



Energy, Mines and
Resources Canada

Énergie, Mines et
Ressources Canada

CANMET

Canada Centre
for Mineral
and Energy
Technology

Centre canadien
de la technologie
des minéraux
et de l'énergie

This document was produced
by scanning the original publication.

Ce document est le produit d'une
numérisation par balayage
de la publication originale.

STANDARD PROCEDURE FOR THE DOCUMENTATION OF COMPUTER PROGRAMS

VERSION I

R. SABOURIN

MINE EVALUATION GROUP

MINING RESEARCH LABORATORIES

MAY 1979

ENERGY RESEARCH PROGRAM
MINING RESEARCH LABORATORIES
REPORT NO. ERP/MRL 79-53(TR)

MICROMEDIA

79-5550

18 pp / 1 fiche

WICKOWEDIA

TABLE OF CONTENTS

| | <u>Page No.</u> |
|---|-----------------|
| 1. INTRODUCTION ----- | 1 |
| 2. STANDARD DESCRIPTION OF A COMPUTER PROGRAM ----- | 2 |
| 2.1 Definition of the Problem ----- | 2 |
| 2.2 Description of the Main Program ----- | 2 |
| 2.2.1 General Description ----- | 2 |
| 2.2.2 Capacity of the Program ----- | 3 |
| 2.2.3 Input ----- | 3 |
| 2.2.3.1 Input Control Cards ----- | 3 |
| 2.2.3.2 Option Cards ----- | 3 |
| 2.2.3.3 Data Cards ----- | 3 |
| 2.2.4 Output ----- | 3 |
| 2.2.5 Operating Details ----- | 4 |
| 2.2.6 Appendix ----- | 4 |
| 3. STANDARD DESCRIPTION OF A SUBROUTINE ----- | 4 |
| 3.1 Mathematical Background ----- | 4 |
| 3.2 General Description ----- | 5 |
| 3.3 Listing of the Subroutine ----- | 5 |
| 4. STANDARDS FOR THE CREATION OF A PERMANENT FILE ----- | 5 |
| 4.1 Definitions of Disk File Parameters ----- | 5 |
| 4.2 Disk File Standard Labels ----- | 6 |
| 4.2.2 Permanent File Name ----- | 6 |
| 4.2.3 Identifier Name ----- | 7 |
| 4.2.4 Retention Period ----- | 8 |
| 4.3 Definition of Tape File Parameters ----- | 8 |
| 4.4 Tape File Standard Labels ----- | 9 |
| 4.4.1 Label Name ----- | 9 |
| 4.4.2 Multi-File Tape ----- | 9 |
| 5. STANDARDS FOR COMMENTARY STATEMENTS IN A PROGRAM ----- | 10 |
| 5.1 Comments in a Program ----- | 10 |
| 5.1.1 General Description ----- | 11 |
| 5.1.2 Program Options ----- | 11 |

Table of Contents (cont'd)

| | <u>Page No.</u> |
|--|-----------------|
| 5.1.3 Dimension and Common Declarations ----- | 11 |
| 5.1.4 Main-Line Program Body ----- | 11 |
| 5.1.5 Format Specifications ----- | 12 |
| 5.2 Comments in a Subroutine ----- | 12 |
| 5.2.1 General Description ----- | 12 |
| 5.2.2 Parameters and Variables Definitions ----- | 12 |
| 5.2.3 Dimension and Common Declaration ----- | 12 |
| 5.2.4 Main-Line Subroutine Body ----- | 13 |
| 5.2.5 Format Specifications ----- | 13 |
| 6. STANDARDS IN USING THE SYSTEM UPDATE ----- | 13 |
| 6.1 Control Cards to Implement Corrections ----- | 13 |
| 6.2 Labels For Correction Decks ----- | 14 |
| 6.3 Labels For a New Source Deck ----- | 14 |
| 6.4 Names of Programs and Subroutines ----- | 14 |

STANDARD PROCEDURE FOR THE DOCUMENTATION
OF COMPUTER PROGRAMS

by

R. Sabourin*

1. INTRODUCTION

The Mine Evaluation Group (MEG) has established, during the past three years, two computer program systems which include FORTRAN and COBOL language programs. These systems are the C.A.D.D. System (Collection and Analysis of Drill-Hole Data) and the Geostatistical System.

The continuous development of new computer programs by MEG necessitates the use of standard procedure for the correction of programs and for the creation of computer files. Furthermore, to facilitate their understanding, a full description of each program must be made available to future users. Obviously the documentation of all these programs should therefore be consistent and follow certain description standards which will be explained in the second section of this report.

The purpose of this report will thus be to suggest a standard procedure to describe computer programs. A convention for the creation of data files, for the introduction of comments in a program and for the usage of the system update will also be proposed.

* Geostatistical Engineer, Mining Research Laboratories, Canada Centre for Mineral and Energy Technology (CANMET), Department of Energy, Mines and Resources, Ottawa, Ontario, Canada.

2. STANDARD DESCRIPTION OF A COMPUTER PROGRAM

Clarity and conciseness are important objectives to bear in mind when describing a program.

First, too many technical terms or unconventional abbreviations which confuse the reader must be avoided. Selection of proper terms is also important.

Secondly, concise and short sentences must be used. One paragraph should not contain more than one idea. Sentences should be formed of informative statements.

It is hoped that the standard procedure which follows will assist in the documentation of computer program reports.

2.1 Definition of the Problem

The problem should be described concisely. Basic mathematical definitions should be given and references should be stated in this section. The purpose of the program (why the program was created) should also be mentioned.

2.2 Description of the Main Program

In general, the main program should not involve calculating procedure. Rather, the program should be divided into several steps each performing a set of specialized computations. Those computations are then performed by sub-routines.

2.2.1 General Description

This section should give the basic features (what kind of results are produced) of the program. Important options of the program should be mentioned. A general explanation of the programs' calculations should be given.

2.2.2 Capacity of the Program

One must give the number of observations the program is able to use. The variables affected by dimensions should be specified and also special situations involving computer memory problems should be described.

2.2.3 Input

The input will normally consist of job control cards, option cards and data cards.

2.2.3.1 Input Control Cards

One or more examples of control cards set up will be given in this section. The control card parameters are explained.

2.2.3.2 Option Cards

This section describes the options in the following manner:

| COLUMNS | CONTENTS | FORMAT |
|-----------|---------------------------------------|--------------------------|
| From - To | Name and Description of the Option | Field Length And Type |

2.2.3.3 Data Cards

The data cards normally contain several observations taken at different times or locations in space. Each observation is punched between specified columns and each card repeats the same sequence of variables.

The sequence of variables on a data card must be described in the following manner:

| COLUMNS | CONTENTS | FORMAT |
|-----------|---------------------------------------|--------------------------|
| From - To | Name and Description of the Option | Field Length And Type |

2.2.4 Output

The output of the program will consist of results shown on a

computer listing. Sometimes, part of the results may be stored on a file.

This section of the report should give a detailed explanation of the results expected from the program.

2.2.5 Operating Details

This section should include such information as:

- The type of programming language used;
- The logical file names of input and output file;
- The expected executing time for a typical problem.

2.2.6 Appendix

Several appendices should be included in the report. These appendices will normally be copies of computer listings. The photocopies should be as clear as possible.

Appendix 1.0 - Listing of the Main Program (With Comment Statements)

Appendix 2.0 - Deck Set-Up For Input Cards. (If It Is Not Already Inside The Report)

Appendix 3.0 - Output Listing of the Results

Appendix 4.0 - Program Modifications

The last appendix should be the most recent version of the program. Any modifications should be clearly identified using the system update.

3. STANDARD DESCRIPTION OF A SUBROUTINE

The subroutines of a program should be described separately from the main program. In fact, several programs could use the same subroutine. The description of a subroutine would normally be shorter than the main program.

The general description can be put as a commentary at the beginning of the subroutine statements.

3.1 Mathematical Background

A summary, if necessary, of the mathematical computations involved in the logical algorithms of the subroutine should be given. The mathematical variables should be formulated. References could be noted in this section. As well as any programming considerations. The author and the date of creation should also be mentioned.

3.2 General Description

The general description should consist of:

- A) Purpose,
- B) Usage - CALL XX (ABC),
- C) Description of the Parameters,
- D) Remarks,
- E) Subroutines and Functions Required,
- F) Method and References.

3.3 Listing of the Subroutine

The photocopy of the subroutine listing should be as clear as possible. Commentary statements should be added in the subroutine.

4. STANDARDS FOR THE CREATION OF A PERMANENT FILE

We will first briefly describe the parameters of two control card language statements (CCL) which are used to create a disk and a tape file. MEG's standard labels of these parameters will then be given.

4.1 Definitions of Disk File Parameters

To create a disk file, one must use the "CATALOG" card. To access the information on a disk file, one must use the "ATTACH" card.

CATALOG, LFN, PFN, ID = name, passwords, retention period.

ATTACH, LFN, PFN, ID = name, passwords, multi-read.

It is obvious that the parameters of both statements are basically the same except for the retention period and the multi-read. These five parameters are the only ones normally used.

LFN: Is the logical file name by which the file is presently known to the job, 1-7 letters or digits beginning with a letter. If omitted, the first 7 characters of PFN are assumed.

PFN: The permanent file name by which the file is known in permanent file manager tables, 1-40 letters or digits. If omitted or null, LFN becomes the permanent file name.

ID = Name: A name is given by the owner or creator of file.

PASSWORDS: The passwords in use by MEG are:

CN: Password for control permission 1 - 9 letters or digits.

TK: Password for turnkey 1 - 9 letters or digits.

RETENTION PERIOD (RP): Indicating the number of days the file is to be retained, 0 - 999. Infinite retention is 999.
To be used only when a file is created (CATALOG).

MULTI-READ INDICATOR(MR): No other job can attach this file while this job is in execution.

(NON-ZERO DIGIT): Other jobs can attach the file immediately after for read purposes only.

4.2 Disk File Standard Labels

4.2.1 Logical File Name

LFN = tapeXX, XX is any integer number. We use tape XX for data files only.

Or

LFN = MEGXXXX, XXXX is the number (e.g. 2050) of a program in MEG's program library. It is used when a compiled copy of a program or a new program library (NEWPL) is needed.

4.2.2 Permanent File Name

The seven first digits (MEG1111) will only be used for a new program library or a compiled copy of a program.

PFN = MEG1111XXX2222T3333C4444455555SYY

1111, is the program number from which the data have been created.

XXX , represents MEG letter codes identifying a deposit.

2222, is the first drill-hole number of this file.

T , represents "to"

3333, is the last drill-hole number of this file.

C , stands for "corrected" information stored on the file

44444, represents the cut-off grade value. It will be used to identify different files of economic intersections calculated by the program MEG0100.

e.g. 0P5 is equal to 0.5 pounds per short ton.

55555, represents a mineralized zone of the deposits or any particular information about the data.

SYY, YY represents the number of sections the file contains.

4.2.3 Identificator Name

NAME = MEGXXXX

For Programs: The NEWPL will be stored on a magnetic tape unless corrections are being applied to the program. In this case, the OLDPL on disk will be purged and a NEWPL created after introducing each new deck of corrections. Once the corrections are completed, the old tape will be released and a new tape will be created with the NEWPL.

XXX = PB, for a binary compiled copy

OR

XXX = PL, for a new program library

For Data Files: When a data file is created, the ID name must identify the data as being binary or coded.

NAME = MEGDTB, for binary data

OR

NAME = MEGDTC, for coded data

4.2.4 Retention Period

The "AUDIT" or "NAUDIT" programs must be run each week to check the expiration date of disk files. The following rules apply for the retention period of newly created files.

- A) No disk files can have an infinite (999) retention period with the exception of the new program library and the compiled copy of a program.
- B) A retention period of 10 days is given for correction files. Automatic purging is done when a new correction file is created.
- C) A retention period of 30, 60 or 90 days can be given to any data files created from MEG's programs.

4.3 Definition of Tape File Parameters

The "LABEL" and "REQUEST" statements with their parameters are:
 LABEL, LFN, W or R RING or NORING, D = HY or PE, L = Z, VSN = vsn. REQUEST,
 LFN, MT or NT, HY or PE, RING or NORING, VSN = vsn.

Read or Write Label:

- R Label is to be read and compared with parameters on the label control statement.
- W Label is to be written.

Write Ring:

- Ring This parameter gives the permission to write on the tape.
- Noring The user is not able to write on the tape.

Density:

- D = HY 800 character per inch (CPI) for 7-track tapes.
- D = PE 1600 character per inch (CPI), for 9-track tapes.

Label :

L = Z Label name, 1-17 characters.

Volume Serial Number:

VSN = vsn , 1-6 characters used to identify the tape for automatic assignment.

MT is required on the request card, if a 7-track tape is used.

NT for a 9-track tape.

4.4 Tape File Standard

4.4.1 Label Name

The only standard for a tape file applies to the label parameter on the label card. The VSN number is defined by the installation when a new tape is created. Only nine track tapes are used.

Label = MEG1111 YYY 222 T 333

1111 , is the name of the program which has created the data file.

If the information to be recorded on the tape is a program, the following characters are not applicable.

yyy , represents MEG letter codes identifying a deposit.

222 , is the first drill-hole number of the file.

T , represents "to"

333 , is the last drill-hole number of the file.

N.B: the maximum number of characters for one label is 17.

4.4.2 Multi-File Tape

For a tape containing more than one sub-file, the following procedure is required:

A) REQUEST,MFN,NTI,PE,MF, E or N, VSN = vsn, RING OR NORING.

MFN , is the multifile name of the multifile tape.

MFN = MEG XXX

With XXX being the MEG letter code of a deposit for data files.

XXX , is not used for a multifile tape containing programs only.

MF , defines a multifile tape.

E , when we read a multifile tape.

N , when we create a multifile tape.

B) LABEL,LFN,R or W, M = MFN, L = LABEL, P = N.

M = MFN , is the multifile name given to the tape when it was created.

P , is the position of the file to be read or written.

If the user wants to add another file on the multifile tape, the new label must be given and W must be used instead of R. N.B.: A tape created from another installation must follow the foreign tape policy of the Computer Science Center (CSC) (see CSC User's Guide, section 3.7.2)

5. STANDARDS FOR COMMENTARY STATEMENTS IN A PROGRAM

This section describes a procedure to add comments in a program. The basic structure of a program will also be discussed.

5.1 Comments In a Program

A program should be structurally divided into the following five groups of statements:

- 1 - General Description of the Program
- 2 - Definition of the Program's Option
- 3 - Dimension and Common Declarations
- 4 - Main-Line Program Body
- 5 - Format Specifications

5.1.1 General Description

A brief description of the important features of the output results should be given. Names of subroutines and functions used as well as important steps of the program are declared.

5.1.2 Program Options

One must describe each option and the format in which it must be read.

5.1.3 Dimension and Common Declarations

The dimensions of each variable are declared with the real or integer fortran statements. Block commons should be used instead of blank common; the dimension of a variable should not be set with a block common. Dimensioned variables are defined in the main program.

The capacity of the program should be given (see section 2.2.2). The subroutines which include block commons should be declared before each of these common statements.

5.1.4 Main-Line Program Body

Every step of the program should be commented. The program should include tests indicating some typical errors introduced by the wrong selection

of program options. The selected options as well as the observations used by the program should be printed. Statement numbers should begin at one hundred and be incremented by five. The three branch arithmetic "IF", when possible, should preferably be used to any two branch or any logical "IF" statement.

The "PRINT" instead of the "WRITE" statement should be used to print information on the output file. The "READ" statement without logical unit should be used to input data.

5.1.5 Format Specifications

Format statements should all be at the end of the main-program or subroutines. The format statement numbers should begin at one and be incremented by one up to ninety-nine.

5.2 Comments In a Subroutine

Basically, the subroutines should be documented in the same way as the main program.

- 1 - General Description
- 2 - Definition of Parameters and Computed Variables
- 3 - Dimension and Common Declarations
- 4 - Main-Line Subroutine Body
- 5 - Format Specification

5.2.1 General Description

The description of the subroutine will be limited to its purpose or function. This purpose should be simple since the subroutine is created to solve only one part of the problem.

5.2.2 Parameters and Variables Definitions

The subroutine parameters should be fully described. Parameters which are also program options should be stated as such and referred to the description given in the main program. The most important variables calculated by the subroutine should be defined. These variables should normally be placed in a block common.

5.2.3 Dimension and Common Declaration

The dimensioned variables will also be declared with a "Real" or an "integer" statement. Parameters dimensioned by the main program will have a dimension of one except for parameters defined with two or three arrays. The exact dimensions of these parameters should be declared.

5.2.4 Main-Line Subroutine Body

This portion of the subroutine will follow the same rules explained in section 5.1.4 of this report.

5.2.5 Format Specifications

Same rules explained in section 5.1.5 of this report apply.

6. STANDARDS IN USING THE SYSTEM UPDATE

The computer programmers of MEG are using the system update to modify an existing program. This system keeps every deck of corrections on the new program library (NEWPL). This feature enables members of MEG to distinguish any program corrections from the original statements.

The following exemplifies the way to identify any deck of corrections as well as individual corrections.

6.1 Control Cards to Implement Corrections

Once a NEWPL is created, it is possible to bring corrections to the program. The following control cards will normally be used to add corrections on the NEWPL.

```
H1234, CM55000, T25, IØ200, P2.
ACCOUNT, XXXXX
ATTACH, ØLDPL, MEG ****, ID = MEGPL, PW = ***, ****, MR = 1.
ATTACH, LGØ, MEG****, ID = MEGPB, PW = ***, ****, MR = 1
UPDATE, N, Q.
FTN, I.
PURGE, ØLDPL.
PURGE, LGØ
CATALØG, NEWPL, MEG****, ID = MEGPL, TK = ***, CN = ****, RP = 999.
CATALØG, LGØ, MEG****, ID = MEGPB, TK = ***, CN = ****, RP = 999.
```

Note that the ATTACH and the CATALØG cards follow the conventions as set out in section 4.0 of this report.

"MEG****" is the program number.

6.2 Labels For Correction Decks

To identify a new correction deck, update uses the following statement: *IDENT DNAME. With DNAME = DXXYYMMDD
 where: XX, is the number of correction decks used during the same day plus one. YY, MM, DD is the year, the month and day of creation of the correction deck.

6.3 Labels For a New Source Deck

The name of the source deck for the main program will be "GEM****", where "****" is the program number.

The name of the source deck for subroutines or functions will be the same as the main program deck name, incremented by one. For example, if the deck name of the main program is GEM2100, the deck name of the first and second subroutines will be GEM2101 and GEM2102.

6.4 Names of Programs and Subroutines

The standard program names will be "MEG****."

The standard subroutine names will be the same as their main program, incremented by one. For example, if the name of the main program is MEG2100, the name of the first and the second subroutine will be MEG2101 and MEG2102, respectively.

