View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by NASA Technical Reports Server

X-564-72-65

PREPRINT

NASA TM X- 65874

PROGRAM DOCUMENTATION STANDARDS

(NASA-TM-X-65874) PROGRAM DOCUMENTATION STANDARDS D.A. Parker, et al (NASA) Mar. 1972 34 p

N72-23157

CSCL 09B

Unclas

G3/08 26831

DONALD A. PARKER

CLINTON A. FRUM

MARCH 1972

GODDARD SPACE FLIGHT CENTER

GREENBELT, MARYLAND

X-564-72-65

PROGRAM DOCUMENTATION STANDARDS

Donald A. Parker
Information Processing Division
Goddard Space Flight Center

Clinton A. Frum
Computer Science Corporation
Silver Spring, Maryland

March 1972

GODDARD SPACE FLIGHT CENTER
Greenbelt, Maryland

PRECEDING PAGE BLANK NOT FILMED

ACKNOWLEDGEMENTS

I would like to express my thanks to Carol Mannan*, Karen Hillman* and Orbie Jones for their cooperation and assistance in the preparation of this document.

*Computer Science Corporation employees.

PRECEDING PAGE BLANK NOT FILMED

ABSTRACT

This style manual has been developed to serve as a reference and guide for system and program documentation. It is intended to set standards for documentation, prescribing the procedures to be followed, format to be used, and information to be produced.

The standards for program documentation specify the extent to which the programmer should support his efforts in writing.

There are several excellent reasons for the necessity of clear and accurate documentation:

- It assists the program development process, saving rewrite time, and serves as a stepping stone to further program modification of programs currently operational.
- It assists in converting programs to a new machine, i.e., in the case of obsolete equipment.
- It provides programmer recognition.
- It serves as a record of the work performed.
- It increases the efficiency of the entire installation.
- It provides the program users with a basic understanding of the program and of the types of formulae, controls, etc., that are included. It also clarifies the purposes and outputs of the program.

Documentation users may be interested in programs from several points of view. Programmers who modify and update programs require the most extensive description of a program. Management, on the other hand, is often interested in only a general description of the program's function, for purposes of review. Operators are interested in a concise description of the program operating instructions and error recovery procedures to enable them to effectively run programs. Program users are interested in a technical description of the program, its input and output, to enable them to set up runs and interpret the results.

This standards manual describes several aspects of program documentation:

- System Description*
- Program Description
- Operation Description
- Machine Utilization
- Program Validation

The first three sections of the manual (System, Program, and Operation descriptions) contain information of particular interest to management, operators, and program users, respectively. The information in these three sections may tend to be repetitive in some respects, but each section has been designed as a self-sufficient description from the management, operator, or user point of view.

*Not to be confused with Operating Systems.

TABLE OF CONTENTS

		Page
Section 1 - System Description		1-1
1.1	Title Page	1-1
1.2	Program Sign-Off Sheet	1-1
1.3	Revision Page	1-1
1.4	Table of Contents	1-5
1.5	Scope and Purpose	1-5
1.6	Authority	1-5
1.7	Problem Definition	1-6
1.8	Problem Analysis	1-6
Section 2 - Program Description		2-1
2.1	Program Summary	2-1
2.2	Program Logic and Flow	2-1
2.3	Technical Description	2-2
2.4	Input	2-2
2.5	Internal Data Areas Description	2-2
2.6	Output	2-5
2.7	Controls and Constraints	2-5
2.8	Program Listing and Flow Charts	2-7
Section 3 - Operation Description		3-1
3.1	Operational Data Flow	3-1
3.2	Machine Set-Up	3-1
3.3	Input Set-Up	3-1
3.4	Operator Instructions	3-1
3.5	Disposition of Output	3-5
3.6	Restart Procedure	3-5
3.7	Program Fault Identification	3-5
	3.7.1 Halts	3-5
	3.7.2 Messages	3-6
	3.7.3 Console Alteration/Sense Switches/Alter Keys	3-6
	3.7.4 Operating Manual/Layouts	3-6
Section 4 - Machine Utilization		4-1
4.1	Program Language	4-1

TABLE OF CONTENTS (Cont'd)

		Page
4.2	Operating System and Version	4-1
4.3	Memory and Equipment Usage	4-1
4.4	Program Run Time	4-1
Section 5 - Program Validation		5-1
5.1	Statement of Method	5-1
5.2	Data Set for Input and Description	5-1
5.3	Program Output and Description	5-2
Section 6 - Appendix		6-1
Appendix		

SECTION 1

SYSTEM DESCRIPTION

1.1 TITLE PAGE

The title page should precede each system description, and should include the following items:

- Program name
- User name, address, and control number
- Programmer/Analyst name, title, and organization
- Date of publication
- Revision number
- Approver's name (Technical Monitor or his designated representative)
- Publishing organization's name and address

A sample title page is found on page 1-2.

1.2 PROGRAM SIGN-OFF SHEET

The program sign-off sheet should immediately follow the title page, and should include a list of people (denoted by their signatures) who must approve the program.

A sample program sign-off sheet is found on page 1-3.

1.3 REVISION PAGE

The revision page should immediately follow the program sign-off sheet, and should be kept current, indicating the dates of all revisions made to the document. It should include the following items:

- Program disposition status

TITLE PAGE

REPROCESSING REPORT PROGRAM

Prepared For

Bernard Narrow

Goddard Space Flight Center

Greenbelt, Maryland

Control No. 8654

Prepared By

Leon Cariaga

Programmer/Analyst

CSC

April 15, 1970

Revision No. 1

APPROVED BY: _____

PROGRAM SIGN-OFF SHEET

QUALITY CONTROL: _____

OPERATIONS: _____

ENGINEERING: _____

BRANCH: _____

OTHER (USER): _____

REVISION PAGE									
Disposition Status	Cost (man years)	Date	Prog/Anal Name	Section Number	Page Number	Revision	Revision Number	Reason	Originator of Revision
Prod.	10.83	8/1/62	T. Jones	2.1	9	Modified EDIT flowchart.	1	Bug	J. Doe
Prod.	8.63	11/15/62	R. Smith	4.4	21	Changed CONTROL to allow an input area of 2500 words.		To permit a larger input base.	J. Doe
Prod.	3.98	4/1/63	Proj. Sup.	3.4	18 19 21 48	Revised output to contain listing of input parameters; input to include tape parity.	3	Improvement	R. May
Prod.	4.012	4/15/63	Proj. Sup.	3.4 Oper. Inst. Sheet	" 33A 8	Monthly entries now input from tape, not cards; operator instructions thus changed.	4	Revision in data base.	R. May

- Revision cost (In terms of man power)
- Date
- Programmer/Analyst's name
- Section number
- Page number
- Revision number
- Revision abstract
- Revision reason
- Revision originator

A sample revision page is found on page 1-4.

1.4 TABLE OF CONTENTS

A table of contents should immediately follow the revision page and should list all major areas of the document.

1.5 SCOPE AND PURPOSE

This discussion is directed toward the layman user, explaining the system's purpose and function, the system's desirability, and the general manner in which the system's goals are achieved. An example of such a discussion follows on page 1-6.

1.6 AUTHORITY

If possible, a copy of the document authorizing the program development should be obtained and included as part of the documentation. Inclusion of this information will indicate the level of authority authorizing the development effort. If the program was developed by the Government, it should show the branch or division authorizing the activity. If the program was developed by a private contractor, a copy of the Statement-of-Work should be included. The

SCOPE AND PURPOSE

The use of large data bases involves considerable problems. Their creation is subject to a large range of errors, and often they must be periodically supplemented with new information in order to retain their utility. The following program is designed to assist the user in maintaining large files stored on magnetic tape, provided the file format adheres to certain restrictions. All entries in the file must be identical in format and must be written in fixed-length blocks on the magnetic tape. The file also must contain no duplicate entries. In any one use of the program, the user may add a new set of entries to an existing file, determine those entries not meeting user-specified restrictions, and/or correct (replace or delete) individual entries of a file.

Statement-of-Work contains the contract number, group, and branch authorizing the development effort, as well as a description of the work to be performed, and program specifications.

1.7 PROBLEM DEFINITION

The problem definition technically describes the problem to be solved in terms of system development (i.e., this is not a program description). Basic equations and formulae, as well as system flowcharts, may be presented. A glossary of terms unique to the problem definition should also be included. A sample of the type of input and output should be given, including input/output source and form. It should be kept in mind that both the problem definition and problem analysis are aimed at the management group which uses documentation for the purposes of review and information only.

A sample problem definition follows on page 1-7.

1.8 PROBLEM ANALYSIS

The problem analysis indicates the manner in which the problem is to be solved in terms of system development (i.e., this is not a program description). A flowchart showing the major operation symbols in their logical context within

PROBLEM DEFINITION

The problem involved in this program concerns the maintenance of large data bases. Maintenance consists of three activities:

a. Editing (determining those entries of a file not meeting user-specified restrictions)
b. Correcting (replacing or deleting individual entries of a file)
c. Adding (adding new sets of entries to an existing file)

The problem, then, is to provide a system which will carry out these three activities.

The problem is solved for specific types of data bases: those stored on magnetic tape in fixed-length blocks and having identically formatted entries, no two of which are identical in content.

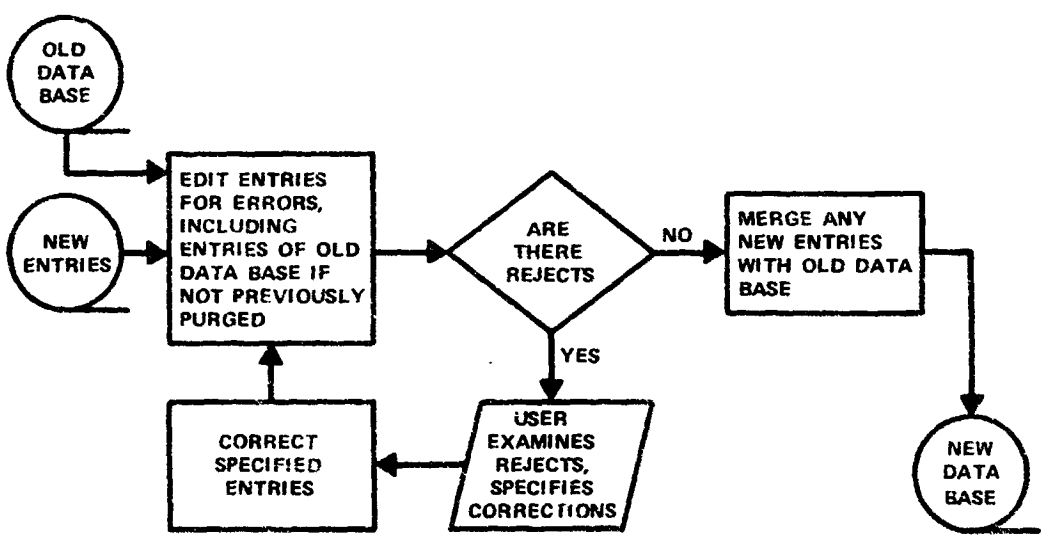
the general construction of the problem should be included. The flowchart should depict the major decision points, inputs and outputs, major starts and stops, and other relative points.

A sample discussion of problem analysis follows on page 1-8.

PROBLEM ANALYSIS

This data base maintenance problem consists of a series of steps. It might be conceived as a never-ending process, starting with the creation of a file and continuing with each update.

The initial problem involves editing and correcting a large master file. As each set of new information arrives, it must also be edited and corrected. It is then merged with the master file. Since the user may make mistakes in his initial correction of a file, he must be allowed to re-edit a file in order to determine if it needs further correction. The following flowchart illustrates the series of steps involved in this problem:



SECTION 2

PROGRAM DESCRIPTION

2.1 PROGRAM SUMMARY

The program summary contains a short description of the program. A separate summary containing a brief description of each associated subroutine is also required. The program and subroutine summaries should include their purposes and functions in relation to the entire system development.

An example of the program summary follows.

> SUMMARY - PROGRAM CONTROL
>
> This system of programs involves three types of tasks: editing, correcting, and adding to a data base. These tasks may or may not be sequential, i.e., a particular use of the program may involve editing and adding to the original data base, without making corrections to any information already included in the data base. The user must also examine the edit output before he can specify any changes to a given data base.
>
> To allow the user to perform any one of the three maintenance operations in any sequence, a CONTROL program is provided. The user specifies the sequence of tasks to be performed. CONTROL then relinquishes control to a subprogram or subprograms, indicating the order of the tasks to be performed.
>
> CONTROL also examines the user's specification of his data base format to determine if the program has enough storage room to handle the data base.

2.2 PROGRAM LOGIC AND FLOW

The program logic and flow should be depicted in a detailed flowchart. The interaction of subroutines, executive routines, etc., should be included.

Standard template symbols (see Appendix) should be observed to indicate:

- Type of operation to be performed
- Initial conditions
- Program modifications
- Explanatory information
- Connection links

This flowchart should develop in detail the contents of each major operation symbol, i.e., arithmetic storing, comparing, branching. Each symbol and its contents must give the necessary information so that the program may be written directly from this flowchart.

An example of a detailed flowchart is shown on page 2-3.

2.3 TECHNICAL DESCRIPTION

The technical description should present all program details, techniques, etc. The technical description should include each program's purpose and function within the system; the type of inputs and outputs; examples of the calculations performed; special program considerations, especially in relation to other programs; and a glossary of terms unique in meaning to this program.

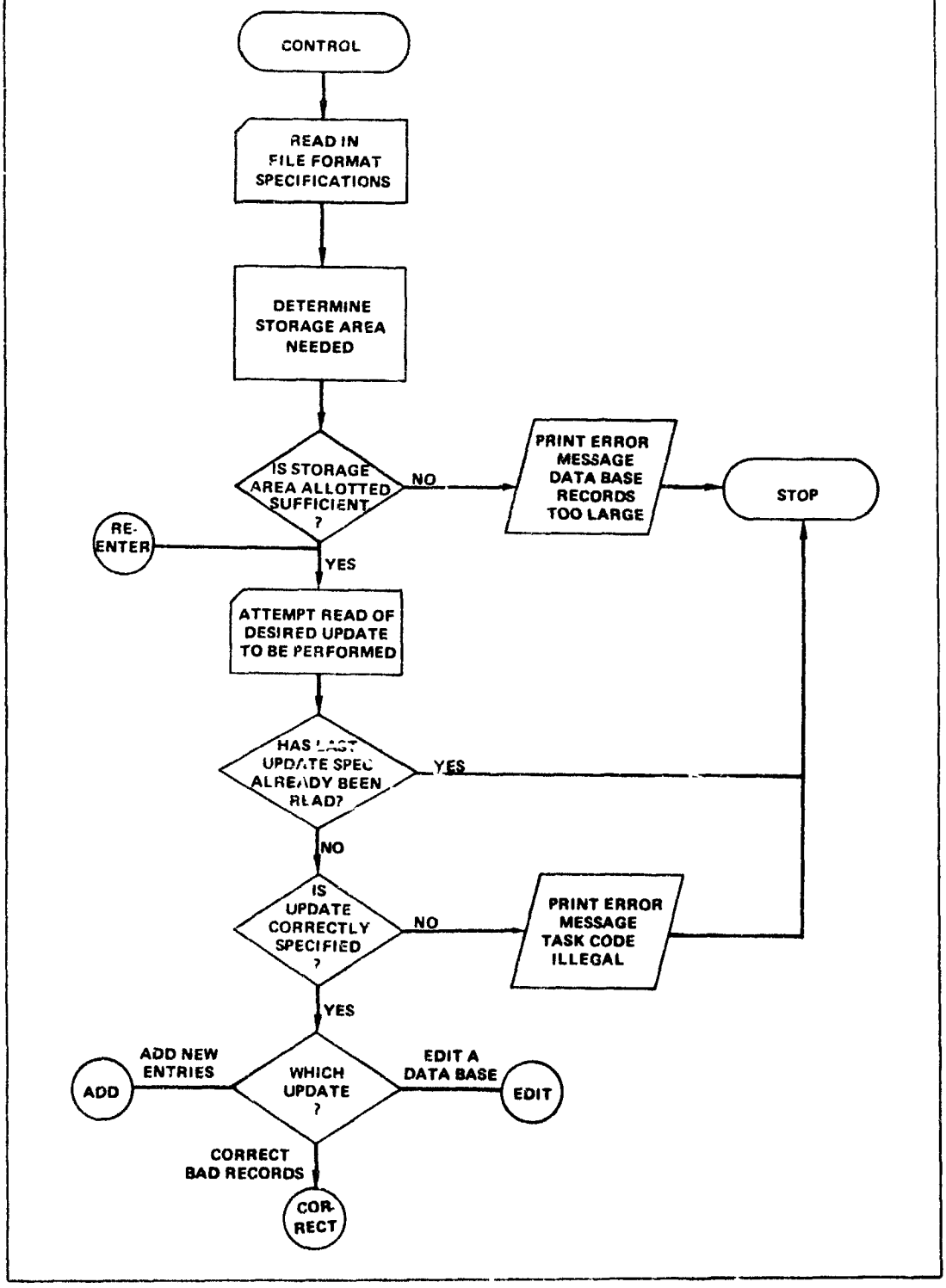
2.4 INPUT

All input should be described in detail, whether card, tape or other, together with a sample of the input type and a description of the input structure. Fields may be described on layout forms. The type, source, composition, limits, and volume should be specified. Sample input should be included in the appendix. A sample card input description follows on page 2-4.

2.5 INTERNAL DATA ARRAYS DESCRIPTION

A list of data arrays should be given, including their functional name; description; the name by which they are referenced; their location relative to the program, subroutine, or common area and position with array or word in

PROGRAM LOGIC AND FLOW



INPUT

1. INPUT CARDS

The data base format and order of tasks to be performed are input on data cards.

a. Card 1

Column	Variable	Description	Name	Format
1-5	Blocking factor	Tells the program how many entries to expect in each section of the data base.	NL	5-digit integer right-justified
6-10	Entry length in characters	Tells the program how long each entry in the data base is, allowing the program to set up an array into which entries can be read.	LR	5-digit integer right-justified

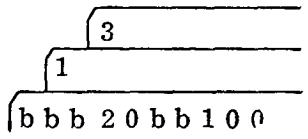
b. Card 2

Column	Variable	Description	Format
1	Update choice	Used as a flag to determine which update task the user wants to carry out. 1 means edit, 2 correct bad records, 3 add new entries.	1-digit integer

2. DECK STRUCTURE

a. JOB
b. FTN
c. Source deck
d. LOAD
e. RUN
f. Data Card 1
g. Data Card 2
: repeat g as often as desired
:
h. EOF

3. EXAMPLE



a. 20 logical records/block
b. Logical record length = 100 words
c. Perform an edit; new entries to master file.

which they are stored; and the type of information stored in them. Data arrays used for more than one type of information or variable should be so indicated. Also, equivalence of arrays or parts of arrays should be noted. A sample description follows on page 2-6.

2.6 OUTPUT

All output should be described in detail, together with a sample of the output produced. Fields may be described on card layout; layout forms should be printed. The type, source, composition, limits, and volume should be specified. The sample output should be included in the Appendix. A sample output description follows on page 2-7.

2.7 CONTROLS AND CONSTRAINTS

In a large program or system of programs, failure of the machine, software, or particular subprogram may not necessitate a complete re-running of the program. The control points at which a program can be restarted should be discussed in detail. Any changes in the input and output created by a restart at a particular control point should also be indicated. The built-in program options should be discussed, and their effect on the restart procedure should be noted.

The program constraints should also be discussed here. Any limitations on the type of input, output, or problem the program is designed to solve should be indicated.

INTERNAL DATA ARRAYS DESCRIPTION

VARIABLE	DESCRIPTION	NAME	LENGTH DECIMAL	LOCATION
Input area	This array is used to store a section of entries extracted from a data base.	IN	2000	COMMON/AR/+0
Edit Masks	This array is used to extract fields from data base entries and compare them to the correct values specified for the fields.	IREQ	300	COMMON/8/+0
Correction cards	This array is used to store the images of erroneous entries and the corrections to be made to them.	ICORR	290	COMMON/8/+10
Edit values or bad entries	This array is used to store the input cards that give the correct values for fields of entries in the data base, and then to store bad entries discovered in the data base.	IERR	200	COMMON/OAR/+10

NOTES: 1. ICORR (1) through ICORR (290) locations are equivalenced to IREQ (11) through IREQ (300) locations.

2. IERR is used by EDIT as an input area for record form specifications which are converted to masks and stored IREQ, leaving IERR free as an output area for erroneous records.

OUTPUT

1. SAMPLE

```
b b b b b 2 4 0 1 6 J O H N S O N , b A N D R E W
                    *
b b b b bX 3 2Q 3 9 7 O L D B O W
           * * *



b b b b b 9 3 8 3 4 7 H A L L E Y , b S A M U E L

b b b b b 8 3 6 1 3 4 2 8 O L D  E A S
                  *
```

2. DESCRIPTION

The erroneous cards are printed out, 3 words per line, for as many lines as the logical record length requires. (In the example, 5-word records require 2 lines.) Each line of the record is followed by a line with stars underscoring the erroneous fields. Logical records are separated by four blank lines. Each line of a single record, with the line's associated underscore, is separated from other lines by a single blank line. Up to 50 erroneous records can be printed out.

3. FORMAT

Record lines are printed out, using the following format:

FORMAT (5X, 3A8)

2.8 PROGRAM LISTING AND FLOW CHARTS

An up-to-date listing and program comments and program flow charts will be prepared and bound in a separate document.

SECTION 3

OPERATION DESCRIPTION

3.1 OPERATIONAL DATA FLOW

The operational data flow should be depicted in a flowchart geared for machine operation. The flowchart has a multiple purpose:

- It defines the input and output.
- It establishes the timing for a sample run, to assist in scheduling and program testing.
- It provides a brief synopsis of the program functions.
- It lists the minimum configuration on which the program can be run.

A sample flowchart follows.

3.2 MACHINE SET-UP

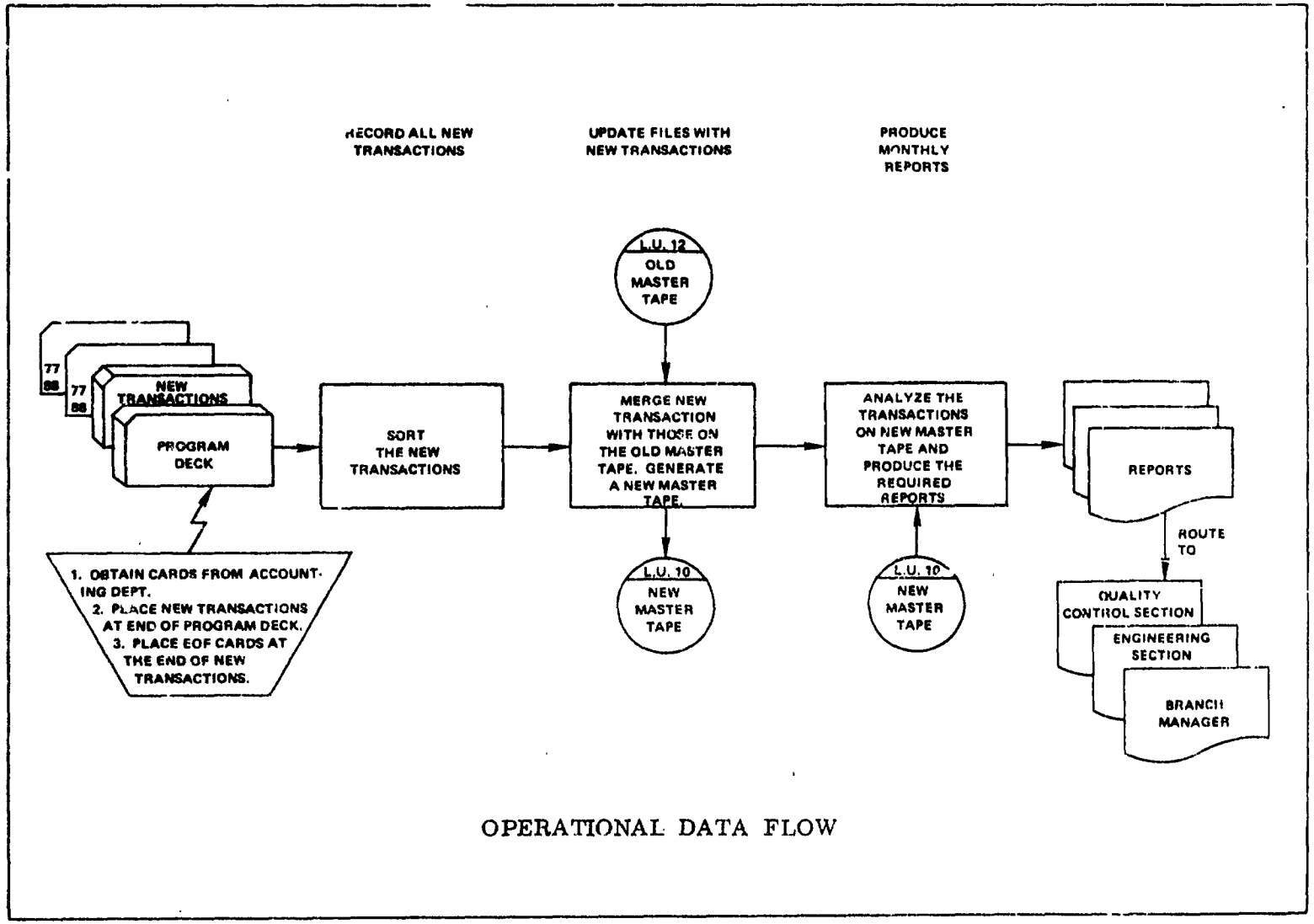
The machine set-up description should provide the operator with a check list of all required inputs and outputs. It shall specify the printer forms; carriage tape; format and sequence of input cards; and the complete set-up of all console switches as a function of the computer and the complexity of its operating environment. A example set-up summary is illustrated on page 3-3.

3.3 INPUT SET-UP

A list of sequenced set-up instructions should follow the machine set-up summary. Any limits on the system configuration (i.e., SYSTEM TAPE is always assumed to be on channel 1, unit 0) should be specified. An example follows on page 3-4.

3.4 OPERATOR INSTRUCTIONS

The operator must be apprised of all messages, halts, and end-of-job conditions which occur in the normal operation of the machine; he can thereby



OPERATIONAL DATA FLOW

3200 INSTRUCTION CARD

RUN NO 1 TIME 1 hr 30 min DATE 7/13/70

NAME DPB PROGRAMMING SECTION CODE 564 PHONE 6419

1.2 R.T
☒ SCOPE ☐ COMET

PRODUCTION PROGRAM

JUMP ☐1 ☐2 ☐3 ☐4 ☐5 ☐6

REAL TIME APPROX TIME STATION ☐ANALOG ☐PRINT ☐PUNCH ☐CARD READER ☐SELECT STOP

	ANALOG		CHANNEL 1			CHANNEL 5		
	A	B	UNIT- 0	UNIT- 1	UNIT- 2	UNIT- 1	UNIT- 2	UNIT
TAPE NO.	6017		SCOPE TAPE	SCRATCH TAPE	SCRATCH TAPE		SCRATCH TAPE	

LOG NO

I P 0 0 5
SPONSOR NUMBER

I N C J B
PROGRAMER ID

E N G 3
PROJECT NUMBER

4 C
CATEGORY TYPE

I A 0 0 0 0
PROGRAM NUMBER

ERROR PROCEDURE

TO TERMINATE NORMALLY, PLEASE DUMP THE COMPUTER MEMORY IF THE PROGRAM FAILS.

SPECIAL INSTRUCTIONS

1. PLEASE SAVE ALL BINARY CARDS PUNCHED BY THE PROGRAM.
2. PLEASE SAVE THE COMPUTER LISTING FROM THE PRINTERS.
3. PLEASE RETURN ALL OUTPUT PRODUCED BY THE PROGRAM TO THE DISPATCH DECK, TO BE PLACED INTO THE OUTPUT BIN ASSIGNED TO CARLOS BROGLIO.

INPUT SET-UP

1. Check for special instructions contained on back of run card.
2. Mount Deposit Accounting Master File (most current date) on channel 5, unit 1.
3. Place 3-ply stock paper in the printer.
4. Request the program from the 3200 dispatch deck.
5. Place program deck in the card reader, followed by the input parameter cards.
6. Place blank cards in the card punch.

rapidly recognize any abnormal conditions which exist and can take the required action with a minimum of waste time.

The operating sequence instructions must be specified, in order of performance. A sample is shown below.

OPERATOR INSTRUCTIONS

Initiate operation:

a) Press AUTOLOAD key to load the system.
b) System will request data; respond and press the "FINISH" key. The system will then be in operation.

Take-Down Procedure:

a) Remove the listing from the printer.
b) Remove the cards from card punch.
c) Remove and save the type out and parameter cards.
d) Remove the master file from Servo on channel 5, unit 1.

3.5 DISPOSITION OF OUTPUT

Normally all output is sent to the dispatcher's desk. In the event that multiple copies of output are produced and are to be delivered to specific areas, e.g., Quality Control, Data Inspector, etc., addresses should be designated.

3.6 RESTART PROCEDURE

The operator must be informed of the sequence of steps to take, in the event of a machine, software, or program failure. He should be aware of any control points within the program from which he can restart the run. He also should be apprised of any output that can be salvaged.

3.7 PROGRAM FAULT IDENTIFICATION

The operator must be apprised of all messages, halts, and end-of-job conditions which occur in the normal operation of the machine; he can thereby rapidly recognize any abnormal conditions which exist and can take the required action with a minimum of wasted time.

3.7.1 Programmable Halts

If the programmed halt is included in the program, it should be carefully documented, using a separate page for each halt. A sample follows below.

PROGRAMMED HALTS

Program No. _____

Halt Description _____

Halt No.	"A" Register	"Q" Register	"P" Register

Message:

Cause:

Corrective Action:

If more than three halts are included as part of a program, a separate index of halts proves extremely useful to the operator in trying to locate the halt in the manual. For example:

Halt No.	Cause	Page
0112	Printer line-up	40
2114	Recurring tape error	75
2234	Tape label error	92

3.7.2 Messages

A listing of all messages should be provided for each program. If a message is followed by a program halt, the halt number should be referenced in the message. The cause of the message should be explained, and if a halt occurs at the same time, the operator action should be indicated. Since halts are documented separately from the messages, the page number of the halt should be indicated in the message explanation.

3.7.3 Console Alteration/Sense Switches/Alter Keys (if applicable)

The documentation should clearly state the purpose of each console switch, and its effect on any section of the program, since it could be turned ON accidentally during program operation.

3.7.4 Operating Manual Layouts

The operating section should be provided with layouts of the input cards and output forms to enable rapid location of input/output errors and to prevent the use of the incorrect card file or incorrect report form.

Tape records or memory layouts do not have to be included in this section since they serve no useful purpose to the operator. However, sample reports and card forms should be included wherever possible. The last layout in the operating manual is the layout of the carriage tape for the printer, if a special tape is required.

SECTION 4

MACHINE UTILIZATION

4.1 PROGRAM LANGUAGE

The source language(s) in which the program was written should be specified.

4.2 OPERATING SYSTEM AND VERSION

The particular machine(s) for which the program was designed and the version of the operating system under which it must be run must be stated.

4.3 MEMORY AND EQUIPMENT USAGE

The programmer must supply information regarding all peripheral equipment used by the program, i.e., card reader, card punch, tape drives, printer, disk, etc. He should also supply a detailed map of areas of core used. An example of a core usage description follows.

Routine	Starting Location	Ending Location	Length (Decimal)
PROGRAM MAIN	77044	77777	733
SUBROUTINE EXEC	75620	76043	223
SUBROUTINE AQVW	73500	75617	2117
COMMON/DATA/	76044	77043	777
COMMON/2/	16000	16273	273
System Routines	00000	15777	15777
			Total 19900

4.4 PROGRAM RUN TIME

A summary of approximate computer timing should be provided by the programmer. The computer time should be broken down into compilation and run time. The run time, if variable, should be estimated on the basis of sample runs. Estimates of operator run time should also be stated. Operator run time may be broken down into input set-up, execution, and take down times.

SECTION 5

PROGRAM VALIDATION

5.1 STATEMENT OF METHOD

Since the program may be used by different people, some indication should be provided as to what extent the program has been checked out. Therefore, a complete description of the validation procedures used should be included. Examples of the data used in the validation of the program (i.e., the type and series numbers of the data tapes that were used) and the corresponding results should be provided to aid the user in verifying the accuracy of the program. Any future modification to the program should be tested, using the input data provided in the example. These test results should then be compared with the previous test results.

It is often not possible for the programmer to check out all the combinations of conditions that might occur in the execution of a program. However, the greater the number of conditions illustrated by a sample, the more valuable it becomes.

In summary, the purpose of the validation section is to describe exactly what occurred during the running of the program and the way in which the program goals were accomplished.

5.2 DATA SET FOR INPUT AND DESCRIPTION

The programmer must give an example(s) and describe the set(s) of input data used in debugging the program. Where it is practical, a brief description should accompany the data set, indicating the portion(s) of the program tested by the input data set. The data set selected for inclusion should be carefully designed to test the program as comprehensively as possible. It may be necessary to include a series of data sets to fully accomplish the validation exercise. This is particularly important, since future program modifications will be validated by testing the program with the examples provided in the program documentation. Card layout forms may be used to describe fields of the input data. The manner in which the data was input should also be specified. Any information that is beneficial and pertinent to the program testing should be included in this section of the program documentation. It should be understood that the content and depth of the discussion will naturally depend upon the complexity of the program being documented. In the event that the validation is

obvious or does not apply, this section of the documentation will contain a statement indicating that it is not applicable.

5.3 PROGRAM OUTPUT AND DESCRIPTION

The output of the sample run should be included here. The method in which it was output should be described; the output should also be interpreted. Criteria establishing that the output is valid should be stated.

SECTION 6

APPENDIX

Any items in program documentation which require extensive or special discussion in regard to purpose or use, i.e., mathematical concepts or items not within the specific categories delineated in this style manual, will be handled separately in appendix form.

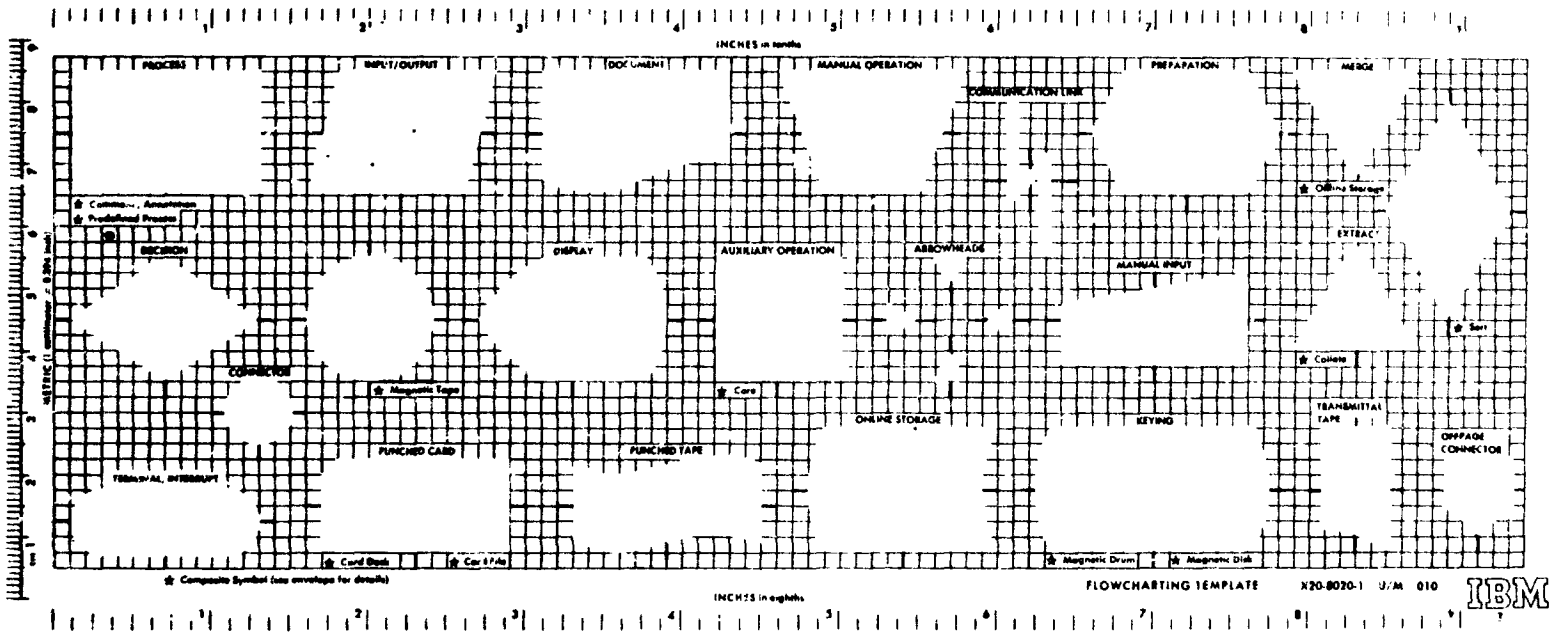
A program description paralleling that of Section 2 of specialized subroutines called by the program and of any support programs should also be included in the appendices. If the programs or subroutines have been documented elsewhere, a brief description of their functions and references to the documentation should be given instead. Input and output samples, as mentioned in Section 2, may also be included in appendix form.

A glossary of all names (variable, subprogram, and label names) used in the flowcharts and program should also appear in the appendix.

A list of pertinent reference material should be provided.

APPENDIX

The standard IBM template symbols for flowcharting use are shown on page A-2.



STANDARD IBM TEMPLATE

NASA–GSFC