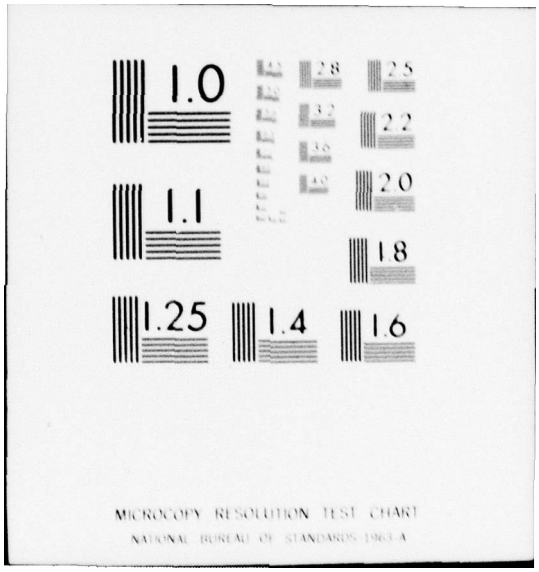
AD-A056 202 NATIONAL MILITARY COMMAND SYSTEM SUPPORT CENTER WASH--ETC F/G 9/2
NMCS INFORMATION PROCESSING SYSTEMS 360 FORMATTED FILE SYSTEM (--ETC(U)
OCT 77

UNCLASSIFIED NMCSSC-CSM-GD-15-74-CHANG NL

1 OF 1
AD
A056 202

END
DATE
FILMED
8-78
DDC


1.0
1.1
1.25
1.4
1.6
2.8
2.5
3.2
2.2
3.6
2.0
4.0
1.8
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

DEFENSE COMMUNICATIONS AGENCY
COMMAND AND CONTROL
TECHINCAL CENTER

(6) NMCS Information Processing System 360 Formatted File System (NIPS 360 FFS). General Description. Change 3.

LEVEL II

IN REPLY REFER TO

(11) 15 October 1977

TO: DISTRIBUTION

AD A002984

SUBJECT: Change 3 to CSM GD15-74, General Description, dated 15 October 1974

(14) NMCSSC-CSM-GD-15-74-CHANGE-3 (12) 19 p.

1. Insert the enclosed change pages and destroy the replaced pages according to applicable security regulations.

2. A list of Effective Pages to verify the accuracy of this manual is enclosed. This list should be inserted before the title page.

3. When this change has been posted, make an entry in the Record of Changes on the inside cover.

FOR THE DIRECTOR:

17 Enclosures
Change 3 pages

J. DOUGLAS POTTER
Assistant to the Director
for Administration

243 000

JOB

DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY PRACTICABLE. THE COPY FURNISHED TO DDC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

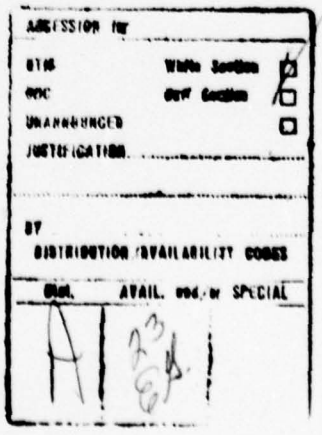
EFFECTIVE PAGES - 30 September 1977

This list is used to verify the accuracy of CSM GD 15-74, after change 3 pages have been inserted. Original pages are indicated by the letter O, change 3 by the numeral 3.

Page No.	Change No.
Title Page	O
ii	O
iii	3
iv	O
1-2	3
3-6	O
7-12	3
13-16	O
17-24	3
25-35	O

CONTENTS

ILLUSTRATIONS



Section 1

INTRODUCTION

The NIPS 360 FFS is an advanced data management system providing powerful, efficient, and flexible data management support to a wide variety of users. NIPS provides the ability to structure files, generate and maintain files, retrieve information and output that information in simple or complex arrays on a variety of output devices. Seven major processing functions are provided:

(1) File Definition (structure);

(2) File Generation and Maintenance;

(3) File Revision;

(4) Data Selection and Ordering (retrieval);

(5) Formal Report Generation (output);

(6) Terminal Processing and Display; and

(7) Utility programs supporting table generation, subroutine loading and data conversion, among others.

NIPS functions under the Primary Control Program (PCP), the multiprogramming with fixed number of tasks (MFT) or the multiprogramming with variable number of tasks (MVT) versions of the IBM System/360 Operating System. NIPS also functions under IBM System/370 System Control Programming Operating System/Virtual Storage 1 (VS1) and Operating System/Virtual Storage 2 (VS2). Using multiprogramming capabilities, NIPS permits online functions to parallel the normal background or batched processing functions. NIPS can use a broad range of hardware since input/output device independence has been retained. Core sizes are relatively noncritical in that NIPS will function within 90K of core memory. Core usage is dynamic, permitting NIPS to make maximum use of the partition or region of core provided by

the operating system. Minimum core requirements for NIPS and the operating system are 128K for single partition systems and 256K for multitasking systems using the online capabilities. The design base was a S/360 Model 50H; however the instruction set utilization was limited to the standard instruction set and the decimal feature instructions.

NIPS will support data files organized with the Indexed Sequential Access Method (ISAM), resident on a direct access storage device (such as an IBM 3330, 2314 or 2311 Disk Storage Device), and the Sequential Access Method (SAM), resident on tape or direct access devices. When installed on S/370 with OS/VS, NIPS will support data files organized under the Virtual Storage Access Method (VSAM) for batch processing.

Section 3

SYSTEM ORGANIZATION

The six major processing components and the general utility component each perform a general task and are associated with a general class of functions to be performed.

The File Structuring (FS), File Revision (FR), and File Maintenance (FM) components are primarily concerned with the creation, reorganization and maintenance of the system files. The Retrieval and Sort Processor (RASP) is the primary information retrieval tool while the Output Processor (OP) component provides for formal report production. Finally, the Terminal Processing (TP) component provides online NIPS programming, file interrogation, maintenance, and output capabilities through a variety of local and remote terminals. These include the IBM 3277, 2250, 2260 and 2265 display terminals, the IBM 1050 dial-up and IBM 2741 terminals. Each component, as well as the utility component, is discussed briefly in this section. The applicability of this system to a specific task area can only be evaluated at the detailed level. This description, therefore, suggests only in the most general terms the true depth of the actual software system.

3.1 File Structure

This system component permits the user to identify his data file and its security classification and to introduce definitions of the fields, groups, and sets that are to make up his data file. Although there is theoretically no limit to the number of unique data elements that can be named in a single file, the system imposes a practical limit of approximately 25,000 names. Structures are limited to a single level of subordination. At that level, however, up to 255 different periodic and/or variable sets may be specified.

There are also size restrictions in the length of a single periodic subset or fixed set. This constraint is 1,000 bytes in the current design.

File records (the collection of the fixed, periodic, and variable sets which are uniquely identified by a record identifier or key) are less easily defined as to size. The system performs dynamically, giving the user a high degree of flexibility. As a worst-case condition, the system may limit the user to 99,000 bytes per file record. However, the system loads into memory only those sets referenced by the job, thereby effectively permitting a significant expansion of record size.

During the file definition process, the user may desire to specify in advance certain automatic functions such as conversion of retrieval literals to coded file values, output data conversions or editing. If file indexing is desired, the user indicates which field or fields will provide the index values. The FS component provides mechanisms for this purpose and records these characteristics in the basic File Format Table (FFT). Record control or key fields may be defined as a single field or a string of multiple fields. The only limitation is that the total record control group may not exceed 244 characters. Periodic subsets may also have data fields assigned for sequencing and control within the record. If the user does not desire to provide these subset control elements, the system will create sequence numbers for the control function and will maintain them as a normal part of the FM function.

FS provides a simple, convenient way for the user to specify his file format and requirements. Using a simple free-format language which uses default options, the user can specify the logical data associations he desires.

Additional limitations to the FS component are as follows:

- A maximum of 50 subroutines/tables may be defined for a file.
- A maximum of 50 edit masks may be defined, each not more than 69 characters, and the

aggregate total of edit characters must not exceed 1,000.

- o A total of 100 fields or groups may be defined for each set.
- o A maximum of 99 fields may be defined in a group.
- o A maximum of 50 groups may be defined in a set.
- o Coordinate fields may consist of 5, 6, 7, 8, 11, or 15 characters only.
- o General system conventions as to names and punctuation must be followed.

3.2 File Revision

The FR component provides the capability to revise the format in which data is stored in a NIPS data file. This is accomplished by the comparison of the FFT describing the file in its current format - to the FFT describing the file in its revised format. FR will generate, as a result of the comparison, FM logic statements which will be used by FM to copy selected data elements into the new format.

FR processes data on a field basis only. It will permit the addition, deletion, and relocation of fields as well as changes to their storage mode, size, and name. Periodic sets may be deleted or relocated. They may also be added, but no data will be included, as a direct result of FR, in either new sets or fields. FR will permit fields from any set in the old file to be split into several sets in the new file. It will not permit fields from multiple sets in the old file to be merged into one set in the new file. There is no way for FR to logically connect the subsets of two existing sets. The FM logic statements generated may be modified by the user to suit his needs.

The only limitation is the need, on the part of the user, to recompile his FM logic statements and any retrievals, outputs, and terminal queries which might be affected as a result of file revision.

3.3 File Maintenance

This component provides the user with a broad and flexible capability for generating and maintaining data files. Using FM component features, the user may add, delete, or change file records, periodic sets or subsets, and variable sets. Also, the user may modify or change file fields and may change (increase or decrease) the volume of data associated with any file record. If indexing is specified for the file, the index data set is also maintained. This maintenance is automatic and transparent to the user. These functions may all be accomplished in parallel, or in a single pass of the master data file; they may be accomplished by virtually any mixture of unsequenced input transaction formats. Multiple file maintenance passes are permitted to support data dependent activities.

Input material may be records which are either fixed or variable in length and may be blocked or unblocked. Storage media for input material may be any supported readable media under the operating system.

All processing is controlled by logic statements provided by the user. These statements may be written in either a macro-like programmer-oriented language (FOCL) or a high-level procedural English-like language (NFL) which is easy to learn and simple to use, yet powerful and flexible enough to efficiently accomplish a wide range of file maintenance functions. The user may perform functions using data from the input transaction, the master data file record, records from other NIPS files, or all three. The Ordinary Maintenance (OM) feature of FM permits the user to write logic statements containing only OM transaction descriptor cards which can perform automatic validation of transaction data and unconditional updating of file fields. The OM statements can be combined with FOOL statements to provide additional user flexibility. All input/output functions are performed automatically by the component; since the languages are indirect, the user makes his references using the field names he has assigned as meaningful titles of data elements. The languages include instructions that permit automatic table validation or conversion or automatic linkage to user-provided, special-purpose subroutines. The indirect-addressing capability,

used for processing field-initiated corrections, is another FM feature.

In addition to maintenance (or generation) of a data file, this component permits the user to create auxiliary output files simultaneously with the maintenance process. The system will permit the generation of two independent print streams, two independent punch streams, and five auxiliary sequentially-organized output streams (tape or direct access volumes) in parallel with the prime processing function.

Generative code techniques are used in the actual data manipulation. To avoid wasteful regeneration, the system provides a complete, automatic library maintenance function for the user's logic statements. Maintenance execution, therefore, may consist of executing prestored logic statements, compiling logic statements and library updating, or combinations of these functions. There is no limit to the number of different statements that can be precompiled and retained in the library, or to the number of statements that can be used in a single file update run.

A data file may be maintained as an ISAM, SAM or VSAM data set. Large SAM files may be segmented into volumes of data records containing a range of record keys. These may then be updated and queried individually or in groups. For a SAM file that is indexed, volume control information is provided so that queries against the file need pass only those volumes containing the candidate records appropriate to the query instead of the entire data file.

Other component functions provide for the creation of summaries, run logs and/or audit trails during file processing. These functions are specified by the user and are not limited to a set of predefined system capabilities. They are as simple or complex as the user's requirements dictate.

Limitations to FM include the following:

- Maximum size of transaction record is 1,000 characters.

- A maximum of 24 logic statements may be compiled in any one execution of FM.

3.4 Retrieval and Sort Processor

The RASP component, primary retriever of the system, is characterized by its voluminous, batched-processing capabilities. Using generative code techniques, RASP includes an automatic library maintenance capability which permits broad utilization of the standing query. Compile-and-go modes of operation permit satisfaction of the ad hoc query requirement and complete intermingling of compile-and-go and standing query processing in a single job process is permitted.

RASP uses a simple English-like, free-format condition/action language which is flexible in notation. By specifying the retrieval condition, the user may select specific records to be retrieved from the data file. Comparison operators provided are the normal equal to, less than, greater than and between. All may be preceded by the negative operator "not". Boolean connectors are permitted, as are up to eight levels of nested parentheses. Two geographic retrieval operators are also provided: irregular area and circle search. The former is of interest in that it provides area-to-area, area-to-line, line-to-line, and point-to-area capabilities and is not constrained to convex area definitions.

The true power of the processor, however, is not directly indicated by the language. The scope of the processor is more directly related to its ability to provide many answer sets, each of different control or sequence, by a single pass of either the entire data file or only those records selected as possible candidates for retrieval by index processing. In RASP, the simplest qualification statement is called a condition. Satisfaction of the condition assigns the output of the file record (or a selected portion of the file record) to a temporary file for later system sequencing. Since the user must frequently output information from a file record in several sequences, each condition statement may have several associated sort statements. Each sort statement is simply the string of fields and/or literals upon which the user desires to sequence the answer records. Conditions may be prestored

machine core. Their primary function is simply to provide interface for TP application programs. Output generated by the application programs is made available to the terminal user for paging in the conversational mode.

NIPS provides the following terminal application programs or components:

a. Quick Inquiry Processor (QUIP) is designed to give the user a powerful data retrieval and display capability.

b. Source Data Automation (SCDA) provides the user with a means of executing a precompiled file maintenance logic statement against an update transaction entered by an operator at an online terminal.

c. Online Source Edit (EDIT) provides the ability to enter, manage and edit NIPS and ALC source code language statements at the terminal and to submit jobs to the operating system for batch processing.

d. Utility programs provide the ability to send messages and data between terminals and to communicate with the computer operator.

The QUIP language consists of a simplified output language and a subset of the RASP language. Although QUIP was designed primarily for terminal application, it can also be operated in the batch mode to provide a simple, efficient tool to process ad hoc queries. When executing QUIP as a problem program in the online terminal environment, it requires a direct access resident ISAM NIPS data file or a direct access resident SAM data file. A NIPS data file organized as either ISAM, SAM or VSAM organization (tape or disk resident) is acceptable when QUIP is executed in the background environment. Where possible, QUIP will attempt to limit the number of data records that must be examined in detail through the use of file indexing in the same manner as RASP. The QUIP language allows user or system formatted output lines. One and two dimensional sum and count matrixes can easily be specified in addition to computational operations. An additional capability allows the terminal user to select an OP RIT from the user library

and display pages of the output report on the terminal. QUIP provides an interfile output capability in the same manner as OP. Data elements extracted from secondary files may be combined with primary file data to produce a composite report.

The SODA component uses a free-format keyword language consisting of control statements which identify files, records, report formats, corrections, and disposition actions. Given a file, SODA will fetch the record identified in the update transaction, and provide both the record and the transaction to the library-stored logic statement which performs the required edit and update function. The record is returned to the data file in updated form under operator control. The user's input transaction is validated by the logic statement during execution. If errors are detected, immediate notification may be given to the terminal operator. The user may correct individual lines of his transaction, using the standard TP correction procedures and repeat the SODA request. The user may alternatively enter key-numbered corrections to individual fields and repeat the request. SODA will maintain file indexes, if present, in the same manner as FM. SODA cannot be used to perform file maintenance on a VSAM data base.

The EDIT component provides the capability to enter and maintain source language statements from the terminal. The source statements may be stored on a library and modified from the terminal. The modified source statements may then be edited and any errors reported back to the terminal operator. When errors have been corrected, a Job Control Language (JCL) stream may be constructed and submitted to the operating system as a background job.

3.7 Utility Components

A large collection of utility support programs is provided by NIPS. A data validation/conversion-table generation capability which assures efficient use of this important data processing tool is provided. Tables formed by the system are modular so that usage on minimum core environments may be achieved. Table sizes may effectively be considered up to approximately 140,000 bytes of argument/function pairs. They provide simple, effective

utilization of validation techniques not otherwise possible. Maximum combined size of the argument and function is 256 bytes.

A utility is provided to allow the placing of load modules, created by compiling user-written subroutines, on a NIPS library. The subroutines will contain the linkage necessary for proper utilization by NIPS. The subroutines may be written in Basic Assembler Language, COBOL, FORTRAN, PL/1, or JOVIAL.

Utilities are provided to convert the modes of NIPS files from ISAM or VSAM to SAM and vice-versa. Other utilities are available to specify and create indexes for existing files, and to transfer index data sets from disk to tape and vice-versa. Utilities are also available to analyze text for keyword selection and to create and maintain keyword dictionaries and tables. Another accepts a retrieval answer file and converts it to a data file containing all of the properties of a NIPS master file.

Section 4

SYSTEM FEATURES

To simplify the training of nonprogramming oriented users, the system makes extensive use of the stored procedure and cataloging capability of the operating system. This results in effectively eliminating the requirement for the typical user to code JCL streams to control his job. The skilled programmer may, however, continue to use fully the flexibilities provided by the operating system.

Modularity of programming ensures simplified maintenance and error correction in the operating software. Relatively unique code generation techniques ensure ease of modification in the user language areas, if needed.

All user-generated material required to support a data file is contained as part of the data file or user library. The FDT, FM logic statements, and data records are all a part of the data file itself. All conversion tables, subroutines, RASP queries, QUIP queries, and OP RITs can be on a single user library. The user's source material may reside in the same library or in a separate library. By causing the library to occupy a direct access device with the data set, a notable degree of file transportability is achieved. A system user may effectively move a single disk pack (data volume permitting) to another system operating with NIPS and be assured that he has all the materials necessary to maintain, retrieve, or display data from his file.

File analysis and run optimization statistics may be produced by FM, RASP, QUIP and OP upon demand. File analysis statistics assist a user in redesigning a file and selecting proper data elements for secondary indexing. Run optimization statistics assist in tuning production jobs to run most efficiently.

Eleven volumes of user-oriented documentation cover the system components, error codes, procedures, techniques and general file concepts. Program Specification Manuals are

available to assist installation system programmers in understanding system capabilities.

Section 5

OPERATING SYSTEM CONSIDERATIONS

NIPS will operate under five operating system configurations:

PCP	Primary Control Program. The standard operating system which processes one job at a time in sequence.
MFT	Multiprogramming with a fixed number of tasks. A partitioned core system allowing up to four jobs to operate concurrently in independent fixed-size partitions.
MVT	Multiprogramming with a variable number of tasks. A regional core system allowing up to 15 jobs to operate concurrently in variable-size regions.
VS1	Operating System/Virtual Storage 1. A virtual storage system which can perform 15 different tasks concurrently.
VS2	Operating System/Virtual Storage 2. A virtual storage system which can perform 63 different tasks concurrently.

NIPS is not restricted to any one level of the operating system and it will use subsequent releases as they become available. Multiprogramming capabilities of the operating system permit multiple jobs to operate concurrently in independent partitions, regions or virtual storage. The data management facilities of the operating system will handle all system and user data storage. NIPS data files can be organized under ISAM for direct access device files or SAM for tape or direct access files or under VSAM for direct access files or the S/370 operating under VS1 or VS2. Data file indexes, where applicable, are organized under the Basic Direct Access Method (BDAM) and reside on a direct

access device. The various system libraries will be stored using Basic Partitioned Access Method (BPAM). The Basic Graphic Programming Services of the operating system are used for the TP component that services the local IBM 2250 and 2260 display terminals. The Basic Telecommunications Access Method (BTAM) is required for remote device support and for local IBM 327C Information Display Systems.

In addition to the basic operating system, the following must be included at system generation time:

a. Programs

1. Assembler - F level
2. Linkage Editor
3. Sort

b. Libraries

1. MACLIB - Macro Library
2. SORTLIB - Sort Library

c. Data Management Access Methods

1. Sequential Access Method
2. Index Sequential Access Method
3. Basic Direct Access Method
4. Basic Partitioned Access Method
5. Basic Graphics Programming Services (needed only for terminal processing support)
6. Basic Telecommunication Access Method (needed only for remote terminal processing support)
7. Virtual Storage Access Method (available only on an S/370 with VS1 or VS2)

Section 6

HARDWARE CONSIDERATIONS

NIPS has been designed and programmed for an IBM System 360 Model 50H (256K core size). It will also operate on a Model 40H and larger models of the System 360 line and Model 135 and larger models of the System 370 line. Without online terminals, it will operate on a 128K core size machine, however, experience has shown that the H core is the minimum for complex applications on non virtual storage systems. NIPS can use magnetic tapes, direct access devices, card reader/punches, online printers, a console typewriter, and IBM 2260, 2265, 2250, 3277, 3275, 1050, and 2741 terminals

A minimum configuration could include three IBM 2311 disk units, (although an IBM 2314 or 3330 disk storage unit is considered highly desirable) a card reader, and an on-line printer. Tape requirements are related to the user's requirements and range from none to a quantity sufficient to perform the largest sort the user may require. Disk sorting is used to the capacity of the direct access devices available to the system.

The system will service a number of IBM 2848 display control units, each of which will support one 1053 printer and eight local 2260 terminals for online processing. Up to 32 IBM 3277 display stations may be attached to the system via each 3275, and 3272 control unit. An unlimited number of local 2250 terminals are supported under these configurations:

a. 2250 -1 Each terminal has its own control unit.

b. 2250 -3 Up to four terminals for each 2840-2 control unit.

Remote IBM 3277, 3275, 2260, 2265, 1050, or 2741 configurations must include appropriate communications facilities and 270X data adapter or transmission control units.