AD-A039 180 DAVID W TAYLOR NAVAL SHIP RESEARCH AND DEVELOPMENT CE--ETC F/G 9/2
EDIT: A FORTRAN PROGRAM MAINTENANCE SYSTEM FOR THE TEXAS INSTRU--ETC(U)
MAR 77 P MORAWSKI

UNCLASSIFIED CMLD-77-07 NL

1 OF 1
AD A039180

END

DATE FILMED
5 - 77

DAVID W. TAYLOR NAVAL SHIP RESEARCH AND DEVELOPMENT CENTER

Bethesda, Md. 20084

EDIT: A FORTRAN PROGRAM MAINTENANCE SYSTEM FOR THE TEXAS INSTRUMENTS ASC-7

Paul Morawski

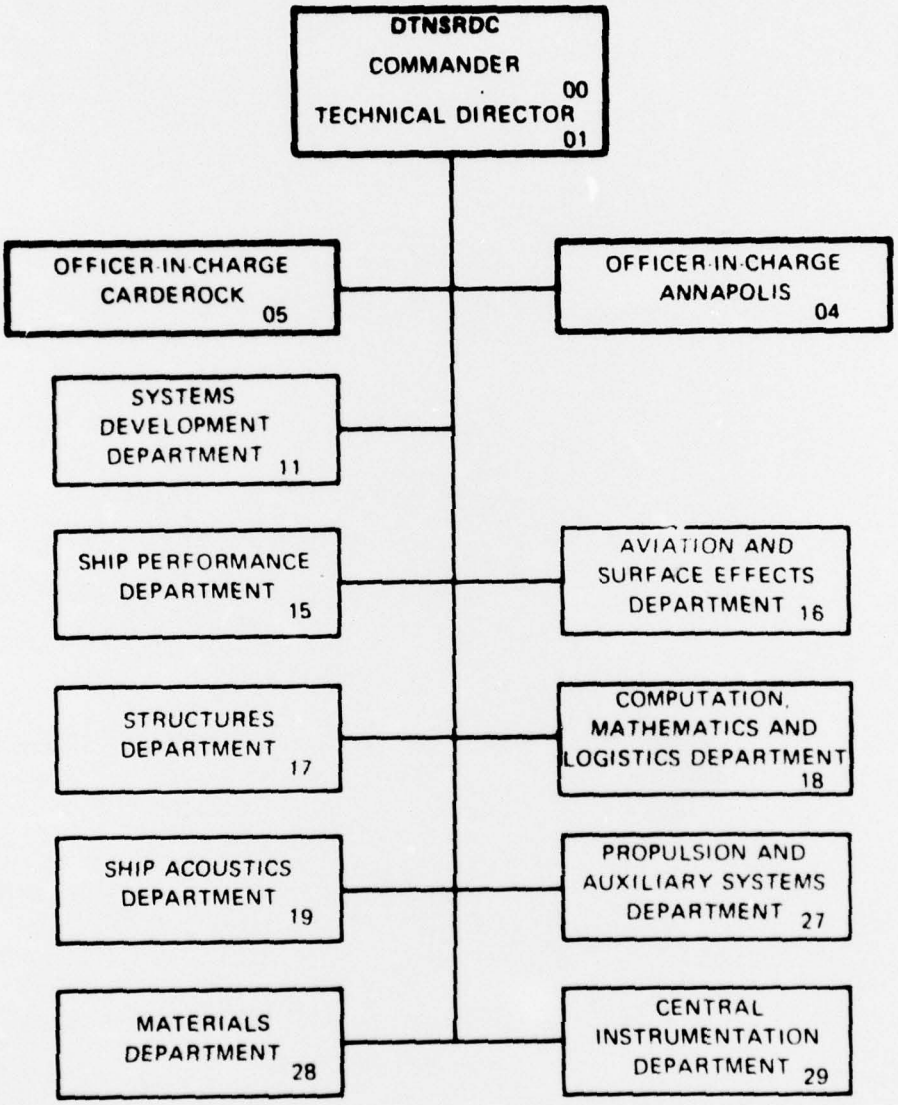
APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED.

COMPUTATION, MATHEMATICS, AND LOGISTICS DEPARTMENT
DEPARTMENTAL REPORT

MARCH 1977

CMLD-77-07

MAJOR DTNSRDC ORGANIZATIONAL COMPONENTS



UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: (14) CMLD-77-07
2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle): (6) EDIT: A FORTRAN PROGRAM MAINTENANCE SYSTEM FOR THE TEXAS INSTRUMENTS ASC-7
5. TYPE OF REPORT & PERIOD COVERED
6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s): (10) Paul Morawski
8. CONTRACT OR GRANT NUMBER(s): (16) F53532 (17)

9. PERFORMING ORGANIZATION NAME AND ADDRESS
David W. Taylor Naval Ship R&D Center
Bethesda, Maryland 20084

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
62760N, ZF53532001
Work Unit 1-1808-009

11. CONTROLLING OFFICE NAME AND ADDRESS (11)

12. REPORT DATE
March 1977

13. NUMBER OF PAGES
~~11~~ (12) 12p.

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

15. SECURITY CLASS. (of this report)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Program Maintenance
Advanced Scientific Computer (ASC)
Macro Programs
Job Specification Language (JSL)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

EDIT is a MACRO program for automatic maintenance of FORTRAN and RATFOR (Rational FORTRAN) programs on the Texas Instruments Advanced Scientific Computer. It has been used extensively to accelerate the creation of complex fluid dynamics programs for execution on the Advanced Scientific Computer (ASC).

EDIT allows the user to more efficiently create and modify programs using the TI-supplied Source Management System, SMS. Object libraries and load modules may also be automatically updated. The program files (source, object,

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

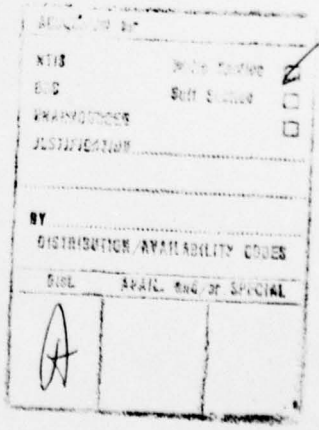
406847

OVER

Item 20. ABSTRACT (cont.)

and load module) have a user-specified number of versions retained as backup. The files may be tape or disk resident. The EDIT system is described and sample usages are presented.

TABLE OF CONTENTS



ABSTRACT

EDIT is a MACRO program for automatic maintenance of FORTRAN and RATFOR (Rational FORTRAN) programs on the Texas Instruments Advanced Scientific Computer. It has been used extensively to accelerate the creation of complex fluid dynamics programs for execution on the Advanced Scientific Computer (ASC).

EDIT allows the user to more efficiently create and modify programs using the TI-supplied Source Management System, SMS. Object libraries and load modules may also be automatically updated. The program files (source, object, and load module) have a user-specified number of versions retained as backup. The files may be tape or disk resident. The EDIT system is described and sample usages are presented.

INTRODUCTION

The Texas Instruments Advanced Scientific Computer (ASC) is a large-scale digital computer which was installed at the Naval Research Laboratory in April 1976. Its vector arithmetic capability and massive central memory make it attractive for use in solving large numerical problems. While the ASC was being used to develop involved fluid mechanics programs, the need for some procedure to automatically update programs became apparent. This deficiency led to the development of a MACRO program, EDIT, which generates the control cards a user needs to update a program library. One EDIT call card and a set of Texas Instruments Source Management System directives (SMS manuals are available from NRL Code 4222) allow a programmer to perform a program update which, if done manually, would require approximately 80 control cards.

EDIT was modeled on a similar program used on the CDC 6000 computers at DTNSRDC. This program, also called EDIT, was developed by Mel Haas of DTNSRDC, Code 1843, in 1972.

DESCRIPTION

The EDIT macro utilizes TI-supplied software (eg., SMS, CIFER, FTN, PDSQSH, etc.) to automate the maintenance of FORTRAN and RATFOR* routines. EDIT stores the source code, object code, and load module for a given program as ASC-cataloged files with a user-specified number of versions.

To use EDIT to maintain a program the user must first catalog a node (i.e., filename) in the ASC tree-structured file catalog system. The user may then use EDIT to perform a "creation run" (flagged by EDIT's OLDSRC and OLDOBJ parameters) to establish his program library. EDIT will catalog three nodes beneath the user supplied node: SRCLIB, OBJLIB, and LOADLIB, for the source code, object code, and load module, respectively. These nodes will have partial access control and full son-add control.† The maximum number of versions of the files is specified by the MXVR parameter on the initial macro call.

EDIT will catalog files containing the source code, object code, and load module, depending on the values of the EDIT parameters IEDIT and CAT. However, these files will not be cataloged if any of the programs called by EDIT end with a termination code which indicates a fatal error. Each time a new version of a file is cataloged, it is automatically flagged by the system as the version to be used by the next EDIT run. When the user-specified number of versions has been cataloged, the oldest version will be replaced by the new version. Thus, a programmer may update and maintain programs by simply supplying update information to EDIT. All file manipulation and library updating are managed by the EDIT macro.

---

* Rational FORTRAN preprocessor, see; Kernighan and Plauger, "Software Tools," Addison-Wesley (1976). Consult DTNSRDC Code 1843, Numerical Fluid Dynamics Branch, (202) 227-1933 for local modifications to RATFOR.

† These terms are used to define specific characteristics of a node. They control who may access the node and who may add sons beneath the node.

INPUT TO EDIT

Input to EDIT is in the form of SMS directives which may either follow the EDIT call or be on a file named by the EDIT EDTINPUT parameter. An SMS $OPTION card is not required, as one is automatically supplied by EDIT ($OPTION A,B,D,L=1). If the user chooses to override the default $OPTION card, the A, B, and D options must still be specified to insure that all libraries will be properly updated. The user is also cautioned that a deck which is deleted from the source library (SRCLIB) is not automatically deleted from the object library (OBJLIB). This situation seldom arises, but may be remedied by using CIFER to manually delete the object code.

EDIT PARAMETERS

The format of the EDIT call is as follows:

/ØEDITØPATH [,IEDIT= {1 / 2 / 3} ,MXVR= {1 / 2 / ⋮ / 64} ,CAT= {YES / NO} ,OLDOBJ= {YES / NO} ,

OLDSRC= {YES / NO} ,DTYP= {TAPE / PAD / HPT} ,RATFOR= {YES / NO} ,FTNOPT=(*) ,

FTVERS=(*) ,SPACE=<integer> ,LNKOPT=(*) ,EDITLIST= <name> ,

LNKINPUT=<name> ,EDTINPUT=<name>]

The quantities between the brackets are all optional; the default values are underlined. The meaning of each of the parameters is given below.

PATH — Existing pathname of program to be edited.

IEDIT
1. Perform source update and compile.
2. Same as IEDIT=1, plus update object library.

3. Same as IEDIT=2, plus build a new load module.

MXVR Number of versions of files to be kept. This parameter is meaningful only for the first EDIT run having CAT=YES.

CAT Command to catalog new files if the run is successful.

OLDOBJ Indicates whether object code has previously been cataloged by EDIT. Should be set to NO the first time EDIT is used with IEDIT≥2 and CAT=YES. Thereafter the default value of OLDOBJ=YES may be used.

OLDSRC Similar to OLDOBJ except that it applies to the source file; OLDSRC should be set to NO the first time EDIT is run with CAT=YES (i.e., a "creation run").

DTYP File storage device desired. This parameter may be different for successive runs, but is meaningful only when CAT=YES.

RATFOR Instructs EDIT that this run updates decks written in RATFOR. Each RATFOR program must be preceded by a card with a sharp sign in column 1 to initiate RATFOR processing.

* See FTN and LNK macros for FORTRAN and LINKAGE EDITOR options. System default options will be supplied if these parameters are omitted.

SPACE Identical to FTN SPACE parameter. System default will be supplied when SPACE is omitted.

EDITLIST Used to name print file. If this parameter is omitted, the print file name will be EDIT.PRT and will be printed automatically.

LNKINPUT Names an optional input file to the LINKAGE EDITOR. This file is required only if the user desires to supply routines to the LINKAGE EDITOR which exists on external libraries.

EDTINPUT Name of file containing the SMS input deck to EDIT. If this parameter is omitted, it is assumed that SMS directives immediately follow the EDIT call card.

EXAMPLES

ESTABLISHING A PROGRAM LIBRARY

```
/ JOB
/ MACASG MACRO,DOD/NAVY/NSRDC/MORAP1/MACLIB
/ LIMIT BAND=30
/ EDIT EXAMPLE/PATH/NAME,OLDOBJ=NO,OLDSRC=NO,IEDIT=2
$MODSET EXAMPLE,USER=JOEUSER
$DECK EXMAIN,LANG=FORTRAN,USER=JOEUSER,ACTION=ADD
      PROGRAM EXMAIN
C
C *** MAIN PROGRAM FOR EXAMPLES
C
      X = 3.
      Y = 2.
      Z = X+Y
      CALL EXSUB(X,Y,Z)
      STOP
      END
/ EOJ
```

This example establishes a source library and an object library beneath the node "EXAMPLE/PATH/NAME", a node which was previously cataloged by the user. The MACASG statement assigns the author's MACRO library (which contains EDIT) to the user's job. IEDIT is set to 2 to inhibit load module creation, since the main program calls a subroutine not yet in the library. The OLDOBJ and OLDSRC parameters are both set to NO to indicate that neither OBJECT nor SOURCE libraries have previously been cataloged for this particular program node.

ADDING A DECK TO AN EXISTING LIBRARY

```
/ JOB
/ MACASG MACRO,DOD/NAVY/NSRDC/MORAP1/MACLIB
/ LIMIT BAND=30
/ EDIT EXAMPLE/PATH/NAME,RATFOR=YES
$MODSET UPDATE1,USER=JOEUSER
$DECK EXSUB,LANG=RATFOR,USER=JOEUSER,ACTION=ADD
#         THIS CARD FLAGS RATFOR DECK
   SUBROUTINE EXSUB(X,Y,Z)
C
C  SUBROUTINE FOR EXAMPLES
C
   R = X*X+Y*Y
   IF (R .EQ. 0.) RETURN          # NOTE THAT THE SHARP
   IF (R .GE. Z)                     # SIGN ALLOWS THE USER TO COMMENT
      {                                 # MORE FREELY
      Z = SIN(Z)
      X = 0.
      Y = 0.
      }
   ELSE Z = COS(Z)
   RETURN
   END
/ EOJ
```

This example adds a deck to the program library created in the first example. The RATFOR=YES option is used to inform EDIT that this run contains references to DECKS which are written in RATFOR. The card with the sharp sign in column 1 is a flag to the RATFOR processor that RATFOR code follows. FORTRAN and RATFOR codes may be freely interspersed, since the END card of each RATFOR program turns the RATFOR processor off. The default EDIT option (IEDIT=3) is used in this case to allow a load module to be built, since this run supplies the subroutine which was missing in the first example.

MAKING A CORRECTION TO AN EXISTING PROGRAM

```
/ JOB
/ MACASG MACRO,DOD/NAVY/NSRDC/MORAP1/MACLIB
/ LIMIT BAND=30
/ EDIT EXAMPLE/PATH/NAME
$MODSET UPDATE2,USER=JANEUSER
$DECK EXMAIN,USER=JANEUSER
$REPLACE 7
      Z=X*Y
/ EOJ
```

This example shows how the user would replace line 7 of the main program of the first example with the desired line, Z = X*Y. Since the default EDIT option is used (IEDIT=3), all libraries, including the load module, would reflect the change.

EXECUTING AN EXISTING PROGRAM

```
/ JOB
/ ASG SYS.LMOD,EXAMPLE/PATH/NAME/LOADLIB,USE=SHR
/ FXQT
/ EOJ
```

This example shows how to execute the program developed in the previous three examples. Since the load module was assigned with the access name SYS.LMOD, the FXQT macro may be used to accomplish a standard FORTRAN execution.

DISTRIBUTION LIST

Copies:			Copies:		
ONR					
1	607	R. Lundegard	5	1892.1	
1	607	M. Cooper	1	1900	M. Sevik
NRL			12	DDC	
1	4221	M. Van Sickle			
1	4222	G. Perez			
15	4222	B. Brooks			
NAVSEC					
2	6136	E. Comstock R. Keane			
DTNSRDC					
1	1500	W. Cummins			
2	1552				
1	1600	R. Chaplin			
1	1700	W. Murray			
1	1720.3	R. Jones			
1	1800				
2	1802	F. Frenkiel F. Theilheimer			
1	1805				
1	1809.3				
1	1820				
1	1840				
1	1843	J. Schot			
30	1843	P. Morawski			
3	1844	S. Dhir M. Golden D. Gignac			
1	1850				
1	1860				
1	1890				

DTNSRDC ISSUES THREE TYPES OF REPORTS

(1) DTNSRDC REPORTS, A FORMAL SERIES PUBLISHING INFORMATION OF PERMANENT TECHNICAL VALUE, DESIGNATED BY A SERIAL REPORT NUMBER.

(2) DEPARTMENTAL REPORTS, A SEMIFORMAL SERIES, RECORDING INFORMATION OF A PRELIMINARY OR TEMPORARY NATURE, OR OF LIMITED INTEREST OR SIGNIFICANCE, CARRYING A DEPARTMENTAL ALPHANUMERIC IDENTIFICATION.

(3) TECHNICAL MEMORANDA, AN INFORMAL SERIES, USUALLY INTERNAL WORKING PAPERS OR DIRECT REPORTS TO SPONSORS, NUMBERED AS TM SERIES REPORTS; NOT FOR GENERAL DISTRIBUTION.