance (OM).
GENFILE	Defines a temporary data set used as a work area for the OM compiler.
LITFILE	Same properties as FMLITFIL but dedicated for OM use only.
EDITFILE	Defines a data set used as a work area by the OM compiler.
FMNFLUD	Same properties as FMFLUD but dedicated for OM use only.
FMTEFILE	Defines a temporary data set containing extended TDD formats for OM.
FMAEFIL	Defines a temporary data set containing extended ADD statement records for OM.
FMSODANQ	This DD statement prevents the update of a data set simultaneously by SODA (Source Data) and FM.
NFLERR	Defines a temporary data set containing error messages reflecting those errors detected by the NFL editing steps.
NFLMACS	Defines a temporary data set containing NFL generated macro prototypes with their corresponding parameters; it is used as input to the macro generator.
ASSEMIN	Defines two concatenated data sets containing generated macros and those values specified in OM DEFINE statements; all used as input to the assembler.

JOB PREPARATION

ISAMWORK	Defines a temporary ISAM data set containing the FFT and logic statement library during sequential file processing.
SAMFILE	Defines the sequential data set to be processed by FM.
FMNDATA	Defines the sequential output data file when record key changes occurred during sequential file processing.
FMSAMOUT	Defines the output sequential data file that is produced by sequential file processing without key changes.
STAT	Defines the temporary data set used by the statistics capability.
TRANST	Defines a temporary data set containing the File Analysis Statistics Transactions data file.
SOURCPRT	Defines a printer output data set used for source listings.
SOURCLIB	Defines a library used to store source material.
SYSIN	Defines the input source statements for FM. This statement must be supplied by the user as //FM.SYSIN DD *.
IXTRANS	This defines the temporary SAM data set used to hold the Index Transaction Records for input to Index Maintenance.
XINDEX	This defines the Index Data Set.

JOB PREPARATION

Procedure: XRASP, XRASPEX (Retrieval and Sort Processor)

STEPCAT	Defines the NIPS user catalog for VSAM processing.
STEPLIB	Defines the NIPS system library.
DATAFILE DATAFIL1 DATAFIL2	Defines the indexed sequential data files to be interrogated by RASP.
SAMFILE SAMFILE1 SAMFILE2	Defines the sequential data files to be interrogated by RASP.
QDFILE	Defines the sequential data set generated by RASP containing the retrieval source statements and the subset of qualified records from the retrieval.
SORTOUT	Used by S/360 sort. At the end of retrieval proper, this data set is the Qualifying Record Table (QRT) generated during retrieval time.
SORTIN	Defines the input data set (QRT) to S/360 sort.
SORTWK01 through SORTWK06	Defines the work data sets used by S/360 sort.
SORTLIB	Defines the S/360 Operating System's sort library.
SYSLMOD	Defines a temporary data set which first contains RASP source statements and finally compiled retrievals. If permanent retrievals are specified, they are copied from SYSLMOD to the data set defined by the SLIB DD statement.
SLIB	Defines a temporary partitioned data set containing temporary retrievals, the user library and system libraries.
DLIB	Defines the user's primary file library.

JOB PREPARATION

TLIB	Defines the user's secondary file library. Library references will always search the library specified by DLIB before searching TLIB.
SYSLIB	Defines the library containing the Gen Code macros used to generate a retrieval.
DPUNCH	Defines the data set which contains the compiled CSECTS.
SYSUT1	Defines the data set containing the statement factoring input and is used by the assembler and linkage editor for work data sets.
SYSUT2	Defines the data set containing the qualification mask records generated by statement factoring for the translator section. Also used by the assembler for a work data set.
SYSUT3	Defines the data set containing translator generated linkage editor control cards. Also used by the assembler as a work data set.
STEREO	Contains the final stereotype records from Edit Pass 2 used as input to the translator.
STROUTDD	Defines the temporary data set used by the input editor phase only and contains intermediate stereotype records.
ASSEMIN	Defines the assembler input data set generated by the translator.
RSIPDS	Defines a partitioned data set used throughout RASP as the run communications data set.
NAMETAB	Defines the temporary data set containing a list of field/group names as detected by the RASP edit. Resolved by Edit Pass 2 against the FFT.
STAT	Defines the temporary data set used by the statistics capability.

JOB PREPARATION

SYSPRINT	Used to print any user data from the RASP component.
SYSOUT	Defines a printer output data set containing S/360 sort messages.
SYSUDUMP	ABEND dump printer output.
XINDEX XINDEX1 XINDEX2	These define the Index Data Sets to be used.
PRINTER	This defines diagnostic messages from Index Processing.
SYSDUMMY	Required to negate the printout from the linkage editor.
SOURCPRT	Defines a printer output data set used for source listings.
SOURCLIB	Defines a library used to store source material.
SYSIN	Defines the input source statements for RASP. This statement must be supplied by the user as //RASP.SYSIN DD *.

JOB PREPARATION

Procedure: XOP, XOPSD, XOPEX, XOPSDEX (Output Processor)

STEPCAT	Defines the NIPS user catalog for VSAM processing.
STEPLIB	Defines the NIPS system library.
DATAFILE DATAFIL1 DATAFIL2	Defines the indexed sequential data files processed by OP.
SAMFILE SAMFILE1 SAMFILE2	Defines the sequential data files processed by OP.
DLIB TLIB	Permanent RITs will be stored on one of these data sets.
SYSPRINT SYSOUT	Defines printer output data sets.
SYSUDUMP	ABEND dump printer output.
SYSUT1 SYSUT2 SYSUT3	Defines temporary work data sets.
SYSPUNCH	Defines a temporary data set containing the object modules resulting from the assembly.
SYSLIB	Defines a data set containing the Gen Code macros used to generate a RIT.
SYSLMOD	Defines an output library for temporary data sets.
SYSTPRT	Defines the printer output data set for the assembler and linkage editor listings.
OPIWCF	Defines a temporary data set containing OP supervisor control cards. Created by OPBEGIN and used by OPCTLPRC.
OPSTST	Defines a temporary data set containing RIT specification decks. Created by OPBEGIN and used by OPTAG.

JOB PREPARATION

OPCREATE	Defines a temporary data set containing the RIT create cards and pointers to the corresponding specification deck. Created by OPBEGIN and used by OPTAG.
OPCOMREC	Defines a temporary data set containing communications for OPTAG. Created by OPBEGIN.
STRUCT	Same as OPSTST.
INSTS	Defines a temporary data set containing the macro calls and instructions comprising a RIT. Created by OPTAG and used by the assembler.
LCF	Defines a temporary data set containing the linkedit control cards for permanent RIT.
LCT	Same as LCF but for temporary RITs.
LITSTR	Defines a temporary data set containing macro definitions generated by OPTAG defining the communications CSECTs of the RITs that have been structured.
NAMDEF	Defines a temporary data set containing a name-definition string defining the attributes, in coded form, of field/group names.
LITAB	Defines a temporary data set containing a table of literals defined in a RIT. Created and used by OPTAG.
RGTAB	Defines a temporary data set containing a table in internal format of the functions of a RIT. Created and used by OPTAG.
CGC	Defines a temporary data set containing a table, by RIT, of CSECT ID used for constructing the link-edit control files.
ERRTAB	Defines a temporary data set containing a table of error codes recognized in the editing of the RIT specification decks.

JOB PREPARATION

DCTNY	Defines a temporary data set for the storage of a dictionary of data fields and literal names in source form.
SUBTAB	Defines a temporary data set created and used by OPTAG for storage of a table of conversion subroutines and tables used by the RIT.
OPLGOGO	Defines a temporary data set containing the preliminary execution table. Created and used by OPCTLPRC.
OPLEXEX	Defines a temporary data set containing the execution table. Created by OPCTLPRC and used by OPPROPER.
OPLEXDIR	Defines a temporary data set containing a directory into the execution table. Created by OPCTLPRC and used by OPPROPER.
OPLINE	Defines a printer output data set for the printed output of a RIT.
OPPUNCH	Defines a punch output data set for the punched output of a RIT.
OPRECORD	Defines a data set for RIT record output.
SLIB	Defines libraries of load modules.
SYSAIN	Defines a concatenation of the data sets defined by the LITSTR and INSTS DD statements.
SYSLIN	Defines the link-edit control file for permanent RITs. (LCF)
SYSTIN	Defines the link-edit control file for temporary RITs (LCT)
LOADR	Defines a temporary data set containing object modules to be link-edited.

JOB PREPARATION

QDFILE	Defines a temporary data set containing those data records which were qualified by RASP.
QRTFILE	Defines the RASP-generated data set containing sorted pointers into the QDFILE.
STAT	Defines the temporary data set used by the statistics capability.
SOURCPRT	Defines a printer output data set used for source listings.
SOURCLIB	Defines a library used to store source material.

JOB PREPARATION

Procedure: XTABGEN (Table Generator)

STEPLIB	Defines the NIPS system library.
SYSUDUMP	ABEND printer output.
SYSPRINT	Defines a printer output data set.
SORTLIB	Defines the S/360 Operating System's sort library.
SORTWK01 through SORTWK06	Defines the S/360 Sort disk work data sets.
SYSOUT	Defines a printer output data set for the S/360 SORT messages.
SYSLMOD	Defines a partitioned data set where the table is to be stored by the linkage editor.
SYSIN	Defines the TABGEN source input statements. This statement must be supplied by the user as //TAB.SYSIN DD *.

JOB PREPARATION

Procedure: XSUBLDR (Subroutine Loader)

STEPLIB	Defines the NIPS system library.
SYSABEND	ABEND dump printer output.
SYSPRINT	Defines printer output data set used to print subroutine loader output.
ASSEMIN	Defines the assembler input data set built by the subroutine loader.
LNKEDIN	Defines the linkage editor input data set containing control cards for that processor.
SYSOUT	Defines a printer data set.
SYSUT1	Defines a work data set used by the assembler and linkage editor.
SYSUT2 SYSUT3	Defines work data sets used by the assembler.
ASMBL.SYSLIB	Defines the macro library containing the macro QUTSUBR.
SYSPUNCH	Defines the data set containing the assembler output CSECTs.
ASMBL.SYSIN	Defines the same data set as ASSEMIN.
SYSLMOD	Defines the partitioned data set onto which the user subroutine is to be loaded.
DPUNCH	Defines the same data set as LNKEDIN.
SYSLIN	Defines the same data set as SYSPUNCH.
LKEDIT.SYSLIB	Defines the partitioned data set containing any load modules called internally by the user subroutine.
MODLIB	Defines the partitioned data set containing the user subroutine input load module.

JOB PREPARATION

SYSIN	Defines the input source deck and must be supplied by the user as //SUB.SYSIN DD *.

Procedure: XSTOIS (File Load Utility)

STEPCAT	Defines the NIPS user catalog for VSAM processing.
STEPLIB	Defines the NIPS system library.
VSMFILE	Defines the VSAM data file to be loaded. This data file must have been previously defined by IDCAMS.
DATAFILE	Defines the indexed sequential data set to be created.
SAMFILE	Defines the sequential data set used to create the ISAM data set.
SYSPRINT	Defines a printer output data set.
SYSUDUMP	ABEND dump printer output.

Procedure: XISTOS (File Unload Utility)

STEPCAT	Defines the NIPS user catalog for VSAM processing.
STEPLIB	Defines the NIPS system library.
VSMFILE	Defines the VSAM data set to be unloaded.
DATAFILE	Defines the indexed sequential data set to be unloaded.
SAMFILE	Defines the sequential data set to be copied.
SYSPRINT	Defines a printer output data set.
SYSUDUMP	ABEND dump printer output.
SAMOUT	Defines the sequential data set to be created.

JOB PREPARATION

Procedure: X1410CON (360 to 1410 Data Conversion)

STEPLIB	Defines the NIPS system library.
SYSPRINT SYSOUT	Defines printer output data sets.
SYSUDUMP	ABEND dump printer output.
DATAFILE	Defines the NIPS 360 FFS ISAM data set to be converted to a 1410 data file.
NEWFILE	Defines the sequential 1410 data file generated by this utility.
SYSIN	Defines the 1410 FFT object deck and must be supplied by the user as //GO.SYSIN DD *.

Procedure: X360CON (1410 to 360 Data Conversion)

SYSPRINT SYSOUT	Defines printer output data sets.
SYSUDUMP	ABEND dump printer output.
DATAFILE	Defines the NIPS 360 FFS ISAM FFT.
FILE1410	Defines the 1410 data file.
NEWFILE	Defines the NIPS 360 FFS sequential data set generated by this utility.
SYSIN	Defines the 1410 FFT object deck and must be supplied by the user as //GO.SYSIN DD *.

JOB PREPARATION

Procedure: XQRTQDF

STEPLIB	Defines the NIPS system library.
SYSPRINT	Defines a printer output data set.
SYSUDUMP	ABEND dump printer output.
QRTFILE	Defines a data set containing sorted pointers into the QDFILE created by RASP.
QDFILE	Defines a data set containing those data records qualified by RASP.
SAMFILE	Defines a sequential data set which has all the properties of a S/360 SAM data set.
SYSIN	Defines the QRTQDF source input statements. This statement must be supplied by the user as //QRTQDF.SYSIN DD *.

JOB PREPARATION

Procedure: XQUIP, XQUIPSD (Quick Inquiry Processor)

STEPCAT	Defines the NIPS user catalog for VSAM processing.
STEPLIB	Defines the NIPS system library.
SYSUT1 SYSUT2 SYSUT3 SYSUT4	Defines temporary work data sets used for structuring the query, and for Index Processing.
SLIB	Defines user and system libraries.
DATAFILE	Defines the indexed sequential or VSAM data set accessed by QUIP in source direct mode.
DATAFIL1 DATAFIL2	Defines the indexed sequential data sets accessed by QUIP as secondary files in Interfile Output.
FFT FFT1 FFT2	Defines the FFT data sets which are associated with non-NIPS files being accessed by QUIP.
SAMFILE	Defines the sequential data set accessed by QUIP in source direct mode.
SYSPRINT	Defines a printer output data set.
PB	Defines a temporary data set containing those data records qualified by QUIP in source direct mode.
KEY	Defines a temporary data set containing the sort keys generated by QUIP for those data records which qualified in source direct mode.
SORTWK01 through SORTWK04	Defines work areas used by the internal QUIP sort when in source direct mode.
SYSUDUMP	ABEND dump printer output.
SOURCPRT	Defines a printer output data set used for source listings.

JOB PREPARATION

SOURCLIB	Defines a library used to store source material.
SYSIN	Defines the QUIP source statement input. This statement must be supplied by the user when QUIP is run in the batch partition. It is coded as //QUIP.SYSIN DD *.
QDFILE	Defines the data set containing the data records qualified by RASP.
QRTFILE	Defines the RASP-generated data set containing sorted pointers (QRT) into the QDFILE.
STAT	Defines the temporary data set used by the statistics capability.
XINDEX	This defines the Index Data Set.

JOB PREPARATION

Procedure: XDMPLIB

STEPCAT	Defines the NIPS user catalog for VSAM processing.
STEPLIB	Defines the NIPS system library.
SYSPRINT	Defines a printer output data set.
SYSUDUMP	ABEND dump printer output.
DATAFILE	Defines an indexed sequential data set.
ISAMWORK	Defines a temporary ISAM data set containing the FFT and logic statement library during sequential file processing.
SAMFILE	Defines the sequential data set to be processed by UTDMPLIB.
SYSIN	Defines the input source control card for UTDMPLIB. This statement must be supplied by the user as //UTDMP.SYSIN DD *.

JOB PREPARATION

Procedure: XSUBCHK (Subroutine Check)

STEPLIB	Defines the NIPS system library.
SYSUDUMP	ABEND dump printer output.
SYSPRINT	Defines a printer output data set.
SYSOUT	Defines a printer output data set.
SLIB	Defines user and system libraries.
SYSIN	Defines the input source deck and must be supplied by the user as //SUBCHK.SYSIN DD *.

JOB PREPARATION

Procedure: XCLASS (File Classification Change)

STEPCAT	Defines the NIPS user catalog for VSAM processing.
STEPLIB	Defines the NIPS system library.
SYSPRINT	Defines a printer output data set.
SYSUDUMP	ABEND dump printer output.
DATAFILE	Defines the ISAM data file to be processed.
SAMFILE	Defines the sequential data set to be processed. This will be both the input and output data set for sequential processing on disk.
UTSAMOUT	Defines the output sequential data file that is produced when tape input is used.
SYSIN	Defines the CLASS source input card. This statement must be supplied by the user as //CLASS.SYSIN DD *.

JOB PREPARATION

Procedure: XUTFSCAN (Field Scan)

STEPCAT	Defines the NIPS user catalog for VSAM processing.
SYSPRINT	Defines a printer output data set.
SYSUDUMP	ABEND dump printer output.
DATAFILE	Defines the ISAM data file.
SAMFILE	Defines the SAM data file.
ISAMWORK	Defines temporary ISAM data set if a SAM data file is input.
SYSUT2	Defines partitioned data set containing members to be scanned.
TRANST	Defines the data set for output transactions.
SYSIN	Defines input stream. This card must be supplied by the user as //UTF.SYSIN DD *.
SOURCPRT	Defines a printer output data set.
SOURCLIB	Defines a library used to store source material.
SYSIN	Defines the input to the UTSOURC utility. This is not overridden if a LIST operation is desired. This statement is overridden if library update is to be performed.

JOB PREPARATION

Procedure: XSP (Index Specification)

STEPCAT	Defines the NIPS user catalog for VSAM processing.
STEPLIB	Defines the NIPS system library.
NEWFILE	Defines the indexed sequential data file created during an FM run.
XINDEX	Defines the Index Data Set.
SORTWK01 through SORTWK06	Defines the S/360 Operating System's Sort work data sets.
SORTLIB	Defines the S/360 Operating System's Sort Library.
SYSOUT	Defines a printer output data set for Sort messages.
SYSPRINT	Defines a printer output data set.
SOURCPRT	Defines a printer output data set for Index Specification messages.
SYSUDUMP	ABEND dump printer output.
SLIB	Defines the user library containing subroutines/ tables.
SAMFILE	Defines a sequential data file
UTSAMOUT	Defines the updated sequential data file.
ISAMWORK	Defines the temporary ISAM data set containing the FFT during sequential file processing in Index Specification.

JOB PREPARATION

<u>Procedure</u>: XTRDISK (Index Transfer - Unload)

STEPCAT	Defines the NIPS user catalog for VSAM processing.
STEPLIB	Defines the NIPS system library.
SYSPRINT	Defines the printed output of XTRDISK. Contains any diagnostic messages.
SYSUDUMP	ABEND dump printer output.
INDEXSAM	Defines the output sequential Index Data Set.
INDEXDAM	Defines the input, direct access, disk-resident Index Data Set.
DATAFILE	Defines the ISAM data file corresponding to the index data set.
SAMFILE	Defines the sequential access data file corresponding to the index data set.

JOB PREPARATION

Procedure: XTRTAPE (Index Transfer - Load)

STEPLIB	Defines the NIPS system library.
SYSPRINT	Defines the printed output of XTRTAPE. Contains any diagnostic messages.
SYSUDUMP	ABEND dump printer output.
INDEXSAM	Defines the input, sequential version of an Index Data Set.
XINDEX	Defines the output, disk-resident, direct access Index Data Set.

JOB PREPARATION

<u>Procedure</u>:	XKA (Keyword Analysis)
STEPCAT	Defines the NIPS user catalog for VSAM processing.
STEPLIB	Defines the NIPS system library.
DATAFILE	Defines the indexed sequential data file to be analyzed.
SAMFILE	Defines the sequential data files to be analyzed.
SLIB	Defines the user library containing subroutine and/or tables.
SYSPRINT	Defines a printer output data set.
SOURCLIB	Defines a library used to store source material.
SOURCPRT	Defines a printer output data set used for source listings.
SYSUDUMP	Defines a printer output data set for ABEND dump output.
SYSOUT	Defines a printer output data set containing S/360 sort messages.
SORTLIB	Defines the S/360 Operating System sort library.
SORTWK01 thru SORTWK04	Defines the work data sets used by S/360 sort.
KANWK01	Defines a temporary work data set.
KMDPRT	Defines a printer output data set.
SYSIN	Defines the input user control statement data set for UTNDXKAN. This statement must be supplied by the user as //XKA.SYSIN DD *.

JOB PREPARATION

Procedure:	XUTODE
STEPLIB	Defines the NIPS system library.
SYSUDUMP	Defines a printer output data set for ABEND.
SNAPSHOT	Defines a printer output data set for debug.
SOURCPRT	Defines a printer output data set for source listings.
ODDPRT	Defines a printer output data set for diagnostics.
SYSPRINT	Defines a printer output data set.
SLIB	Defines the library where the user compiled format will be stored.
SOURCLIB	Defines a library used to store source material.

JOB PREPARATION

Procedure:	UTNDXKMD
STEPLIB	Defines the NIPS System Library.
SLIB	Defines the user library containing tables.
KMDWK01 thru KMDWK04	Defines UTNDXKMD work data sets.
KMDPRT	Defines message and display device, UTNDXKMD.
SYSOUT	Sort message output device.
SYSPRINT	Printer output device.
SORTLIB	S/360 OS Sort Library.
SORTWK01 thru SORTWK06	Sort work data sets.
SYSUDUMP	ABEND dump device.
SYSIN	SYSIN device.

JOB PREPARATION

Appendix F

PROCEDURE LISTINGS

```
//XCLASS        PROC  A=A,BSZFILE=,BSZNEWF=,                          00000100
//              CL=',',CL1=',',DEN=,ISAM='DUMMY.FILE',                00000200
//              JOBLIB='FFS.JOBLIB',LAB=SL,NDISP=KEEP,RGN=60K,        00000300
//              PTFJOBL='PTF.JOBLIB',                                 00000400
//              SAM='DUMMY.FILE',SAMOUT='DUMMY,',TRCH=,               00000500
//              UISAM=2314,USAM='(TAPE9,,DEFER)',                     00000600
//              VISAM=,VSAM=,VSMOUT=                                  00000700
//*                                                                   00000800
//*  CHARLES W. HICKISCH  MAJOR,USA  PROJ CODE=763NIPS  BRANCH=431   00000900
//*   DATE=MARCH 1,1974                                               00001000
//*                                                                   00001100
//CLASS        EXEC  PGM=UTCLASS,REGION=&RGN                          00001200
//STEPLIB DD DSN=&PTFJOBL,DISP=SHR                                    00001300
//        DD DSN=&JOBLIB,DISP=SHR                                     00001400
//SYSPRINT DD SYSOUT=(&A,&CL)                                         00001500
//SYSUDUMP DD SYSOUT=(&A,&CL1)                                        00001600
//DATAFILE DD DSN=&ISAM,UNIT=&UISAM,VOL=&VISAM,DISP=SHR               00001700
//SAMFILE DD    DSN=&SAM.S,UNIT=&USAM,VOL=&VSAM,DISP=(SHR,KEEP),      00001800
//              DCB=(RECFM=VB,LRECL=1000,BLKSIZE=&BSZFILE,TRTCH=&TRCH,00001900
//              DEN=&DEN),LABEL=(,&LAB)                               00002000
//UTSAMOUT DD   &SAMOUT.DCB=(RECFM=VB,LRECL=1000,BLKSIZE=&BSZNEWF,    00002100
//              TRTCH=&TRCH,DEN=&DEN),DISP=(,&NDISP),LABEL=(,&LAB),   00002200
//              UNIT=&USAM,VOL=&VSMOUT                                00002300
```

NIPS 360 FFS PROCEDURES

```
//XDMPLIB       PROC  A=A,BSZFILE=,                                      00000100
//              CL=',',CL1=',',DEN=,ISAM='DUMMY.FILE',                   00000200
//              JOBLIB='FFS.JOBLIB',LAB=SL,RGN=60K,                      00000300
//              PTFJOBL='PTF.JOBLIB',                                    00000400
//              SAM='DUMMY.FILE',STG=NIPW,TRCH=,                         00000500
//              UISAM='(2314,P)',USAM='(TAPE9,,DEFER)',                  00000600
//              VISAM=,VSAM=                                             00000700
//*                                                                      00000800
//*  CHARLES W. HICKISCH  MAJOR,USA  PROJ CODE=763NIPS  BRANCH 431      00000900
//*   DATE=MARCH 1,1974                                                  00001000
//*                                                                      00001100
//UTDMP     EXEC  PGM=UTDMPLIB,REGION=&RGN                               00001200
//STEPLIB DD DSN=&PTFJOBL,DISP=SHR                                       00001300
//        DD DSN=&JOBLIB,DISP=SHR                                        00001400
//SYSPRINT DD SYSOUT=(&A,&CL)                                            00001500
//SYSUDUMP DD SYSOUT=(&A,&CL1)                                           00001600
//DATAFILE  DD  DSNAME=&ISAM,UNIT=&UISAM,VOLUME=&VISAM,DISP=SHR          00001700
//ISAMWORK  DD  UNIT=&STG,DCB=DSORG=IS,SPACE=(CYL,(10))                  00001800
//SAMFILE   DD  DSNAME=&SAM.S,UNIT=&USAM,VOLUME=&VSAM,DISP=(SHR,KEEP),  C00001900
//              LABEL=(,&LAB),DCB=(RECFM=VB,LRECL=1000,BLKSIZE=&BSZFILE, 00002000
//              TRTCH=&TRCH,DEN=&DEN)                                    00002100
```

```
//XFM  PROC  A=A,AUXBUFN=2,AUXSP=6,BLKSIZE=560,B=B,                    00000100
//             VSDSN='DUMMY.FILE',NEWVSM='DUMMY.FILE',                 00000300
//             BSZFILE=,BSZNEWF=,                                     X00000500
//             CL=',',CL1=',',CL2=',',CYLOFL=1,DEN=,                   00000600
//             EROPT=ABE,GEN='DUMMY,',                                 00000700
//             INDEX=1,ISAM='DUMMY.FILE',                              00000800
//             INXSP=1,JOBLIB='FFS.JOBLIB',JOBMAC='FFS.JOBMACRO',      00000900
//             LAB=SL,LIB='DUMMY.FILE',LIB1='DUMMY.FILE',             X00001000
//             MCTSP=5,NBRBLK=200,OVFLOW=1,PRIME=5,RGN=98K,            00001100
//             PTFJOBL='PTF.JOBLIB',PTFJOBM='PTF.JOBMACRO',            00001200
//             SAM='DUMMY.FILE',SAMOUT='DUMMY,',SDISP=SHR,SORTSP=10,   00001300
//             SOURCL='DUMMY.FILE',STG=NIPW,TDISP=MOD,TRANS='&&TRANS', 00001400
//             TRANSP=200,TRBUFNO=4,TRCH=,UISAM='(2314,P)',            00001500
//             ULIB=2314,ULIB1=2314,USAM='(TAPE9,,DEFER)',             00001600
//             USOURCL=2314,UTRANS=NIPW,VINDEX='REF=*.DATAFILE',       00001700
//             VISAM=,VLIB=,VLIB1=,VOVFLOW='REF=*.DATAFILE',           00001800
//             VPRIME='REF=*.DATAFILE',VSAM=,VSMMCT=,VSMOUT=,VSOURCL=, 00001900
//             VTRANS=,XDISP=SHR,XINDEX='DUMMY.FILE',                  00002000
//             XUNIT=2314,XVOL=                                        00002100
//*                                                                    00002200
//*  CHARLES W. HICKISCH  MAJOR,USA  PROJ CODE=763NIPS  BRANCH=431    00002300
//*   DATE=MARCH 1, 1974                                               00002400
//*                                                                    00002500
//FM     EXEC  PGM=FM,REGION=&RGN                                      00002600
//STEPLIB DD DSN=&PTFJOBL,DISP=SHR                                     00002800
//       DD DSN=&JOBLIB,DISP=SHR                                       00002900
//FMSTAT   DD SYSOUT=(&A,&CL)                                          00003000
//SYSOUT DD SYSOUT=(&A,&CL)                                            00003100
//SYSLIST DD SYSOUT=(&A,&CL)                                           00003200
//SOURCPRT DD SYSOUT=(&A,&CL)                                          00003300
//SYSUDUMP DD SYSOUT=(&A,&CL1)                                         00003400
//SYSPRINT DD SYSOUT=(&A,&CL2)                                         00003500
//VSMFILE  DD  DSN=&VSDSN,DISP=SHR                                     00003600
//NEWVSM   DD  DSN=&NEWVSM,DISP=SHR                                    00003700
//DATAFILE  DD DSNAME=&ISAM,UNIT=&UISAM,VOLUME=&VISAM,DISP=SHR         00003800
//NEWFILE DD &GEN.DCB=(DSORG=IS,CYLOFL=&CYLOFL,BLKSIZE=&BSZNEWF),     X00003900
//             DISP=(,KEEP,DELETE),SPACE=(CYL,&INDEX),                X00004000
//             DSNAME=&ISAM.A(INDEX),UNIT=&UISAM,VOLUME=&VINDEX        00004100
//       DD    &GEN.DCB=*.NEWFILE,SPACE=(CYL,&PRIME),                 X00004200
//             DSNAME=&ISAM.A(PRIME),UNIT=&UISAM,VOLUME=&VPRIME,      X00004300
//             DISP=(,KEEP,DELETE)                                     00004400
//       DD    &GEN.DCB=*.NEWFILE,SPACE=(CYL,&OVFLOW),                X00004500
//             DSNAME=&ISAM.A(OVFLOW),UNIT=&UISAM,VOLUME=&VOVFLOW,    X00004600
//             DISP=(,KEEP,DELETE)                                     00004700
//FMCOMM    DD UNIT=&STG,SPACE=(TRK,(1,1))                             00004800
//FMFLUD    DD UNIT=&STG,SPACE=(CYL,(&MCTSP,5))                        00004900
//FMTRANS  DD UNIT=&STG,SPACE=(CYL,(&AUXSP,5)),DCB=BUFNO=&TRBUFNO      00005000
//FMLABELS  DD UNIT=&STG,SPACE=(CYL,(1,1))                             00005100
//FMCM      DD UNIT=&STG,SPACE=(TRK,(20,5))                            00005200
//FMCMFILE  DD UNIT=&STG,SPACE=(TRK,(20,5))                            00005300
//FMLITFIL  DD UNIT=&STG,SPACE=(TRK,(20,5))                            00005400
//SYSLIB DD DSN=&PTFJOBM,DISP=SHR                                      00005500
//       DD DSN=&JOBMAC,DISP=SHR                                       00005600
//          DD DSNAME=SYS1.MACLIB,DISP=SHR                             00005700
//FMLEIN    DD UNIT=&STG,SPACE=(TRK,(25,5))                            00005800
//SYSLMOD   DD UNIT=&STG,SPACE=(CYL,(4,1,10)),DCB=SYS1.LINKLIB,       X00005900
//             LABEL=EXPDT=66366                                       00006000
```

```
//TEMLSREC  DD UNIT=&STG,SPACE=(TRK,(20,5))                                00006100
//SORTLIB DD DSNAME=SYS1.SORTLIB,DISP=SHR                                  00006200
//SORTIN    DD UNIT=&STG,SPACE=(TRK,(&TRANSP,150)),DCB=BUFNO=&TRBUFNO      00006300
//SORTWK01 DD  SPACE=(CYL,(&SORTSP),,CONTIG),UNIT=(&STG,SEP=SORTIN)        00006400
//SORTWK02 DD  SPACE=(CYL,(&SORTSP),,CONTIG),                             X00006500
//             UNIT=(&STG,SEP=SORTWK01),SEP=SORTWK01                       00006600
//SORTWK03 DD  SPACE=(CYL,(&SORTSP),,CONTIG),                             X00006700
//             UNIT=(&STG,SEP=SORTWK02),SEP=SORTWK02                       00006800
//SORTWK04 DD  SPACE=(CYL,(&SORTSP),,CONTIG),                             X00006900
//             UNIT=(&STG,SEP=(SORTWK01,SORTWK03)),                       X00007000
//             SEP=(SORTWK01,SORTWK03)                                     00007100
//SORTWK05 DD  SPACE=(CYL,(&SORTSP),,CONTIG),                             X00007200
//             UNIT=(&STG,SEP=(SORTWK02,SORTWK04)),                       X00007300
//             SEP=(SORTWK02,SORTWK04)                                     00007400
//SORTWK06 DD  SPACE=(CYL,(&SORTSP),,CONTIG),                             X00007500
//             UNIT=(&STG,SEP=(SORTWK01,SORTWK03,SORTWK05)),              X00007600
//             SEP=(SORTWK01,SORTWK03,SORTWK05)                            00007700
//SORTOUT   DD DISP=(OLD,PASS),DSNAME=*.SORTIN,VOLUME=REF=*.SORTIN,       *00007800
//             DCB=(LRECL=1500,BLKSIZE=1504,RECFM=VB,BUFNO=2)              00007900
//FMLEFILE  DD DSNAME=*.SORTWK04,VOLUME=REF=*.SORTWK04,DISP=(OLD,PASS)    00008000
//FMMACRO   DD DSNAME=*.SORTWK05,VOLUME=REF=*.SORTWK05,DISP=(OLD,PASS)    00008100
//FMOUTMAC  DD DSNAME=*.SORTWK06,VOLUME=REF=*.SORTWK06,DISP=(OLD,PASS)    00008200
//TAPEIN    DD DSNAME=*.SORTIN,VOLUME=REF=*.SORTIN,DISP=(OLD,PASS)        00008300
//TAPEOUT   DD DSNAME=*.SORTIN,VOLUME=REF=*.SORTIN,DISP=(OLD,PASS),       *00008400
//             DCB=(LRECL=1500,BLKSIZE=1504,RECFM=VB)                      00008500
//RECSIN DD DSNAME=*.FMFLUD,VOLUME=REF=*.FMFLUD,DISP=(OLD,PASS)           00008600
//RECSOUT   DD DSNAME=*.RECSIN,VOLUME=REF=*.RECSIN,DISP=(OLD,PASS),       X00008700
//             DCB=(LRECL=1000,BLKSIZE=1004,RECFM=VB)                      00008800
//RECSWK01 DD DSNAME=*.SORTWK01,VOLUME=REF=*.SORTWK01,DISP=(OLD,PASS)     00008900
//RECSWK02 DD DSNAME=*.SORTWK02,VOLUME=REF=*.SORTWK02,DISP=(OLD,PASS)     00009000
//RECSWK03 DD DSNAME=*.SORTWK03,VOLUME=REF=*.SORTWK03,DISP=(OLD,PASS)     00009100
//RECSWK04 DD DSNAME=*.SORTWK04,VOLUME=REF=*.SORTWK04,DISP=(OLD,PASS)     00009200
//RECSWK05 DD DSNAME=*.SORTWK05,VOLUME=REF=*.SORTWK05,DISP=(OLD,PASS)     00009300
//RECSWK06 DD DSNAME=*.SORTWK06,VOLUME=REF=*.SORTWK06,DISP=(OLD,PASS)     00009400
//TRANS     DD DUMMY,DISP=OLD,DCB=(EROPT=&EROPT,BUFNO=&TRBUFNO)           00009500
//FMSETTBL DD UNIT=&STG,SPACE=(TRK,(1,1))                                 00009600
//FMAUXOP   DD  DSN=*.FMTRANS,VOLUME=REF=*.FMTRANS,DISP=(OLD,PASS),       00009700
//      DCB=(LRECL=1000,BLKSIZE=1004,BUFNO=&AUXBUFN)                      00009800
//PUNCH     DD SYSOUT=&B,DCB=BUFNO=&AUXBUFN                               00009900
//AUX1 DD DUMMY,DCB=(RECFM=VB,BLKSIZE=1004,LRECL=1000,BUFNO=&AUXBUFN)     00010000
//AUX2 DD DUMMY,DCB=(RECFM=VB,BLKSIZE=1004,LRECL=1000,BUFNO=&AUXBUFN)     00010100
//AUX3 DD DUMMY,DCB=(RECFM=VB,BLKSIZE=1004,LRECL=1000,BUFNO=&AUXBUFN)     00010200
//AUX4 DD DUMMY,DCB=(RECFM=VB,BLKSIZE=1004,LRECL=1000,BUFNO=&AUXBUFN)     00010300
//AUX5 DD DUMMY,DCB=(RECFM=VB,BLKSIZE=1004,LRECL=1000,BUFNO=&AUXBUFN)     00010400
//FMTAUX    DD UNIT=&STG,SPACE=(TRK,(10,5)),DCB=BUFNO=&AUXBUFN            00010500
//SLIB      DD DSNAME=*.SYSLMOD,DISP=(OLD,PASS),VOLUME=REF=*.SYSLMOD      00010600
//  DD  DSNAME=&LIB.L,UNIT=&ULIB,VOLUME=&VLIB,DISP=SHR                    00010700
//  DD  DSNAME=&LIB1.L,UNIT=&ULIB1,VOLUME=&VLIB1,DISP=SHR                 00010800
//          DD DSNAME=&JOBLIB,DISP=SHR                                    00010900
//SOURCLIB  DD  DISP=&SDISP,DSN=&SOURCL.L,VCL=&VSOURCL,UNIT=&USOURCL      00011000
//OMMACS  DD  UNIT=&STG,SPACE=(CYL,(5,1))                                 00011100
//GENFILE   DD  UNIT=&STG,SPACE=(CYL,(3,1))                               00011200
//LITFILE   DD  UNIT=&STG,SPACE=(CYL,(3,1))                               00011300
//EDITFILE  DD  UNIT=&STG,SPACE=(CYL,(3,1))                               00011400
//FMNFLUD   DD  UNIT=&STG,SPACE=(CYL,(5,1))                               00011500
//FMTEFILE  DD  UNIT=&STG,SPACE=(TRK,(5,1))                               00011600
//FMAEFILE  DD  UNIT=&STG,SPACE=(TRK,(5,1))                               00011700
```

```
//FMSODANQ  DD  DISP=SHR,DSNAME=&ISAM.D,VOLUME=REF=*.STEPLIB                00011800
//NFLERR  DD  DSNAME=*.SORTWK04,VOLUME=REF=*.SORTWK04,DISP=(MOD,PASS)      00011900
//NFLMACS  DD  DSNAME=*.SORTWK03,VOLUME=REF=*.SORTWK03,DISP=(OLD,PASS)     00012000
//ASSEMIN  DD  DSNAME=*.FMLITFIL,VOLUME=REF=*.FMLITFIL,DISP=(OLD,PASS)     00012100
//         DD  DSNAME=*.SORTWK06,VOLUME=REF=*.SORTWK06,DISP=(OLD,PASS)     00012200
//ISAMWORK  DD  UNIT=&STG,DCB=DSORG=IS,SPACE=(CYL,(10))                    00012300
//SAMFILE DD DSNAME=&SAM.S,UNIT=&USAM,VOLUME=&VSAM,DISP=(SHR,KEEP),       X00012400
//              LABEL=(,&LAB),DCB=(RECFM=VB,LRECL=1000,BLKSIZE=&BSZFILE,X00012500
//              TRTCH=&TRCH,DEN=&DEN)                                      00012600
//FMSAMOUT DD  &SAMOUT.DCB=(RECFM=VB,LRECL=1000,BLKSIZE=&BSZNEWF,         X00012700
//              DEN=&DEN,TRTCH=&TRCH),DISP=(NEW,KEEP),                     00012800
//              LABEL=(,&LAB),                                             00012900
//              UNIT=&USAM,VOLUME=(PRIVATE,RETAIN,&VSMOUT)                 00013000
//FMNDATA  DD  &SAMOUT.DCB=*.FMSAMOUT,DISP=(,KEEP),LABEL=(,&LAB),         X00013100
//              UNIT=&USAM,VOLUME=(PRIVATE,RETAIN,&VSMMCT)                 00013200
//STAT  DD  DSN=*.FMCM,VOL=REF=*.FMCM,DISP=(OLD,PASS)                      00013300
//TRANST  DD  DSN=&TRANS,VOL=&VTRANS,UNIT=&UTRANS,DISP=&TDISP,             00013400
//             SPACE=(TRK,1)                                               00013500
//INDEXPRT  DD  SYSOUT=(&A,&CL)                                            00013600
//IXTRANS DD DISP=(NEW,PASS),UNIT=&STG,SPACE=(CYL,(&INXSP,1))              00013700
//XINDEX DD DSN=&XINDEX.X,DISP=(&XDISP,KEEP),UNIT=&XUNIT,VOL=&XVOL,        00013800
//              DCB=(BLKSIZE=&BLKSIZE,RECFM=F,KEYLEN=4,DSORG=DA),          X00013900
//              SPACE=(&BLKSIZE,&NBRBLK)                                   00014000
```

```
//XFMEX  PROC  A=A,AUXBUFN=2,AUXSP=6,B=B,BLKSIZE=560,                  00000100
//             BSZFILE=,BSZNEWF=,                                     X00000200
//             CHKDSP=DELETE,CHKID='&&CHKID',CHKSP=0,                  00000300
//             CHKST=NEW,CL=',',CL1=',',CL2=',',DEN=,EROPT=ABE,        00000400
//             INXSP=1,ISAM='DUMMY.FILE',JOBLIB='FFS.JOBLIB',          00000500
//             LAB=SL,LIB='DUMMY.FILE',LIB1='DUMMY.FILE',             X00000600
//             MCTSP=5,NBRBLK=200,NRMDSP=DELETE,RGN=98K,               00000700
//             PTFJOBL='PTF.JOBLIB',                                   00000710
//             SAM='DUMMY.FILE',SAMOUT='DUMMY,',SORTSP=10,             00000800
//             STG=NIPW,TDISP=MOD,TRANS='&&TRANS',                     00000900
//             TRANSP=200,TRCH=,UCHK=NIPW,UISAM='(2314,P)',            00001000
//             ULIB=2314,ULIB1=2314,USAM='(TAPE9,,DEFER)',             00001100
//             UTRANS=NIPW,VCHK=,VISAM=,VLIB=,VLIB1=,                  00001200
//             VSAM=,VSMMCT=,VSMOUT=,VTRANS=,                          00001300
//             XDISP=SHR,XINDEX='DUMMY.FILE',XUNIT=2314,XVOL=          00001400
//*                                                                    00001500
//*  CHARLES W. HICKISCH  MAJOR,USA  PROJ CODE=763NIPS  BRANCH=431    00001600
//*   DATE=MARCH 1, 1974                                               00001700
//*                                                                    00001800
//FM     EXEC  PGM=FM,REGION=&RGN                                      00001900
//STEPLIB  DD  DSN=&PTFJOBL,DISP=SHR                                   00002000
//         DD  DSN=&JOBLIB,DISP=SHR                                    00002010
//FMSTAT   DD SYSOUT=(&A,&CL)                                          00002100
//SYSOUT DD SYSOUT=(&A,&CL)                                            00002200
//SYSUDUMP DD SYSOUT=(&A,&CL1)                                         00002300
//SYSPRINT DD SYSOUT=(&A,&CL2)                                         00002400
//SOURCPRT DD SYSOUT=(&A,&CL)                                          00002500
//DATAFILE  DD DSNAME=&ISAM,UNIT=&UISAM,VOLUME=&VISAM,DISP=SHR         00002600
//FMCOMM    DD UNIT=&STG,SPACE=(TRK,(1,1)),                           X00002700
//             DISP=(,&NRMDSP,&CHKDSP),DSNAME=&CHKID.A                 00002800
//FMFLUD    DD UNIT=&STG,SPACE=(CYL,(&MCTSP,5)),                      X00002900
//             DISP=(,&NRMDSP,&CHKDSP),DSNAME=&CHKID.B                 00003000
//FMTRANS  DD UNIT=&STG,SPACE=(CYL,(&AUXSP,5)),                        X00003100
//             DISP=(,&NRMDSP,&CHKDSP),DSNAME=&CHKID.C                 00003200
//SYSLMOD DD LABEL=EXPDT=66366,SPACE=(CYL,(4,1,10)),DCB=SYS1.LINKLIB,  X00003300
//             UNIT=&STG,DSNAME=&CHKID.D,DISP=(,&NRMDSP,&CHKDSP)       00003400
//SORTLIB DD DSNAME=SYS1.SORTLIB,DISP=SHR                              00003500
//SORTIN    DD UNIT=&STG,SPACE=(TRK,(&TRANSP,150)),                   X00003600
//             DISP=(,&NRMDSP,&CHKDSP),DSNAME=&CHKID.E                 00003700
//SORTWK01 DD  SPACE=(CYL,(&SORTSP),,CONTIG),UNIT=(&STG,SEP=SORTIN),  X00003800
//             DISP=(,&NRMDSP,&CHKDSP),DSNAME=&CHKID.F                 00003900
//SORTWK02 DD  SPACE=(CYL,(&SORTSP),,CONTIG),                         X00004000
//             DISP=(,&NRMDSP,&CHKDSP),DSNAME=&CHKID.G,                X00004100
//             UNIT=(&STG,SEP=SORTWK01),SEP=SORTWK01                   00004200
//SORTWK03 DD  SPACE=(CYL,(&SORTSP),,CONTIG),                         X00004300
//             DISP=(,&NRMDSP,&CHKDSP),DSNAME=&CHKID.H,                X00004400
//             UNIT=(&STG,SEP=SORTWK02),SEP=SORTWK02                   00004500
//SORTWK04 DD  SPACE=(CYL,(&SORTSP),,CONTIG),                         X00004600
//             DISP=(,&NRMDSP,&CHKDSP),DSNAME=&CHKID.I,                X00004700
//             UNIT=(&STG,SEP=(SORTWK01,SORTWK03)),                    X00004800
//             SEP=(SORTWK01,SORTWK03)                                 00004900
//SORTWK05 DD  SPACE=(CYL,(&SORTSP),,CONTIG),                         X00005000
//             DISP=(,&NRMDSP,&CHKDSP),DSNAME=&CHKID.J,                X00005100
//             UNIT=(&STG,SEP=(SORTWK02,SORTWK04)),                    X00005200
//             SEP=(SORTWK02,SORTWK04)                                 00005300
//SORTWK06 DD SPACE=(CYL,(&SORTSP),,CONTIG),                           00005400
//             DISP=(,&NRMDSP,&CHKDSP),DSNAME=&CHKID.K,                00005500
```

```
//              UNIT=(&STG,SEP=(SORTWK01,SORTWK03,SORTWK05)),                 00005600
//              SEP=(SORTWK01,SORTWK03,SORTWK05)                              00005700
//SORTOUT   DD DISP=(OLD,PASS),DSNAME=*.SORTIN,VOLUME=REF=*.SORTIN,          *00005800
//              DCB=(LRECL=1500,BLKSIZE=1504,RECFM=VB)                        00005900
//RECSIN DD DSNAME=*.FMFLUD,VOLUME=REF=*.FMFLUD,DISP=(OLD,PASS)               00006000
//RECSOUT   DD DSNAME=*.RECSIN,VOLUME=REF=*.RECSIN,DISP=(OLD,PASS),          X00006100
//              DCB=(LRECL=1000,BLKSIZE=1004,RECFM=VB)                        00006200
//RECSWK01 DD DSNAME=*.SORTWK01,VOLUME=REF=*.SORTWK01,DISP=(OLD,PASS)        00006300
//RECSWK02 DD DSNAME=*.SORTWK02,VOLUME=REF=*.SORTWK02,DISP=(OLD,PASS)        00006400
//RECSWK03 DD DSNAME=*.SORTWK03,VOLUME=REF=*.SORTWK03,DISP=(OLD,PASS)        00006500
//RECSWK04 DD DSNAME=*.SORTWK04,VOLUME=REF=*.SORTWK04,DISP=(OLD,PASS)        00006600
//RECSWK05 DD DSNAME=*.SORTWK05,VOLUME=REF=*.SORTWK05,DISP=(OLD,PASS)        00006700
//RECSWK06 DD DSNAME=*.SORTWK06,VOLUME=REF=*.SORTWK06,DISP=(OLD,PASS)        00006800
//TRANS     DD DUMMY,DISP=OLD,DCB=EROPT=&EROPT                                00006900
//FMSETTBL DD UNIT=&STG,SPACE=(TRK,(1,1)),                                   X00007000
//              DISP=(,&NRMDSP,&CHKDSP),DSNAME=&CHKID.L                       00007100
//FMAUXOP DD UNIT=&STG,SPACE=(CYL,(&AUXSP,5)),                               X00007200
//   DCB=(LRECL=1000,BLKSIZE=1004,BUFNO=&AUXBUFN),                            00007300
//              DISP=(,&NRMDSP,&CHKDSP),DSNAME=&CHKID.M                       00007400
//PUNCH      DD SYSOUT=&B,DCB=BUFNO=&AUXBUFN                                 00007500
//AUX1   DD DUMMY,DCB=(RECFM=VB,BLKSIZE=1004,LRECL=1000,BUFNO=&AUXBUFN)      00007600
//AUX2   DD DUMMY,DCB=(RECFM=VB,BLKSIZE=1004,LRECL=1000,BUFNO=&AUXBUFN)      00007700
//AUX3   DD DUMMY,DCB=(RECFM=VB,BLKSIZE=1004,LRECL=1000,BUFNO=&AUXBUFN)      00007800
//AUX4   DD DUMMY,DCB=(RECFM=VB,BLKSIZE=1004,LRECL=1000,BUFNO=&AUXBUFN)      00007900
//AUX5   DD DUMMY,DCB=(RECFM=VB,BLKSIZE=1004,LRECL=1000,BUFNO=&AUXBUFN)      00008000
//FMTAUX    DD UNIT=&STG,SPACE=(TRK,(10,5)),                                 X00008100
//              DCB=BUFNO=&AUXBUFN,                                           00008200
//              DISP=(,&NRMDSP,&CHKDSP),DSNAME=&CHKID.N                       00008300
//SLIB      DD DSNAME=*.SYSLMOD,DISP=(OLD,PASS),VOLUME=REF=*.SYSLMOD         00008400
//   DD   DSNAME=&LIB.L,UNIT=&ULIB,VOLUME=&VLIB,DISP=SHR                     00008500
//   DD   DSNAME=&LIB1.L,UNIT=&ULIB1,VOLUME=&VLIB1,DISP=SHR                  00008600
//          DD DSNAME=&JOBLIB,DISP=SHR                                       00008700
//FMSODANQ   DD  DISP=SHR,DSNAME=&ISAM.D,VOLUME=REF=*.STEPLIB                00008800
//ISAMWORK   DD  UNIT=&STG,DCB=DSORG=IS,SPACE=(CYL,(10)),                    X00008900
//              DISP=(,&NRMDSP,&CHKDSP),DSNAME=&CHKID.O                       00009000
//SAMFILE DD DSNAME=&SAM.S,UNIT=&USAM,VOLUME=&VSAM,DISP=(SHR,KEEP),          X00009100
//              LABEL=(,&LAB),DCB=(RECFM=VB,LRECL=1000,BLKSIZE=&BSZFILE,X00009200
//              TRTCH=&TRCH,DEN=&DEN)                                         00009300
//FMSAMOUT DD   &SAMOUT.DCB=(RECFM=VB,LRECL=1000,BLKSIZE=&BSZNEWF,           X00009400
//              DEN=&DEN,TRTCH=&TRCH),DISP=(NEW,KEEP),                        00009500
//              LABEL=(,&LAB),                                                00009600
//              UNIT=&USAM,VOLUME=(PRIVATE,RETAIN,&VSMOUT)                    00009700
//FMNDATA   DD   &SAMOUT.DCB=*.FMSAMOUT,DISP=(,KEEP),LABEL=(,&LAB),          X00009800
//              UNIT=&USAM,VOLUME=(PRIVATE,RETAIN,&VSMMCT)                    00009900
//CHECKDD DD DSNAME=&CHKID,DISP=(&CHKST,&NRMDSP,&CHKDSP),UNIT=&UCHK,         00010000
//              VOLUME=&VCHK,SPACE=(CYL,(&CHKSP,1))                           00010100
//STAT    DD  UNIT=&STG,SPACE=(TRK,(20,5))                                   00010200
//TRANST   DD   DSN=&TRANS,VOL=&VTRANS,UNIT=&UTRANS,DISP=&TDISP,             00010300
//             SPACE=(TRK,1)                                                 00010400
//INDEXPRT   DD  SYSOUT=(&A,&CL)                                             00010500
//IXTRANS DD DISP=(NEW,PASS),UNIT=&STG,SPACE=(CYL,(&INXSP,1))                00010600
//XINDEX DD DSN=&XINDEX.X,DISP=(&XDISP,KEEP),UNIT=&XUNIT,VOL=&XVOL,          00010700
//              DCB=(BLKSIZE=&BLKSIZE,RECFM=F,KEYLEN=4,DSORG=DA),            X00010800
//              SPACE=(&BLKSIZE,&NBRBLK)                                      00010900
```

```
//XFR     PROC  A=A,AUXSP=6,B=B,BLKSIZE=560,BSZFILE=,BSZNEWF=,          X00000100
//              VSDSN='DUMMY.FILE',VSOLDF='DUMMY.FILE',                 00000300
//              CL=',',CL1=',',CL2=',',                                 X00000500
//              DEN=,INXSP=1,ISAM='DUMMY.FILE',                         00000600
//              JOBLIB='FFS.JOBLIB',JOBMAC='FFS.JOBMACRO',              00000700
//              LAB=SL,LIB='DUMMY.FILE',LIB1='DUMMY.FILE',              X00000800
//              MCTSP=8,NEWFFT='DUMMY.FILE',NBRBLK=200,                 00000900
//              NEWUSAM='(TAPE9,,DEFER)',                               00001000
//              PTFJOBL='PTF.JOBLIB',PTFJOBM='PTF.JOBMACRO',            00001100
//              RGN=100K,SAM='DUMMY.FILE',SORTSP=10,STG=NIPW,           00001200
//              TDISP=MOD,TRANS='&&TRANS',TRANSP=200,TRANTYP=ISAM,      00001300
//              TRCH=,UFFT='(2314,P)',UISAM='(2314,P)',                 00001400
//              ULIB=2314,ULIB1=2314,                                   00001500
//              USAM='(TAPE9,,DEFER)',UTRANS=NIPW,VFFT=,VISAM=,         00001600
//              VLIB=,VLIB1=,VSAM=,VSMMCT=,VSMOUT=,VSORT='REF=*.SORTIN', 00001700
//              VTRANS=,XDISP=SHR,XINDEX='DUMMY.FILE',                  00001800
//              XUNIT=2314,XVOL=                                        00001900
//*                                                                     00002000
//*  CHARLES W. HICKISCH  MAJOR,USA  PRCJ CODE=763NIPS  BRANCH=431     00002100
//*   DATE=MARCH 1, 1974                                                00002200
//*                                                                     00002300
//FR      EXEC  PGM=FR,REGION=&RGN                                      00002400
//STEPLIB   DD  DSN=&PTFJOBL,DISP=SHR                                   00002600
//          DD  DSN=&JOBLIB,DISP=SHR                                    00002700
//SYSLIST   DD  SYSOUT=(&A,&CL)                                         00002800
//SYSOUT DD SYSOUT=(&A,&CL)                                             00002900
//SYSUDUMP DD SYSOUT=(&A,&CL1)                                          00003000
//SYSPRINT DD SYSOUT=(&A,&CL2)                                          00003100
//SYSPUNCH DD  SYSOUT=&B                                                00003200
//OLDVSM   DD  DSN=&VSOLDF,DISP=SHR                                     00003300
//VSMFILE   DD  DSN=&VSDSN,DISP=SHR                                     00003400
//OLDFILE   DD  DSNAME=&ISAM,UNIT=&UISAM,VOLUME=&VISAM,DISP=SHR         00003500
//DATAFILE DD  DSNAME=&NEWFFT,UNIT=&UFFT,VOLUME=&VFFT,DISP=SHR          00003600
//SAMFILE   DD  DSNAME=&SAM.S,UNIT=&USAM,VOLUME=&VSAM,LABEL=(,&LAB),    C00003700
//              DCB=(RECFM=VB,LRECL=1000,BLKSIZE=&BSZFILE,DEN=&DEN,     X00003800
//              TRTCH=&TRCH),DISP=(SHR,PASS)                            00003900
//TDDRCDS   DD  UNIT=&STG,SPACE=(CYL,(3,1))                             00004000
//POOLRCDS DD  UNIT=&STG,SPACE=(CYL,(3,1))                              00004100
//FRVSNA    DD  UNIT=&STG,SPACE=(CYL,(5,1))                             00004200
//FRVSNB    DD  UNIT=&STG,SPACE=(CYL,(5,1))                             00004300
//FRGENCD   DD  UNIT=&STG,SPACE=(TRK,1)                                 00004400
//SORTLIB   DD  DSNAME=SYS1.SORTLIB,DISP=SHR                            00004500
//FMCOMM    DD  UNIT=&STG,SPACE=(TRK,(1,1))                             00004600
//FMFLUD    DD  UNIT=&STG,SPACE=(CYL,(&MCTSP,10))                       00004700
//FMTRANS   DD  UNIT=&STG,SPACE=(CYL,(&AUXSP,5))                        00004800
//FMLABELS DD  UNIT=&STG,SPACE=(CYL,(1,1))                              00004900
//FMCM      DD  UNIT=&STG,SPACE=(TRK,(20,5))                            00005000
//FMCMFILE DD  UNIT=&STG,SPACE=(TRK,(20,5))                             00005100
//FMLITFIL DD  UNIT=&STG,SPACE=(TRK,(20,5))                             00005200
//FMLEFILE DD  UNIT=&STG,SPACE=(CYL,(5,2))                              00005300
//FMMACRO   DD  UNIT=&STG,SPACE=(CYL,(8,2))                             00005400
//FMOUTMAC DD  UNIT=&STG,SPACE=(CYL,(8,2))                              00005500
//SYSLIB    DD  DSN=&PTFJOBM,DISP=SHR                                   00005600
//          DD  DSN=&JOBMAC,DISP=SHR                                    00005700
//          DD  DSNAME=SYS1.MACLIB,DISP=SHR                             00005800
//FMLEIN    DD  UNIT=&STG,SPACE=(TRK,(25,5))                            00005900
//SYSLMOD   DD  UNIT=&STG,SPACE=(CYL,(4,1,10)),DCB=&JOBLIB,             C00006000
```

```
//                LABEL=EXPDT=66366                                          00006100
//TEMLSREC DD     UNIT=&STG,SPACE=(TRK,(20,5))                               00006200
//SORTIN   DD     UNIT=&STG,SPACE=(TRK,(&TRANSP,20))                         00006300
//SORTWK01 DD     SPACE=(CYL,(&SORTSP),,CONTIG),UNIT=&STG                    00006400
//SORTWK02 DD     SPACE=(CYL,(&SORTSP),,CONTIG),UNIT=(&STG,SEP=SORTWK01)     00006500
//SORTWK03 DD     SPACE=(CYL,(&SORTSP),,CONTIG),UNIT=(&STG,SEP=SORTWK02)     00006600
//SORTWK04 DD     SPACE=(CYL,(&SORTSP),,CONTIG),                            C00006700
//                UNIT=(&STG,SEP=(SORTWK01,SORTWK03))                        00006800
//SORTWK05 DD     SPACE=(CYL,(&SORTSP),,CONTIG),                            C00006900
//                UNIT=(&STG,SEP=(SORTWK02,SORTWK04))                        00007000
//SORTWK06 DD     SPACE=(CYL,(&SORTSP),,CONTIG),                            C00007100
//                UNIT=(&STG,SEP=(SORTWK03,SORTWK05))                        00007200
//SORTOUT  DD     DSNAME=*.SORTIN,VOLUME=REF=*.SORTIN,DISP=(OLD,PASS),      C00007300
//                DCB=(LRECL=1500,BLKSIZE=1504,RECFM=VB)                     00007400
//TAPEIN   DD     DSNAME=*.SORTIN,VOLUME=&VSORT,DISP=(OLD,PASS)              00007500
//TAPEOUT  DD     DSNAME=*.SORTIN,VOLUME=&VSORT,DISP=(OLD,PASS),            C00007600
//                DCB=(LRECL=1500,BLKSIZE=1504,RECFM=VB)                     00007700
//RECSIN   DD     DSNAME=*.FMFLUD,VOLUME=REF=*.FMFLUD,DISP=(OLD,PASS)        00007800
//RECSOUT  DD     DSNAME=*.RECSIN,VOLUME=REF=*.RECSIN,DISP=(OLD,PASS),      C00007900
//                DCB=(LRECL=1000,BLKSIZE=1004,RECFM=VB)                     00008000
//RECSWK01 DD     DSNAME=*.SORTWK01,VOLUME=REF=*.SORTWK01,DISP=(OLD,PASS)    00008100
//RECSWK02 DD     DSNAME=*.SORTWK02,VOLUME=REF=*.SORTWK02,DISP=(OLD,PASS)    00008200
//RECSWK03 DD     DSNAME=*.SORTWK03,VOLUME=REF=*.SORTWK03,DISP=(OLD,PASS)    00008300
//RECSWK04 DD     DSNAME=*.SORTWK04,VOLUME=REF=*.SORTWK04,DISP=(OLD,PASS)    00008400
//RECSWK05 DD     DSNAME=*.SORTWK05,VOLUME=REF=*.SORTWK05,DISP=(OLD,PASS)    00008500
//RECSWK06 DD     DSNAME=*.SORTWK06,VOLUME=REF=*.SORTWK06,DISP=(OLD,PASS)    00008600
//TRANS    DD     DDNAME=&TRANTYP                                            00008700
//VSAM     DD     DSN=&VSOLDF,DISP=SHR                                       00008800
//ISAM     DD     DSNAME=&ISAM,UNIT=&UISAM,VOLUME=&VISAM,DISP=SHR            00008900
//SAM      DD     DSNAME=&SAM.S,UNIT=&USAM,VOLUME=&VSAM,LABEL=(,&LAB),      C00009000
//                DCB=*.SAMFILE,DISP=(SHR,KEEP)                              00009100
//ASSEMIN  DD     DSNAME=*.FMOUTMAC,VOLUME=REF=*.FMOUTMAC,DISP=(OLD,PASS)    00009200
//FMSETTBL DD     UNIT=&STG,SPACE=(TRK,(1,1))                                00009300
//FMAUXOP  DD     DSNAME=*.FMTRANS,VOLUME=REF=*.FMTRANS,DISP=(OLD,PASS),     00009400
//                DCB=(LRECL=1000,BLKSIZE=1004)                              00009500
//SLIB     DD     DSNAME=*.SYSLMOD,DISP=(OLD,PASS),VOLUME=REF=*.SYSLMOD      00009600
//         DD     DSNAME=&LIB.L,UNIT=&ULIB,VOLUME=&VLIB,DISP=SHR             00009700
//         DD     DSNAME=&LIB1.L,UNIT=&ULIB1,VOLUME=&VLIB1,DISP=SHR          00009800
//         DD     DSNAME=&JOBLIB,DISP=SHR                                    00009900
//FMSAMOUT DD     UNIT=&NEWUSAM,VOLUME=(PRIVATE,RETAIN,&VSMOUT),            C00010000
//                LABEL=(,&LAB),DISP=(NEW,KEEP),                             00010100
//                DCB=(RECFM=VB,LRECL=1000,BLKSIZE=&BSZNEWF,DEN=&DEN,       X00010200
//                TRTCH=&TRCH)                                               00010300
//FMNDATA  DD     UNIT=&NEWUSAM,VOLUME=(PRIVATE,RETAIN,&VSMPCT),            C00010400
//                LABEL=(,&LAB),DISP=(,KEEP),DCB=*.FMSAMOUT                  00010500
//STAT  DD  DSN=*.FMCM,VOL=REF=*.FMCM,DISP=(OLD,PASS)                        00010600
//TRANST  DD  DSN=&TRANS,VOL=&VTRANS,UNIT=&UTRANS,DISP=&TDISP,               00010700
//            SPACE=(TRK,1)                                                  00010800
//INDEXPRT  DD  SYSOUT=(&A,&CL)                                              00010900
//IXTRANS DD DISP=(NEW,PASS),UNIT=&STG,SPACE=(CYL,(&INXSP,1))                00011000
//XINDEX DD DSN=&XINDEX.X,DISP=(&XDISP,KEEP),UNIT=&XUNIT,VOL=&XVOL,          00011100
//                DCB=(BLKSIZE=&BLKSIZE,RECFM=F,KEYLEN=4,DSORG=DA),         X00011200
//                SPACE=(&BLKSIZE,&NBRBLK)                                   00011300
```

NIPS 360 FFS PROCEDURES

```
//XFS     PROC  A=A,CL=',',CL1=',',BSZNEWF=1004,INDEX=1,                X00000100
//              ISAM='DUMMY.FILE',                                      X00000200
//              JOBLIB='FFS.JOBLIB',                                     00000300
//              LIB='DUMMY.FILE',LIB1='DUMMY.FILE',                      00000400
//              NDISP=KEEP,OVFLOW=1,PRIME=5,RGN=60K,SDISP=SHR,           00000500
//              PTFJOBL='PTF.JOBLIB',                                    00000600
//              SORTSP=3,SOURCL='DUMMY.FILE',STG=NIPW,                   00000700
//              UISAM=2314,ULIB=2314,ULIB1=2314,USOURCL=2314,            00000800
//              VISAM=,VLIB=,VLIB1=,VSOURCL=                             00000900
//*                                                                      00001000
//*  CHARLES W. HICKISCH  MAJOR,USA  PROJ CODE=763NIPS  BRANCH=431      00001100
//*   DATE=MARCH 1, 1974                                                 00001200
//*                                                                      00001300
//FS      EXEC   PGM=FSPHASE,REGION=&RGN                                 00001400
//STEPLIB  DD  DSN=&PTFJOBL,DISP=SHR                                     00001500
//         DD  DSN=&JOBLIB,DISP=SHR                                      00001600
//SLIB    DD DSN=&LIB.L,DISP=SHR,VOLUME=&VLIB,UNIT=&ULIB                 00001700
//        DD DSN=&LIB1.L,UNIT=&ULIB1,VOLUME=&VLIB1,DISP=SHR              00001800
//        DD DSN=&JOBLIB,DISP=SHR                                        00001900
//SOURCLIB  DD  DISP=&SDISP,DSN=&SOURCL.L,VCL=&VSOURCL,UNIT=&USOURCL     00002000
//NEWFILE DD DSN=&ISAM.(INDEX),UNIT=&UISAM,VOLUME=&VISAM,               *00002100
//              SPACE=(CYL,&INDEX),DCB=(DSORG=IS,BLKSIZE=&BSZNEWF),     X00002200
//              DISP=(,&NDISP,DELETE)                                    00002300
//        DD DSN=&ISAM.(PRIME),UNIT=&UISAM,VOLUME=REF=*.NEWFILE,        *00002400
//              SPACE=(CYL,&PRIME),DCB=*.NEWFILE,DISP=(,&NDISP,DELETE)   00002500
//        DD DSN=&ISAM.(OVFLOW),UNIT=&UISAM,VOLUME=REF=*.NEWFILE,       *00002600
//              SPACE=(CYL,&OVFLOW),DCB=*.NEWFILE,DISP=(,&NDISP,DELETE)  00002700
//SORTLIB DD DSN=SYS1.SORTLIB,DISP=SHR                                   00002800
//SORTIN DD UNIT=&STG,SPACE=(CYL,(1,1)),                                *00002900
//        DCB=(RECFM=VB,BLKSIZE=1004,LRECL=1000)                         00003000
//SORTWK01 DD SPACE=(CYL,(&SORTSP),,CONTIG),UNIT=(&STG,SEP=SORTIN)       00003100
//SORTWK02 DD SPACE=(CYL,(&SORTSP),,CONTIG),UNIT=(&STG,SEP=SORTWK01),   *00003200
//              SEP=SORTWK01                                             00003300
//SORTWK03 DD SPACE=(CYL,(&SORTSP),,CONTIG),UNIT=(&STG,SEP=SORTWK02),   *00003400
//              SEP=SORTWK02                                             00003500
//SORTWK04 DD SPACE=(CYL,(&SORTSP),,CONTIG),                            *00003600
//              UNIT=(&STG,SEP=(SORTWK01,SORTWK03)),                    *00003700
//              SEP=(SORTWK01,SORTWK03)                                  00003800
//SORTWK05 DD SPACE=(CYL,(&SORTSP),,CONTIG),                            *00003900
//              UNIT=(&STG,SEP=(SORTWK02,SORTWK04)),                    *00004000
//              SEP=(SORTWK02,SORTWK04)                                  00004100
//SORTWK06 DD SPACE=(CYL,(&SORTSP),,CONTIG),                            *00004200
//              UNIT=(&STG,SEP=(SORTWK01,SORTWK03,SORTWK05)),           *00004300
//              SEP=(SORTWK01,SORTWK03,SORTWK05)                         00004400
//SORTOUT DD    UNIT=&STG,SPACE=(CYL,(1,1)),                            *00004500
//              DCB=(RECFM=VB,BLKSIZE=1004,LRECL=1000)                   00004600
//SYSOUT DD DUMMY                                                        00004700
//SYSPRINT DD SYSOUT=(&A,&CL)                                            00004800
//SYSUDUMP DD SYSOUT=(&A,&CL1)                                           00004900
//SOURCPRT DD SYSOUT=(A,&CL)                                             00005000
//SYSUT1 DD DISP=(,PASS),DSN=IXSPDS,UNIT=&STG,SPACE=(TRK,(5,1))          00005100
//FSSTEP2  EXEC PGM=FSABMOD,COND=(111,NE,FS),REGION=&RGN                 00005200
//STEPLIB DD DSN=&JOBLIB,DISP=SHR                                        00005300
//SYSOUT DD SYSOUT=(&A,&CL)                                              00005400
//SYSPRINT DD SYSOUT=(&A,&CL)                                            00005500
//DELNEW  DD    DSNAME=*.FS.NEWFILE,UNIT=&UISAM,VOLUME=REF=*.FS.NEWFILE, *00005600
//              DISP=(OLD,DELETE)                                        00005700
```

```
//XISTOS PROC  A=A,BSZFILE=,BSZNEWF=,CC=,CL=',',CL1=',',DEN=,
//             ISAM='DUMMY.FILE',JOBLIB='FFS.JOBLIB',LAB=SL,NDISP=KEEP,
//             ODISP=KEEP,OLDSAM='DUMMY.FILE',OLDVSAM=,
//             OSDISP=KEEP,RGN=60K,SAM='DUMMY.FILE',SEQNO=1,
//             PTFJOBL='PTF.JOBLIB',
//             TRCH=,UISAM='(2314,P)',USAM='(TAPE9,,DEFER)',
//             VISAM=,VSAM=
//*
//*  CHARLES W. HICKISCH  MAJOR,USA  PROJ CODE=763NIPS  BRANCH=431
//*   DATE=MARCH 1, 1974
//*
//STP1 EXEC PGM=UTBLDSAM,REGION=&RGN,PARM='&CC'
//STEPLIB  DD  DSN=&PTFJOBL,DISP=SHR
//         DD  DSN=&JOBLIB,DISP=SHR
//DATAFILE  DD  DSNAME=&ISAM,UNIT=&UISAM,VOLUME=&VISAM,
//       DISP=(SHR,&ODISP,KEEP),DCB=BUFNO=5
//SAMFILE  DD  DSNAME=&OLDSAM.S,UNIT=&USAM,DISP=(SHR,&OSDISP),
//             VOL=&OLDVSAM,DCB=(RECFM=VB,LRECL=1000,BLKSIZE=&BSZFILE,
//       DEN=&DEN,TRTCH=&TRCH,BUFNO=5),LABEL=(&SEQNO,&LAB)
//SAMOUT  DD  DSNAME=&SAM.S,UNIT=&USAM,VOLUME=&VSAM,DISP=(,&NDISP),
//             DCB=(*.SAMFILE,BLKSIZE=&BSZNEWF),LABEL=(&SEQNO,&LAB)
//SYSPRINT DD SYSOUT=(&A,&CL)
//SYSUDUMP DD SYSOUT=(&A,&CL1)
```

NIPS 360 FFS PROCEDURES

```
//XKA     PROC  A=A,CL=',',CL1=',',JOBLIB='FFS.JOBLIB',RGN=100K,
//              PTFJOBL='PTF.JOBLIB',
//              STG=NIPW,SORTSP=10,
//              LIB=,ULIB=2314,VLIB=,
//              ISAM='DUMMY.FILE',UISAM='(2314,P)',VISAM=,
//              SAM='DUMMY.FILE',USAM='(TAPE9,,DEFER)',VSAM=,
//              LAB=SL,DEN=3,TRCH=
//*
//*                       PARAMETERS
//*
//* A           SYSOUT - DEFAULT=PRINTER
//* CL          SYSOUT CLASSIFICATION  - REQUIRED
//* CL1         SYSUDUMP CLASSIFICATION - REQUIRED
//* JOBLIB      STEPLIB DSN - DEFAULT=FFS.JOBLIB
//* RGN         STEP REGION - DEFAULT=175K
//*PTFJOBL      STEPLIB DSN - DEFAULT=PTF.JOBLIB
//* STG         TEMP STORAGE UNIT - DEFAULT=NIPW
//* SORTSP      SORTWKNN CYLINDERS - DEFAULT=10
//* LIB         USER LIBRARY DSN - REQUIRED
//* ULIB        USER LIBRARY UNIT - DEFAULT=2314
//* VLIB        USER LIBRARY 'SER=SERIAL' - REQUIRED
//*
//* ISAM        DSNAME IF ISAM DATA FILE. CMIT IF SAM FILE.
//* UISAM       ISAM UNIT CLASS
//* VISAM       ISAM VOL SERIAL
//*
//* SAM         DSNAME IF SAM DATA FILE. OMIT IF ISAM FILE.
//* USAM        SAM UNIT CLASS
//* VSAM        SAM VOL SERIAL
//* LAB         SAM VOL LABEL IF NOT STANDARD
//* DEN         SAM VOL DENSITY IF NOT 1600 BPI.
//* TRCH        SAM CONVERSION IF M-TRACK VOL.
//*
//*  CHARLES W. HICKISCH  MAJOR,USA  PRCJ CODE=763NIPS  BRANCH=431
//*   DATE=MARCH 1,1974
//*
//XKA     EXEC  PGM=UTNDXKAN,
//              REGION=&RGN
//*
//STEPLIB  DD   DSN=&PTFJOBL,
//              DISP=SHR
//         DD   DSN=&JOBLIB,
//              DISP=SHR
//*
//*
//DATAFILE DD   DSN=&ISAM,
//              DISP=SHR,
//              UNIT=&UISAM,
//              VOL=&VISAM
//*
//SAMFILE  DD   DSN=&SAM.S,
//              UNIT=&USAM,
//              VOL=&VSAM,
//              DISP=(SHR,KEEP),
//              LABEL=(,&LAB),
//              DCB=(DEN=&DEN,TRTCH=&TRCH)
//*
```

```
//SLIB     DD  DSN=&LIB.L,                                          00005400
//             UNIT=&ULIB,                                          00005500
//             VOL=&VLIB,                                           00005600
//             DISP=SHR                                             00005700
//*                                                                 00005800
//        DD   DSN=&JOBLIB,DISP=SHR                                 00005900
//*                                                                 00006000
//KANWK01  DD  UNIT=&STG,                                           00006100
//             DISP=(,PASS),                                        00006200
//             SPACE=(CYL,(&SORTSP,&SORTSP))                        00006300
//*                                                                 00006400
//SYSOUT   DD  SYSOUT=(&A,&CL)                                      00006500
//*                                                                 00006600
//SYSPRINT DD  SYSOUT=(&A,&CL)                                      00006700
//*                                                                 00006800
//KMDPRT   DD  SYSOUT=(&A,&CL),SPACE=(CYL,(2,1))                    00006900
//*                                                                 00007000
//SYSUDUMP DD  SYSOUT=(&A,&CL1)                                     00007100
//*                                                                 00007200
//SORTLIB  DD  DSN=SYS1.SORTLIB,                                    00007300
//             DISP=SHR                                             00007400
//*                                                                 00007500
//SORTWK01 DD  UNIT=&STG,                                           00007600
//             SPACE=(CYL,(&SORTSP),,CONTIG)                        00007700
//*                                                                 00007800
//SORTWK02 DD UNIT=(&STG,SEP=SORTWK01),SEP=SORTWK01,                00007900
//             SPACE=(CYL,(&SORTSP),,CONTIG)                        00008000
//*                                                                 00008100
//SORTWK03 DD UNIT=(&STG,SEP=SORTWK02),SEP=SORTWK02,                00008200
//             SPACE=(CYL,(&SORTSP),,CONTIG)                        00008300
//*                                                                 00008400
//SORTWK04 DD UNIT=(&STG,SEP=(SORTWK01,SORTWK03)),                  00008500
//            SEP=(SORTWK01,SORTWK03),                              00008600
//             SPACE=(CYL,(&SORTSP),,CONTIG)                        00008700
//*                                                                 00008800
//SORTWK05 DD  UNIT=(&STG,SEP=(SORTWK02,SORTWK04)),                 00008900
//             SEP=(SORTWK02,SORTWK04),                             00009000
//             SPACE=(CYL,(&SORTSP),,CONTIG)                        00009100
//*                                                                 00009200
//SORTWK06 DD  UNIT=(&STG,SEP=(SORTWK01,SORTWK03,SORTWK05)),        00009300
//             SEP=(SORTWK01,SORTWK03,SORTWK05),                    00009400
//             SPACE=(CYL,(&SORTSP),,CONTIG)                        00009500
//*                                                                 00009600
```

NIPS 360 FFS PROCEDURES

```
//XKM     PROC  A=A,CL=',',CL1=',',JOBLIB='FFS.JOBLIB',RGN=100K,
//              PTFJOBL='PTF.JOBLIB',
//              STG=NIPW,SORTSP=10,
//              LIB=,ULIB=2314,VLIB=,DLIB='MCD,KEEP',
//              PRISP=2,SECSP=2,BLKSI=3478
//*
//*                       PARAMETERS
//*
//* A           SYSOUT - DEFAULT=PRINTER
//* CL          SYSOUT CLASSIFICATION  - REQUIRED
//* CL1         SYSUDUMP CLASSIFICATION - REQUIRED
//* JOBLIB      STEPLIB DSN - DEFAULT=FFS.JOBLIB
//* RGN         STEP REGION - DEFAULT=175K
//*PTFJOBL      STEPLIB DSN - DEFAULT=PTF.JOBLIB
//* STG         TEMP STORAGE UNIT - DEFAULT=NIPW
//* SORTSP      SORTWKNN CYLINDERS - DEFAULT=10
//* LIB         USER LIBRARY DSN - REQUIRED
//* ULIB        USER LIBRARY UNIT - DEFAULT=2314
//* VLIB        USER LIBRARY 'SER=SERIAL' - REQUIRED
//* DLIB        USER LIBRARY DISP - DEFAULT='MOD,KEEP'
//* PRISP       WORK SPACE PRIMARY CYLS - DEFAULT=2
//* SECSP       WORK SPACE SECONDARY CYLS - DEFAULT=2
//* BLKSI       WORK SPACE DCB BLKSIZE - DEFAULT=94
//*
//*  CHARLES W. HICKISCH  MAJOR,USA  PROJ CODE=763NIPS  BRANCH=431
//*   DATE=MARCH 1,1974
//*
//XKM     EXEC  PGM=UTNDXKMD,
//              REGION=&RGN
//*
//STEPLIB  DD   DSN=&PTFJOBL,
//              DISP=SHR
//         DD   DSN=&JOBLIB,
//              DISP=SHR
//*
//SLIB     DD   DSN=&LIB.L,
//              UNIT=&ULIB,
//              VOL=&VLIB,
//              DISP=(&DLIB)
//*
//KMDWK01  DD   UNIT=&STG,
//              DISP=(,PASS),
//              SPACE=(CYL,(&PRISP,&SECSP)),
//              DCB=(RECFM=FBS,LRECL=94,BLKSIZE=&BLKSI)
//*
//KMDWK02  DD   UNIT=&STG,
//              DISP=(,PASS),
//              SPACE=(CYL,(&PRISP,&SECSP)),
//              DCB=(RECFM=FBS,LRECL=94,BLKSIZE=&BLKSI)
//*
//KMDWK03  DD   UNIT=&STG,
//              DISP=(,PASS),
//              SPACE=(TRK,(1,1))
//*
//KMDWK04  DD   UNIT=&STG,
//              DISP=(,PASS),
//              SPACE=(TRK,(5,5)),
```

```
//              DCB=(RECFM=FBS,LRECL=72,BLKSIZE=720)                  00005400
//*                                                                   00005500
//KMDWK05  DD   UNIT=&STG,                                            00005600
//              DISP=(,PASS),                                         00005700
//              SPACE=(CYL,(&PRISP,&SECSP)),                          00005800
//              DCB=(RECFM=FBS,LRECL=94,BLKSIZE=&BLKSI)               00005900
//*                                                                   00006000
//SYSOUT   DD   SYSOUT=(&A,&CL)                                       00006100
//*                                                                   00006200
//SYSPRINT DD   SYSOUT=(&A,&CL)                                       00006300
//KMDPRT   DD   SYSOUT=(&A,&CL),SPACE=(CYL,(2,1))                     00006400
//*                                                                   00006500
//SYSUDUMP DD   SYSOUT=(&A,&CL1)                                      00006600
//*                                                                   00006700
//SORTLIB  DD   DSN=SYS1.SORTLIB,                                     00006800
//              DISP=SHR                                              00006900
//*                                                                   00007000
//SORTWK01 DD   UNIT=&STG,                                            00007100
//              SPACE=(CYL,(&SORTSP),,CONTIG)                         00007200
//*                                                                   00007300
//SORTWK02 DD UNIT=(&STG,SEP=SORTWK01),SEP=SORTWK01,                  00007400
//              SPACE=(CYL,(&SORTSP),,CONTIG)                         00007500
//*                                                                   00007600
//SORTWK03 DD UNIT=(&STG,SEP=SORTWK02),SEP=SCRTWK02,                  00007700
//              SPACE=(CYL,(&SORTSP),,CONTIG)                         00007800
//*                                                                   00007900
//SORTWK04 DD UNIT=(&STG,SEP=(SORTWK01,SORTWK03)),                    00008000
//             SEP=(SORTWK01,SORTWK03),                               00008100
//              SPACE=(CYL,(&SORTSP),,CONTIG)                         00008200
//*                                                                   00008300
//SORTWK05 DD   UNIT=(&STG,SEP=(SORTWK02,SORTWK04)),                  00008400
//              SEP=(SORTWK02,SORTWK04),                              00008500
//              SPACE=(CYL,(&SORTSP),,CONTIG)                         00008600
//*                                                                   00008700
//SORTWK06 DD   UNIT=(&STG,SEP=(SORTWK01,SCRTWK03,SORTWK05)),         00008800
//              SEP=(SORTWK01,SORTWK03,SORTWK05),                     00008900
//              SPACE=(CYL,(&SORTSP),,CONTIG)                         00009000
//*                                                                   00009100
```

```
//XOP     PROC  A=A,B=B,BSZFILE=,BSZFIL1=,BSZFIL2=,                          00000100
//              CL=',',CL1=',',CL2=',',                                       00000200
//              ISAM='DUMMY.FILE',ISAM1='DUMMY.FILE',ISAM2='DUMMY.FILE',      00000300
//              JOBLIB='FFS.JOBLIB',JOBMAC='FFS.JOBMACRO',                    00000400
//              LAB=SL,                                                       00000500
//              LIB='DUMMY.FILE',LIB1='DUMMY.FILE',LIB2='DUMMY.FILE',         00000600
//              LIBDISP=SHR,                                                  00000700
//              PTFJOBL='PTF.JOBLIB',                                         00000710
//              PTFJOBM='PTF.JOBMACRO',                                       00000720
//              QDF='&&QDFILE',QDISP=PASS,QRT='&&QRTFILE',                    00000800
//              RGN=100K,                                                     00000900
//              SAM='DUMMY.FILE',SAM1='DUMMY.FILE',SAM2='DUMMY.FILE',         00001000
//              SDISP=SHR,SOURCL='DUMMY.FILE',STG=NIPW,                       00001100
//              TDISP=MOD,TRANS='&&TRANS',                                    00001200
//              UISAM='(2314,P)',                                             00001300
//              ULIB=2314,ULIB1=2314,ULIB2=2314,                              00001400
//              UQDF=NIPW,UQRT=NIPW,                                          00001500
//              USAM='(TAPE9,,DEFER)',USOURCL=2314,                           00001600
//              UTRANS=NIPW,                                                  00001700
//              VISAM=,VISAM1=,VISAM2=,                                       00001800
//              VLIB=,VLIB1=,VLIB2=,                                          00001900
//              VQDF=,VQRT=,                                                  00002000
//              VSAM=,VSAM1=,VSAM2=,                                          00002100
//              VSOURCL=,VTRANS=                                              00002200
//*                                                                           00002300
//*  CHARLES W. HICKISCH  MAJOR,USA  PROJ CODE=763NIPS  BRANCH=431           00002400
//*   DATE=MARCH 1, 1974                                                      00002500
//*                                                                           00002600
//OP EXEC PGM=OP,REGION=&RGN                                                  00002700
//STEPLIB  DD  DSN=&PTFJOBL,DISP=SHR                                          00002800
//         DD  DSN=&JOBLIB,DISP=SHR                                           00002810
//DATAFILE DD DSNAME=&ISAM,DISP=SHR,UNIT=&UISAM,VOLUME=&VISAM                 00002900
//DATAFIL1 DD DSNAME=&ISAM1,DISP=SHR,UNIT=&UISAM,VOLUME=&VISAM1               00003000
//DATAFIL2 DD DSNAME=&ISAM2,DISP=SHR,UNIT=&UISAM,VOLUME=&VISAM2               00003100
//SAMFILE  DD DSNAME=&SAM.S,DISP=SHR,UNIT=&USAM,VOLUME=&VSAM,                X00003200
//             LABEL=(,&LAB),DCB=BLKSIZE=&BSZFILE                             00003300
//SAMFILE1 DD DSNAME=&SAM1.S,DISP=SHR,UNIT=&USAM,VOLUME=&VSAM1,               X00003400
//             LABEL=(,&LAB),DCB=BLKSIZE=&BSZFIL1                             00003500
//SAMFILE2  DD DSNAME=&SAM2.S,DISP=SHR,UNIT=&USAM,VOLUME=&VSAM2,              *00003600
//             LABEL=(,&LAB),DCB=BLKSIZE=&BSZFIL2                             00003700
//DLIB  DD DSNAME=&LIB.L,DISP=&LIBDISP,VOLUME=&VLIB,UNIT=&ULIB                00003800
//TLIB  DD DSNAME=&LIB1.L,DISP=SHR,VOLUME=&VLIB1,UNIT=&ULIB1                  00003900
//SOURCPRT DD SYSOUT=(&A,&CL)                                                 00004000
//SOURCLIB  DD  DISP=&SDISP,DSN=&SOURCL.L,VOL=&VSOURCL,UNIT=&USOURCL          00004100
//SYSPRINT DD SYSOUT=(&A,&CL),DCB=(LRECL=133,BLKSIZE=665)                     00004200
//SYSOUT   DD SYSOUT=(&A,&CL)                                                 00004300
//SYSUDUMP DD SYSOUT=(&A,&CL1)                                                00004400
//SYSUT1   DD UNIT=&STG,SPACE=(TRK,(200,20))                                  00004500
//SYSUT2   DD UNIT=(&STG,SEP=SYSUT1),SPACE=(TRK,(200,20))                     00004600
//SYSUT3   DD UNIT=(&STG,SEP=(SYSUT1,SYSUT2)),SPACE=(TRK,(200,20))            00004700
//SYSPUNCH DD UNIT=&STG,SPACE=(TRK,(50,10)),                                  X00004800
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=400)                            00004900
//SYSLIB   DD  DSN=&PTFJOBM,DISP=SHR                                          00005000
//         DD  DSN=&JOBMAC,DISP=SHR                                           00005010
//SYSLMOD  DD UNIT=&STG,SPACE=(TRK,(50,25,5)),DCB=SYS1.LINKLIB,               X00005100
//             LABEL=RETPD=0000                                               00005200
//SYSTPRT DD SYSOUT=(&A,&CL),DCB=(RECFM=FB,LRECL=121,BLKSIZE=605)             00005300
```

```
//OPIWCF    DD UNIT=&STG,SPACE=(TRK,(2,1)),DCB=(BLKSIZE=400,LRECL=80,    *00005400
//                RECFM=FB)                                              00005500
//OPSTST    DD UNIT=&STG,SPACE=(TRK,(10,2))                              00005600
//OPCREATE  DD UNIT=&STG,SPACE=(TRK,(1,1))                               00005700
//OPCOMREC  DD UNIT=&STG,SPACE=(TRK,(1,1))                               00005800
//INSTS     DD UNIT=&STG,SPACE=(TRK,(50,5))                              00005900
//LCF       DD UNIT=&STG,SPACE=(TRK,(10,5))                              00006000
//LCT       DD UNIT=&STG,SPACE=(TRK,(10,5))                              00006100
//LITSTR    DD UNIT=&STG,SPACE=(TRK,(50,5))                              00006200
//NAMDEF    DD DSNAME=*.SYSUT1,VOLUME=REF=*.SYSUT1,DISP=(OLD,PASS)       00006300
//LITAB     DD DSNAME=*.SYSUT2,VOLUME=REF=*.SYSUT2,DISP=(OLD,PASS)       00006400
//RGTAB     DD DSNAME=*.SYSUT3,VOLUME=REF=*.SYSUT3,DISP=(OLD,PASS)       00006500
//CGC       DD UNIT=&STG,SPACE=(TRK,(1,1))                               00006600
//ERRTAB    DD UNIT=&STG,SPACE=(TRK,(8,5))                               00006700
//DCTNY     DD DSNAME=*.SYSPUNCH,VOLUME=REF=*.SYSPUNCH,DISP=(OLD,PASS)   00006800
//SUBTAB    DD UNIT=&STG,SPACE=(TRK,(1,1))                               00006900
//OPLGOGO   DD DSNAME=*.SYSUT1,VOLUME=REF=*.SYSUT1,DISP=(OLD,PASS)       00007000
//OPLEXEX   DD DSNAME=*.SYSUT2,VOLUME=REF=*.SYSUT2,DISP=(OLD,PASS)       00007100
//OPLEXDIR  DD DSNAME=*.SYSUT3,VOLUME=REF=*.SYSUT3,DISP=(OLD,PASS)       00007200
//OPLINE     DD SYSOUT=(&A,&CL2),DCB=LRECL=133                           00007300
//OPPUNCH    DD SYSOUT=&B                                                00007400
//SLIB DD DSNAME=*.SYSLMOD,VOLUME=REF=*.SYSLMOD,DISP=(OLD,PASS)          00007500
// DD DSNAME=*.DLIB,DISP=SHR,UNIT=&ULIB,VOLUME=&VLIB                     00007600
// DD DSNAME=*.TLIB,DISP=SHR,UNIT=&ULIB1,VOLUME=&VLIB1                   00007700
//    DD DSNAME=&LIB2.L,DISP=SHR,UNIT=&ULIB2,VOLUME=&VLIB2
//       DD  DSN=&PTFJOBL,DISP=SHR
// DD DSNAME=&JOBLIB,DISP=SHR
//SYSAIN DD DSNAME=*.LITSTR,VOLUME=REF=*.LITSTR,DISP=(OLD,PASS),         00008000
//                DCB=(RECFM=FB,LRECL=80,BLKSIZE=400,BUFNO=5)            00008100
//   DD DSNAME=*.INSTS,VOLUME=REF=*.INSTS,DISP=(OLD,PASS),               00008200
//                DCB=(RECFM=FB,LRECL=80,BLKSIZE=400,BUFNO=5)            00008300
//SYSLIN DD DSNAME=*.LCF,VOLUME=REF=*.LCF,DISP=(OLD,PASS)                00008400
//SYSTIN DD DSNAME=*.LCT,VOLUME=REF=*.LCT,DISP=(OLD,PASS)                00008500
//LOADR  DD DSNAME=*.SYSPUNCH,VOLUME=REF=*.SYSPUNCH,DISP=(OLD,PASS),    X00008600
//                DCB=(RECFM=FB,LRECL=80,BLKSIZE=400)                    00008700
//QDFILE DD DSNAME=&QDF,UNIT=&UQDF,VOLUME=&VQDF,DISP=(SHR,&QDISP,KEEP)  00008800
//QRTFILE DD DSNAME=&QRT,UNIT=&UQRT,VOLUME=&VQRT,DISP=(SHR,&QDISP,KEEP) 00008900
//STAT DD DSN=*.INSTS,VOL=REF=*.INSTS,DISP=(OLD,PASS)                    00009000
//TRANST DD DSN=&TRANS,VOL=&VTRANS,UNIT=&UTRANS,                         00009100
//        DISP=&TDISP,SPACE=(TRK,1)                                      00009200
```

```
//XOPEX  PROC  A=A,B=B,                                                  00000100
//             CL=',',CL1=',',CL2=',',                                   00000200
//             CHKDSP=DELETE,CHKID='&&CHKID',CHKSP=0,CHKST=NEW,          00000300
//             ISAM='DUMMY.FILE',ISAM1='DUMMY.FILE',ISAM2='DUMMY.FILE',  00000400
//             JOBLIB='FFS.JOBLIB',                                      00000500
//             LIB='DUMMY.FILE',LIB1='DUMMY.FILE',LIB2='DUMMY.FILE',     00000600
//             LIBDISP=SHR,                                              00000700
//             NRMDSP=DELETE,                                            00000800
//             PTFJOBL='PTF.JOBLIB',                                     00000810
//             QDF='&&QDFILE',QDISP=PASS,QRT='&&QRTFILE',                00000900
//             RGN=100K,                                                 00001000
//             STG=NIPW,                                                 00001100
//             TDISP=MOD,TRANS='&&TRANS',                                00001200
//             UCHK=NIPW,                                                00001300
//             UISAM='(2314,P)',                                         00001400
//             ULIB=2314,ULIB1=2314,ULIB2=2314,                          00001500
//             UQDF=NIPW,UQRT=NIPW,                                      00001600
//             UTRANS=NIPW,                                              00001700
//             VISAM=,VISAM1=,VISAM2=,                                   00001800
//             VLIB=,VLIB1=,VLIB2=,                                      00001900
//             VQDF=,VQRT=,                                              00002000
//             VCHK=,VTRANS=                                             00002100
//*                                                                      00002200
//*   CHARLES W. HICKISCH  MAJOR,USA  PROJ CODE=763NIPS  BRANCH=431     00002300
//*   DATE=MARCH 1, 1974                                                00002400
//*                                                                      00002500
//OP EXEC PGM=OP,REGION=&RGN                                             00002600
//STEPLIB  DD  DSN=&PTFJOBL,DISP=SHR                                     00002700
//         DD  DSN=&JOBLIB,DISP=SHR                                      00002710
//DATAFILE DD DSNAME=&ISAM,DISP=SHR,UNIT=&UISAM,VOLUME=&VISAM            00002800
//DATAFIL1 DD DSNAME=&ISAM1,DISP=SHR,UNIT=&UISAM,VOLUME=&VISAM1          00002900
//DATAFIL2 DD DSNAME=&ISAM2,DISP=SHR,UNIT=&UISAM,VOLUME=&VISAM2          00003000
//DLIB  DD DSNAME=&LIB.L,DISP=&LIBDISP,VOLUME=&VLIB,UNIT=&ULIB           00003100
//TLIB  DD DSNAME=&LIB1.L,DISP=SHR,VOLUME=&VLIB1,UNIT=&ULIB1             00003200
//SOURCPRT DD SYSOUT=(&A,&CL)                                            00003300
//PRINTER  DD SYSOUT=(&A,&CL)                                            00003400
//SYSOUT   DD SYSOUT=(&A,&CL)                                            00003500
//SYSPRINT DD SYSOUT=(&A,&CL)                                            00003600
//SYSUDUMP DD SYSOUT=(&A,&CL1)                                           00003700
//OPLINE    DD SYSOUT=(&A,&CL2),DCB=LRECL=133                            00003800
//OPIWCF   DD UNIT=&STG,SPACE=(TRK,(2,1)),DCB=(BLKSIZE=400,LRECL=80,     *00003900
//             RECFM=FB),DSNAME=&CHKID.A,DISP=(,&NRMDSP,&CHKDSP)         00004000
//OPCOMREC DD UNIT=&STG,SPACE=(TRK,(1,1)),                               X00004100
//             DISP=(,&NRMDSP,&CHKDSP),DSNAME=&CHKID.B                   00004200
//OPLGOGO  DD UNIT=&STG,SPACE=(TRK,(20,1)),                              X00004300
//             DISP=(,&NRMDSP,&CHKDSP),DSNAME=&CHKID.C                   00004400
//OPLEXEX  DD UNIT=(&STG,SEP=(PLGOGO),SPACE=(TRK,(20,1)),                X00004500
//             DISP=(,&NRMDSP,&CHKDSP),DSNAME=&CHKID.D                   00004600
//OPLEXDIR DD UNIT=(&STG,SEP=(OPLGOGO,OPLEXEX)),SPACE=(TRK,(20,1)),      X00004700
//             DISP=(,&NRMDSP,&CHKDSP),DSNAME=&CHKID.E                   00004800
//OPPUNCH   DD SYSOUT=&B                                                 00004900
//SLIB DD DSNAME=*.DLIB,DISP=SHR,UNIT=&ULIB,VOLUME=&VLIB                 00005000
// DD DSNAME=*.TLIB,DISP=SHR,UNIT=&ULIB1,VOLUME=&VLIB1
//    DD DSNAME=&LIB2.L,DISP=SHR,UNIT=&ULIB2,VOLUME=&VLIB2
//      DD  DSN=&PTFJOBL,DISP=SHR
// DD DSNAME=&JOBLIB,DISP=SHR
//QDFILE DD DSNAME=&QDF,UNIT=&UQDF,VOLUME=&VQDF,DISP=(SHR,&QDISP,KEEP)  00005400
```

```
//QRTFILE DD DSNAME=&QRT,UNIT=&UQRT,VOLUME=&VQRT,DISP=(SHR,&QDISP,KEEP) 00005500
//CHECKDD DD DSNAME=&CHKID,DISP=(&CHKST,&NRMDSP,&CHKDSP),UNIT=&UCHK,     00005600
//              VOLUME=&VCHK,SPACE=(CYL,(&CHKSP,1))                      00005700
//STAT DD UNIT=&STG,SPACE=(CYL,(2,1))                                    00005800
//TRANST DD DSN=&TRANS,VOL=&VTRANS,UNIT=&UTRANS,                         00005900
//        DISP=&TDISP,SPACE=(TRK,1)                                      00006000
```

```
//XOPSD  PROC  A=A,B=B,BSZFILE=,BSZFIL1=,BSZFIL2=,                          00000100
//             CL=',',CL1=',',CL2=',',                                      00000200
//             ISAM='DUMMY.FILE',ISAM1='DUMMY.FILE',ISAM2='DUMMY.FILE',     00000300
//             JOBLIB='FFS.JOBLIB',JOBMAC='FFS.JOBMACRO',                   00000400
//             LAB=SL,                                                      00000500
//             LIB='DUMMY.FILE',LIB1='DUMMY.FILE',LIB2='DUMMY.FILE',        00000600
//             LIBDISP=SHR,                                                 00000700
//             PTFJOBL='PTF.JOBLIB',PTFJOBM='PTF.JOBMACRC',                 00000800
//             RGN=100K,                                                    00000900
//             SAM='DUMMY.FILE',SAM1='DUMMY.FILE',SAM2='DUMMY.FILE',        00001000
//             SDISP=SHR,SOURCL='DUMMY.FILE',STG=NIPW,                      00001100
//             TDISP=MOD,TRANS='&&TRANS',                                   00001200
//             UISAM='(2314,P)',                                            00001300
//             ULIB=2314,ULIB1=2314,ULIB2=2314,                             00001400
//             USAM='(TAPE9,,DEFER)',USOURCL=2314,                          00001500
//             UTRANS=NIPW,                                                 00001600
//             VISAM=,VISAM1=,VISAM2=,                                      00001700
//             VLIB=,VLIB1=,VLIB2=,                                         00001800
//             VSAM=,VSAM1=,VSAM2=,                                         00001900
//             VSOURCL=,VTRANS=                                             00002000
//*                                                                         00002100
//*   CHARLES W. HICKISCH  MAJOR,USA  PROJ CODE=763NIPS  BRANCH=431        00002200
//*   DATE=MARCH 1, 1974                                                   00002300
//*                                                                         00002400
//OP EXEC PGM=OP,REGION=&RGN                                                00002500
//STEPLIB  DD  DSN=&PTFJOBL,DISP=SHR                                        00002600
//         DD  DSN=&JOBLIB,DISP=SHR                                         00002700
//DATAFILE DD DSNAME=&ISAM,DISP=SHR,UNIT=&UISAM,VOLUME=&VISAM               00002800
//DATAFIL1 DD DSNAME=&ISAM1,DISP=SHR,UNIT=&UISAM,VOLUME=&VISAM1             00002900
//DATAFIL2 DD DSNAME=&ISAM2,DISP=SHR,UNIT=&UISAM,VOLUME=&VISAM2             00003000
//SAMFILE DD DSNAME=&SAM.S,DISP=SHR,UNIT=&USAM,VOLUME=&VSAM,               X00003010
//             LABEL=(,&LAB),DCB=BLKSIZE=&BSZFILE                           00003100
//SAMFILE1 DD DSNAME=&SAM1.S,DISP=SHR,UNIT=&USAM,VOLUME=&VSAM1,            X00003200
//             LABEL=(,&LAB),DCB=BLKSIZE=&BSZFIL1                           00003300
//SAMFILE2  DD DSNAME=&SAM2.S,DISP=SHR,UNIT=&USAM,VOLUME=&VSAM2,           *00003400
//             LABEL=(,&LAB),DCB=BLKSIZE=&BSZFIL2                           00003500
//DLIB  DD DSNAME=&LIB.L,DISP=&LIBDISP,VOLUME=&VLIB,UNIT=&ULIB              00003600
//TLIB  DD DSNAME=&LIB1.L,DISP=SHR,VOLUME=&VLIB1,UNIT=&ULIB1                00003700
//SOURCPRT DD SYSOUT=(&A,&CL)                                               00003800
//SOURCLIB  DD  DISP=&SDISP,DSN=&SOURCL.L,VOL=&VSOURCL,UNIT=&USOURCL        00003900
//SYSPRINT DD SYSOUT=(&A,&CL),DCB=(LRECL=133,BLKSIZE=665)                   00004000
//SYSOUT   DD SYSOUT=(&A,&CL)                                               00004100
//SYSUDUMP DD SYSOUT=(&A,&CL1)                                              00004200
//SYSUT1   DD UNIT=&STG,SPACE=(TRK,(200,20))                                00004300
//SYSUT2   DD UNIT=(&STG,SEP=SYSUT1),SPACE=(TRK,(200,20))                   00004400
//SYSUT3   DD UNIT=(&STG,SEP=(SYSUT1,SYSUT2)),SPACE=(TRK,(200,20))          00004500
//SYSPUNCH DD UNIT=&STG,SPACE=(TRK,(50,10)),                                X00004600
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=400,BUFNO=5)                  00004700
//SYSLIB   DD  DSN=&PTFJOBM,DISP=SHR                                        00004800
//             DD  DSN=&JOBMAC,DISP=SHR                                     00004900
//SYSLMOD  DD UNIT=&STG,SPACE=(TRK,(50,25,5)),DCB=SYS1.LINKLIB,             X00005000
//             LABEL=RETPD=0000                                             00005100
//SYSTPRT DD SYSOUT=(&A,&CL),DCB=(RECFM=FB,LRECL=121,BLKSIZE=605)           00005200
//OPIWCF   DD UNIT=&STG,SPACE=(TRK,(2,1)),DCB=(BLKSIZE=400,LRECL=80,        *00005300
//             RECFM=FB)                                                    00005400
//OPSTST   DD UNIT=&STG,SPACE=(TRK,(10,2))                                  00005500
//OPCREATE DD UNIT=&STG,SPACE=(TRK,(1,1))                                   00005600
```

```
//OPCOMREC DD UNIT=&STG,SPACE=(TRK,(1,1))                                00005700
//INSTS    DD UNIT=&STG,SPACE=(TRK,(50,5))                             00005800
//LCF      DD UNIT=&STG,SPACE=(TRK,(10,5))                             00005900
//LCT      DD UNIT=&STG,SPACE=(TRK,(10,5))                             00006000
//LITSTR   DD UNIT=&STG,SPACE=(TRK,(50,5))                             00006100
//NAMDEF   DD DSNAME=*.SYSUT1,VOLUME=REF=*.SYSUT1,DISP=(OLD,PASS)      00006200
//LITAB    DD DSNAME=*.SYSUT2,VOLUME=REF=*.SYSUT2,DISP=(OLD,PASS)      00006300
//RGTAB    DD DSNAME=*.SYSUT3,VOLUME=REF=*.SYSUT3,DISP=(OLD,PASS)      00006400
//CGC      DD UNIT=&STG,SPACE=(TRK,(1,1))                              00006500
//ERRTAB   DD UNIT=&STG,SPACE=(TRK,(8,5))                              00006600
//SUBTAB   DD UNIT=&STG,SPACE=(TRK,(1,1))                              00006700
//DCTNY    DD DSNAME=*.SYSPUNCH,VOLUME=REF=*.SYSPUNCH,DISP=(OLD,PASS)  00006800
//OPLGOGO  DD DSNAME=*.SYSUT1,VOLUME=REF=*.SYSUT1,DISP=(OLD,PASS)      00006900
//OPLEXEX  DD DSNAME=*.SYSUT2,VOLUME=REF=*.SYSUT2,DISP=(OLD,PASS)      00007000
//OPLEXDIR DD DSNAME=*.SYSUT3,VOLUME=REF=*.SYSUT3,DISP=(OLD,PASS)      00007100
//OPLINE   DD SYSOUT=(&A,&CL2),DCB=LRECL=133                           00007200
//OPPUNCH    DD SYSOUT=&B                                              00007300
//SLIB DD DSNAME=*.SYSLMOD,VOLUME=REF=*.SYSLMOD,DISP=(OLD,PASS)        00007400
// DD DSNAME=*.DLIB,DISP=SHR,UNIT=&ULIB,VOLUME=&VLIB                   00007500
// DD DSNAME=*.TLIB,DISP=SHR,UNIT=&ULIB1,VOLUME=&VLIB1                 00007600
//    DD DSNAME=&LIB2.L,DISP=SHR,UNIT=&ULIB2,VOLUME=&VLIB2             00007700
//      DD  DSN=&PTFJOBL,DISP=SHR                                      00007800
// DD DSNAME=&JOBLIB,DISP=SHR                                          00007900
//SYSAIN DD DSNAME=*.LITSTR,VOLUME=REF=*.LITSTR,DISP=(OLD,PASS),       00008000
//              DCB=(RECFM=FB,LRECL=80,BLKSIZE=400,BUFNO=5)            00008100
//  DD DSNAME=*.INSTS,VOLUME=REF=*.INSTS,DISP=(OLD,PASS),              00008200
//              DCB=(RECFM=FB,LRECL=80,BLKSIZE=400,BUFNO=5)            00008300
//SYSLIN DD DSNAME=*.LCF,VOLUME=REF=*.LCF,DISP=(OLD,PASS)              00008400
//SYSTIN DD DSNAME=*.LCT,VOLUME=REF=*.LCT,DISP=(OLD,PASS)              00008500
//LOADR  DD DSNAME=*.SYSPUNCH,VOLUME=REF=*.SYSPUNCH,DISP=(OLD,PASS),   'X00008600
//              DCB=(RECFM=FB,LRECL=80,BLKSIZE=400)                    00008700
//STAT DD DSN=*.INSTS,VOL=REF=*.INSTS,DISP=(OLD,PASS)                  00008800
//TRANST DD DSN=&TRANS,VOL=&VTRANS,UNIT=&UTRANS,                       00008900
//       DISP=&TDISP,SPACE=(TRK,1)                                     00009000
```

```
//XOPSDEX PROC A=A,B=B,BSZFILE=,BSZFIL1=,BSZFIL2=,                      00000100
//            CL=',',CL1=',',CL2=',',                                   00000200
//            CHKDSP=DELETE,CHKID='&&CHKID',CHKSP=0,CHKST=NEW,          00000300
//            ISAM='DUMMY.FILE',ISAM1='DUMMY.FILE',ISAM2='DUMMY.FILE',  00000400
//            JOBLIB='FFS.JOBLIB',                                      00000500
//            LAB=SL,                                                   00000600
//            LIB='DUMMY.FILE',LIB1='DUMMY.FILE',LIB2='DUMMY.FILE',     00000700
//            LIBDISP=SHR,                                              00000800
//            NRMDSP=DELETE,                                            00000900
//            PTFJOBL='PTF.JOBLIB',                                     00000910
//            RGN=100K,                                                 00001000
//            SAM='DUMMY.FILE',SAM1='DUMMY.FILE',SAM2='DUMMY.FILE',     00001100
//            STG=NIPW,                                                 00001200
//            TDISP=MOD,TRANS='&&TRANS',                                00001300
//            UCHK=NIPW,                                                00001400
//            UISAM='(2314,P)',                                         00001500
//            ULIB=2314,ULIB1=2314,ULIB2=2314,                          00001600
//            USAM='(TAPE9,,DEFER)',                                    00001700
//            UTRANS=NIPW,                                              00001800
//            VISAM=,VISAM1=,VISAM2=,                                   00001900
//            VLIB=,VLIB1=,VLIB2=,                                      00002000
//            VSAM=,VSAM1=,VSAM2=,                                      00002100
//            VCHK=,VTRANS=                                             00002200
//*                                                                     00002300
//*   CHARLES W. HICKISCH  MAJOR,USA  PROJ CODE=763NIPS  BRANCH=431     00002400
//*   DATE=MARCH 1, 1974                                                00002500
//*                                                                     00002600
//OP EXEC PGM=OP,REGION=&RGN                                            00002700
//STEPLIB  DD  DSN=&PTFJOBL,DISP=SHR                                    00002800
//         DD  DSN=&JOBLIB,DISP=SHR                                     00002810
//DATAFILE DD DSNAME=&ISAM,DISP=SHR,UNIT=&UISAM,VOLUME=&VISAM           00002900
//DATAFIL1 DD DSNAME=&ISAM1,DISP=SHR,UNIT=&UISAM,VOLUME=&VISAM1         00003000
//DATAFIL2 DD DSNAME=&ISAM2,DISP=SHR,UNIT=&UISAM,VOLUME=&VISAM2         00003100
//SAMFILE  DD DSNAME=&SAM.S,DISP=SHR,UNIT=&USAM,VOLUME=&VSAM,          X00003200
//            LABEL=(,&LAB),DCB=BLKSIZE=&BSZFILE                        00003300
//SAMFILE1 DD DSNAME=&SAM1.S,DISP=SHR,UNIT=&USAM,VOLUME=&VSAM1,         X00003400
//            LABEL=(,&LAB),DCB=BLKSIZE=&BSZFIL1                        00003500
//SAMFILE2  DD DSNAME=&SAM2.S,DISP=SHR,UNIT=&USAM,VOLUME=&VSAM2,        *00003600
//            LABEL=(,&LAB),DCB=BLKSIZE=&BSZFIL2                        00003700
//DLIB  DD DSNAME=&LIB.L,DISP=&LIBDISP,VOLUME=&VLIB,UNIT=&ULIB          00003800
//TLIB  DD DSNAME=&LIB1.L,DISP=SHR,VOLUME=&VLIB1,UNIT=&ULIB1            00003900
//SOURCPRT DD SYSOUT=(&A,&CL)                                           00004000
//PRINTER  DD SYSOUT=(&A,&CL)                                           00004100
//SYSOUT   DD SYSOUT=(&A,&CL)                                           00004200
//SYSPRINT DD SYSOUT=(&A,&CL),DCB=LRECL=133                             00004300
//SYSUDUMP DD SYSOUT=(&A,&CL1)                                          00004400
//OPLINE  DD SYSOUT=(&A,&CL2),DCB=LRECL=133                             00004500
//OPIWCF   DD UNIT=&STG,SPACE=(TRK,(2,1)),DCB=(BLKSIZE=400,LRECL=80,    *00004600
//            RECFM=FB),DSNAME=&CHKID.A,DISP=(,&NRMDSP,&CHKDSP)         00004700
//OPCOMREC DD UNIT=&STG,SPACE=(TRK,(1,1)),                              X00004800
//            DISP=(,&NRMDSP,&CHKDSP),DSNAME=&CHKID.B                   00004900
//OPLGOGO  DD UNIT=&STG,SPACE=(TRK,(20,1)),                             X00005000
//            DISP=(,&NRMDSP,&CHKDSP),DSNAME=&CHKID.C                   00005100
//OPLEXEX  DD UNIT=(&STG,SEP=OPLGOGO),SPACE=(TRK,(20,1)),               X00005200
//            DISP=(,&NRMDSP,&CHKDSP),DSNAME=&CHKID.D                   00005300
//OPLEXDIR DD UNIT=(&STG,SEP=(OPLGOGO,OPLEXEX)),SPACE=(TRK,(20,1)),     X00005400
//            DISP=(,&NRMDSP,&CHKDSP),DSNAME=&CHKID.E                   00005500
```

```
//OPPUNCH   DD SYSOUT=&B                                                00005600
//SLIB DD DSNAME=*.DLIB,DISP=SHR,UNIT=&ULIB,VOLUME=&VLIB                00005700
// DD DSNAME=*.TLIB,DISP=SHR,UNIT=&ULIB1,VOLUME=&VLIB1                  00005800
//    DD DSNAME=&LIB2.L,DISP=SHR,UNIT=&ULIB2,VOLUME=&VLIB2
//      DD  DSN=&PTFJOBL,DISP=SHR
// DD DSNAME=&JOBLIB,DISP=SHR
//CHECKDD DD DSNAME=&CHKID,DISP=(&CHKST,&NRMDSP,&CHKDSP),UNIT=&UCHK,    00006100
//              VOLUME=&VCHK,SPACE=(CYL,(&CHKSP,1))                     00006200
//STAT DD UNIT=&STG,SPACE=(CYL,(2,1))                                   00006300
//TRANST DD DSN=&TRANS,VOL=&VTRANS,UNIT=&UTRANS,                        00006400
//        DISP=&TDISP,SPACE=(TRK,1)                                     00006500
```

NIPS 360 FFS PROCEDURES

```
//XQRTQDF PROC A=A,BSZNEWF=1004,CL=',',CL1=',',DEN=,                    00000100
//             JOBLIB='FFS.JOBLIB',                                     00000200
//             LAB=,QDF='&&QDFILE',QDISP=PASS,                          00000300
//             PTFJOBL='PTF.JOBLIB',                                    00000310
//             QRT='&&QRTFILE',RGN=60K,SAM='DUMMY',TRCH=,               00000400
//             UQDF=NIPW,UQRT=NIPW,USAM='(TAPE9,,DEFER)',               00000500
//             VQDF=,VQRT=,VSAM=                                        00000600
//*                                                                     00000700
//*   CHARLES W. HICKISCH  MAJOR,USA  PRCJ CODE=763NIPS  BRANCH=431    00000800
//*   DATE=MARCH 1, 1974                                                00000900
//*                                                                     00001000
//QRTQDF EXEC PGM=UTQRTQDF,REGION=&RGN                                  00001100
//STEPLIB  DD  DSN=&PTFJOBL,DISP=SHR                                    00001200
//         DD  DSN=&JOBLIB,DISP=SHR                                     00001210
//SYSPRINT DD SYSOUT=(&A,&CL)                                           00001300
//SYSUDUMP DD SYSOUT=(&A,&CL1)                                          00001400
//QRTFILE DD DSNAME=&QRT,UNIT=&UQRT,VOLUME=&VQRT,DISP=(SHR,&QDISP,KEEP) 00001500
//QDFILE DD DSNAME=&QDF,UNIT=&UQDF,VOLUME=&VQDF,DISP=(SHR,&QDISP,KEEP)  00001600
//SAMFILE DD DSNAME=&SAM.S,UNIT=&USAM,VOLUME=&VSAM,DISP=(,KEEP),       *00001700
//             DCB=(TRTCH=&TRCH,DEN=&DEN,BLKSIZE=&BSZNEWF),             00001800
//             LABEL=(,&LAB)                                            00001900
```

```
//XQUIP  PROC  A=A,                                                  00000100
//             CL=',',CL1=',',CL2=',',                               00000200
//             ISAM1='DUMMY.FILE',ISAM2='DUMMY.FILE',                00000300
//             JOBLIB='FFS.JOBLIB',                                  00000400
//             LIB='DUMMY.FILE',LIB1='DUMMY.FILE',                   00000500
//             PTFJOBL='PTF.JOBLIB',                                 00000510
//             QRT='&&QRTFILE',                                      00000600
//             QDF='&&QDFILE',QDISP=PASS,                            00000700
//             RGN=92K,SDISP=SHR,                                    00000800
//             SOURCL='DUMMY.FILE',                                  00000900
//             STG=NIPW,TDISP=MOD,TRANS='&&TRANS',                   00001000
//             UISAM1='(2314,P)',UISAM2='(2314,P)',                  00001100
//             ULIB=2314,ULIB1=2314,                                 00001200
//             UQDF=NIPW,                                            00001300
//             UQRT=NIPW,                                            00001400
//             USOURCL=2314,UTRANS=NIPW,                             00001500
//             VISAM1=,VISAM2=,                                      00001600
//             VLIB=,VLIB1=,                                         00001700
//             VQDF=,                                                00001800
//             VQRT=,                                                00001900
//             VSOURCL=,VTRANS=                                      00002000
//*                                                                  00002100
//*   CHARLES W. HICKISCH  MAJOR,USA  PROJ CODE=763NIPS BRANCH=431   00002200
//*   DATE=MARCH 1, 1974                                             00002300
//*                                                                  00002400
//QUIP EXEC PGM=IPBASE,REGION=&RGN                                   00002500
//STEPLIB  DD  DSN=&PTFJOBL,DISP=SHR                                 00002600
//         DD  DSN=&JOBLIB,DISP=SHR                                  00002610
//SYSUT1  DD  SPACE=(TRK,(50,10)),UNIT=&STG                          00002700
//SYSUT2  DD  SPACE=(TRK,(1,1)),UNIT=&STG                            00002800
//SYSUT3  DD  SPACE=(TRK,(10,10)),UNIT=(&STG,SEP=SYSUT1)             00002900
//SLIB       DD DSNAME=&LIB.L,UNIT=&ULIB,VOLUME=&VLIB,DISP=SHR       00003000
//           DD DSNAME=&LIB1.L,UNIT=&ULIB1,VOLUME=&VLIB1,DISP=SHR    00003100
//           DD DSNAME=&JOBLIB,DISP=SHR                              00003200
//QDFILE DD DSNAME=&QDF,UNIT=&UQDF,VOLUME=&VQDF,DISP=(SHR,&QDISP,KEEP) 00003300
//QRTFILE DD DSNAME=&QRT,UNIT=&UQRT,VOLUME=&VQRT,DISP=(SHR,&QDISP,KEEP) 00003400
//DATAFIL1 DD DSN=&ISAM1,UNIT=&UISAM1,VOL=&VISAM1,DISP=SHR           00003500
//DATAFIL2 DD DSN=&ISAM2,UNIT=&UISAM2,VOL=&VISAM2,DISP=SHR           00003600
//SOURCLIB  DD  DISP=&SDISP,DSN=&SOURCL.L,VOL=&VSOURCL,UNIT=&USOURCL 00003700
//SOURCPRT DD SYSOUT=(&A,&CL)                                        00003800
//SYSUDUMP DD SYSOUT=(&A,&CL1)                                       00004100
//SYSPRINT DD SYSOUT=(&A,&CL2)                                       00004200
//STAT  DD DSN=*.SYSUT1,VOL=REF=*.SYSUT1,DISP=(OLD,PASS)             00004300
//TRANST DD DSN=&TRANS,VOL=&VTRANS,UNIT=&UTRANS,DISP=&TDISP,         00004400
//         SPACE=(TRK,1)                                             00004500
```

```
//XQUIPSD PROC A=A,BSZFILE=,
//             CL=',',CL1=',',CL2=',',
//             FFT='DUMMY.FILE',
//             FFT1='DUMMY.FILE',FFT2='DUMMY.FILE',
//             ISAM='DUMMY.FILE',
//             ISAM1='DUMMY.FILE',ISAM2='DUMMY.FILE',
//             JOBLIB='FFS.JOBLIB',
//             LAB=SL,
//             LIB='DUMMY.FILE',LIB1='DUMMY.FILE',
//             PTFJOBL='PTF.JOBLIB',
//             QDFSP='(5,1)',
//             QRTSP='(2,1)',
//             RGN=92K,
//             SAM='DUMMY.FILE',SDISP=SHR,
//             SORTSP='(2,1)',
//             SOURCL='DUMMY.FILE',
//             STG=NIPW,TDISP=MOD,TRANS='&&TRANS',
//             UFFT='(2314,P)',
//             UFFT1='(2314,P)',UFFT2='(2314,P)',
//             UISAM='(2314,P)',
//             UISAM1='(2314,P)',UISAM2='(2314,P)',
//             ULIB=2314,ULIB1=2314,
//             USAM='(TAPE9,,DEFER)',
//             USOURCL=2314,UTRANS=NIPW,
//           VFFT=,VFFT1=,VFFT2=,
//             VISAM=,
//             VISAM1=,VISAM2=,
//             VLIB=,VLIB1=,
//             VSAM=,
//             VSOURCL=,VTRANS=,
//             XINDEX='DUMMY.FILE',
//             XUNIT=2314,XVOL=
//*
//*   CHARLES W. HICKISCH  MAJOR,USA  PROJ CODE=763NIPS  BRANCH=431
//*   DATE=MARCH 1, A974
//*
//QUIP EXEC PGM=IPBASE,REGION=&RGN
//STEPLIB  DD  DSN=&PTFJOBL,DISP=SHR
//         DD  DSN=&JOBLIB,DISP=SHR
//SYSUT1  DD  SPACE=(TRK,(50,10)),UNIT=&STG
//SYSUT2  DD  SPACE=(TRK,(1,1)),UNIT=&STG
//SYSUT3  DD  SPACE=(TRK,(10,10)),UNIT=(&STG,SEP=SYSUT1)
//SYSUT4  DD  SPACE=(CYL,(0,0)),UNIT=&STG
//SLIB       DD DSNAME=&LIB.L,UNIT=&ULIB,VOLUME=&VLIB,DISP=SHR
//           DD DSNAME=&LIB1.L,UNIT=&ULIB1,VCLUME=&VLIB1,DISP=SHR
//           DD DSNAME=&JOBLIB,DISP=SHR
//DATAFILE   DD DISP=SHR,DSNAME=&ISAM,UNIT=&UISAM,VOLUME=&VISAM
//DATAFIL1 DD DSN=&ISAM1,UNIT=&UISAM1,VOL=&VISAM1,DISP=SHR
//DATAFIL2 DD DSN=&ISAM2,UNIT=&UISAM2,VOL=&VISAM2,DISP=SHR
//SAMFILE    DD DISP=SHR,DSNAME=&SAM.S,UNIT=&USAM,VOLUME=&VSAM,
//             LABEL=(,&LAB),DCB=(RECFM=VB,LRECL=1000,BLKSIZE=&BSZFILE)
//FFT    DD    DSN=&FFT,
//             UNIT=&UFFT,
//             VOLUME=&VFFT,
//             DISP=SHR
//FFT1   DD    DSN=&FFT1,
//             UNIT=&UFFT1,
```

```
//              VOLUME=&VFFT1,
//              DISP=SHR
//FFT2    DD    DSN=&FFT2,
//              UNIT=&UFFT2,
//              VOLUME=&VFFT2,
//              DISP=SHR
//SOURCLIB  DD  DISP=&SDISP,DSN=&SOURCL.L,VCL=&VSOURCL,UNIT=&USOURCL      00004800
//SOURCPRT DD SYSOUT=(&A,&CL)                                            00004900
//SYSUDUMP DD SYSOUT=(&A,&CL1)                                           00005000
//SYSPRINT DD SYSOUT=(&A,&CL2)                                           00005100
//PB        DD  UNIT=&STG,SPACE=(CYL,&QDFSP)                             00005300
//KEY       DD  UNIT=&STG,SPACE=(CYL,&QRTSP)                             00005400
//SORTWK01  DD  UNIT=&STG,SPACE=(CYL,&SCRTSP)                            00005500
//SORTWK02  DD  UNIT=&STG,SPACE=(CYL,&SCRTSP)                            00005600
//SORTWK03  DD  UNIT=&STG,SPACE=(CYL,&SCRTSP)                            00005700
//SORTWK04  DD  UNIT=&STG,SPACE=(CYL,&SORTSP)                            00005800
//STAT  DD DSN=*.SYSUT1,VOL=REF=*.SYSUT1,DISP=(OLD,PASS)                 00005900
//TRANST DD DSN=&TRANS,VOL=&VTRANS,UNIT=&UTRANS,DISP=&TDISP,             00006000
//          SPACE=(TRK,1)                                                00006100
//XINDEX  DD  DSN=&XINDEX.X,UNIT=&XUNIT,VCL=&XVOL,DISP=SHR               00006200
```

```
//XRASP PROC A=A,BSZFILE=,BSZFIL1=,BSZFIL2=,                          00000100
//          CL=',',CL1=',',ISAM='DUMMY.FILE',ISAM1='DUMMY.FILE',      00000200
//          ISAM2='DUMMY.FILE',JOBLIB='FFS.JOBLIB',                   00000300
//          JOBMAC='FFS.JOBMACRO',LAB=SL,LIB='DUMMY.FILE',            00000400
//          LIBDISP=SHR,LIB1='DUMMY.FILE',LIB2='DUMMY.FILE',          00000500
//            PTFJOBL='PTF.JOBLIB',PTFJOBM='PTF.JOBMACRO',            00000510
//          QDF='&&QDFILE',QDFSP='(50,10)',QDISP=PASS,QRT='&&QRTFILE', 00000600
//          QRTSP='(10,10)',RGN=100K,SAM='DUMMY.FILE',                00000700
//          SAM1='DUMMY.FILE',SAM2='DUMMY.FILE',SDISP=SHR,SORTSP=20,  00000800
//          SOURCL='DUMMY.FILE',STG=NIPW,TDISP=MOD,TRANS='&&TRANS',   00000900
//          UISAM='(2314,P)',ULIB=2314,ULIB1=2314,ULIB2=2314,         00001000
//          UQDF=NIPW,UQRT=NIPW,USAM='(TAPE9,,DEFER)',USOURCL=2314,   00001100
//          UTRANS=NIPW,VISAM=,VISAM1=,VISAM2=,VLIB=,VLIB1=,VLIB2=,   00001200
//          VQDF=,VQRT=,VSAM=,VSAM1=,VSAM2=,VSOURCL=,VTRANS=,         00001300
//          XDISP=SHR,XINDEX='DUMMY.FILE',XINDEX1='DUMMY.FILE',       00001400
//          XINDEX2='DUMMY.FILE',XUNIT=2314,XVOL=,XVOL1=,XVOL2=       00001500
//*                                                                   00001600
//*   CHARLES W. HICKISCH  MAJOR,USA  PRCJ CODE=763NIPS BRANCH=431    00001700
//*   DATE=MARCH 1, 1974                                              00001800
//*                                                                   00001900
//RASP EXEC PGM=RSEXEC,REGION=&RGN                                    00002000
//STEPLIB DD DISP=SHR,DSN=&PTFJOBL                                    00002100
//          DD  DISP=SHR,DSN=&JOBLIB                                  00002110
//DATAFILE DD DSNAME=&ISAM,DISP=SHR,UNIT=&UISAM,VOLUME=&VISAM         00002200
//DATAFIL1 DD DSNAME=&ISAM1,DISP=SHR,UNIT=&UISAM,VOLUME=&VISAM1       00002300
//DATAFIL2 DD DSNAME=&ISAM2,DISP=SHR,UNIT=&UISAM,VOLUME=&VISAM2       00002400
//SAMFILE  DD DSNAME=&SAM.S,DISP=SHR,UNIT=&USAM,VOLUME=&VSAM,        X00002500
//              LABEL=(,&LAB),DCB=BLKSIZE=&BSZFILE                    00002600
//SAMFILE1 DD DSNAME=&SAM1.S,DISP=SHR,UNIT=&USAM,VOLUME=&VSAM1,      X00002700
//              LABEL=(,&LAB),DCB=BLKSIZE=&BSZFIL1                    00002800
//SAMFILE2  DD DSNAME=&SAM2.S,DISP=SHR,UNIT=&USAM,VOLUME=&VSAM2,     *00002900
//              LABEL=(,&LAB),DCB=BLKSIZE=&BSZFIL2                    00003000
//QDFILE    DD DSNAME=&QDF,UNIT=(&UQDF,SEP=DATAFILE),VOLUME=&VQDF,   X00003100
//              SPACE=(CYL,&QDFSP),DISP=(,&QDISP,DELETE)              00003200
//SORTOUT   DD DSNAME=&QRT,UNIT=(&UQRT,SEP=(DATAFILE,QDFILE)),       X00003300
//              VOLUME=&VQRT,SPACE=(CYL,&QRTSP),DISP=(,&QDISP,DELETE), X00003400
//              DCB=(RECFM=FB,LRECL=1011,BLKSIZE=1015)                00003500
//SORTIN DD DSNAME=*.SORTOUT,VOLUME=REF=*.SORTOUT,DISP=(OLD,PASS)     00003600
//SORTWK01 DD  SPACE=(CYL,(&SORTSP),,CONTIG),UNIT=(&STG,SEP=SORTIN)   00003700
//SORTWK02 DD  SPACE=(CYL,(&SORTSP),,CONTIG),                        X00003800
//              UNIT=(&STG,SEP=SORTWK01),SEP=SORTWK01                 00003900
//SORTWK03 DD  SPACE=(CYL,(&SORTSP),,CONTIG),                        X00004000
//              UNIT=(&STG,SEP=SORTWK02),SEP=SORTWK02                 00004100
//SORTWK04 DD  SPACE=(CYL,(&SORTSP),,CONTIG),                        X00004200
//              UNIT=(&STG,SEP=(SORTWK01,SORTWK03)),                  X00004300
//              SEP=(SORTWK01,SORTWK03)                               00004400
//SORTWK05 DD  SPACE=(CYL,(&SORTSP),,CONTIG),                        X00004500
//              UNIT=(&STG,SEP=(SORTWK02,SORTWK04)),                  X00004600
//              SEP=(SORTWK02,SORTWK04)                               00004700
//SORTWK06 DD  SPACE=(CYL,(&SORTSP),,CONTIG),                        X00004800
//              UNIT=(&STG,SEP=(SORTWK01,SORTWK03,SORTWK05)),         X00004900
//              SEP=(SORTWK01,SORTWK03,SORTWK05)                      00005000
//SYSLMOD  DD UNIT=&STG,SPACE=(TRK,(40,20,20)),DCB=SYS1.LINKLIB,     X00005100
//              LABEL=EXPDT=66366                                     00005200
//DLIB  DD DSNAME=&LIB.L,DISP=&LIBDISP,VOLUME=&VLIB,UNIT=&ULIB        00005300
//TLIB  DD DSNAME=&LIB1.L,DISP=SHR,VOLUME=&VLIB1,UNIT=&ULIB1          00005400
//SLIB  DD DSNAME=*.SYSLMOD,DISP=(OLD,PASS),VOLUME=REF=*.SYSLMOD      00005500
```

```
// DD DSNAME=*.DLIB,DISP=SHR,UNIT=&ULIB,VOLUME=&VLIB                    00005600
// DD DSNAME=*.TLIB,DISP=SHR,UNIT=&ULIB1,VOLUME=&VLIB1                  00005700
//    DD DSNAME=&LIB2.L,DISP=SHR,UNIT=&ULIB2,VOLUME=&VLIB2              00005800
// DD DSNAME=&JOBLIB,DISP=SHR                                           00005900
//SORTLIB DD    DISP=SHR,DSNAME=SYS1.SORTLIB                           00006000
//SYSLIB    DD  DISP=SHR,DSN=&PTFJOBM                                  00006100
//          DD  DISP=SHR,DSN=&JOBMAC                                   00006110
//        DD    DISP=SHR,DSNAME=SYS1.MACLIB                            00006200
//DPUNCH DD DSNAME=*.SORTWK04,VOLUME=REF=*.SORTWK04,DISP=(OLD,PASS)    00006300
//SYSUT1 DD DSNAME=*.SORTWK01,VOLUME=REF=*.SORTWK01,DISP=(OLD,PASS)    00006400
//SYSUT2 DD DSNAME=*.SORTWK05,VOLUME=REF=*.SORTWK05,DISP=(OLD,PASS)    00006500
//STEREO.DD DSNAME=*.SORTWK06,VOLUME=REF=*.SORTWK06,DISP=(OLD,PASS)    00006600
//STROUTDD DD   UNIT=&STG,SPACE=(TRK,(50,20))                          00006700
//SYSUT3 DD DSNAME=*.SORTWK06,VOLUME=REF=*.SORTWK06,DISP=(OLD,PASS),   00006800
//              DCB=(BLKSIZE=80,LRECL=80)                              00006900
//ASSEMIN DD    UNIT=&STG,SPACE=(TRK,(50,5)),DCB=BLKSIZE=400,         C00007000
//              DISP=(MOD,DELETE)                                      00007100
//RSIPDS DD     UNIT=&STG,SPACE=(TRK,(3,1,1))                          00007200
//NAMETAB DD DSNAME=*.ASSEMIN,VOLUME=REF=*.ASSEMIN,DISP=(OLD,PASS)     00007300
//SOURCPRT  DD  SYSOUT=(&A,&CL)                                        00007400
//SOURCLIB  DD  DISP=&SDISP,DSN=&SOURCL.L,VOL=&VSOURCL,UNIT=&USOURCL   00007500
//SYSPRINT DD SYSOUT=(&A,&CL)                                          00007600
//SYSOUT DD SYSOUT=(&A,&CL)                                            00007700
//SYSUDUMP DD SYSOUT=(&A,&CL1)                                         00007800
//STAT DD DSN=*.ASSEMIN,VOL=REF=*.ASSEMIN,DISP=(OLD,PASS)              00007900
//TRANST DD DSN=&TRANS,VOL=&VTRANS,UNIT=&UTRANS,DISP=&TDISP,           00008000
//          SPACE=(TRK,(1))                                            00008100
//XINDEX    DD DSN=&XINDEX.X,VOL=&XVOL,UNIT=&XUNIT,DISP=&XDISP         00008200
//XINDEX1   DD  DSN=&XINDEX1.X,VOL=&XVOL1,UNIT=&XUNIT,DISP=&XDISP      00008300
//XINDEX2   DD  DSN=&XINDEX2.X,VOL=&XVOL2,UNIT=&XUNIT,DISP=&XDISP      00008400
//PRINTER DD SYSOUT=(&A,&CL)                                           00008500
//&YSDUMMY DD DUMMY                                                    00008600
```

```
//XRASPEX PROC A=A,BSZFILE=,BSZFIL1=,BSZFIL2=,                          00000100
//          CHKDSP=DELETE,CHKID='&&CHKID',CHKSP=0,CHKST=NEW,            00000200
//          CL=',',CL1=',',ISAM='DUMMY.FILE',ISAM1='DUMMY.FILE',        00000300
//          ISAM2='DUMMY.FILE',JOBLIB='FFS.JOBLIB',LAB=SL,              00000400
//          LIB='DUMMY.FILE',LIBDISP=SHR,LIB1='DUMMY.FILE',             00000500
//          LIB2='DUMMY.FILE',NRMDSP=DELETE,QDF='&&QDFILE',             00000600
//            PTFJOBL='PTF.JOBLIB',                                     00000610
//          QDFSP='(50,10)',QDISP=PASS,QRT='&&QRTFILE',                 00000700
//          QRTSP='(10,10)',RGN=100K,SAM='DUMMY.FILE',                  00000800
//          SAM1='DUMMY.FILE',SAM2='DUMMY.FILE',SORTSP=20,STG=NIPW,     00000900
//          TDISP=MOD,TRANS='&&TRANS',UCHK=NIPW,UISAM='(2314,P)',       00001000
//          ULIB=2314,ULIB1=2314,ULIB2=2314,UQDF=NIPW,UQRT=NIPW,        00001100
//          USAM='(TAPE9,,DEFER)',UTRANS=NIPW,VCHK=,VISAM=,VISAM1=,     00001200
//          VISAM2=,VLIB=,VLIB1=,VLIB2=,VQDF=,VQRT=,VSAM=,VSAM1=,       00001300
//          VSAM2=,VTRANS=,XDISP=SHR,XINDEX='DUMMY.FILE',               00001400
//          XINDEX1='DUMMY.FILE',XINDEX2='DUMMY.FILE',XUNIT=2314,       00001500
//          XVOL=,XVOL1=,XVOL2=                                         00001600
//*                                                                     00001700
//*   CHARLES W. HICKISCH  MAJOR,USA  PROJ CODE=763NIPS  BRANCH=431     00001800
//*   DATE=MARCH 1, 1974                                                00001900
//*                                                                     00002000
//RASP EXEC PGM=RSEXEC,REGION=&RGN                                      00002100
//STEPLIB  DD  DSN=&PTFJOBL,DISP=SHR                                    00002200
//         DD  DSN=&JOBLIB,DISP=SHR                                     00002210
//DATAFILE DD DSNAME=&ISAM,DISP=SHR,UNIT=&UISAM,VOLUME=&VISAM           00002300
//DATAFIL1 DD DSNAME=&ISAM1,DISP=SHR,UNIT=&UISAM,VOLUME=&VISAM1         00002400
//DATAFIL2 DD DSNAME=&ISAM2,DISP=SHR,UNIT=&UISAM,VOLUME=&VISAM2         00002500
//SAMFILE  DD DSNAME=&SAM.S,DISP=SHR,UNIT=&USAM,VOLUME=&VSAM,          X00002600
//            LABEL=(,&LAB),DCB=BLKSIZE=&BSZFILE                        00002700
//SAMFILE1 DD DSNAME=&SAM1.S,DISP=SHR,UNIT=&USAM,VOLUME=&VSAM1,         X00002800
//            LABEL=(,&LAB),DCB=BLKSIZE=&BSZFIL1                        00002900
//SAMFILE2  DD DSNAME=&SAM2.S,DISP=SHR,UNIT=&USAM,VOLUME=&VSAM2,        *00003000
//            LABEL=(,&LAB),DCB=BLKSIZE=&BSZFIL2                        00003100
//QDFILE    DD DSNAME=&QDF,UNIT=(&UQDF,SEP=DATAFILE),VCLUME=&VQDF,      X00003200
//            SPACE=(CYL,&QDFSP),DISP=(,&QDISP,&CHKDSP)                 00003300
//SORTOUT   DD DSNAME=&QRT,UNIT=(&UQRT,SEP=(DATAFILE,QDFILE)),          X00003400
//            VOLUME=&VQRT,SPACE=(CYL,&QRTSP),DISP=(,&QDISP,&CHKDSP),   X00003500
//            DCB=(RECFM=VB,LRECL=1011,BLKSIZE=1015)                    00003600
//SORTIN DD DSNAME=*.SORTOUT,VOLUME=REF=*.SORTOUT,DISP=(OLD,PASS)       00003700
//SORTWK01 DD  SPACE=(CYL,(&SORTSP),,CONTIG),UNIT=(&STG,SEP=SORTIN),    X00003800
//            DISP=(,&NRMDSP,&CHKDSP),DSNAME=&CHKID.A                   00003900
//SORTWK02 DD  SPACE=(CYL,(&SORTSP),,CONTIG),                           X00004000
//            DISP=(,&NRMDSP,&CHKDSP),DSNAME=&CHKID.B,                  X00004100
//            UNIT=(&STG,SEP=SORTWK01),SEP=SORTWK01                     00004200
//SORTWK03 DD  SPACE=(CYL,(&SORTSP),,CONTIG),                           X00004300
//            DISP=(,&NRMDSP,&CHKDSP),DSNAME=&CHKID.C,                  X00004400
//            UNIT=(&STG,SEP=SORTWK02),SEP=SORTWK02                     00004500
//SORTWK04 DD  SPACE=(CYL,(&SORTSP),,CONTIG),                           X00004600
//            DISP=(,&NRMDSP,&CHKDSP),DSNAME=&CHKID.D,                  X00004700
//            UNIT=(&STG,SEP=(SORTWK01,SORTWK03)),                      X00004800
//            SEP=(SORTWK01,SORTWK03)                                   00004900
//SORTWK05 DD  SPACE=(CYL,(&SORTSP),,CONTIG),                           X00005000
//            DISP=(,&NRMDSP,&CHKDSP),DSNAME=&CHKID.E,                  X00005100
//            UNIT=(&STG,SEP=(SORTWK02,SORTWK04)),                      X00005200
//            SEP=(SORTWK02,SORTWK04)                                   00005300
//SORTWK06 DD  SPACE=(CYL,(&SORTSP),,CONTIG),                           X00005400
//            DISP=(,&NRMDSP,&CHKDSP),DSNAME=&CHKID.F,                  X00005500
```

```
//                UNIT=(&STG,SEP=(SORTWK01,SORTWK03,SORTWK05)),          X00005600
//                SEP=(SORTWK01,SORTWK03,SORTWK05)                        00005700
//SYSLMOD DD LABEL=EXPDT=66366,SPACE=(TRK,(40,20,20)),DCB=SYS1.LINKLIB,X00005800
//                UNIT=&STG,DSNAME=&CHKID.G,DISP=(,&NRMDSP,&CHKDSP)       00005900
//DLIB  DD DSNAME=&LIB.L,DISP=&LIBDISP,VOLUME=&VLIB,UNIT=&ULIB           00006000
//TLIB  DD DSNAME=&LIB1.L,DISP=SHR,VOLUME=&VLIB1,UNIT=&ULIB1             00006100
//SLIB  DD DSNAME=*.SYSLMOD,DISP=(OLD,PASS),VOLUME=REF=*.SYSLMOD         00006200
// DD DSNAME=*.DLIB,DISP=SHR,UNIT=&ULIB,VOLUME=&VLIB                      00006300
// DD DSNAME=*.TLIB,DISP=SHR,UNIT=&ULIB1,VOLUME=&VLIB1                    00006400
//   DD DSNAME=&LIB2.L,DISP=SHR,UNIT=&ULIB2,VOLUME=&VLIB2                 00006500
// DD DSNAME=&JOBLIB,DISP=SHR                                             00006600
//SORTLIB DD   DISP=SHR,DSNAME=SYS1.SORTLIB                              00006700
//SYSUT1 DD DSNAME=*.SORTWK01,VOLUME=REF=*.SORTWK01,DISP=(OLD,PASS)      00006800
//STEREO DD DUMMY                                                        00006900
//STROUTDD DD   UNIT=&STG,SPACE=(TRK,(50,20)),                          X00007000
//                DISP=(,&NRMDSP,&CHKDSP),DSNAME=&CHKID.H                00007100
//RSIPDS DD     UNIT=&STG,SPACE=(TRK,(3,1,1)),                          X00007200
//                DISP=(,&NRMDSP,&CHKDSP),DSNAME=&CHKID.I                00007300
//NAMETAB DD UNIT=&STG,DCB=BLKSIZE=400,SPACE=(TRK,(5,1)),               X00007400
//                DISP=(,&NRMDSP,&CHKDSP),DSNAME=&CHKID.J                00007500
//SYSOUT   DD SYSOUT=(&A,&CL)                                            00007600
//SYSPRINT DD SYSOUT=(&A,&CL)                                            00007700
//SYSUDUMP DD SYSOUT=(&A,&CL1)                                           00007800
//CHECKDD DD DSNAME=&CHKID,DISP=(&CHKST,&NRMDSP,&CHKDSP),UNIT=&UCHK,     00007900
//                VOLUME=&VCHK,SPACE=(CYL,(&CHKSP,1))                    00008000
//STAT DD UNIT=&STG,SPACE=(CYL,(2,1))                                    00008100
//TRANST DD DSN=&TRANS,VOL=&VTRANS,UNIT=&UTRANS,DISP=&TDISP,             00008200
//        SPACE=(TRK,(1))                                                00008300
//XINDEX   DD DSN=&XINDEX.X,VOL=&XVOL,UNIT=&XUNIT,DISP=&XDISP            00008400
//XINDEX1  DD  DSN=&XINDEX1.X,VOL=&XVOL1,UNIT=&XUNIT,DISP=&XDISP         00008500
//XINDEX2  DD  DSN=&XINDEX2.X,VOL=&XVOL2,UNIT=&XUNIT,DISP=&XDISP         00008600
//SOURCPRT     DD   SYSOUT=(&A,&CL)                                      00008700
```

NIPS 360 FFS PROCEDURES

```
//XRESTANS      PROC  A=A,CL=',',LAB=BLP,QDF='&&QDFILE',                 00000010
//              QDFSP='(50,10)',QDISP=PASS,QRT='&&QRTFILE',              00000020
//              QRTSP='(10,10)',UQDF=NIPW,UQRT=NIPW,                     00000030
//              VQDF=,VQRT=,VTAPE=                                       00000040
//*                                                                      00000050
//*   CHARLES W. HICKISCH  MAJOR,USA  PROJ CODE=763NIPS  BRANCH=431     00000060
//*   DATE=MARCH 1, 1974                                                 00000070
//*                                                                      00000080
//RSTQRT  EXEC  PGM=IEBGENER                                             00000090
//SYSUT1 DD DSNAME=QRTXXX,UNIT=TAPE9,VOLUME=&VTAPE,DISP=(OLD,PASS),     X00000100
//              LABEL=(,&LAB),DCB=(RECFM=VB,LRECL=1000,BLKSIZE=1004)     00000110
//SYSUT2 DD  DSNAME=&QRT,UNIT=&UQRT,VOLUME=&VQRT,DISP=(,&QDISP,DELETE),X00000120
//              SPACE=(CYL,&QRTSP),                                     X00000130
//              DCB=(RECFM=VB,LRECL=1000,BLKSIZE=1004)                   00000140
//SYSPRINT DD SYSOUT=(&A,&CL)                                            00000150
//SYSIN  DD  DUMMY                                                       00000160
//RSTQDF  EXEC  PGM=IEBGENER,COND=(0,NE,RSTQRT)                          00000170
//SYSUT1 DD DSNAME=QDFXXX,VOLUME=REF=*.RSTQRT.SYSUT1,                   X00000180
//              DISP=(OLD,KEEP),LABEL=(2,&LAB),                         X00000190
//              DCB=(RECFM=VB,LRECL=1000,BLKSIZE=1004)                   00000200
//SYSUT2 DD  DSNAME=&QDF,UNIT=&UQDF,VOLUME=&VQDF,DISP=(,&QDISP,DELETE),X00000210
//              SPACF=(CYL,&QDFSP),                                     X00000220
//              DCB=(RECFM=VB,LRECL=1000,BLKSIZE=1004)                   00000230
//SYSPRINT DD SYSOUT=(&A,&CL)                                            00000240
//SYSIN  DD  DUMMY                                                       00000250
```

NIPS 360 FFS PROCEDURES

```
//XRESTLIB      PROC  A=A,CL=',',LAB=BLP,LIB=NONE,NDISP=KEEP,          00000010
//                    SEQNO=1,STG=NIPW,ULIB=2314,VLIB=,VTAPE=,VWORK=   00000020
//*                                                                    00000030
//*   CHARLES W. HICKISCH  MAJOR,USA  PRCJ CODE=763NIPS  BRANCH=431    00000040
//*   DATE=MARCH 1, 1974                                               00000050
//*                                                                    00000060
//RESTLIB  EXEC  PGM=IEHMOVE                                           00000070
//SYSUT1 DD UNIT=&STG,SPACE=(TRK,40),VOLUME=&VWORK                     00000080
//DISK   DD  UNIT=&ULIB,VOLUME=&VLIB,DISP=OLD                          00000090
//TAPE   DD  DSNAME=&LIB.L,UNIT=TAPE9,VOLUME=&VTAPE,                  X00000100
//                DCB=(RECFM=FB,LRECL=80,BLKSIZE=800),                X00000110
//                LABEL=(&SEQNO,&LAB),DISP=(OLD,&NDISP)                00000120
//SYSPRINT DD SYSOUT=(&A,&CL)                                          00000130
```

```
//XSAVFANS     PROC  A=A,CL=',',LAB=BLP,QDF='&&QDFILE',                 00000010
//              QDISP=PASS,QRT='&&QRTFILE',UCDF=NIPW,UQRT=NIPW,         00000020
//              VQDF=,VQRT=,VTAPE=                                      00000030
//*                                                                     00000040
//*   CHARLES W. HICKISCH  MAJOR,USA  PROJ CODE=763NIPS  BRANCH=431     00000050
//*   DATE=MARCH 1, 1974                                                00000060
//*                                                                     00000070
//CPYQRT  EXEC  PGM=IEBGENER                                            00000080
//SYSPRINT DD SYSOUT=(&A,&CL)                                           00000090
//SYSUT1  DD  DSNAME=&QRT,UNIT=&UQRT,VOLUME=&VQRT,                     X00000100
//              DISP=(OLD,&QDISP,KEEP),                                X00000110
//              DCB=(RECFM=VB,LRECL=1000,BLKSIZE=1004)                  00000120
//SYSUT2 DD DSNAME=QRTXXX,UNIT=TAPE9,VOLUME=&VTAPE,DISP=(,PASS),       X00000130
//              LABEL=(,&LAB),DCB=(RECFM=VB,LRECL=1000,BLKSIZE=1004)    00000140
//SYSIN  DD  DUMMY                                                      00000150
//CPYQDF  EXEC  PGM=IEBGENER,COND=(0,NE,CPYQRT)                         00000160
//SYSUT1  DD  DSNAME=&QDF,UNIT=&UQDF,VOLUME=&VQDF,                     X00000170
//              DISP=(OLD,&QDISP,KEEP),                                X00000180
//              DCB=(RECFM=VB,LRECL=1000,BLKSIZE=1004)                  00000190
//SYSUT2 DD DSNAME=QDFXXX,VOLUME=REF=*.CPYQRT.SYSUT2,                  X00000200
//              DISP=(OLD,KEEP),LABEL=(2,&LAB),                        X00000210
//              DCB=(RECFM=VB,LRECL=1000,BLKSIZE=1004)                  00000220
//SYSPRINT DD SYSOUT=(&A,&CL)                                           00000230
//SYSIN  DD  DUMMY                                                      00000240
```

NIPS 360 FFS PROCEDURES

```
//XSAVELIB      PROC  A=A,CL=',',LAB=BLP,LIB=NONE,NDISP=KEEP,          00000010
//              SEQNO=1,STG=NIPW,ULIB=2314,VLIB=,VTAPE=,VWORK=         00000020
//*                                                                    00000030
//*   CHARLES W. HICKISCH  MAJOR,USA  PROJ CODE=763NIPS  BRANCH=431    00000040
//*   DATE=MARCH 1, 1974                                               00000050
//*                                                                    00000060
//SAVELIB  EXEC  PGM=IEHMOVE                                           00000070
//SYSUT1 DD UNIT=&STG,SPACE=(TRK,40),VOLUME=&VWORK                     00000080
//DISK   DD  UNIT=&ULIB,VOLUME=&VLIB,DISP=OLD                          00000090
//TAPE   DD  DSNAME=&LIB.L,UNIT=TAPE9,VOLUME=&VTAPE,                  X00000100
//              DCB=(RECFM=FB,LRECL=80,BLKSIZE=800),                  X00000110
//              LABEL=(&SEQNO,&LAB),DISP=(OLD,&NDISP)                  00000120
//SYSPRINT DD SYSOUT=(&A,&CL)                                          00000130
```

NIPS 360 FFS PROCEDURES

```
//XSP          PROC  A=A,BLKSIZE=560,BSZFILE=,BSZNEWF=,
//               CL=',',CL1=',',DEN=,ISAM='DUMMY.FILE',
//               JOBLIB='FFS.JOBLIB',LAB=SL,LIB='DUMMY.FILE',
//               NBRBLK=200,RGN=100K,SAM='DUMMY.FILE',
//               PTFJOBL='PTF.JOBLIB',
//               SAMOUT='DUMMY,',SORTSP=10,STG=NIPW,TRCH=,
//               UISAM=2314,ULIB=2314,USAM='(TAPE9,,DEFER)',
//               VISAM=,VLIB=,VSAM=,VSMOUT=,XDISP=OLD,
//               XINDEX='DUMMY.FILE',XUNIT=2314,XVOL=
//*****************************************************************
//***                                                            **
//***    THIS PROC IS USED TO GENERATE OR UPDATE A DISK RESIDENT  **
//*** INDEX DATA SET BASED ON THE DISK RESIDENT ISAM DATA FILE.   **
//***                                                            **
//***    //STEPNAME    EXEC  XSP,ISAM=AAAAAAA,VISAM='SER=XXXXXX',  **
//***                  XVOL='SER=YYYYYY',XDISP=(BBB),NBRBLK=DDDD   **
//***                                                            **
//*   CHARLES W. HICKISCH  MAJOR,USA   PROJ CODE=763NIPS  BRANCH=431
//*   DATE=MARCH 1, 1974
//***                                                            **
//*****************************************************************
//*                                                              **
//UTXSP     EXEC  PGM=UTNDXSPC,REGION=&RGN
//STEPLIB   DD  DISP=SHR,DSN=&PTFJOBL
//          DD  DISP=SHR,DSN=&JOBLIB
//INDEXPRT   DD  SYSOUT=(&A,&CL)
//ISAMWORK     DD    UNIT=&STG,SPACE=(CYL,(10)),DCB=(DSORG=IS,RECFM=VB,
//               LRECL=1000,BLKSIZE=1004,OPTCD=IYLR,CYLOFL=2)
//NEWFILE      DD    DSN=&ISAM,DISP=SHR,VOL=&VISAM,UNIT=&UISAM
//SAMFILE      DD    DSN=&SAM.S,UNIT=&USAM,VOLUME=&VSAM,
//               DISP=(SHR,KEEP),LABEL=(,&LAB),DCB=(RECFM=VB,LRECL=1000,
//               BLKSIZE=&BSZFILE,TRTCH=&TRCH,DEN=&DEN)
//SLIB         DD    DSN=&LIB.L,VOLUME=&VLIB,UNIT=&ULIB,DISP=SHR
//               DD DSN=&JOBLIB,DISP=SHR
//SORTLIB      DD    DSN=SYS1.SORTLIB,DISP=SHR
//SORTWK01     DD    UNIT=&STG,SPACE=(CYL,(&SORTSP),,CONTIG)
//SORTWK02     DD    UNIT=&STG,SPACE=(CYL,(&SORTSP),,CONTIG)
//SORTWK03     DD    UNIT=&STG,SPACE=(CYL,(&SORTSP),,CONTIG)
//SORTWK04     DD    UNIT=&STG,SPACE=(CYL,(&SORTSP),,CONTIG)
//SORTWK05     DD    UNIT=&STG,SPACE=(CYL,(&SORTSP),,CONTIG)
//SORTWK06     DD    UNIT=&STG,SPACE=(CYL,(&SORTSP),,CONTIG)
//SOURCPRT     DD    SYSOUT=(&A,&CL)
//SYSOUT       DD    SYSOUT=(&A,&CL1)
//SYSPRINT     DD    SYSOUT=(&A,&CL)
//SYSUDUMP     DD    SYSOUT=(&A,&CL1)
//UTSAMOUT     DD    &SAMOUT.DCB=(RECFM=VB,LRECL=1000,BLKSIZE=&BSZNEWF,
//               DEN=&DEN,TRTCH=&TRCH),DISP=(,KEEP),LABEL=(,&LAB),
//               UNIT=&USAM,VOLUME=(PRIVATE,RETAIN,&VSMOUT),DSN=&SAM.S
//XINDEX   DD  DSN=&XINDEX.X,DISP=(&XDISP,KEEP),UNIT=&XUNIT,
//               VOL=&XVOL,SPACE=(&BLKSIZE,&NBRBLK),
//               DCB=(BLKSIZE=&BLKSIZE,RECFM=F,KEYLEN=4,DSORG=DA)
```

NIPS 360 FFS PROCEDURES

```
//XSTOIS PROC  A=A,BSZFILE=,BSZNEWF=,CC=,CL=',',CL1=',',CYLOFL=1,DEN=,
//             INDEX=1,ISAM='DUMMY.FILE',JOBLIB='FFS.JCBLIB',
//             LAB=SL,NDISP=KEEP,OVFLOW=5,PRIME=40,RGN=60K,
//             PTFJOBL='PTF.JOBLIB',
//             SAM='DUMMY.FILE',SEQNO=1,TRCH=,UISAM='(2314,P)',
//             USAM='(TAPE9,,DEFER)',VISAM='SER=CANCEL',
//             VOVFLOW='REF=*.DATAFILE',VPRIME='REF=*.DATAFILE',
//             VSAM=
//*
//*   CHARLES W. HICKISCH  MAJOR,USA   PROJ CODE=763NIPS  BRANCH=431
//*   DATE=MARCH 1, 1974
//*
//STP1  EXEC PGM=UTBLDISM,REGION=&RGN,PARM='&CC'
//STEPLIB  DD  DISP=SHR,DSN=&PTFJOBL
//         DD  DISP=SHR,DSN=&JOBLIB
//DATAFILE  DD  DCB=(DSORG=IS,CYLOFL=&CYLOFL,BUFNO=5,BLKSIZE=&BSZNEWF),
//          VOLUME=&VISAM,
//          DSNAME=&ISAM.(INDEX),SPACE=(CYL,&INDEX),UNIT=&UISAM,
//          DISP=(,&NDISP,DELETE)
//          DD  DCB=*.DATAFILE,VOLUME=&VPRIME,
//          DSNAME=&ISAM.(PRIME),SPACE=(CYL,&PRIME),UNIT=&UISAM,
//          DISP=(,&NDISP,DELETE)
//          DD  DCB=*.DATAFILE,VOLUME=&VOVFLOW,
//          DSNAME=&ISAM.(OVFLOW),SPACE=(CYL,&OVFLOW),UNIT=&UISAM,
//          DISP=(,&NDISP,DELETE)
//SAMFILE  DD  DSNAME=&SAM.S,DISP=SHR,UNIT=&USAM,VOLUME=&VSAM,
//       LABEL=(&SEQNO,&LAB),DCB=(RECFM=VB,LRECL=1000,BLKSIZE=&BSZFILE,
//       DEN=&DEN,TRTCH=&TRCH,BUFNO=5)
//SYSPRINT DD SYSOUT=(&A,&CL)
//SYSUDUMP DD SYSOUT=(&A,&CL1)
```

NIPS 360 FFS PROCEDURES

```
//XSUBCHK       PROC  A=A,CL=',',CL1=',',CL2=',',JOBLIB='FFS.JOBLIB',     0000010C
//              PTFJOBL='PTF.JOBLIB',                                     0000011C
//              LIB=,ULIB=2314,VLIB=                                      0000020C
//*                                                                       0000030C
//*   CHARLES W. HICKISCH  MAJOR,USA   PROJ CODE=763NIPS  BRANCH=431      0000040C
//*                                                                       0000060C
//SUBCHK    EXEC PGM=UTSUBCHK                                             0000070C
//STEPLIB DD DSN=&PTFJOBL,DISP=SHR                                        0000080C
//          DD  DISP=SHR,DSN=&JOBLIB                                      00000810
//SLIB    DD DSN=&LIB.L,VOL=&VLIB,UNIT=&ULIB,DISP=SHR                     00000900
//SYSDMY DD SYSOUT=(&A,&CL)                                               00001000
//SYSUDUMP DD SYSOUT=(&A,&CL1)                                            0000110C
//SYSPRINT DD SYSOUT=(&A,&CL2)                                            00001200
```

```
//XSUBLDR       PROC  A=A,CL=',',CL1=',',JOBLIB='FFS.JOBLIB',
//              JOBMAC='FFS.JOBMACRO',LIB=NONE,LIBDISP=OLD,
//              LIBSP='(2,1,5)',MODLIB='&TEMP',RGN=60K,
//              PTFJOBL='PTF.JOBLIB',PTFJOBM='PTF.JOBMACRC',
//              STG=NIPW,ULIB=2314,VLIB=,BLK=7294,MDISP=DELETE
//*
//*   CHARLES W. HICKISCH  MAJOR,USA  PROJ CODE=763NIPS  BRANCH=431
//*   DATE=MARCH 1, 1974
//*
//SUB     EXEC  PGM=UTSUBLDR,REGION=&RGN
//STEPLIB  DD  DISP=SHR,DSN=&PTFJOBL
//          DD  DISP=SHR,DSN=&JOBLIB
//SYSUDUMP DD SYSOUT=(&A,&CL1)
//SYSPRINT DD SYSOUT=(&A,&CL)
//ASSEMIN DD    UNIT=&STG,SPACE=(CYL,01),DSNAME=&AIN,DISP=(,PASS),
//              DCB=(BLKSIZE=400)
//LNKEDIN DD    UNIT=&STG,SPACE=(CYL,01),DSNAME=&CLIN,DISP=(,PASS),
//              DCB=(BLKSIZE=400)
//SYSOUT   DD SYSOUT=(&A,&CL)
//ASMBL  EXEC  PGM=IEUASM,COND=(0,LT,SUB),REGION=&RGN
//SYSUT1 DD SPACE=(1700,(400,50)),UNIT=&STG
//SYSUT2 DD SPACE=(1700,(400,50)),UNIT=(&STG,SEP=SYSUT1)
//SYSUT3 DD SPACE=(1700,(400,50)),UNIT=(&STG,SEP=(SYSUT1,SYSUT2))
//SYSLIB   DD  DISP=SHR,DSN=&PTFJOBM
//          DD  DISP=SHR,DSN=&JOBMAC
// DD DISP=SHR,DSN=SYS1.MACLIB
//SYSPRINT DD SYSOUT=(&A,&CL)
//SYSPUNCH DD   UNIT=&STG,SPACE=(CYL,5),DSNAME=&LKIN,DISP=(,PASS),
//              DCB=(BLKSIZE=400)
//SYSIN  DD     DISP=(OLD,DELETE),DSNAME=&AIN
//LKEDT  EXEC   PGM=IEWL,PARM='LIST,XREF',REGION=&RGN,
//              COND=((0,LT,ASMBL),(0,LT,SUB))
//SYSUT1 DD     SPACE=(TRK,(50,5)),UNIT=&STG
//SYSPRINT DD SYSOUT=(&A,&CL)
//SYSLMOD  DD   DSNAME=&LIB.L,UNIT=&ULIB,VOLUME=&VLIB,
//              SPACE=(CYL,&LIBSP),DISP=(&LIBDISP,KEEP),
//              DCB=(RECFM=U,BLKSIZE=&BLK)
//DPUNCH DD     DSNAME=&LKIN,DISP=(OLD,DELETE)
//SYSLIN DD     DSNAME=&CLIN,DISP=(OLD,DELETE)
//SYSLIB DD     DISP=SHR,DSNAME=&JOBLIB
//MODLIB DD     DSNAME=&MODLIB,DISP=(OLD,&MDISP)
```

```
//XTABGEN       PROC  A=A,BLK=7294,CL=',',CL1=',',CL2=',',                00000100
//              JOBLIB='FFS.JOBLIB',LIB=NONE,LIBDISP=OLD,                 00000200
//              LIBSP='(2,1,5)',RGN=80K,SORTSP=8,STG=NIPW,ULIB=2314,      00000300
//              PTFJOBL='PTF.JOBLIB',                                     00000310
//              VLIB=                                                     00000400
//*                                                                       00000500
//*   CHARLES W. HICKISCH  MAJOR,USA  PROJ CODE=763NIPS  BRANCH=431       00000600
//*   DATE=MARCH 1, 1974                                                  00000700
//*                                                                       00000800
//TAB  EXEC PGM=UTTABGEN,REGION=&RGN                                      00000900
//STEPLIB  DD  DISP=SHR,DSN=&PTFJOBL                                      00001000
//         DD  DISP=SHR,DSN=&JOBLIB                                       00001010
//SYSOUT   DD SYSOUT=(&A,&CL)                                             00001100
//SYSUDUMP DD SYSOUT=(&A,&CL1)                                            00001200
//SYSPRINT DD SYSOUT=(&A,&CL2)                                            00001300
//SORTLIB  DD  DISP=SHR,DSNAME=SYS1.SORTLIB                               00001400
//SORTWK01  DD  SPACE=(CYL,(&SORTSP),,CONTIG),UNIT=&STG                   00001500
//SORTWK02 DD SPACE=(CYL,(&SORTSP),,CONTIG),UNIT=(&STG,SEP=SORTWK01),     X00001600
//              SEP=SORTWK01                                              00001700
//SORTWK03 DD SPACE=(CYL,(&SORTSP),,CONTIG),UNIT=(&STG,SEP=SORTWK02),     X00001800
//              SEP=SORTWK02                                              00001900
//SORTWK04 DD SPACE=(CYL,(&SORTSP),,CONTIG),UNIT=(&STG,SEP=(SORTWK01,     X00002000
//              SORTWK03)),SEP=(SORTWK01,SORTWK03)                        00002100
//SORTWK05 DD SPACE=(CYL,(&SORTSP),,CONTIG),UNIT=(&STG,SEP=(SORTWK02,     X00002200
//              SORTWK04)),SEP=(SORTWK02,SORTWK04)                        00002300
//SORTWK06 DD SPACE=(CYL,(&SORTSP),,CONTIG),UNIT=(&STG,SEP=(SORTWK01,     X00002400
//              SORTWK03,SORTWK05)),SEP=(SORTWK01,SORTWK03,SORTWK05)      00002500
//SYSLMOD DD DSNAME=&LIB.L,VOLUME=&VLIB,UNIT=&ULIB,SPACE=(CYL,&LIBSP),    X00002600
//              DISP=(&LIBDISP,KEEP),DCB=(RECFM=U,BLKSIZE=&BLK)           00002700
```

```
//XTP     PROC  A=A,                                                    00000100
//            CL=',',CL1=',',CL2=U,CL3=C,CL4=S,CL5=T,                   00000200
//               FFT='DUMMY.FILE',
//               FFT1='DUMMY.FILE',FFT2='DUMMY.FILE',
//               JOBLIB='FFS.JOBLIB',                                   00000300
//               LIB='DUMMY.FILE',                                      00000400
//               LIB1='DUMMY.FILE',                                     00000500
//               PTFJOBL='PTF.JOBLIB',                                  00000600
//               RGN=82K,                                               00000800
//               STG=NIPW,TPIMQ=TPIMQ,                                  00000900
//               TPDUMP='SYSOUT=(A,,)',                                 00001000
//               UFFT='(2314,P)',
//               UFFT1='(2314,P)',UFFT2='(2314,P)',
//               ULIB=2314,ULIB1=2314,                                  00001100
//               UOMQ=2314,                                             00001200
//             VFFT=,VFFT1=,VFFT2=,
//               VOMQ='REF=*.INMSGQ',                                   00001300
//               VLIB=,VLIB1=,                                          00001400
//               XINDEX='DUMMY.FILE',                                   00001500
//               XUNIT=2314,XVOL=                                       00001600
//*                                                                     00001700
//*   CHARLES W. HICKISCH  MAJOR,USA  PROJ CODE=763NIPS  BRANCH=431     00001800
//*   DATE=MARCH 1, 1974                                                00001900
//*                                                                     00002000
//TPMONSUP    EXEC PGM=UTTPDRVR,REGION=&RGN                             00002100
//STEPLIB DD DSN=&PTFJOBL,DISP=SHR                                      00002200
//        DD DSN=&JOBLIB,DISP=SHR                                       00002300
//SYSUT1 DD SPACE=(TRK,(0,5)),UNIT=&STG                                 00002400
//SYSUT2 DD SPACE=(TRK,(0,20)),UNIT=&STG                                00002500
//SYSUT3 DD SPACE=(TRK,(0,5)),UNIT=(&STG,SEP=SYSUT1)                    00002600
//SYSUT4 DD SPACE=(CYL,(0,1)),UNIT=&STG                                 00002700
//SLIB DD DSNAME=&LIB.L,UNIT=&ULIB,VOLUME=&VLIB,DISP=SHR                00002800
//            DD DSNAME=&LIB1.L,UNIT=&ULIB1,VOLUME=&VLIB1,DISP=SHR      00002900
//      DD DSNAME=&JOBLIB,DISP=SHR                                      00003000
//DATAFILE DD DISP=SHR,DSNAME=DUMMY.FILE,UNIT=(,P,DEFER)                00003100
//DATAFIL2  DD  DISP=SHR,DSN=DUMMY.FILE,UNIT=(2314,P,DEFER)             00003300
//DATAFIL3  DD  DISP=SHR,DSN=DUMMY.FILE,UNIT=(2314,P,DEFER)             00003400
//SAMFILE   DD  DISP=SHR,DSN=DUMMY.FILES,UNIT=(2314,P,DEFER)            00003500
//DATAFILA  DD  DSN=DUMMY.FILE67,UNIT=(NIPW,2),SPACE=(TRK,(0)),DISP=NEW 00003200
//FFT     DD     DSN=&FFT,
//               UNIT=&UFFT,
//               VOLUME=&VFFT,
//               DISP=SHR
//FFT1    DD     DSN=&FFT1,
//               UNIT=&UFFT1,
//               VOLUME=&VFFT1,
//               DISP=SHR
//FFT2    DD     DSN=&FFT2,
//               UNIT=&UFFT2,
//               VOLUME=&VFFT2,
//               DISP=SHR
//EDCONSOL  DD  SPACE=(TRK,(5,,4)),UNIT=&STG                            00003600
//EDITLIB   DD  DSN=DUMMY.FILEL,DISP=SHR                                00003700
//INMSGQ    DD  DISP=SHR,DSNAME=&TPIMQ                                  00003800
//OMSGQ DD DISP=SHR,DSNAME=&TPIMQ                                       00003900
//OUTMSGQ   DD  SPACE=(TRK,0),VOLUME=&VOMQ,UNIT=&UOMQ                   00004000
//AMSGQ DD UNIT=&STG,SPACE=(TRK,0)                                      00004100
```

```
//SDCONSOL DD UNIT=&STG,SPACE=(230C,(10,,4))                          00004200
//SDKNSET DD SPACE=(CYL,(1,1)),UNIT=&STG,DCB=(RECFM=F,BLKSIZE=1004)   00004300
//STATRECS  DD  SYSOUT=(&A,&CL)                                       00004400
//SYSLMOD DD SPACE=(TRK,(20,,8)),UNIT=&STG,DCB=FFS.JOBLIB,            00004500
// LABEL=EXPDT=66366                                                  00004600
//SYSPRINT DD SYSOUT=(&A,&CL)                                         00004700
//SYSABEND DD SYSOUT=(&A,&CL1)                                        00004800
//TPDUMP  DD  SYSOUT=(&A,&CL1)                                        00004900
//EDITDUMP DD SYSOUT=(&A,&CL1)                                        00005000
//SNAPSHOT DD SYSOUT=(&A,&CL1)                                        00005010
//PL1DUMP  DD SYSOUT=(&A,&CL1)                                        00005020
//SYSONLIN DD &TPDUMP                                                 00005100
//XINDEX  DD  DSN=&XINDEX.X,UNIT=&XUNIT,VOL=&XVOL,DISP=SHR            00005200
//SHARDASD DD DSN=NIPS.TPJOBQ,DISP=(MOD,KEEP)                         00005300
//SYSIN  DD  DUMMY                                                    00005400
//U DD SYSOUT=(&A,&CL2)                                               00005500
//C DD SYSOUT=(&A,&CL3)                                               00005600
//S DD SYSOUT=(&A,&CL4)                                               00005700
//T DD SYSOUT=(&A,&CL5)                                               00005800
//SUBFILE DD UNIT=&STG,SPACE=(CYL,(0,1,10))                           00005900
//MENUSET DD DSN=DUMMY.FILEL,DISP=SHR                                00005910
```

```
//XTRDISK      PROC  A=A,CL=',',CL1=',',JOBLIB='FFS.JOBLIB',
//           LAB=SL,RGN=60K,
//           PTFJOBL='PTF.JOBLIB',
//           XFDISP=SHR,XFNAME=,XFUNIT=2314,XFVOL=,
//           ISAM='DUMMY.FILE',UISAM='(2314,P)',VISAM=,
//           SAM='DUMMY.FILE',USAM='(TAPE9,,DEFER)',VSAM=,
//           XTNAME=,XTUNIT='(TAPE9,,DEFER)',XTVOL=,
//           XTLAB=SL,STAT=NO,
//           BSZFILE=,TRCH=,DEN=
//*******************************************************************
//*                                                                 *
//*      THIS PROC IS USED TO TRANSFER A DISK-RESIDENT INDEX DATA SET *
//* TO TAPE.  THIS OPERATION CONDENSES THE INDEX DATA SET.  THE TAPE *
//* SO CREATED IS A SEQUENTIAL DATA SET CONSISTING OF VARIABLE LENGTH *
//* BLOCKED RECORDS THAT CONTAIN THE SOURCE DATA AND CONTROL        *
//* INFORMATION FOR SUBSEQUENTLY RECONSTRUCTING THE SOURCE DATA.    *
//*                                                                 *
//*    //STEPNAME EXEC XTRDISK,XFNAME=WW,XTNAME=XX,XFVOL=YY,XTVOL=ZZ *
//*                                                                 *
//*   CHARLES W. HICKISCH  MAJOR,USA  PROJ CODE=763NIPS  BRANCH=431
//*   DATE=MARCH 1, 1974
//*                                                                 *
//*******************************************************************
//*                                                                 *
//XTR    EXEC  PGM=UTNDXTFR,PARM='DISK,&STAT',REGION=&RGN
//STEPLIB  DD  DISP=SHR,DSN=&PTFJOBL
//         DD  DISP=SHR,DSN=&JOBLIB
//INDEXDAM     DD  DSN=&XFNAME,UNIT=&XFUNIT,VOL=&XFVOL,DISP=&XFDISP
//INDEXPRT     DD   SYSOUT=(&A,&CL)
//INDEXSAM DD DSN=&XTNAME,UNIT=&XTUNIT,VOL=&XTVOL,DISP=(NEW,KEEP),
//           LABEL=(,&XTLAB)
//SYSPRINT     DD   SYSOUT=(&A,&CL)
//SYSUDUMP     DD   SYSOUT=(&A,&CL1)
//DATAFILE DD  DSN=&ISAM,
//           UNIT=&UISAM,
//           VOLUME=&VISAM,
//           DISP=SHR
//SAMFILE   DD  DSN=&SAM.S,
//           UNIT=&USAM,
//           VOLUME=&VSAM,
//           DISP=(SHR,KEEP),
//           LABEL=(,&LAB),
//           DCB=(RECFM=VB,LRECL=1000,BLKSIZE=&BSZFILE,
//           TRTCH=&TRCH,DEN=&DEN)
```

NIPS 360 FFS PROCEDURES

```
//XTRTAPE       PROC  A=A,CL=',',CL1=',',JOBLIB='FFS.JOBLIB',          00000100
//              XFLAB=SL,NBRBLK=50,RGN=60K,XFNAME=,                    00000200
//              PTFJOBL='PTF.JOBLIB',                                  00000300
//              XFUNIT='(TAPE9,,DEFER)',XFVOL=,XFDISP=OLD,            X00000400
//              XTDISP='(NEW,KEEP)',XTNAME=,XTUNIT=2314,XTVOL=         00000500
//*********************************************************************** 00000600
//*                                                                   * 00000700
//*      THIS PROC IS USED TO RECONSTRUCT A DISK-RESIDENT INDEX DATA  * 00000800
//* SET FROM A PREVIOUSLY UNLOADED SEQUENTIAL VERSION OF THE INDEX    * 00000900
//* DATA SET.                                                         * 00001000
//*                                                                   * 00001100
//*     //STEPNAME EXEC XTRTAPE,XFNAME=WW,XTNAME=XX,XFVCL=YY,XTVOL=ZZ * 00001200
//*                                                                   * 00001300
//*   CHARLES W. HICKISCH  MAJOR,USA  PRCJ CODE=763NIPS  BRANCH=431    00001400
//* DATE=MARCH 1,1974                                                  00001500
//*                                                                   * 00001600
//*********************************************************************** 00001700
//XTR           EXEC PGM=UTNDXTFR,PARM=TAPE,REGION=&RGN                00001800
//STEPLIB  DD   DISP=SHR,DSN=&PTFJOBL                                  00001900
//         DD   DISP=SHR,DSN=&JOBLIB                                   00002000
//INDEXPRT      DD    SYSOUT=(&A,&CL)                                  00002100
//INDEXSAM  DD  DSN=&XFNAME,UNIT=&XFUNIT,VOL=&XFVOL,DISP=&XFDISP,     X00002200
//              LABEL=(,&XFLAB)                                        00002300
//SYSPRINT      DD    SYSOUT=(&A,&CL)                                  00002400
//SYSUDUMP      DD    SYSOUT=(&A,&CL1)                                 00002500
//XINDEX        DD    DSN=&XTNAME,UNIT=&XTUNIT,VOL=&XTVOL,            *00002600
//              DISP=&XTDISP,SPACE=(560,&NBRBLK),                     C00002700
//              DCB=(BLKSIZE=560,RECFM=F,KEYLEN=4,DSORG=DA)            00002800
```

NIPS 360 FFS PROCEDURES

```
//XUTFSCAN      PROC  A=A,BSZFILE=,                                    00000100
//              CL=',',CL1=',',DEN=,ISAM='DUMMY.FILE',                 00000200
//              JOBLIB='FFS.JOBLIB',LAB=SL,LIB='DUMMY.FILE',           00000300
//              LIB1='DUMMY.FILE',RGN=98K,SAM='DUMMY.FILE',            00000400
//              PTFJOBL='PTF.JOBLIB',                                  00000410
//              SORTSP=10,STG=NIPW,TBLK=500,TDISP=MOD,                 00000500
//              TRANS='DUMMY.FILE',TRANSP=200,TRCH=,UISAM='(2314,P)',  00000600
//              ULIB=2314,ULIB1=2314,USAM='(TAPE9,,DEFER)',            00000700
//              UTRANS=2314,VISAM=,VLIB=,VLIB1=,VSAM=,VTRANS=          00000800
//*                                                                    00000900
//*   CHARLES W. HICKISCH  MAJOR,USA  PROJ CODE=763NIPS  BRANCH=431    00001000
//*   DATE=MARCH 1, 1974                                               00001100
//*                                                                    00001200
//UTFLDSCN  EXEC  PGM=UTFLDSCN,REGION=&RGN                             00001300
//STEPLIB  DD  DISP=SHR,DSN=&PTFJOBL                                   00001400
//         DD  DISP=SHR,DSN=&JOBLIB                                    00001410
//SYSPRINT  DD SYSOUT=(&A,&CL)                                         00001500
//SYSUDUMP  DD SYSOUT=(&A,&CL1)                                        00001600
//SYSOUT    DD SYSOUT=(&A,&CL1)                                        00001700
//DATAFILE  DD DSNAME=&ISAM,UNIT=&UISAM,VOLUME=&VISAM,DISP=SHR         00001800
//SORTLIB  DD  DSNAME=SYS1.SORTLIB,DISP=SHR                            00001900
//SORTIN    DD UNIT=&STG,SPACE=(TRK,(&TRANSP,150))                     00002000
//SORTWK01  DD SPACE=(CYL,(&SORTSP),,CONTIG),UNIT=(&STG,SEP=SORTIN)    00002100
//SORTWK02  DD SPACE=(CYL,(&SORTSP),,CONTIG),                          00002200
//             UNIT=(&STG,SEP=SORTWK01),SEP=SORTWK01                   00002300
//SORTWK03  DD SPACE=(CYL,(&SORTSP),,CONTIG),                          00002400
//             UNIT=(&STG,SEP=SORTWK02),SEP=SORTWK02                   00002500
//SORTWK04  DD SPACE=(CYL,(&SORTSP),,CONTIG),                          00002600
//             UNIT=(&STG,SEP=(SORTWK01,SORTWK03)),                    00002700
//             SEP=(SORTWK01,SORTWK03)                                 00002800
//SORTWK05  DD SPACE=(CYL,(&SORTSP),,CONTIG),                          00002900
//             UNIT=(&STG,SEP=(SORTWK02,SORTWK04)),                    00003000
//             SEP=(SORTWK02,SORTWK04)                                 00003100
//SORTWK06  DD SPACE=(CYL,(&SORTSP),,CONTIG),                          00003200
//             UNIT=(&STG,SEP=(SORTWK01,SORTWK03,SORTWK05)),           00003300
//             SEP=(SORTWK01,SORTWK03,SORTWK05)                        00003400
//ISAMWORK  DD UNIT=&STG,DCB=DSORG=IS,SPACE=(CYL,(10))                 00003500
//SAMFILE   DD DSNAME=&SAM.S,UNIT=&USAM,VOLUME=&VSAM,DISP=(SHR,KEEP),  00003600
//             LABEL=(,&LAB),DCB=(RECFM=VB,LRECL=1000,BLKSIZE=&BSZFILE, 00003700
//             TRTCH=&TRCH,DEN=&DEN)                                   00003800
//SYSUT2    DD DSNAME=&LIB.L,UNIT=&ULIB,VOLUME=&VLIB,DISP=SHR          00003900
//          DD DSNAME=&LIB1.L,UNIT=&ULIB1,VOLUME=&VLIB1,DISP=SHR       00004000
//TRANST  DD  DSNAME=&TRANS,UNIT=&UTRANS,VOLUME=&VTRANS,DISP=&TDISP,   00004100
//             DCB=(RECFM=FB,LRECL=50,BLKSIZE=&TBLK)                   00004200
```

NIPS 360 FFS PROCEDURES

```
//XUTODE PROC  A=A,                                                  00000100
//             CL=',',                                               00000200
//             JOBLIB='FFS.JOBLIB',                                  00000300
//             PTFJOBL='PTF.JOBLIB',                                 00000400
//             LIB='DUMMY.FILE',                                     00000500
//             RGN=66K,                                              00000600
//             SDISP=SHR,                                            00000700
//             SNAP=,                                                00000800
//             SOURCL='DUMMY.FILE',                                  00000900
//             ULIB=2314,                                            00001000
//             USOURCL=2314,                                         00001100
//             VLIB=,                                                00001200
//             VSOURCL=                                              00001300
//XUTODE EXEC  PGM=UTODE,REGION=&RGN,PARM=&SNAP                      00001400
//STEPLIB DD DSN=&PTFJOBL,DISP=SHR                                   00001500
//         DD  DISP=SHR,DSN=&JOBLIB                                  00001600
//SYSUDUMP DD  SYSOUT=(&A,&CL)                                       00001700
//SNAPSHOT DD  SYSOUT=(&A,&CL)                                       00001800
//SOURCPRT DD  SYSOUT=(&A,&CL)                                       00001900
//ODDPRT   DD  SYSOUT=(&A,&CL)                                       00002000
//SYSPRINT  DD  SYSOUT=(&A,&CL)                                      00002100
//SLIB   DD    DISP=OLD,DSN=&LIB.L,UNIT=&ULIB,VOL=&VLIB              00002200
//SOURCLIB DD  DISP=&SDISP,DSN=&SOURCL.L,UNIT=&USOURCL,VOL=&VSOURCL  00002300
```

NIPS 360 FFS PROCEDURES

```
//XUTSOURC      PROC  CL=',',CL1=',',JOBLIB='FFS.JOBLIB',                 00000100
//              PTFJOBL='PTF.JOBLIB',                                     00000110
//              NAME=,SDISP=SHR,SOURCL=,USOURCL=2314,VSOURCL=             00000200
//*                                                                       00000300
//*   CHARLES W. HICKISCH  MAJOR,USA  PROJ CODE=763NIPS  BRANCH=431       00000400
//*   DATE=MARCH 1, 1974                                                  00000500
//*                                                                       00000600
//SOURC EXEC PGM=UTSOURC                                                  00000700
//STEPLIB  DD  DISP=SHR,DSN=&PTFJOBL                                      00000800
//         DD  DISP=SHR,DSN=&JOBLIB                                       00000810
//SOURCPRT DD SYSOUT=(A,&CL)                                              00000900
//SOURCLIB  DD  DISP=&SDISP,DSN=&SOURCL.L,VOL=&VSOURCL,UNIT=&USOURCL      00001000
//SYSUDUMP DD SYSOUT=(A,&CL1)                                             00001100
//SYSIN  DD DISP=SHR,DSN=&SOURCL.L(&NAME),UNIT=&USOURCL,VOL=&VSCURCL      00001200
```

DISTRIBUTION

CCTC CODES	COPIES
C124 (Reference and Record)	3
C124 (Record Copy) Stock	6
C240	20
C315	1
C341 (Maintenance Contractor)	10
C341 (Stock)	70

EXTERNAL

Director of Administrative Services, Office of the Joint Chiefs of Staff Attn: Chief, Personnel Division, Room 1A724, The Pentagon Washington, D.C. 20301	1
Director for Personnel, J-1, Office of the Joint Chiefs of Staff, Attn: Chief, Data Service Office, Room 1B738C, The Pentagon, Washington, D.C. 20301	1
Director for Operations, J-3, Office of the Joint Chiefs of Staff, Attn: P & AD, Room 2B870, The Pentagon, Washington, D.C. 20301	1
Director for Operations, J-3, Office of the Joint Chiefs of Staff, Attn: Deputy Director for Operations (Reconnaisance and Electronic Warfare) Room 2D921, The Pentagon, Washington, D.C. 20301	1
Director for Logistics, J-4, Office of the Joint Chiefs of Staff, Room 2E828, The Pentagon, Washington, D.C. 20301	1
Chief, Studies Analysis and Gaming Agency, Attn: Chief, Force Analysis Branch, Room 1D928A, The Pentagon, Washington, D.C. 20301	1
Automatic Data Processing, Liaison Office National Military Command Center, Room 2D901A, The Pentagon, Washington, D.C. 20301	1

EXTERNAL	COPIES
Automatic Data Processing Division Supreme Headquarters Allied Powers, Europe Attn: SA & P Branch, APO New York 09055------	1
Director, Defense Communications Agency, Office Of MEECN System Engineering, Attn: Code 960T, Washington, D.C. 20301----------	1
Director, Defense Communications Engineering Center, Hybrid Simulation Facility, 1860 Wiehl Avenue, Reston, VA 22070----------	1
Director, Defense Intelligence Agency Attn: DS - 5C2 Washington, D.C. 20301----------	5
Commander-in-Chief, Pacific, Attn: J6331, FPO San Francisco, 96610----------	1
Commander-in-Chief, US Army Europe and Seventh Army ATTN: OPS APO New York 09403---	1
Commanding General, US Army Forces Command, Attn: Data Support Division, Building 206, Fort McPherson, GA 30303----------	1
Commander, Fleet Intelligence Center, Europe, Box 18, Naval Air Station, Jacksonville, Florida 32212----------	1
Commanding Officer, Naval Air Engineering Center, Ground Support Equipment Department, SE 314, Building 76-1, Philadelphia, PA 19112	1
Commanding Officer, Naval Security Group Command, 3801 Nebraska Avenue, N.W. Attn: GP22, Washington, D.C. 20390----------	1
Commanding Officer, Navy Ships Parts Control Center, Attn: Code 712, Mechanicsburg, PA 17055	1
Headquarters, US Marine Corps, Attn: System Design and Programming Section (MC-JSMD-7) Washington, D.C. 20380----------	1

EXTERNAL	COPIES
Commanding Officer, US Army Forces Command Intelligence Center, Attn: AFIC-PD, Fort Bragg, NC 28307	1
Commander, US Army Foreign Science and Technology Center, Attn: AMXSJ-CS, 220 Seventh Street NE, Charlottsville, VA 22212	1
Commanding Officer, US Army Security Agency, Command Data Systems Activity (CDSA) Arlington Hall Station, Arlington, VA 22212	1
Commanding Officer, US Army Security Agency Field Station - Augsburg, Attn: IAEADP, APO New York 09458	1
Commander, Fleet Intelligence Center, Atlantic, Attn: DPS, Norfolk, VA 23511	1
Commander, Fleet Intelligence Center, Pacific, Box 500, Pearl Harbor, HI 96860	1
Air Force Operations Center, Attn: Systems Division (XOOCSC) Washington, D.C. 20301	1
Commander, Armed Forces Air Intelligence Training Center, TTMNIM (360 FFS), Lowry AFB, Co 80230	1
Commander, Air Force Data Services Center, Attn: Director of System Support, Washington, D.C. 20330	1
Commander-in-Chief, US Air Forces in Europe, Attn: ACDI APO New York 09332	1
Commander, USAF Tactical Air Command, Langley AFB, VA 23665	1
Commander, Space and Missile Test Center, Attn: (ROCA) Building 7000, Vandenberg, AFB, CA 93437	1

EXTERNAL	COPIES
Naval Air Systems Command, Naval Air Station, Code 13999, Jacksonville, Florida 32212------	1
Commanding General, US Army Computer Systems Command, Attn: Support Operations Directorate, Fort Belvoir, VA----------	1
Defense Documentation Center, Cameron Station, Alexandria, VA 22314----------	12
TOTAL	159

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER CSM UM 15-78	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) NMCS Information Processing System 360 Formatted File System (NIPS 360 FFS) - Users Manual Vol VIII - Job Preparation		5. TYPE OF REPORT & PERIOD COVERED 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s)		8. CONTRACT OR GRANT NUMBER(s) DCA 100-77-C-0065
9. PERFORMING ORGANIZATION NAME AND ADDRESS International Business Machines, Corp. Rosslyn, Virginia		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS National Military Command System Support Center The Pentagon, Washington, D.C. 20301		12. REPORT DATE 1 September 1978 13. NUMBER OF PAGES 187
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

~~Copies of this document may be obtained from the Defense Documentation Center, Cameron Station, Alexandria, Virginia 22314.~~

This document has been approved for public release and sale; its distribution is unlimited.

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

18. SUPPLEMENTARY NOTES

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

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

This document familiarizes the user with the features available using NIPS 360 FFS Job Preparation procedures. It describes symbolic parameters and file naming conventions, illustrates general approaches to running jobs using single or multiple data bases and file libraries. Job Control Language (JCL) examples for each system component are given.

This document supersedes CSM UM 15-78, Vol VIII.

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

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)