The DOCTOR system. A DOCumenTary ORiented programming system

J.K. Gibson

Research Department

January 1982

This paper has not been published and should be regarded as an Internal Report from ECMWF.

Permission to quote from it should be obtained from the ECMWF.



European Centre for Medium-Range Weather Forecasts Europäisches Zentrum für mittelfristige Wettervorhersage Centre européen pour les prévisions météorologiques à moyen

1. INTRODUCTION

DOCTOR is a programming standard born out of the advantages and shortcomings found by the author during his experience with other programming standards. Two standards in particular have been used extensively - the OLYMPUS system (Roberts, 1974) and the Control Data Corporation programming standard. In addition, many useful ideas and concepts are based on Frank's (1971) documentation processor. (DOCK, an Internal/External Documentation Processorm is a CDC Proprietary Product, copyright Control Data Corporation, T971). Since the DOCTOR system was first described (Gibson, 1980), it has been developed and modified, and is currently being used for ECMWF's new forecasting system. DOCTOR's application will be described with particular reference to FORTRAN. There is no reason why, suitably modified, some or all of its principles cannot be used in conjunction with any programming language.

elegación de emiliares de esta la esta companya de la factoria de la constanción del constanción de la constanción de la

1.1 Basic aims of the system and wroter transmission of the system and or

DOCTOR attempts to: bespecies of most sentendary was levered algore

- a) provide well presented code.
- b) produce source code following a standard structure.
 - c) set up points of reference for external documentation.
 - d) enable the inclusion within the source code of documentation which can be extracted mechanically.
 - e) allow maximum communication between routines by storing universal variables in structured pools or common blocks.
 - f) facilitate the recognition of variable types, and the differentiation between local variables, variables in common blocks, and dummy arguments to routines.
 - g) provide a set of utility routines for copying vectors, resetting arrays, etc.

1.2 Documentation

Documentation is a means of retaining the ability of code to be understood. The production of documentation is a skill at least as important as the skill required to design and generate the code itself. Good documentation increases the value of code - it assists maintenance, aids understanding, and can be invaluable if language to language re-coding should ever be necessary. One of the biggest problems with documentation is keeping it up to date. When documentation takes the form of typed or printed material there is always a delay in producing amendments. There is also a danger of changing the code, but not the documentation. This danger may be reduced if much of the documentation is incorporated in the text of the source code. If facilities are provided to extract the documentation from the source code, delays in producing amendments are overcome, and labour costs reduced. The DOCTOR system provides a simple set of conventions to enable sufficient documentation to be included in the souce code of each routine to enable it to be understood and used by another user. These conventions are such that simple extraction programmes may be written to process the source code and produce the documentation. In addition, routines are structured into numbered sections and sub-sections, thus facilitating cross references to lists of equations and symbols etc., which could not easily be included in the source code. Such additional documentation should be detailed in the section of the header comments titled REFERENCE (See 2.3.1 below).

2. CODING CONVENTIONS

Code should follow a modular structure, each module or routine fulfilling a stated purpose. Modules should be divided into numbered sections and sub-sections. Communication between modules should not involve long parameter lists - shared data should be made available through shared data pools, common blocks, etc.

2.1 Naming conventions

THE STATE OF THE S

The type of a variable is indicated by the first letter of the variable's name according to Fig. 1.

+ ! !	PREFIX	TYPE
!	I,J,K,M,N	! INTEGER
!	H 4.78	! INTEGER (used for HOLLERITH)
!	L (! LOGICAL TO THE PARTY OF THE P
• 42 20	4 .4 44.40	! CHARACTER
1	ALL OTHERS	1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 -
+		

Fig. 1 Variable types

In addition, a prefix convention is used to indicate the STATUS of a variable, to enable recognition of local variables, COMMON variables, loop control variables, dummy arguments, parameters, etc., according to Fig. 2.

A "Hollerith" type is included for ease of communicating character data via COMMON blocks and including characters in records to be read or written via unformatted input/output. Conversion between "Hollerith" and CHARACTER type would be performed in FORTRAN by means of internal files.

The default DOCTOR type convention may be established in FORTRAN by the statement:

IMPLICIT INTEGER (H), LOGICAL (L), CHARACTER *8 (Y)

LOCAL VARIABLES INTEGER I REAL Z INTEGER (HOLLERITH) HO LOGICAL LO CHARACTER YO REAL P(But not PP) INTEGER (HOLLERITH) HA LOGICAL LA CHARACTER YO PARAMETER INTEGER J(But not JP) REAL PF LOGICAL LP REAL PF REAL PF CHARACTER YP GLOBAL OR COMMON INTEGER ANY INTEGER PREFIX NOT INCLUDED ABOVE LOGICAL L, BUT NOT LO, LA, OR LP	STA: Para de la compania d Para de la compania del compania de la compania del compania de la compania del compania del compania de la compania del		. I . TYPE Pilopie nei tyeksieren en k	! PREFIX
INTEGER (HOLLERITH) HO LOGICAL LO CHARACTER YO INTEGER K REAL P(But not PP) INTEGER (HOLLERITH) HA LOGICAL LA LOGICAL LA CHARACTER YA LOGICAL JEUT NOT JP) PARAMETER INTEGER JP REAL PP REAL PP CHARACTER YP INTEGER LOGICAL LP CHARACTER YP REAL PP CHARACTER YP INTEGER ANY INTEGER PREFIX NOT INCLUDED ABOVE INTEGER (HOLLERITH) H, BUT NO HO, INTEGER LOGICAL LP INTEGER (HOLLERITH) H, BUT NO HO, INTEGER (HOLLERITH) H, BUT NOT LO, LOGICAL LA, OR LP LOGICAL LA, OR LP LOGICAL LA, OR LP				-+
INTEGER (HOLLERITH) HO		sus a Security and		
DUMMY ARGUMENTS LOGICAL LO CHARACTER YO REAL P(But not PP) INTEGER (HOLLERITH) HA LOGICAL LA LOGICAL LA LOGICAL JA LOGICAL J			I SECTION	1 3335999
DUMMY ARGUMENTS CHARACTER YO REAL		a stance of the stance of the stance of	! INTEGER (HOLLERITH)	! НО
DUMMY ARGUMENTS INTEGER K P(But not PP) REAL P(But not PP) INTEGER (HOLLERITH) HA LOGICAL LA CHARACTER YA CHARACTER J(But not JP) PARAMETER INTEGER JP REAL PP REAL PP CHARACTER YP CHARACTER YP CHARACTER ANY INTEGER PREFIX NOT INCLUDED ABOVE REAL ANY REAL PREFIX NOT INCLUDED ABOVE INTEGER (HOLLERITH) H, BUT NO HO, HA,OR HP LOGICAL L, BUT NOT LO, LA, OR LP LA, OR L			! LOGICAL	! M. MLO.
REAL ! P(But not PP) INTEGER (HOLLERITH) ! HA LOGICAL ! LA CHARACTER ! YA LOOP CONTROL ! INTEGER ! J(But not JP) REAL ! PP REAL ! PP REAL ! PP LOGICAL ! LP LOGICAL ! LP LOGICAL ! LP CHARACTER ! YP CHARACTER ! ANY INTEGER PREFIX NOT INCLUDED ABOVE REAL ! ANY REAL PREFIX NOT INCLUDED ABOVE REAL ! ANY REAL PREFIX NOT INCLUDED ABOVE INTEGER (HOLLERITH) ! H, BUT NO HO, HA,OR HP LOGICAL ! L, BUT NOT LO, LA, OR LP CHARACTER ! Y, BUT NOT YO,		$\label{eq:continuous} \chi = \chi^{(4)(3)} + \chi^{(5)(4)(3)}$! CHARACTER	I YO
INTEGER (HOLLERITH) LOGICAL LOGICAL CHARACTER J(But not JP) INTEGER I	DUMMY	ARGUMENTS	! INTEGER	! ! K
INTEGER (HOLLERITH)			! REAL [8/2] 4/2/2/2	! ! P(But not PP)
CHARACTER			! INTEGER (HOLLERITH)	1
CHARACTER		for a second of the second	!	! *!
LOOP CONTROL I INTEGER J(But not JP) REAL PARAMETER LOGICAL			Parkagyah dikabahahan Sugaha	!
PARAMETER INTEGER I		• .	! CHARACTER	! YA
REAL	LOOP C	ONTROL	! INTEGER	! J(But not JP)
LOGICAL ! LP ! LOGICAL ! LP ! CHARACTER ! YP ! INTEGER ! ANY INTEGER PREFIX ! NOT INCLUDED ABOVE ! INTEGER (HOLLERITH) ! H, BUT NO HO, ! ! HA,OR HP ! LOGICAL ! L, BUT NOT LO, ! LA, OR LP ! CHARACTER ! Y, BUT NOT YO, ! 'Y, BUT NOT YO,	PARAME	TER	! INTEGER	! JP
LOGICAL				l PP
CHARACTER	한미국병국 (1947) -	r spiroitet od		
CHARACTER YP ANY INTEGER PREFIX INOT INCLUDED ABOVE INOT INCLUDED ABOVE INOT INCLUDED ABOVE INTEGER (HOLLERITH) H, BUT NO HO, INTEGER (HOLLERITH) HA, OR HP INTEGER INTEG	A STANLEY	ederold gradien		
GLOBAL OR COMMON INTEGER INTEGER INTEGER PREFIX INTEGER PREFIX INTELUDED ABOVE INTEGER (HOLLERITH) INTEGER PREFIX INTEGER PREFI			! CHARACTER	! YP
! ! NOT INCLUDED ABOVE ! ANY REAL PREFIX ! NOT INCLUDED ABOVE ! NOT INCLUDED ABOVE ! NOT INCLUDED ABOVE ! HA, OR HP ! LOGICAL ! L, BUT NOT LO, ! LA, OR LP ! CHARACTER ! Y, BUT NOT YO, !	CLODAL	OR COMMON		
! REAL ! ANY REAL PREFIX ! NOT INCLUDED ABOVE ! INTEGER (HOLLERITH) ! H, BUT NO HO, ! HA,OR HP ! LOGICAL ! L, BUT NOT LO, ! LA, OR LP ! CHARACTER ! Y, BUT NOT YO, !	GLOBAL	OR COMMON	! INTEGER	! ANY INTEGER PREFIX
! NOT INCLUDED ABOVE ! INTEGER (HOLLERITH) ! H, BUT NO HO, !		•	!	! NOT INCLUDED ABOVE
! NOT INCLUDED ABOVE ! INTEGER (HOLLERITH) ! H, BUT NO HO, ! ! HA,OR HP ! LOGICAL ! L, BUT NOT LO, ! ! LA, OR LP ! CHARACTER ! Y, BUT NOT YO, !				! ANY REAL PREFIX
! INTEGER (HOLLERITH) ! H, BUT NO HO, ! ! HA,OR HP ! LOGICAL ! L, BUT NOT LO, ! ! ! LA, OR LP ! LA, OR LP ! CHARACTER ! Y, BUT NOT YO, !		oc gotorodases	An To the end to but well .	I NOW THAT ITEM TRANSPORT
! INTEGER (HOLLERITH) ! H, BUT NO HO, ! HA,OR HP ! LOGICAL ! L, BUT NOT LO, ! LA, OR LP ! LA, OR LP ! CHARACTER ! Y, BUT NOT YO, !	arthyr og N	April villing me	· Wang talah yang berandan bir 1986 yek	
! HA,OR HP ! LOGICAL ! L, BUT NOT LO, ! ! LA, OR LP ! CHARACTER ! Y, BUT NOT YO, !			! INTEGER (HOLLERITH)	! H, BUT NO HO,
LOGICAL L, BUT NOT LO, L L, BUT NOT LO, L LA, OR LP L CHARACTER LY, BUT NOT YO, L LA			1	
! ! LA, OR LP ! CHARACTER ! Y, BUT NOT YO, !		3 1 1 3 3 3 8 F	The instance of desperon	The second of the second of the second
! CHARACTER ! Y, BUT NOT YO,			! LOGICAL	L, BUT NOT LO,
! CHARACTER ! Y, BUT NOT YO, !				- ·
I segment to the				
			! CHARACTER !	
				! YA, OR YP

Fig. 2 Variables by status

The naming conventions enable the type and status of each variable in the code to be recognized from its name and allowed to be recognized from the contract of the contrac

the top of the property of the property that the property of the transport of the property of

2.2 Conventions for comments were been tweened at the restriction of the particular of the comments were been tweened at the comments of the comments and the comments are the comments of the comments and the comments are the comments of the comments are the comments are the comments of the comments are the comments are the comments of the comments are the comme

All comments within the code should take the form of titles, clauses, or sentences terminated by full stops or periods. Abbreviations terminated by full stops (e.g. etc.) should be either avoided or coded without the full stop (eg etc). Proper names appearing in the middle of a clause or sentence should be prefixed by an asterisk (*), indicating an upper case first letter to the document extraction programme. Alternatively, proper names may be parenthesised by asterisks (eg *FORTRAN*) to indicate that the whole word is to be extracted as upper case letters. Header titles and section titles should be "underlined" by coding - under each character in the next consecutive statement (including punctuation characters). Underlined titles will be recognised by the document extraction programme and printed in upper case. A single letter followed by a full stop will be assumed to be an initial, and extracted as upper case.

With this convention, the following rules apply to an upper/lower case documentation extraction programme.

- a) The first letter after each full stop is UPPER CASE.
- b) Words that are underlined are UPPER CASE.
- c) Words backeted by asterisks are UPPER CASE.
- d) Words prefixed by a single asterisk have first letter UPPER CASE only.
- e) Single letters followed by full stops are UPPER CASE.
- f) All other letters are LOWER CASE.

production of the second of the production of the content of the second of the content of the second of the content of the second of the content of the cont

a thaile in the anneal said of the engineers to separate as any of the last term of the Asset (as the

and 2.3 Code structure of an other control of the structure of the structu

Routines are divided into a preamble or "header", sections, and subsections.

The header contains comments which describe the routine; the main body of the code is divided into numbered sections and subsections.

2.3.1 Header comments and the content of approximation of a section to the section of the sectio

The comments at the head of each routine should take the form of

- a) A C**** card containing a title beginning with the subroutine name.
- to the b) The name of the author and the date written was an of how to make any
- c) Modification details (name, date, and reason for modification)
- d) headed sections, giving the following information:

 PURPOSE. the function of the routine

 INTERFACE. how the routine receives its data and returns its

 results.

 METHOD. how the results are obtained

 REFERENCES. references to external documentation, if any.

Each headed section begins with a comment card containing the title key-word followed by a full stop. The following card should underline the title and the full stop.

The headed section following the PURPOSE section should contain C** in columns 1 to 3. This enables the header comments up to this section to be extracted as OVERVIEW documentation, while the complete header comments would

be extracted as EXTERNAL or INTERNAL documentation. (OVERVIEW, EXTERNAL and INTERNAL levels of documentation are defined in Section 3). An example of a routine header is given in Fig. 3.

di Sandi 183 di Gillande Cilodi pagamenti di edi

```
and the property of the contract of the contra
                                           SUBROUTINE EG1 (HATEXT, KLEN)
С
C**** *EG1* - ROUTINE TO PRINT A TEXT ARRAY and I have an a first series
С
                   TORRESON TO THE ECC. M. W.F. ST & 13/1/82% - SHARE BOOKED IN THE SEC.
С
С
C SECTIONODIFIED BY COLUMN J.K.GIBSON COLUMN 16/11/82 ACTO DEMONSTRATE THE WAY
                                                                                                                                                                                                                                                                                                                                                                  IN WHICH CHANGES SHOULD
С
                                                          of a large a temperature to appreciate the permitted of
C
С
                                            PURPOSE. I populate at the Sand population in the section of the s
С
 C
 С
                                                                                 *EG1* PRINTS A *HOLLERITH* ARRAY CONTAINING 8 CHARACTERS PER
 С
 C
 С
                              AN INTERFACE. TECH NOW NO ADMINST SER OF SELECTION DESCRIPTION SERVICES AND AN ARCHITECTURE.
 C**
 С
                                                                                                                                                                                                         Japan Awar Kanang mgang and and Audion for community that
  C
                                                                                  *CALL* *EG1(HATEXT, KLEN)*
  C
  C
                                                                                                                                                                                                 - *HOLLERITH* ARRAY CONTAINING TEXT.
                                                                                                                         *HATEXT*
             Harbon brief of the *KLEN* Start - LENGTH OF *HATEXT. * of Start Table
  C
   C
                erupa (METHOD) in Minde de la lave e la lapudit de la lave e la lapud
   С
   С
   C
                                                                                    *HATEXT* IS PRINTED USING FORMAT *(10A8).*
                                                                                                                                                                        and the end over the property of the state of the state of
   C
                                                EXTERNALS.
    С
                                               C
    C
                        The Property of None : Property of the Control of t
    C
    С
    С
                                                REFERENCE.
     C
     C
                                                                                     NONE.
     C
      C
```

Fig. 3. Routine header

The header comments should normally be followed by

- a) an IMPLICIT statement
- b) common blocks
- c) the main body of the code.

2.3.2 Common blocks

Common blocks should begin with a C* card containing a title beginning with the common block name. This should be followed by the name of the author and the date written, and modification details. If an editor capable of allowing conditional code is available, a section of comments can be conditionally included giving a table of information under the headings:

_	VARIABLE	TYPE	PURPOSE
* 4	HAMPO IN AMERICAN SE	7000	eranne balancia (k. 1888). Ki
С			

It may also be considered desirable to include the FORTRAN source code of some common blocks in the extracted external or internal level documentation. This is indicated by bracketing such code by C*** cards. Common blocks should be "ruled off" from one another by including a comment card containing minus signs in columns 7 to 72 at the end of each block. Figure 4 illustrates the format for a common block. The CDC UPDATE conditional statements *IF DEF, DOC and *ENDIF may be used to restrict the inclusion of the table of variables, and the inclusion of the C*** brackets unless "DOC" is defined to UPDATE. A common block with conditional statements is illustrated in Fig. 5.

```
С
C *
                *COMMON* *COMEGI* - EXAMPLE OF A COMMON BLOCK.
                                           a length makel. Displicite and Automote makes and the value of our edit
C
                                                                                              13/1/82.
                J.K.GIBSON E.C.M.W.F.
C
                                                             ega, epse Edinole bas (bereduna er Idaara
С
                COMMON /COMEGI/
           I NDVIN. NDVOUT, a HMSG spring . Hair amountains has a TO A Ra
C
                 VARIABLE TYPE Les PURPOSE De la la company & 1 d
C *
С
C
                                                INTEGER LOGICAL UNIT FOR INPUT.
INTEGER LOGICAL UNIT FOR OUTPUT.
               *NDVOUT*
                 *NDVIN*
 С
С
                                              HOLLERITH ARRAY CONTAINING A TEXT MESSAGE.
С
 С
               DIMENSION
               I HMSG(10)
 C
 \mathbf{C}
                                                       Fig. 4. Common block format
 C* *COMMON* *COMEG1* - EXAMPLE OF A COMMON BLOCK.
                 J.K.GIBSON E.C.M.W.F. 13/1/82.
 С
              MODIFIED BY J.K.GIBSON 16/1/82 - TO ILLUSTRATE HOW *CDC*
       *UPDATE* DIRECTIVES MAY

There are and integrate samples samples and a second-second BE USED TO CONTROL THE
                                                                                                                                   DOCUMENTATION PRODUCED.
 *IF DEF, DOC
 C***
                 COMMON /COMEGI/
 *ENDIF
            I NDVIN, NDVOUT, HMSG
 *IF DEF, DOC
 C * * *
 C *
                                                                                 PURPOSE.
                  VARIABLE
                                                    TYPE
                  _____
 С
 С
                                                 INTEGER LOGICAL UNIT FOR INPUT.

LOGICAL UNIT FOR OUTPUT.
                  *NDVIN*
  С
                  *NDVOUT*
  С
                                                 HOLLERITH ARRAY CONTAINING A TEXT MESSAGE.
                  *HMSG*
 С
  С
  C***
  *ENDIF
               THMSG(10) to the Algebra of the transfer of the Algebra decided decided and the Algebra decided and th
  *IF DEF,DOC
  C***
  *ENDIF
```

Fig. 5. Common block with conditional statements (CDC update form)

2.3.3 Main body of the code

The main body of the code should be divided into sections. Each section should be numbered, and should begin with

- a) A C* card containing the section number and title
- b) A comment card underlining the title.
- c) A CONTINUE statement numbered in accordance with the section number.

 (i.e. N00 for section N).

Sections should be "ruled off" from one another by a comment card containing minus signs in columns 7 to 72.

Sections should be divided into subsections. Subsections should be numbered such that N.M is subsection M of section N. Each subsection should begin with:

- a) A C* card containing the subsection number and title
- b) A CONTINUE statement numbered NMO.

eg

C
C* 1.1 SET LOGICAL SWITCHES.
110 CONTINUE

2.3.4 Additional comments

Blank comment cards (ie C in column 1 only) should be included as necessary to improve the layout of documentation and to enhance the readability of the code. Very few additional comments will be necessary, as a listing of the section and subsection titles should represent the steps in the coded algorithm. Additional comments are useful to document branches in the code,

and are necessary in the case of any unusual coding construction. The aim should be to provide comments which, when extracted, list the algorithm coded step by step. Cross references should be to section and subsection numbers, not to statement numbers in the code

ea

GO TO 300

C*

BRANCH TO 3.

Figure 6 illustrates the code structure defined.

11

TALL STATES

```
SUBROUTINE EG2(KDVIN, KDVOUT)
 \mathbf{C}
 C**** *EG2* - ROUTINE TO CONTROL A PROCESS.
 С
       J.K.GIBSON E.C.M.W.F. 13/1/82.
 С
 С
       MODIFIED BY J.K.GIBSON 16/1/82 - CALLS TO *SUB2* WERE
С
С
                                                   ADDED TO EXTEND THE
     化化环酸铁矿 人名阿尔利 化电镀矿 医二甲甲基二甲酚 化铁铁铁矿 电电影 化二十二氢甲烷烷基 网络欧洲亚洲亚洲
С
                                                   SCOPE OF THE ROUTINE.
С
                                      There is also be a marked a company of the com-
С
       PURPOSE.
С
С
С
            *EG2* CONTROLS A PROCESS. IT IS AN EXAMPLE OF THE *DOCTOR*
C
       PROGRAMMING SYSTEM.
C
C**
      INTERFACE.
С
С
С
            *CALL* *EG1(KDVIN,KDVOUT)*
С
С
                 *KDVTN*
                             - LOGICAL UNIT FOR INPUT DATA.
C
                 *KDVOYUT* - LOGICAL UNIT FOR OUTPUT DATA.
С
C
      METHOD.
С
С
С
            VARIOUS SUBROUTINES ARE CALLED TO PERFORM THE TASKS REQUIRED.
С
С
      EXTERNALS.
С
С
С
           *SETCOM* - SUBROUTINE TO SET DEFAULT VALUES.
                    - SUBROUTINE TO DECODE INPUT PARAMETERS.
С
           *GETP*
                    - SUBROUTINE TO PERFORM PART 1 OF PROCESS.
- SUBROUTINE TO PERFORM PART 2 OF PROCESS.
C
           *SUB1*
            *SUB2*
C
С
           *OUTPUT* - SUBROUTINE TO PRINT RESULTS.
С
С
      REFERENCE.
С
      ------
C
С
           A DESCRIPTION OF THE PROCESS MAY BE FOUND ON FILE 234.
С
      NUMBER 567.
C
      IMPLICIT INTEGER(H), LOGICAL(L), CHARACTER*8(Y)
С
C*
      *COMMON* *COMEG2* - VARIABLES USED IN THE COMPUTATION.
С
С
      J.K.GIBSON
                     E.C.M.W.F. 13/1/82.
C
      COMMON / COMEG2/
                NUM2,
     I NUM1,
                            NUM3.
     R COEF1,
                 COEF2,
                            COEF3.
     L LFLAGS
С
C*
      VARIABLE
                               PURPOSE.
                    TYPE
С
C
C
       *NUM1*
                  INTEGER
                              NUMBER OF TYPE 1 PRODUCTS.
С
       *NUM2*
                  INTEGER NUMBER OF TYPE 2 PRODUCTS.
С
       *NUM3*
                  INTEGER
                             NUMBER OF TYPE 3 PRODUCTS.
```

COEFFICIENT FOR TYPE 1 PRODUCT.

C

COEF1

REAL

```
COEFFICIENT FOR TYPE 2 PRODUCT.
                                                                                                                                                                         REAL
C
                                                                 *COEF2*
                                                                                                                                                                                                                                                                                   COEFFICIENT FOR TYPE 3 PRODUCT.
                                                                 *COEF3*
                                                                                                                                                                        REAL
С
                                                                 *LFLAGS*
                                                                                                                                                                         LOGICAL
                                                                                                                                                                                                                                                                                   LOGICAL SWITCHES.
С
C
                                                      DIMENSION
                                               L LFALGS(20)
С
С
C
                                  erome to the 1. General to OPEN FILES. The state of the subtaint appropriate to the control of the state of t
C *
                                                                                                                                                                             ____
С
                                                                                                                                                                                                                                                                respectively. The interest residence properties are not all the interest effects of
C
                    100 CONTINUE
                                                        OPEN(NDVIN) The Tolk form the model of the transfer of the first open and the transfer of the 
                                                         OPEN(NDVOUT, STATUS='OLD')
  С
 С
 С
                                                                                                                                                                             CONTROL THE PROCESS.
 C *
                                                                                                                                                                               _____
 С
  С
                    200 CONTINUE
  С
                         2.1 SET DEFAULT VALUES AND DECODE PARAMETERS.
                     210 CONTINUE
                                                            CALL SETCOM(NDVIN)
                                                            CALL GETP
  С
                                                                                                                                                                                        PART 1 OF THE PROCESS.
  C *
                       220 CONTINUE
                                                             CALL SUB1
                                                             IF (.NOT.LFLAGS(1)) GO TO 220
                                                                                                                                                                                          BRANCH BACK TO 2.2 FOR NEXT ITERATION IF PART 1
   C*
                                                                                                                                                                                          HAS NOT CONVERGED. About it of the means of the mean of the converge of the co
                       222 CONINUE
   С
                                                                                                                                                                                         PART 2 OF THE PROCESS.
   C*
                                                                                                             2.3
                                                                                                                                                                                                                                             in the military of the control of th
                        230 CONTINUE
                                                              CALL SUB2
                                                              IF (.NOT.LFLAGS(2)) GO TO 230
                                                                                                                                                                                            BRANCH BACK TO 2.3 FOR NEXT ITERATION IF PART 2
     C*
                                                                                                                                                                                            HAS NOT CONVERGED.
   C
     С
                        232 CONTINUE
     С
     С
     С
                                                     als mall 3. weed PRINT THE RESULTS. We seed the section as the above a fit to indeed
    C*
                                                                                                                                                                                  ____
     С
     С
                         300 CONTINUE
                                                                CALL OUTPUT(NDVOUT)
     С
     С
     С
                                                                                                                                                                          CLOSE FILES. The state of the s
      C*
      С
      C
                          400 CONTINUE
                                                                CLOSE(NDVIN)
                                                                 CLOSE (NDVOUT)
      С
        С
        C
                                                                 RETURN
```

Fig. 6. Subroutine coded to DOCTOR standard

3. EXTRACTION OF DOCUMENTATION

Some of the coding conventions defined in Section 2 are intended to enable documentation to be extracted by means of a fixed set of rules. Using these rules it is possible to write a simple documentation extraction programme to process the source code and extract the documentation. A programme called DOC is available at ECMWF for this purpose.

3.1 Levels of documentation

Four levels of documentation are defined - OVERVIEW, SPECIAL, EXTERNAL, and INTERNAL. Users may request three levels - OVERVIEW, EXTERNAL or INTERNAL. Sections of documentation within the source code at each lvel begin with a "trigger" and end with a "terminator".

3.1.1 Overview documentation

Overview documentation is triggered by a C**** card, and terminated at the first subsequent non-comment card, unless a trigger for another documentation level is encountered first. Normally overview documentation will contain the title and purpose of each routine.

3.1.2 Special documentation

Special documentation is triggered by a C*** card, and terminated by a subsequent C*** card. The special level of documentation is used to enable source code other than comments to be extracted as part of EXTERNAL or INTERNAL documentation. It should be used to allow the extraction of code which sets the values of critical constants, etc.

3.1.3 External documentation

External documentation is triggered by a C** card, and terminated at the next subsequent triggger or non-comment card. It is assumed that a request to extract external documentation implies the extraction of overview and special documentation as well. The documentation extracted at the external level is

intended to provide sufficient information for an external user to use the routines documented.

3.1.4 Internal documentation

Internal documentation is triggered by a C* card, and terminated at the next subsequent trigger or non-comment card. A request to extract internal documentation implies the extraction of overview, special, and external documentation as well. The internal documentation should contain the basic steps of the algorithms coded, and should be sufficient to enable maintenance and modification of the code.

4. UTILITY ROUTINES

The concept of utility routines is taken from the OLYMPUS system (Roberts, 1974). The purpose of utility routines is to provide the user with a set of tools for performing common routine tasks, such as copying or re-setting arrays.

With the advent of the X3J3 FORTRAN 77 standard, formatted input/output in FORTRAN has been considerably simplified, especially where text strings are involved. Thus many of the standard OLYMPUS utilities are not included in the basic set of utilities for DOCTOR. (It seems pointless to inclue a routine such as

CALL MESSAGE (48H text of message)

to perform the single statement

WRITE (NDVOUT, '(A)') 'text of message')

Figure 5 lists a basic set of DOCTOR utility routines, their CALL sequence, and their purpose. This basic set should be available to users of the DOCTOR system. It may be supplemented as desired.

1-55 30 CALL SEQUENCE OF STANDARD AND A	! The compage of PURPOSE the problem of the first exercit.
! CALL RESETR (PA, KLEN, PVAL)	! Set the first KLEN value of PA to !
	! PVAL. !
! CALL RESETI (KA, KLEN, KVAL)	! Set the first KLEN values of !
•	ls va same to allocate and constant of the con
: CALL RESETL (LA, KLEN, LAVAL)	! Set the first KLEN values of !
1	l molton com one no propositive constitute l
: CALL RESETH (HA, KLEN, HAVAL)	! look to have the state of !! Set the first KLEN values of !!
	l'acoda de la cidade la completa de combo de la segund. L'HA to HAVAL
! CALL COPYR (PA,K1,PB,K2,KLEN)	! Copy KLEN values from !
!!	! ! PA(K1) to PB(K2). !
! CALL COPYI (KA,K1,KB,K2,KLEN)	! Copy KLEN values from !
<u>.</u>	lender of Kentige of Mark in Agree of the little of the control of
! CALL COPYL (LAA,K1,LAB,K2,KLEN)	Copy KLEN values from
	LAA(K1) to LAB(K2).
! CALL COPYH (HAA,K1,HAB,K2,KLEN)	Copy KLEN values from
! !	! ! HAA(K1) to HAB(K2). !
	Scale the first KLEN values of !
!	hwalfallan ib. Milanafelighera iyoni kid ibni mila I PA by PB.
	Scale the first KLEN values of !
	languaren arrivaren 1981 Proposition of the control of the contr
! CALL SIGNR (PA,KLEN)	Negate the first KLEN values of PA. !
! CALL SIGNI (KA, KLEN)	Negate the first KLEN values of KA. !
! !	to the tip the parties the constraint we can
Toward in the period by the end of the	The transporter of the second

Fig.7 DOCTOR utility routines

REFERENCES

- Frank, R.H. 1971 DOCK an INTERNAL/EXTERNAL documentation processor (Copyright Control Data Corp.). (Extracted from the source code by courtesy of Frank Stevens, CDC).
- Gibson, J.K. 1980 Programming Systems, Docmentation, OLYMPUS,, DOCTOR. ECMWF Tech.Memo.No.20, pp.61.
- Roberts, K.V. 1974 The Olympus Programming System Computer Physics Communication, 7, 237-240.

Current Status of Meteorological Bulletins

The following indicates the status of Meteorological Bulletins at 29.10.1991:-

Reference	Title	Status
M0.	General	
м0.0	Introduction	Not issued
м1.	Basic Functional Design of ECMWF's Meteorological Operational System (EMOS)	
M1.0	General description	
M1.0/1	Current and Planned Meteorological Applications Systems at ECMWF	10/85 original version
M1.0/2	Standards for Software Development	06/86 oeiginal version
M1.1 M1.1/1 M1.1/2 M1.1/3 M1.1/4	Data Bases General Message Data Base Seports Data Base Fields Data Base Products Data Base	Not issued Not issued Not issued 01/87 original version (needs updating) Not issued
M1.1/6	Data Base for Trajectory Models	10/91 revision 1
M1.2	Process control	Not issued
M1.3 M1.3/1	Data Acquisition General	06/87 original version
M1.4	Pre-processing	00/00
M1.4/1	General Decoding	09/89 original version 01/91 original version
M1.4/2 M1.4/3	Data checking and validation	05/90 original version
M1.5	Analysis	See Research Manuals
M1.6	Forecast	See Research Manuals
M1.7	Post-processing	06/87 original version
M1.7/1	Post-processing and dissemination	
M1.8	Dissemination	moved to M3.
M1.9	Archives	
M1.9/1	General	Not issued
M1.9/2	MARS User Guide	07/91 revision 9
M1.10	Metview	Not issued

М2.	Guide to the ECMWF Forecasting System	See Research Manuals
м3.	The dissemination of ECMWF products	
м3.0	Introduction to ECMWF's dissemination service	07/91 original version
M3.1	The dissemination of ECMWF products to Member States	07/91 original version
м3.2	User Guide to ECMWF products	<pre>in preparation; until ready, see ECMWF User Guide to ECMWF products version</pre>

Not issued

M1.11 Data Monitoring