Don

SGP Memo 15

DOCUMENTATION STANDARDS FOR PDP 11 SOFTWARE

Version 1. May 13,1982

This section outlines the documentation standards for all PDP 11 software and includes requirements, suggestions and examples. ALL modules will be required to conform to the requirements before they will be accepted for inclusion during an update. Suggestions are just that - they are not absolutely demanded but are strongly encouraged. The examples can be used as models for preparing documentation (they conform to the requirements and include most of the suggestions as well). Example 1 shows the preferred format.

REQUIREMENTS

1. All code will be documented;
2. All documentation pertaining to a given module or subroutine will be contained in the source code for that module; i.e. the code will be self-documenting;
3. The self-documentation will conform to the features provided by the DOC program (see below). In particular, it must use the CD (or ;D) and CE (or ;E) features to define the block of comments to be included into the documentation and to allow creation of a master list of modules;
4. Documentation for a module must include at least the following:
   (a) The name of the module or subroutine and a one line description of what it does (i.e. a CD line);
   (b) The name of the author, and the release date of the current version;
   (c) The module version number;
   (d) Description of what the module does, including algorithms if necessary. Include remarks or generally helpful comments;
   (e) The source language (eg. F4P, FORTRAN 77, Macro etc);
   (f) The FORTRAN calling sequence (subprograms) or command line (main programs);
   (g) A description of the calling parameters, including the data type, units and short description of each parameter. The description of parameters should be in the same order as in the calling sequence;
   (h) A list of the names of all INCLUDE'd files and comment(s) on their function;
   (i) A list of the names of all external functions and subroutines referenced by the module (intrinsic functions may be omitted);
   (j) A cumulative audit trail giving, for each revision, the version number, the author, the date of revision and a short description of the revision.
5. For main programs, in addition to the above, the documentation must include an overlay description if an overlay is needed.

   None of the above items may be omitted; if, for example, there are no external references, then the documentation should so indicate (see Example 1 below).

SUGGESTIONS

1. The order of items in the documentation should be as described above;
2. The documentation for a given module should consist of a single block of comments at the beginning of the source file;
3. The description of parameters should include some notation to indicate whether a parameter is changed (or might be changed) by the module (see example 1 for one method);
4. The name of a module should be placed in a box at the top of the documentation, with spaces between the characters (see examples).

The DOC Program

The documentation part of the DOC program is shown as Example 3 below. The complete Fortran code of DOC is on MAPPER in the file called [210,20]DOC.FTN .

The DOC program will extract documentation from source code, create a file of the extracted documentation and create a table of contents file with one line descriptions of modules, sorted alphabetically. Documentation is defined to be all those lines between a line having the letters CD or ;D as the first two characters of the line and a line having the letters CE or ;E as the first two characters of the line. Both are inclusive; the lines with CD and CE are both included in the documentation. Lines whose first two letters are CD or ;D will be extracted to go into the table of contents file.

The first two characters of each documentation line will be stripped off, so that the CD and CE will not appear in the documentation file. Tab characters will be converted to blanks by DOC on the assumption that tab stops are at columns 9, 17, 25, ...

DOC will accept multiple files as input so that an accurate table of contents file can be generated for a package of modules. The output files are called DOC.DOC (for the documentation) and TOC.DOC (for the table of contents). These files are only created if the /SP switch (see Example 3 below) has been specified in the filespec; otherwise the output is sent to the terminal in the order TOC.DOC and then DOC.DOC.

DOC extracts all lines between the CD and the CE; this means that executable code as well as comment lines can be put into documentation. One example of the use of this feature is shown in Example 2 below, where the subroutine declaration and the INCLUDE statement lie between the CD line and the CE line and so are put into the documentation. Similarly, a programmer might want to include the actual variable declarations, at least for the subroutine formal parameters, into the documentation as shown in the example.

DOC will extract more than one block of documentation for each module (it will turn "on" the documentation extraction every time it encounters a CD line and turn "off" the documentation when it encounters a CE line). Thus, documentation does not necessarily have to come at the beginning of the code. However, it is suggested that a single block be used at the beginning of the program unless there are compelling reasons to do otherwise. Remember that DOC stops extracting lines only when it encounters a CE line so, if you omit this line, the whole of the source code will be included into the documentation.

DOC will format the output documentation file so that each module begins on a new page, with the page header being the characters in columns 4-16 of the CD line . It does this by starting a new page every time it encounters a CD line with non-blank characters in columns 4 through 16; lines with columns 4 through 16 blank will not start a new page and their contents will not be put into the table of contents file. Thus, blocks of documentation interspersed with code will not necessarily have a separate page for each block and a programmer may like to use this feature to, say, compact the documentation for a main program and its "proprietary" subprograms into one block of documentation.

For the CD lines, DOC will copy the line to the table of contents file exactly as it is written (except for the CD). To keep the table orderly, the CD line should have the letters CD in the first two columns, a space, the module name in columns 4 through 16 and a short description of the module in columns 17 through 80.

Example 1:

```
CD DAT11F       PDP11 to FITS format conversion and write or compare
C
C        --------------------
C        !   D A T 1 1 F    !
C        --------------------
C
C        Author:  Jim Torson             Date: Apr 29, 1982
C
C        Version:  9                     Language:  F4P
C
C        Description:
C          This routine converts the PDP-11 format disk map to FITS
C          format and either writes it out to tape or compares it with
C          data read from the tape.
C
C        Calling sequence:
C          CALL DAT11F(FID,TID,FILADR,COMPAR,NO32,TSTART,ERR)
C
C        Parameters:  (* indicates parameter changed by this routine)
C          FID(100)     I*2       File ID of file to read
C          TID(100)     I*2       Tape ID of tape to write or read
C       *  FILADR       I*4       File address at which to queue next read
C          COMPAR       L*1       .TRUE. to compare disk and tape
C          NO32         L*1       .TRUE. if 32-bit IMPS maps are to be
C                                  converted to 16-bit on the tape
C          TSTART       I*2       For writing to tape, this provides the
C                                  index of the tape buffer (TBUF) where
C                                  we are to start putting converted
C                                  pixel values.Ignored when comparing
C                                  disk and tape
C       *  ERR          L*1       Returns .TRUE. if there was an error
C
C        Included files:  None
C
C        Externals:  None
C
C        Revisions:
C          V.7    J.Torson     Feb. 19, 1982    Added compare feature
C            8    J.Torson     Feb. 28, 1982    Fixed bug in 32-bit to
C                                               16-bit conversion
CE           9    D.Retallack  Apr. 29, 1982    Revised documentation block
C
C
      SUBROUTINE DAT11F(FID,TID,FILADR,COMPAR,NO32,TSTART,ERR)
C
       ____The source code for the routine....
```

Example 2:

```
CD FFT          Fast Fourier Transform - Radix 2, Complex
C
C       -------------
C       !  F F T   !
C       -------------
C
C       Author: D.S. Retallack           Date: Oct 10 '81
C
C       Version: 3                       Language: Fortran 66
C
C       Created Mar 26 '75 D.Retallack Version 1  Initial version.
C       Revised Apr 12 '78 D.Retallack Version 2  Fixed bug in testing
C                                                 last cycle.
C       Revised Oct 10 '81 D.Retallack Version 3  Limited to N<=1024
C                                                 points (total, all
C                                                 variates) to save
C                                                 space in work array.
C
C       Included files:  CONST.DCL
C
C       Externals: None
C
C       Description:
C       This routine performs a complex fast Fourier transform of
C       one variable in multi-dimensional data using the procedures
C       outlined by R.C.Singleton in Comm.A.C.M.,11,773(1968).
C
C       Method:
C       Computes the fast Fourier transform for one variable of
C       dimension 2**M in a multi-variate transform.  N is the number
C       of data points, i.e. N = N1 * N2 * N3 * ... * NP for a P-variate
C       transform and KS = NK * NK+1 * ... * NP, where NK = 2**M is
C       the dimension of the current variable.  Arrays A(N) and B(N)
C       originally contain the real and imaginary components of the
C       data in normal order.  Multi-variate data is stored according
C       to the usual convention, eg. A(J,K,L) is in
C            A( (J-1)*N2*N3 + (K-1)*N3 + L ) for J=1,2,...,N1 ,
C       K=1,2,...,N2 and L=1,2,...,N3.  On exit, the real and imag-
C       inary components of the resulting Fourier coefficients for
C       the current variable are in A and B in reverse binary order.
C       Continuing the above example, if the "column" variable is
C       the current one (N2), column
C            K = K(M-1)*2**(M-1) + K(M-2)*2**(M-2) +...+K(0)
C       is permuted to position
C            K(0)*2**(M-1) + K(1)*2**(M-2) +...+K(M-2)*2 + K(M-1)
C       A separate procedure (Subroutine REORDR) may be used to
C       permute the results to normal order between transform steps
```

```
C       or all at once at the end.
C       If  N = KS = 2**M, the single variate transform:
C
C         (XJ + i*YJ) = SUM,K=0,N { (AK + i*BK)*EXP(2*pi*J*K/N) }
C
C       for J=1,2,...,N is computed where (A + i*B) represent the
C       initial values and (X + i*Y) represent the transformed
C       values.  i = SQRT(-1).
C       NOTE: This routine computes the +i transform; for the
C             -i transform (i.e. the inverse), change the sign
C             of all elements of B.
C
C
      SUBROUTINE FFT(A,B,N,M,KS)
      REAL*4 A(N)               ! REAL PART OF INPUT & OUTPUT VECTOR
      REAL*4 B(N)               ! IMAGINARY PART OF VECTOR
      INTEGER*4 N               ! DIMENSION OF INPUT ARRAY (NO. OF
                                ! POINTS TO BE TRANSFORMED
      INTEGER*4 M               ! ORDER; I.E.  N = 2**M
      INTEGER*4 KS              ! SEE DESCRIPTION ABOVE
C
C  FOLLOWING FILE CONTAINS SOME CONSTANT DEFINITIONS (PI, ROOT 2 ETC.)
C
      INCLUDE 'CONST.DCL'        ! CONSTANT DEFINITIONS
C
CE
C
                                .
                                .
                        The rest of the source code
                                .
                                .
```

Example 3:

```
CD DOC          Documentation extraction program
C
C       ---------------------
C       !     D O C         !
C       ---------------------
C
C       Written by Eric Graham
C
C       Version 1.
C
C       February 12, 1982
C
C       Written in FORTRAN 4P
C
C       This program extracts documentation comments from one or more
C       files and either lists the result on the terminal or spools it
C       to the DEC-10 line printer.  It also produces an alphabetically
C       sorted table of contents file with one entry per module.
C
C       Refer to the PDP 11 Staff Manual for details of the rules for
C       documenting source code so that DOC can extract documentation.
C
C
C
C       The program is installed as ...DOC and is invoked as
C
C       MCR>DOC command-line
C
C       Where command-line can take the following forms:              ?
C
C               filespec/SP
C               filespec
C               @filespec/SP
C               @filespec
C               /SP
C
C       or the command-line may be left out.
C
C       If the command-line contains the switch /SP the documentation
C       will be spooled, otherwise it will be written to the terminal.
C       If it is spooled, it is also left in a file DOC.DOC (for the
C       documentation) and in a file TOC.DOC (for the table of contents).
C
C       If a filespec is preceded by an '@', then that file will
C       be searched for the names of files to be documented.
C       Otherwise the filespec itself will be documented.
C       If the file extension is not explicitly given, then files
```

```
C       with extensions .FTM .FTN .MAC .DCL .DTA .PRE will, if they
C       exist, be documented. (If an FTM file exists, the FTN will
C       not be used).
C
C       If the command-line does not contain a filespec, then
C       input is taken from the terminal until a control-Z is detected.
C
C       The filespec may contain the names of more than one file
C       separated by commas.
C
C       Externals:  GETCOM, GETLIN, PUTLIN, TOF
C
      INCLUDE 'DOC.DCL'
C
C       Revisions:
C         Version 1  E.Graham  Feb. 12, 1982  Original version
CE
                              .
                              .
                       The program source code
                              .
                              .
```